The Stable Framework™

Michael J. Berry

www.StableFramework.org
www.StableInstitute.org
ACKNOWLEDGMENT

I acknowledge and thank Dr. Bret Swan for his encouragement while creating The Stable Framework™.
Forward

Michael Berry’s book entitled *The Stable Framework™* is an excellent read for organizations in the IT industry that desire to evolve quicker than their competitors and ultimately become the future dominators within their industry.

Often, and for good reason, the IT industry is compared to the manufacturing industry in hopes that IT can use traditional manufacturing tools to evolve at a greater pace. What people don’t usually understand is that the manufacturing industry started out with a 2% production success rate and that it took 220 years to get from a 2% to a 99.99964% success rate, which is a sustainable capability for a modern manufacturer. Another interesting fact is that Eli Whitney, the first American manufacturer, missed his first contracted delivery date for 12,000 muskets to the US Government by 12 years.

In time, manufacturers started to understand the need for generating organizational cultures that support analytical tools. If organizational culture does not support analytical tools, then the analytical tools do not work. Hence, for an organization to be capable of effectively using analytical tools to generate profound knowledge about how to improve its capabilities, it needs to have a supportive culture in place.

If the IT industry evolves at the same rate as the manufacturing industry, it will reach similar success rates around 2188 which is 220 years after John Tukey first published the term “Software.” Of course, the hope continues to be that if we can transfer lessons learned from manufacturing to IT this evolution will go much quicker—and I believe it will—however, it is not as easy as one may think to transfer these lessons.

Many books try to teach the IT industry how to use heavy-duty analytical principles even though most IT companies are not yet ready for them. In other words, before their processes are predictable. What makes Michael’s book unique is that he does not introduce these highly analytical tools, as he recognizes that the IT industry must first achieve process predictability before advanced mathematics can apply. The Stable Framework™ is a book written to address the current needs of the IT industry. Michael brings out the tools that were and are used in the manufacturing industry to create predictability. If IT organizations will
effectively implement this knowledge into their organizations, they will have put themselves on a course to become the powerhouses of the future.

I consider Michael to be a visionary to what the IT industry must become and what it is going to take to get there.

—John L. Lee
2016 Shingo Research and Professional Publication award winning author of: *Rising Above It All: The Art and Science of Organizational Transformation*
# Table of Contents

**Forward**.................................................................................................................. v

**Table of Contents**..................................................................................................... vii

**Table of Figures** ....................................................................................................... xiii

**Introduction**............................................................................................................... 1

The Assembly Line.........................................................................................................1
Early Quality Management - Inspection .......................................................................2
TQM and Continual Improvement .................................................................................3
The Beginnings of the Toyota Production System (TPS)..............................................4
Advanced Quality Management – Prevention over Inspection ...............................6
Value Streams ................................................................................................................6
The Toyota Production System ....................................................................................7
Quality Goes Mainstream .........................................................................................8
Operational Excellence ...............................................................................................9
The “Agile” Mindset for Operational Excellence ....................................................9
The (much-needed) “Stable” Mindset for Operational Excellence ............................11
Software as a Manufacturing Process ......................................................................12

**Chapter 1 - The Problem** ....................................................................................... 13

The Hidden Factory .....................................................................................................14
Towards Operational Excellence ................................................................................15

**Chapter 2 - The Hidden Factory in IT** .................................................................. 17

Technical Debt .............................................................................................................19
Quality Debt................................................................................................................19
The “Stable Proposition” ............................................................................................20
Quality Competency ....................................................................................................20

**Chapter 3 - Systems Thinking and Value Streams** .............................................. 23

The Value Proposition ...............................................................................................23
Systems Thinking ......................................................................................................24
System Interactions ....................................................................................................25
The Two Vectors of System Excellence ....................................................................25
Mapping a System ......................................................................................................26
System-Flow Diagrams .............................................................................................27
Systematization ...........................................................................................................27
Value-stream Maps ......................................................................................................29
The Upstream Supplier ...............................................................................................31
The Downstream Customer .......................................................................................32
# TABLE OF CONTENTS

**Chapter 9 - The Present Domain** ................................................................. 95
  - The Sprint Board and Kanban Board ................................................. 95
  - Daily Kaizen Stand-up ................................................................. 97
  - Measuring Team Velocity ............................................................. 98
  - Process Katas and Kata Cards ....................................................... 99
  - Ad-Hoc Kaizen Teams ................................................................. 101
  - CCAPA Committee ..................................................................... 101
  - Change Control Committee ........................................................ 101
  - Training, Retraining, and Cross-training ...................................... 102
  - 5S Housekeeping ........................................................................ 102
  - Alert System ................................................................................ 104
  - Production Change Log ............................................................... 105
  - Suggestion Box .......................................................................... 105

**Chapter 10 - The Past Domain** ............................................................... 107
  - Master Cycle Totals .................................................................... 107
  - Cumulative System Performance ............................................... 112
  - Historical Logs .......................................................................... 114
  - The Performance Console: Tying All Three Domains Together ...... 115

**Chapter 11 - The Four Core Meetings in Stable** ................................. 117
  - The Master Cycle Planning Meeting .......................................... 117
  - The Daily Kaizen Stand-up Meeting ........................................... 118
  - The Master Cycle Review Meeting .............................................. 119
  - The Master Cycle Retrospective Meeting .................................... 120

**Chapter 12 - The Process Owner** ............................................................ 121
  - Customer Engagement ............................................................... 123
  - Process Measures & Service Levels .......................................... 124
  - The Process Kata ........................................................................ 125
  - The Coaching Kata .................................................................... 126
  - The Future Domain and the Process Owner ................................. 136
  - The Present Domain and the Process Owner ............................... 139
  - The Past Domain and the Process Owner .................................... 141

**Chapter 13 - The Master Chief** ............................................................... 147
  - Leadership .................................................................................. 148
  - Leadership Coaching Styles ....................................................... 149
  - Leadership vs. Management ....................................................... 150
  - Leadership as Influence .............................................................. 150
  - The Leader’s Two Teams ............................................................. 151
  - The Three Concerns of Team Leadership .................................... 151
Leadership Styles ................................................................. 156
The Stable Framework™ “Servant Champion” Leadership Model .... 157
The Team Empowerment Model ............................................. 174
Three Models of Excellent Teams ........................................... 184
The Future Domain and the Master Chief .............................. 185
The Present Domain and the Master Chief .............................. 187
The Past Domain and the Master Chief ................................. 190

Chapter 14 - Committees, Groups, and Kaizen Events ............... 193
Change Control Committee ................................................... 193
CCAPA Committee ............................................................... 193
Quarterly Business Review .................................................... 194
Kaizen Teams ........................................................................ 194
Master Kaizen Meeting, or Kaizen of Kaizens .......................... 195
Gemba Walk .......................................................................... 196

Chapter 15 – Stable Hybrid Project Management ....................... 197
Plan-Based Project Management ............................................. 197
Agile Project Management ...................................................... 198
The Hybrid Project Management Approach ............................ 199
Sustained Continual Improvement with The Stable Framework™ ... 206
Improved Project Estimation with The Stable Framework™ ........ 207

Chapter 16 - Stable Operations .............................................. 209
Operations Using the Stable Framework™ .............................. 210

Chapter 17 - Stable Implementations .................................... 215
Implementation Using the Stable Framework™ ....................... 215

Chapter 18 - Stable DevOps .................................................. 219
DevOps Using the Stable Framework™ .................................. 221

Chapter 19 - Stable DevSecOps ............................................. 225
Example DevSecOps Training Kata Card .............................. 226
Example DevSecOps Component Design/Code Review Kata Card ... 227
Example DevSecOps Network Security Kata Card .................. 227
Example DevSecOps Testing DevOps Kata Card ..................... 227
Example DevSecOps DevOps Kata Card ................................. 228

Chapter 20 - Stable Support Desk ......................................... 229
External Support Using the Stable Framework ....................... 229
Internal Support Using the Stable Framework ....................... 229

Chapter 21 - Stable Product Development .............................. 231
Product Development Using the Stable Framework™ ...................... 232

Chapter 22 - Stable Agile Development ................................. 237
  Agile Product Development Using the Stable Framework™ ............ 239

Chapter 23 - Stable Portfolio Management ............................. 241
  Portfolio Management Using The Stable Framework™ ............... 242

Chapter 24 - Getting Started with Stable ............................... 255
  The Path to Stability ............................................................... 255
  The Technology Adoption Lifecycle ....................................... 257
  The Value Realization Curves ................................................. 258
  Quality Planning ................................................................. 259
  Initial Setup ........................................................................ 259
  Taking Over as Master Chief ............................................... 262
  Sustaining the Stable Framework™ ...................................... 262

Chapter 25 - Data Measuring & Modeling Tools .................... 265
  Data Measuring Tools .......................................................... 265
  Data Modeling Tools ............................................................ 275

Glossary of Terms ............................................................... 287

Bibliography ........................................................................... 297

Index ..................................................................................... 301
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Hidden Factory</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>U.S. Software Report (Standish Group)</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>Exponential Cost of Fixing Software Defects</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>Stages of Competency</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>W. Edwards Deming’s System-Model</td>
<td>24</td>
</tr>
<tr>
<td>6</td>
<td>The S.I.P.O.C. Model</td>
<td>26</td>
</tr>
<tr>
<td>7</td>
<td>Software Implementation Process Map</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>Value Stream Map</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>Standardize-Do-Check-Act</td>
<td>38</td>
</tr>
<tr>
<td>10</td>
<td>Plan-Do-Check-Act</td>
<td>39</td>
</tr>
<tr>
<td>11</td>
<td>System Learning Using PDCA</td>
<td>40</td>
</tr>
<tr>
<td>12</td>
<td>Quality Management</td>
<td>52</td>
</tr>
<tr>
<td>13</td>
<td>The Two Roles in The Stable Framework™</td>
<td>70</td>
</tr>
<tr>
<td>14</td>
<td>The Three Domains of The Stable Framework™</td>
<td>74</td>
</tr>
<tr>
<td>15</td>
<td>The Stable Framework™ Master Cycle</td>
<td>76</td>
</tr>
<tr>
<td>16</td>
<td>Plan-Do-Check-Act</td>
<td>78</td>
</tr>
<tr>
<td>17</td>
<td>The Future Domain</td>
<td>85</td>
</tr>
<tr>
<td>18</td>
<td>Service Register</td>
<td>86</td>
</tr>
<tr>
<td>19</td>
<td>Master Services List</td>
<td>87</td>
</tr>
<tr>
<td>20</td>
<td>Supplier Services List</td>
<td>88</td>
</tr>
<tr>
<td>21</td>
<td>Process Asset Matrix</td>
<td>90</td>
</tr>
<tr>
<td>22</td>
<td>Example System Schedule</td>
<td>92</td>
</tr>
<tr>
<td>23</td>
<td>The Present Domain</td>
<td>95</td>
</tr>
<tr>
<td>24</td>
<td>Sprint Board</td>
<td>96</td>
</tr>
<tr>
<td>25</td>
<td>Kanban Board</td>
<td>96</td>
</tr>
<tr>
<td>26</td>
<td>Process Kata Cards</td>
<td>99</td>
</tr>
<tr>
<td>27</td>
<td>5S Housekeeping</td>
<td>103</td>
</tr>
<tr>
<td>28</td>
<td>The Past Domain</td>
<td>107</td>
</tr>
<tr>
<td>29</td>
<td>A SLAM (Service Level Attainment Monitoring) Chart</td>
<td>108</td>
</tr>
<tr>
<td>30</td>
<td>Advanced SLAM Chart Containing SPC Data</td>
<td>108</td>
</tr>
<tr>
<td>31</td>
<td>Incident Register</td>
<td>109</td>
</tr>
<tr>
<td>32</td>
<td>Cycle Totals... the Nbr of Items Completed per Cycle</td>
<td>111</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>33</td>
<td>A Workload Analysis Pareto Chart</td>
<td>111</td>
</tr>
<tr>
<td>34</td>
<td>A Hastings Chart</td>
<td>112</td>
</tr>
<tr>
<td>35</td>
<td>Cumulative System Performance Chart</td>
<td>113</td>
</tr>
<tr>
<td>36</td>
<td>The Performance Console</td>
<td>115</td>
</tr>
<tr>
<td>37</td>
<td>Example Kata Card</td>
<td>128</td>
</tr>
<tr>
<td>38</td>
<td>Cause and Effect, or Ishikawa Diagram</td>
<td>132</td>
</tr>
<tr>
<td>39</td>
<td>A3 Problem Solving Template</td>
<td>134</td>
</tr>
<tr>
<td>40</td>
<td>The Future Domain</td>
<td>136</td>
</tr>
<tr>
<td>41</td>
<td>The Present Domain</td>
<td>140</td>
</tr>
<tr>
<td>42</td>
<td>The Past Domain</td>
<td>141</td>
</tr>
<tr>
<td>43</td>
<td>The Performance Console</td>
<td>142</td>
</tr>
<tr>
<td>44</td>
<td>Service Level Attainment Monitor (SLAM) Chart</td>
<td>143</td>
</tr>
<tr>
<td>45</td>
<td>The Blake-Mouton Managerial Grid</td>
<td>154</td>
</tr>
<tr>
<td>46</td>
<td>The Stable “Servant Champion” Leadership Model</td>
<td>158</td>
</tr>
<tr>
<td>47</td>
<td>Team Development Model (Bruce Tuckman)</td>
<td>172</td>
</tr>
<tr>
<td>48</td>
<td>The Future Domain</td>
<td>186</td>
</tr>
<tr>
<td>49</td>
<td>The Present Domain</td>
<td>187</td>
</tr>
<tr>
<td>50</td>
<td>The Performance Console</td>
<td>189</td>
</tr>
<tr>
<td>51</td>
<td>The Past Domain</td>
<td>191</td>
</tr>
<tr>
<td>52</td>
<td>Work Breakdown Structure (WBS)</td>
<td>201</td>
</tr>
<tr>
<td>53</td>
<td>Project Schedule &amp; Gantt Chart</td>
<td>202</td>
</tr>
<tr>
<td>54</td>
<td>Project Kanban Board</td>
<td>203</td>
</tr>
<tr>
<td>55</td>
<td>Reporting Project Progress with a Gantt &amp; Burnup Chart</td>
<td>204</td>
</tr>
<tr>
<td>56</td>
<td>WBS Showing Cost Allocations</td>
<td>205</td>
</tr>
<tr>
<td>57</td>
<td>Operations Model</td>
<td>210</td>
</tr>
<tr>
<td>58</td>
<td>Operations Model + Stable Framework™</td>
<td>212</td>
</tr>
<tr>
<td>59</td>
<td>SLAM Chart</td>
<td>213</td>
</tr>
<tr>
<td>60</td>
<td>Kanban Board</td>
<td>213</td>
</tr>
<tr>
<td>61</td>
<td>Implementation Model</td>
<td>215</td>
</tr>
<tr>
<td>62</td>
<td>Implementation Model using The Stable Framework™</td>
<td>216</td>
</tr>
<tr>
<td>63</td>
<td>DevOps Model</td>
<td>220</td>
</tr>
<tr>
<td>64</td>
<td>DevOps Using The Stable Framework™</td>
<td>222</td>
</tr>
<tr>
<td>65</td>
<td>DevSecOps</td>
<td>225</td>
</tr>
<tr>
<td>66</td>
<td>Product Development V-Model</td>
<td>231</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>67</td>
<td>Product Dev. V-Model Using The Stable Framework™</td>
<td>233</td>
</tr>
<tr>
<td>68</td>
<td>PMI-ACP® Scrum Model</td>
<td>237</td>
</tr>
<tr>
<td>69</td>
<td>PMI-ACP® Scrum Model Using the Stable Framework™</td>
<td>238</td>
</tr>
<tr>
<td>70</td>
<td>Portfolio Management Model</td>
<td>241</td>
</tr>
<tr>
<td>71</td>
<td>Portfolio Management Using the Stable Framework™</td>
<td>243</td>
</tr>
<tr>
<td>72</td>
<td>Service Performance Inventory</td>
<td>245</td>
</tr>
<tr>
<td>73</td>
<td>Product Performance Inventory</td>
<td>246</td>
</tr>
<tr>
<td>74</td>
<td>Master Project Schedule</td>
<td>247</td>
</tr>
<tr>
<td>75</td>
<td>Portfolio Prioritization Model</td>
<td>249</td>
</tr>
<tr>
<td>76</td>
<td>Opportunity Profile Radar Chart</td>
<td>251</td>
</tr>
<tr>
<td>77</td>
<td>Project Prioritization Matrix</td>
<td>251</td>
</tr>
<tr>
<td>78</td>
<td>Portfolio Kanban</td>
<td>252</td>
</tr>
<tr>
<td>79</td>
<td>The Technology Adoption Lifecycle</td>
<td>257</td>
</tr>
<tr>
<td>80</td>
<td>The Value Realization Curve</td>
<td>258</td>
</tr>
<tr>
<td>81</td>
<td>Burn-up Chart</td>
<td>265</td>
</tr>
<tr>
<td>82</td>
<td>Capability Chart</td>
<td>266</td>
</tr>
<tr>
<td>83</td>
<td>Statistical Process Control Chart</td>
<td>267</td>
</tr>
<tr>
<td>84</td>
<td>Hastings Chart</td>
<td>269</td>
</tr>
<tr>
<td>85</td>
<td>Histogram</td>
<td>269</td>
</tr>
<tr>
<td>86</td>
<td>Pareto Chart</td>
<td>270</td>
</tr>
<tr>
<td>87</td>
<td>Radar Chart</td>
<td>271</td>
</tr>
<tr>
<td>88</td>
<td>Scatter Diagram</td>
<td>271</td>
</tr>
<tr>
<td>89</td>
<td>Sensitivity Analysis</td>
<td>272</td>
</tr>
<tr>
<td>90</td>
<td>Simple SLAM Chart</td>
<td>273</td>
</tr>
<tr>
<td>91</td>
<td>Complex SLAM Chart</td>
<td>273</td>
</tr>
<tr>
<td>92</td>
<td>Kata Card, or Checksheet</td>
<td>275</td>
</tr>
<tr>
<td>93</td>
<td>Network Diagram</td>
<td>275</td>
</tr>
<tr>
<td>94</td>
<td>Affinity Diagram</td>
<td>276</td>
</tr>
<tr>
<td>95</td>
<td>Cause and Effect, or Ishikawa Diagram (Theme Based)</td>
<td>277</td>
</tr>
<tr>
<td>96</td>
<td>Cause and Effect, or Ishikawa Diagram (Process Based)</td>
<td>277</td>
</tr>
<tr>
<td>97</td>
<td>Context Diagram</td>
<td>278</td>
</tr>
<tr>
<td>98</td>
<td>Flowchart</td>
<td>280</td>
</tr>
<tr>
<td>99</td>
<td>UML Sequence Diagram used as a Flow Chart</td>
<td>281</td>
</tr>
<tr>
<td>100</td>
<td>Gantt Chart</td>
<td>281</td>
</tr>
</tbody>
</table>
Figure 101 - Interrelationship Diagram .............................................. 282
Figure 102 - Matrix Diagram .............................................................. 283
Figure 103 - Prioritization Matrix ....................................................... 284
Figure 104 - Process Decision Program (PDPC) Chart ......................... 285
Figure 105 - Tree Diagram ................................................................. 285
Introduction

Deming warned us. We are in a crisis.

The first interchangeable part design in recorded history was attributed to Jean-Baptiste Vaquette de Gribeauval, the French cannon maker. In the 1700’s his design helped Napoleon win many battles by creating lighter-weight cannons that were faster to produce and more resilient because they had interchangeable parts. A side benefit to this development allowed two broken cannons on the battlefield to be combined to make one functional cannon again. Napoleon may have been a strategy genius, but having better cannons, and more of them, was an enormous tactical advantage.

Inspired by this innovation, Honoré Blanc demonstrated his ability to create muskets with interchangeable parts in the late 1700’s. Afraid of upsetting the labor guilds, government leaders in Europe were skeptical and unreceptive to investing in Blanc’s concept. An American ambassador to France named Thomas Jefferson was the exception. He realized this capability would free America from dependence on European governments for military supplies.

Jefferson tried to convince Blanc to move to America and further develop his concept there, but Blanc was unwilling. Jefferson eventually returned to the states and convinced President George Washington to fund the idea. In 1798 a contract was issued to Eli Whitney for 12,000 muskets. Whitney was known for inventing and manufacturing the Cotton Gin which transformed the cotton industry in the southern states. Although Eli Whitney produced all the muskets ordered, they were delivered 12 years late.

The Assembly Line

In 1855, Samuel Colt, inspired by a steam ship’s paddle-wheel, invented the 6-shooter revolver and designed a factory with the help of Eli Whitney’s son to produce the guns using interchangeable parts and a new idea called an assembly line. With this combination, Colt was able to produce 5000 revolvers a year, giving a huge advantage to the Texas Rangers and the American Army, enabling them to win the Mexican-
American war.

Samuel Colt’s factory caught the attention of Henry Ford, who copied those ideas and mass-produced many types of automobiles. Henry Ford’s factory employment model—offering a steady job with an eight-hour workday—was the beginning of the middle class in America.

The "American System of Manufacturing" quickly became a worldwide standard. Industrialists around the world raced to duplicate the concept of using an assembly line and interchangeable parts to mass produce goods.

In 1911 Frederick Taylor introduced the concept of “Scientific Management,” demonstrating that pre-planned work tasks on assembly lines can be scientifically optimized and tuned to match each worker’s “Personal Coefficient” to enable sustainable, optimized production. This combination benefited both the worker and the owner because the operation could be tuned to produce the maximum possible output and the best quality of work for the smallest combined costs of human effort, natural resources, and other capital expenses. As management and employees worked together to widen the gap between top-line income versus bottom-line expenses, the organization could afford to pay the employees better wages. If they collaborated, they would both win.

Henry Ford successfully implemented Taylor’s management techniques to decrease costs and increase production. Using these techniques, the Model T Ford became the most successful and least expensive automobile they produced at that time.

*Early Quality Management - Inspection*

Although production yields increased, quality management at that time was in its infancy. Inspectors waited at the end of assembly lines to separate non-conforming goods from sellable goods. Non-conforming goods were set aside for someone to examine and then rework, discard, or occasionally forward on to an unlucky customer.

In 1924, Walter Shewhart introduced the “Statistical Process Control Chart” while working for a manufacturing supplier to Bell Laboratories. Using process data sampling, Shewhart demonstrated that every repeatable process contains measurable natural variation which made it
INTRODUCTION

possible to detect if a failure in the process was caused by inherent randomness, or by an external cause.

This was important because the approach used for addressing either type of failure should be different. The latter is risk-prevention based, while the former requires better production tools and methods. Shewhart also introduced the world to Systems Thinking, which is understanding everything around us as an ecosystem of systems, subsystems, and supersystems, each one complete with inputs, transformational throughputs, and outputs. He introduced the famous “Plan-Do-Check-Act” model, and explained how it is used to gain a better understanding of dynamics within these systems. During this time, he also worked with a fellow statistician, Joseph Juran, to develop an idea called TQM, or Total Quality Management.

TQM and Continual Improvement

TQM is the idea that quality is the responsibility of everyone—not just the inspector at the end of the assembly line—and that quality efforts should be continual and customer focused.

In time, Juran developed a quality philosophy known today as the “Juran Trilogy.” This includes Quality Planning, Quality Control, and Quality Improvement. In addition, he made popular the use of Pareto Charts and the Pareto Principle (80/20 rule) in defect management to emphasize the importance of “the vital few” issues being more important to address than “the trivial many.” Juran also identified the need for better people management within work environments to achieve better quality. He emphasized the importance of management theory in improving production yields, making quality principles relevant beyond manufacturing and into service industries. Juran also developed the “Cost of Quality Analysis,” where quality-related financial data can be assessed for each value proposition an organization provides. He is famous for publishing the Quality Control Handbook, which is still a standard today.

Several years later in 1927, a young statistician named W. Edwards Deming met Walter Shewhart and Joseph Juran and merged Shewhart’s ideas of Statistical Process Control and Systems Thinking with Juran’s ideas of people management. Deming proposed that people were just one of many variable factors within a system and that while the employees
should be responsible for performing a job correctly, management should be responsible for minimizing variation within the system. Deming provided data to show that typically 85% of manufacturing problems were attributed to bad management.

Deming approached the major U.S. automobile manufacturers with his innovative concepts but found no interest. The prominent management thinking of the day was to find and hire good workers and then incentivize or threaten them in order to meet quotas. They were reluctant to accept Deming’s ideas which pointed blame at management for most production problems. The “get it done” quota mentality took over in America at the expense of advanced quality initiatives and customer satisfaction.

At this point, WWII started and ended. The great industrial nations of the world were devastated, except for the United States. Immediately after the war, American manufacturers were able to export and sell almost everything they could build. Quality was becoming much less important and was certainly not a primary concern. In fact, as the quota mentality grew, an attitude started to brew in America that “quality kills.”

The Beginnings of the Toyota Production System (TPS)

In 1949, the U.S. Army asked Deming to assist with census work being conducted in Japan. While there, Deming gave a landmark speech in August 1950 to a room full of Japanese industrialists about Systems Thinking, Statistical Process Control, and Total Quality Management. He was soon asked by many of the manufacturers to stay in Japan and help them rebuild their factories using those ideas. He stayed for 25 years.

Several months before this landmark speech, the U.S. Army sent senior engineers Taiichi Ohno and Eiji Toyota to visit Henry Ford’s automobile plant in Dearborn, Michigan. The Korean War had started, and the Army wanted the Toyota Motor Company to build trucks for the U.S. troops in Korea because it was cheaper to ship them from Japan than from Michigan.

The engineers returned in awe. While Toyota had produced only 2,500 cars in its 13-year history, Ford was manufacturing 8,000 cars a day. Over the next few decades, Taiichi Ohno and Eiji Toyota applied the mass production and other quality ideas from the Ford plant, but adjusted
them with a focus on better team structures and continual improvement practices. Together, these engineers created the foundation for the Toyota Production System (TPS).

TPS evolved to include a list of eight sources of waste, a streamlined “Just-In-Time” supplier delivery system, and innovations such as self-directed cross-functional teams, Kanban cards, inventory-pull systems, and other innovations. Eiji Toyota and Taiichi Ohno eventually became acquainted with Deming and then Juran through the organization known as JUSE, or the Japanese Union of Scientists and Engineers. Juran arrived in Japan in 1955. Together, Deming and Juran helped Toyota and other companies add Statistical Process Control, Systems Thinking, Management Theory, the Juran Trilogy, and the Cost of Quality Analysis to their quality practices.

During this time, Taiichi Ohno and Eiji Toyota observed how quality improved tremendously when employees were respected and encouraged to contribute and grow. After all, they reasoned, how can people continually improve an environment if they are not continually improving themselves?

Also in 1955, the Toyota Motor Company hired an independent consultant named Shigeo Shingo to combine all of these concepts and teach them to thousands of Toyota employees. After his contract ended, Shingo went on to apply these principles at Mitsubishi, reducing the time needed to produce a supertanker from four months down to two, and setting a world record. A major TPS innovation Shingo used extensively was a source inspection technique called the “Poka-Yoke.” This is a “mistake-proofing” tool developed to ensure a component or a step is produced or performed correctly. Using an innovative tool like this ensured zero defects and eliminated most of the need for Statistical Process Control. He later had his training books translated into English and became a fundamental catalyst in introducing the West to Asian manufacturing techniques that we know today as the Toyota Production System (TPS).

In 1962 Kaoru Ishikawa, a Japanese shipyard quality manager, developed and introduced the concept of Quality Circles to JUSE. Quality Circles are activities where workers that do similar work meet at the end of each shift in small groups to discuss recent improvements, and identify, analyze, and solve work-related problems. Today we call these “Kaizen Circles.”
Kaoru Ishikawa later introduced the Ishikawa Diagram, in 1982, for performing root cause analyses of challenging problems experienced during production.

While Deming stayed in Japan, helping various companies improve their manufacturing processes, technologies continued to advance in America.

**Advanced Quality Management – Prevention over Inspection**

Philip Crosby became the senior quality engineer for the Pershing missile manufacturing process in the 1960s. His experience there led to the publication of a popular book in 1979 called *Quality is Free*, where he emphasized the DIRFT concept, or “Doing It Right the First Time.” This concept emphasized preventative-based quality, a zero-defects goal, a new way of thinking about quality deficiencies called the “price of nonconformance,” and a new definition for quality: “conformance to product and customer requirements.”

**Value Streams and Flow**

In 1984, Eli Goldratt published his famous book *The Goal*, where he introduced the “Theory of Constraints.” This is a concept that emphasizes the importance of prioritizing the flow of value through a manufacturing environment and suggests that companies should continually identify and remove any bottlenecks found within their processes until they can produce products as fast as their customers consume them.

This flow of value, called a "value-stream," has become another focal point for efficiency. Companies are now re-structuring their entire operations around this concept. Instead of arranging teams, activities, and assets in easy to manage similar groups--which tends to encumber the flow of value--they are making the smooth flow of value the priority. This requires cross-functional teams with unobstructed access to activities and assets.

In 1986, Alfie Kohn published a compelling book called *No Contest*, expressing the enormous benefits of people cooperating with each other instead of competing. He pointed out that our Western American culture
is addicted to competition, and that we would accomplish much more together if we learned to cooperate rather than compete. Deming became a big fan and advocate of Alfie Kohn’s ideas.

That same year William B. Smith, Jr. and Dr. Mikel J. Harry developed a set of tools at Motorola, based on Shewhart’s Statistical Process Control techniques, for identifying and minimizing variation in a process. They called their new program Six Sigma. It was soon adopted as a core business strategy by Motorola and then General Electric before going mainstream.

**The Toyota Production System - Lean**

In 1988, a young MIT student named John Krafcik wrote a paper about his observations made while visiting several factories in Japan. He observed factory workers actively engaged in quality improvement activities for each repeatable assembly line step. In his paper, he called what he observed "Lean." This name has now become the western version of the practices and philosophies derived from the Toyota Production System.

Although the Western outlook on quality was maturing, Japan had a 30-year head start, and a culture better tuned for group work and continual improvement. As Japan joined the world economy, the crisis Deming would soon warn us about began.

The first major victim of the quality revolution was the U.S. electronics industry, which died in the 70s. American electronics manufacturers such as RCA, Zenith, and Magnavox lost their businesses to Sony, NEC, Panasonic and other Asian companies. Nobody seemed to ask why. American electronics manufacturers simply assumed the Asians were just willing to work for lower wages, which lowered the price of their well-built products.

The second major victim of the quality revolution was the U.S. auto industry, which in the 1980s realized its products were statistically inferior to Asian competitors and were now starting to lose market share to Asian companies.

By 1982, the Ford Motor Company was in full crisis mode. The company had accumulated $3 billion in losses during the previous four years and
management knew they needed to change something. In desperation, Ford sought out Deming and asked for his help. Thanks to a timely NBC television broadcast that aired in June of 1980 called “If Japan Can, Why Can’t We?” Deming publicly showcased the improvements that revolutionized Japan and challenged American companies to follow suit. He spent the next few years helping Ford re-educate its management teams, helping them create a culture of quality, and introduce proven quality principles. By 1986, Ford had become the most profitable car company in America, beating out General Motors and Chrysler for the first time since 1920.

Also in 1982, Deming published his famous book, Out of the Crisis, in which he warned modern industrial nations that we are now in a crisis. The landscape has changed. No longer will mediocre quality levels maintained to fulfill production quotas keep a company profitable in a modern world-market. No longer will punctuality, and following top-down direction be all that is required to keep a company competitive.

Deming emphasized a shift in management philosophy away from the “quota” mentality and towards an “improvement” mentality. Instead of encouraging line workers to achieve a daily quota, he taught management to encourage their line workers to innovate in ways that would produce more products, with fewer defects, in a shorter amount of time.

In addition he encouraged a culture shift in favor of cross-functional teamwork, employee empowerment, and continual improvement. Deming pointed out the solution to progress is not expensive robots and computer technology. Those systems add costs to the final products. Anybody can build a better product eventually using expensive technology. The trick is to build a better product for less cost, not more.

**Quality Goes Mainstream**

In 1986, the Space Shuttle Challenger exploded 73 seconds into flight in front of the entire world, killing all seven crew-members. It was the worst accident in the history of the U.S. Space Program. Three months later, the number four reactor at the Russian Chernobyl Nuclear Power Plant melted, triggering the worst nuclear disaster in world history in terms of costs and casualties—even greater than Fukushima. Suddenly, the world got serious about quality.
The next year, in 1987, a series of quality initiatives emerged. ISO 9000 was published by the International Organization for Standardization. The Malcolm Baldridge award was created by the U.S. Government signaling the importance of “Process and Quality Control.” The Software Engineering Institute (SEI) started developing a software development quality standard called the Capability Maturity Model integrated (CMMi).

Two years later the British Government published the Information Technology Infrastructure Library, or ITIL, and COBIT was released a decade later in 1996.

Most of these programs originated from government entities. They were created primarily for internal use, military, and government vendors. Although they have gained popularity in some industries, they were designed to ensure quality and thoroughness, rather than value flow, or production speed. A common criticism for all of them is the enormous amount of training, process additives, and additional personnel required for a reasonable implementation. In addition, certification costs are prohibitive for all but the largest companies.

Operational Excellence

Deming and all of these quality pioneers were really just helping their organizations achieve Operational Excellence, or OpEx. Operational Excellence has two primary objectives: producing customer value, and producing it efficiently. We might think of these twin objectives as effectiveness or “external quality”, and efficiency or “internal quality.” Every product or service-oriented company everywhere is attempting to produce what is most valuable to their customer base, and attempting to do so while minimizing waste along the way. World class producers are relentlessly focused on these two objectives. Deming spent his career educating organization about them.

The “Agile” Mindset for Operational Excellence

In 1986, the same year of the Challenger and Chernobyl catastrophes, Japanese Management professors Hirotaka Takeuchi and Ikujiro Nonaka published an article in Harvard Business Review called “The New New Product Development Game.” This article highlighted the benefits of
cross-functional teams working intensely together at Toyota, Honda, NEC, Epson and other lean Japanese companies, generating continual improvement. Takeuchi and Nonaka described this method of working together as “Scrum,” which is a term borrowed from Rugby where the players huddle closely, interlocking arms.

Many I survey report one-third or more of their teams effort is lost to re-doing work that wasn’t done right the first time around.

Jeff Sutherland would eventually come across this article and, together with Ken Schwaber, develop and introduce “Scrum” to the IT community in 1995. Scrum combines many of the concepts described previously to enable development groups to prioritize customer value while staying flexible to pivot quickly as customer needs change.

This “Agile” mindset would soon become mainstream and be immortalized in the signing of the Agile Manifesto in 2001 by Sutherland, Schwaber, and 12 other software industry pioneers.

Most Agile professionals today would be surprised to learn the roots of Scrum, the most common Agile framework, came from the Toyota Production System. Other TPS inspired Agile fixtures include information radiators, daily stand-up meetings, retrospectives, self-empowered teams, single-piece flow Kanban theory, short-feedback cycles, and continuous customer involvement.

Since 2001, Technology companies have adopted Agile practices to address the first OpEx objective--delivering value. This now mature “Agile Mindset” empowers organizations with essential practices needed to prioritize customer value and make quick changes when needed. Thanks to consistent multi-level planning, regular customer involvement, and short feedback cycles, Agile organizations are able to hyper-focus on their customers’ needs, produce what’s most valuable to them, and benefit accordingly.

Using these techniques, Agilists have improved software delivery significantly. Customers have received what is most valuable to them in incremental installments, instead of a complete-but-less-accurate
solution delivered far into the future.

**The (much-needed) “Stable” Mindset for Operational Excellence**

The second OpEx goal remains unaddressed in IT, until now. Two decades after the Agile Manifesto was signed, we have enough history to conclude that for the average adopter, Agile has improved software project success rates by about 9%, and reduced project failure rates by about 9%. These factors are better for aggressive adopters. What hasn’t improved since that time is project performance.

Projects are still about as late and over-budget as they have ever been, even before the signing of the Agile Manifesto in 2001. In addition, rework, scrapped code, and fire-fighting continue to be significant factors internally plaguing most tech organizations. Many I survey report one third or more of their team’s effort is lost to re-doing work that wasn’t done right the first time around.

While the Agile community continually seeks better ways for people to work together, the next evolution needed in IT performance management is a different focus—protecting our workflows from human failure. In 2001 Larry Smith gave us “Shift-left” Thinking for improving quality by taking preventative steps earlier in the value stream. Google recently gave us *Site Reliability Engineering*, or SRE, for building robust systems. The *Stable Framework™* can be thought of as *Process Reliability Engineering*, or PRE. Combined with SRE, we have *SPRE*, or Site & Process Reliability Engineering.

Indeed, the 12th and final principle behind the Agile Manifesto reads “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.” This is where Agile ends, and Stable begins.
Software as a Manufacturing Process

Anything repeatable can be improved. If we think of Software Development, DevOps, Operations, Implementation, and other IT functions as “assembly lines of decisions,” and each repeatable step within these assembly lines as systems, having supplier-fed inputs, transformational throughputs, and customer-facing outputs, a context emerges for augmenting both Agile and non-Agile Development and Operations teams with additional quality practices currently enjoyed in leading Lean manufacturing companies. If applied correctly, these tools empower organizations to achieve Operational Excellence within IT and beyond.

The upside is huge. Reclaiming nearly 33% of a departments effort over time means you can reclaim 33% of your personnel budget, monetize your projects 33% faster, and reduce your related support staff burden significantly. Add these up and think about the possibilities.

To this end, the Stable Framework™ was developed as a simple, out-of-the box model for relentless and sustainable process improvement.

Stable can be executed as an Agile accessory—some call this “Stable Agile,” or as a standalone set of practices for any IT or engineering group who wants to minimize unwanted cost and time expense due to unnecessary mistakes.

The combination of Agile and Stable practices, packaged within the Stable Framework™, provide adopters with the transparency and conscientiousness necessary to achieve the two goals of Operational Excellence and become world-class competitors.

To every hard-working, sincere, and well-meaning team working in a chaotic environment, following the practices in this book will improve your workplace, your products, your career, and fundamentally change your life.
Chapter 1 - The Problem

In Dr. Atul Gawande’s book, The Checklist Manifesto, we read about the tragic story of the first B-17 airplane crash.

On October 30, 1935, the Boeing Company was demonstrating its newest large warplane to what is now the U.S. Air Force at Wright-Patterson Air Force Base, near Dayton, Ohio. The aircraft was a marvel of engineering. It was the first aircraft with four engines. The wings were so large they contained crawlspace allowing the crew to access them during flight. The plane could carry heavy cargo and even had a kitchenette for long distances. The most striking feature was an array of armaments making it what some called a flying battleship, or fortress.

That Wednesday morning the plane was queued up to launch and demonstrate its majestic abilities in front of a set of Air Force senior observers. Onboard, piloting the craft was the Air Force senior test pilot, Major Ployer P. Hill, the Boeing Company’s senior engineer, Les Tower, and several additional observers.

The plane took off and entered a steep climb. It headed almost straight up then tipped to one side. Stalling, it abruptly fell to the ground, crashing dramatically in front of the spectators and exploding in a ball of fire. Both Les Tower and Major Hill died that day.

The evidence submitted to the inspection board indicated that the root cause of the crash was that a gust lock mechanism protecting the flaps from moving around while on the ground had not been disengaged before taking off, making it impossible for the pilot to control the aircraft once it left the ground. Someone had forgotten to unfasten the lock before takeoff, and there was no process or additional mechanism in place for the pilot to check for this. In fact, at that time, pilots were supposed to get proper training once and then operate an aircraft thereafter from memory. A retrospective analysis by Boeing concluded that the number of distinct attributes in these newer aircraft designs had evolved to such a degree that there were just too many factors to manage from a pilot’s memory alone. Boeing’s solution was to create a pilot’s checklist and require its use before takeoff on all future flights.

The B-17 Flying Fortress eventually became a huge success, with a total of 12,726 units produced for the war. As a result of this catastrophe the pilot’s checklist became a critical tool used to manage complexity for
THE STABLE FRAMEWORK™

every major industry and is a primary component in most published quality frameworks.

The Hidden Factory

Like a modern, complex aircraft, today’s IT environments have a variety of complex components that must be coordinated correctly to produce value efficiently. Entire teams work together to collect, enrich, and handoff physical and intellectual assets from suppliers to customers in an ongoing stream of effort. Every step along the way has its own set of attribute data. Some simple. Some complex.

Sometimes the communication is good, sometimes it is not. Some decisions get made based on assumptions. Some of them are correct. Some are not. Few, if any, processes are systematized, making an improvement only temporarily beneficial until someone forgets about it. The same discovery is “relearned” later by another team who mistakenly thinks they are continually improving. All of this creates value, rework, and scrap. Customers pay for all three, but they would prefer not to pay for rework or scrap. We call rework, scrap, re-direction, delays and other forms of waste the “Hidden Factory” within a Production or Operations environment. This Hidden Factory is the problem the Stable Framework™ addresses.

The Hidden Factory

![Diagram of the Hidden Factory]

Figure 1 - The Hidden Factory
Towards Operational Excellence

As mentioned in the introduction, Operational Excellence is the combination of effectiveness, and then efficiency. Every company, everywhere is pursuing Operational Excellence, whether they have the vocabulary for it or not. Effectiveness is delivering precisely what the customer values most, and efficiency is delivering it with minimal byproduct.

While an Agile mindset helps organizations achieve effectiveness, a Stable mindset helps organizations achieve efficiency,

By reducing mistakes, downtime, bottlenecks, confusion, and other factors slowing down workflow, the Stable Framework™ augments Agile to enable organization to achieve Operational Excellence, and dominate their industry.
Chapter 2 - The Hidden Factory in IT

The Standish Group International is a consulting firm that publishes a report card for the United States software industry every few years. With software being close to a $225 billion industry in the U.S., it’s interesting to examine the size of its Hidden Factory. It’s impossible to measure this number exactly, but the Standish Group’s data provides some valuable insight.

One metric Standish publishes is the ratio of successful versus failed versus late projects. The firm defines a successful project as a project delivered on time, within budget, and containing all the targeted scope. Standish’s data showed that only a small percentage of software projects met this definition. Also, a large percentage of projects were considered “failed,” meaning they were completely scrapped.

This includes projects completed and deployed and promptly rejected by the customer. The data showed that during the past ten years this number was about 20% of all projects. In addition, this data showed that about 45% of the projects were significantly late.

While failed projects are usually rejected because the customer can’t use the finished product, late projects are late because of the Hidden Factory.
The data shows that after Agile was introduced into the development community in 2001, project failure rates diminished about nine percent, and project success rates increased about nine percent, but since the advent of Agile methodologies the percentage of late projects hasn’t changed even one whole percentage point. In other words, Agile has helped the industry produce more acceptable software, but not in a more predictable manner. You might say Agile helped with effectiveness, but not efficiency. Both are required to achieve Operational Excellence.

Agile development patterns, which prioritize value and flexibility, help the industry lower project failure rates by involving the customer throughout the development process, and by providing multiple pivot points at various levels of planning. This has been extremely helpful.

The next evolution the IT industry needs is a Lean-based quality program such as Stable to address the Hidden Factory in IT. This solution would lower the costs of developing and supporting software and services by minimizing mistakes, scrap, and rework. When the industry matures in this area, we will see lower project late rates.

I’ve asked hundreds of IT professionals to estimate the amount of effort within their environment that is lost to the Hidden Factory. Think for a minute about your own environment. What percent would you estimate your Hidden Factory to be? Most say about 33%.

Now put a dollar value on that number. 33% of your entire staff’s payroll aggregate every month. What would you do to reclaim that cost?

In addition, the higher up in authority you are, the less Hidden Factory you are made aware of. A fair number of mistakes and rework get hidden from those above. So how can we address these problems?

Stable provides the flexibility of Agile, combined with the tools to practically eliminate the unwanted Hidden Factory. How much money and time is that worth to your organization?
CHAPTER 2 - THE HIDDEN FACTORY IN IT

**Technical Debt**

The term Technical Debt is sometimes confused with the term Hidden Factory, but they are not the same thing.

Technical Debt is a term coined by Ward Cunningham, the programmer who also created the first wiki. Technical Debt is a concept in software development that suggests a rework cost will probably be needed in the future to compensate for an earlier decision to use coding or architecture shortcuts.

While Technical Debt doesn’t always need a resolution, the Hidden Factory comes from inaccurate work that must always be corrected.

**Quality Debt**

Often, project problems are not discovered until later in the production, implementation, or the service cycle. Studies conducted in 1999 and

![Figure 3 - Exponential Cost of Fixing Software Defects](image-url)
2004 underscore the reality of the hidden cost of problems discovered later. The studies found that as defects are created in a workflow, they become exponentially more expensive to fix the further down the lifecycle they travel. This means that a defect created in the Requirements phase is relatively inexpensive to fix if found during that step in the process. However, once the team has coded the wrong requirement, the same defect becomes more expensive to fix. If that same problem requirement is found during testing, then more people are involved and the rework consumes more company resources. This effect, called “Quality Debt,” is accumulated all the way down to where the final customer is impacted.

The purpose of a quality program is to create a framework that enables employees to find potential problems within the same step in which they occur, instead of falling back to “We’ll deal with that problem later.”

**The “Stable Proposition”**

As organizations become exposed to mature quality concepts outlined in the Stable Framework™, a natural resistance often materializes from people new to risk management. “All that stuff will slow us down.”

While this claim may be applicable to any particular step along a value stream, Stable proposes the time invested upfront by the team managing risk will prevent unwanted mistakes made along the way, saving the team from unscheduled problems and the ensuing time wasted downstream dealing with the Hidden Factory. A little bit of upfront care saves the team from a multitude of downstream grief. This is the “Stable Proposition.”

Applied properly, the percent of Hidden Factory efforts in Stable processes should be reduced down to single digit numbers, or eliminated completely.

**Quality Competency**

The “Stages of Competency” model is a tool developed by Noel Burch in the 1970s. The model suggests awareness emerges in these four stages:
A. Individuals are initially unaware of how little they know, being unconscious of their incompetence.

B. As they begin to mature and become aware that tools and methodologies are available to help them achieve their goals, they recognize their incompetence and gradually become conscious of their incompetence.

C. As they begin working toward and showing improvement, they become consciously competent.

D. After much learning, experience, and success, they master the skill and reach unconscious competence...a Zen-like state.

Most organizations I visit are in need of a simple quality program. In the Stages of Competency learning model, they are in quadrant A or B. These companies are often unaware that patterns exist to transform their chaotic work environment into a stable work environment where they can operate with fewer mistakes, and therefore, must less waste.

**Stages of Competency**

![Stages of Competency Diagram]

**Figure 4 - Stages of Competency**

Chaotic environments allow problems to grow undetected until they reach crisis mode and then require a hero to solve. Frequent heroic acts,
as well as a recurring need for hot-fixes, are tell-tale signs of a chaotic environment.

Chaos can exist even among highly skilled people. Like the skilled Air Force test pilot, there is so much complexity in today’s IT environments, that it is impossible for even a well-trained professional to repeatedly get everything right.

In contrast to a typical, rushed chaotic IT environment, picture your people looking at a performance console, reporting that there are no emergencies to fix. Picture them looking at performance indicator charts that show all service level agreements have been met and customer satisfaction trends are up. Picture them discussing remaining variances in their standard work results and working daily as a team to improve performance. Picture an orderly, clean environment where the team is learning and improving together. What you’re picturing is an out-of-the-box quality program called the Stable Framework™.
Customers want value. They want a solution to solve a problem they have, or to lighten a burden in their environment, or to improve their work and their lives. Businesses provide these solutions in the form of “Value Propositions.” It’s critical to understand that value is defined by the customer. It doesn’t matter what the business thinks is valuable, value is defined entirely by their customers.

The Value Proposition

There are many attributes a customer considers when evaluating a value proposition from a potential supplier. Here are some common considerations:

Utility Value is a measure of how well the product or service matches the intended use by the customer. The safest way to ensure strong Utility Value is to involve the customer in the concept, design, and acceptance testing of a product or service. Deming taught us that it doesn’t matter what we think is valuable, it’s what the customer thinks is valuable that sells.

Warranty Value is the invisible trust associated with the seller of the product or service, and the customer’s impression of the seller’s ability to fix or replace that product or service upon failure. If the price is the same, it’s always better to buy from a larger company or seller who will still be in business next year and can repair or replace the item for you.

Durability is a measure of how well a product or service withstands use. A proper quality program helps an organization build a robust or durable product and build it right the first time. Through sustained continual improvement the product or service is made to be more robust over time.

Price is what each customer evaluates, in terms of total cost, while contemplating whether to buy a product or service. Customers want to solve a problem without the risks or time involved with creating a solution themselves. In the end, they want to feel that what they purchased was more valuable to them than the money itself.

Accessibility means how easily a product or service can be acquired.
Customers will pay more for a product or service that is easier to obtain and implement.

A quality program helps:

- Develop utility value by engaging customers to identify what basic needs and improvements are most urgent.
- Improve the durability and stability of any product or service offered by continually improving past problem areas and developing and maintaining process controls.
- Reduce costs over time by minimizing rework, scrap, unwanted delays and unwanted variations in a product or service.
- Improve accessibility by increasing the predictability of release schedules as a result of minimizing variations within the system.

**Systems Thinking**

Deming taught us to think in terms of the whole system and to understand how variation works within it. He taught that any change made to a component or process step potentially impacts the whole system it is in, and possibly the external systems interacting with it. He emphasized that the entire system must be understood before making changes to it.

Deming taught about systems using the preceding diagram. He explained that a system has a starting trigger, input materials and suppliers, internal process steps that add value to the materials, product or service outputs, customers, and a customer feedback loop.
A healthy system has a design and re-design component which adds value based on continual customer feedback, and continual sustained process improvement.

**System Interactions**

Systems interact with other systems. They often contain subsystems, and likely are a part of a supersystem. A domain of all systems considered is called an ecosystem.

An ATM system, for example, services a customer walking up to it and inserting his or her card to withdraw cash. The customer is the consumer system, and the bank computer containing the account information is a supplier system. The ATM is a system providing a service to connect the other two systems, any time of day. The PIN number authentication process could be thought of as a sub-system, and the entire bank corporation as a supersystem, or even an ecosystem.

It’s important to realize a system can’t understand itself. One must be outside of a system, talking with a customer to understand what the system needs to do. Fish in a fishbowl don’t know they are in water. Humans thought the Earth was flat until they began studying the cosmos and triangulating external coordinates. You must get a perspective from outside of a system to understand it. This is a fundamental principle in system analysis and design.

**The Two Vectors of System Excellence**

Deming taught that a system must be continually improved. There are two vectors that govern system improvement. The first system improvement vector is how our system can adapt to better match the expectations of our customers. The only way to accomplish this is to communicate with them relentlessly. Customer needs will evolve over time, so this communication and adjustment process should be ongoing.

After the customer’s needs are being met, the second system improvement vector is to maintain the targeted level of service, but to deliver it faster, with less internal effort, which lowers costs over time.

To facilitate these vectors, consumer research from customer interactions
must flow into a design and re-design system component, continually improving the output value for the customer, or reducing the effort, time, or cost to the business while working the system.

We call the first vector effectiveness, and the second vector efficiency. Together, they represent the pursuit of excellence, and thus comprise the two objectives of Operational Excellence.

**Mapping a System**

To understand the components of a system, we must map the system. There are three approaches to mapping a system. The first is to use a S.I.P.O.C. Model to identify the components of a system. The S.I.P.O.C. Model shows the Suppliers, Inputs, Processes, Outputs, and Customers of a system. This can be useful for general discussions and evaluations.

You can describe an entire value stream using this diagram. In this case, we are describing an on-site software implementation process. It may be that this level of detail is enough for documenting a process or a system you are modeling.

Modeling a value stream in this manner allows new team members to quickly understand all of the players involved with a process on which they are being trained.
System-Flow Diagrams

The second approach to mapping a system is by using a flowchart. This is useful if you need more detail to describe the asset movement, chain-of-custody, decision points, or division of labor within your system.

To create a flowchart, or “System-flow Diagram,” we start by identifying the trigger. The trigger is the starting point for a process. Common examples of triggers are new sales orders, pull requests, new Service Desk tickets, end-of-business-day cycles, or end-of-month cycles.

Once we've identified the trigger, we then outline the process steps needed to fulfill the request. In the previous diagram you can see a Software Implementation Process Map. This map is organized by horizontal lines, called Swim Lanes, which indicate who is responsible for each step, and then vertical lines, which represent phases of the process.

In this example, there are two process phases that occur sequentially to complete this value proposition in this system. The first phase, “Phase 1: Site Planning,” is triggered when the sales contract is signed. The second process, “Phase 2: Site Implementation,” is triggered when the on-site customer manager approves the implementation plan. You can think of the transition point between these two phases as a Phase-gate.

The value delivery for this system is complete when the on-site customer manager signs off the completed implementation.

Systematization

In order to be sure these processes are performed correctly, we’ll establish Standard Operating Procedures and checklists for each process step.

A Standard Operating Procedure describes a process and should be as simple as possible, because IT tends to evolve quicker than other disciplines. It’s important to indicate the necessary steps, but keep descriptions as brief as possible. The target reader should be someone skilled in the art, but new to the environment. Diagrams, simple bullet lists, SIPOC Models, or even photos or video instruction will work, too.

Think of checklists as certificates of proof the process was performed correctly. They are not instructional checklists so much as they are risk
Figure 7 - Software Implementation Process Map
management checklists, indicating the required steps have been taken, and therefore all known risks have been checked before this work item can move on to the next step in the process.

**Value-stream Maps**

If you want more detail about where your system contains waste, a third option is to create a Value-stream Map. A Value-stream Map is like a System-flow Diagram except that each step contains additional timing, dependability, and value-add information.

For each step we’ll assign the typical Step Time (ST), Wait Time (WT), Roll Through Percent (RT%), Availability Percent (AV%), and indicate if the step is considered a Value Add (VA) to the customer, or not. A description of each of these terms follows:

- **Step Time** (ST) is the typical time each step takes. Anybody who performs that step regularly will know this number.
- **Wait Time** (WT) is how long the employee waits before taking that step. It may be zero, or it may be several days because they need to collect information or wait for the right time to begin.
- **Roll Through %** (RT%) is how often the step is successful the first time. The goal is to get every process step performed correctly the first time. We call the percentage of times the step has to be repeated over time “non-rolled throughput yield.”
- **Availability %** (AV%) is how often all the assets required to perform the step are available to you, compared to how often they should be available to you. This could be people, equipment, or information.
- **Value Add** (VA) meaning does the step add something the customer would pay for?

Note that we’ve established a standard unit of measure—days—and indicated it on the map. This map helps to identify obvious areas where improvements can be made throughout the value stream.

Notice that most steps have an upstream supplier and a downstream customer. Upstream suppliers represent anyone executing the process step just before yours. Their work may be to gather data, or something more tangible, and give it to you. We call these “Supplier Processes.”
Figure 8 - Value Stream Map
There are several customers in this scenario. There is, of course, the customer at the end of the value stream. There is the employee at the next process step, who is also a customer. A third customer is another employee who may take over the job when you are absent or leave the company. And finally, if you work in a regulated industry the industry auditors are an additional customer. Any of these processes receiving assets or information from your process we call “Receiver Processes.”

We’ll now take a closer look at interacting with upstream suppliers and downstream customers.

*The Upstream Supplier*

Upstream suppliers provide the materials and information to you, which you need to complete your work. This makes you their customer. We call this relationship between you and all your suppliers the Supply Chain. Managing the quality of these incoming assets is an important step in a flowing process.

Supply chain interactions go through three evolutionary phases. In the first phase, you must check everything being received to be sure it is correct. Any inaccurate receivables must be returned for rework.

In the second phase, the upstream supplier should be taught about quality, and how to apply quality concepts to deliver improved goods. Eventually, when the quality levels of the incoming deliverables become consistently acceptable, and the completed quality checklists arrive with the deliverables, only spot-checking is necessary.

The third phase is the most streamlined. This is where the supplier’s quality standards are proven consistent enough that the incoming packages don’t need to be quality checked. At this point they just need to have the completed quality checklists arrive with the deliverables, as proof of supplier quality control. It takes time, but this state is what you want your supplier relationship to mature into. If a failure detected in production ends up being attributed to a phase three supplier, you must restart the process at phase one again.

All these activities fall under the Incoming Quality Assessment (IQA) efforts. It’s important to nurture the relationships with your suppliers so that you can trust each other and help each other.
Deming often asked companies these types of questions:

- “How have you helped your suppliers better cater to your needs?”
- “Have you shown your suppliers how you use their products?”
- “Have you shared your quality practices with your suppliers?”
- “Have you passed customer recommendations on to your suppliers?”

The Downstream Customer

When interacting with the downstream customer, whether it be the business customer at the end of the line, or simply the next employee receiving your work package, you should always be interested in sampling the customer’s satisfaction level.

There are several dimensions to customer satisfaction that have been refined through extensive research in the 1980’s and 1990’s. The SERVQUAL model provides these five dimensions of customer satisfaction that you may want to include in your employee training and customer surveys. These five dimensions have the greatest impact on the customer experience as they consume a service:

- **Reliability**: the ability to perform the promised service dependably and accurately.
- **Assurance**: the knowledge and courtesy of employees and their ability to convey trust and confidence.
- **Tangibles**: the appearance of physical facilities, equipment, personnel, and communication materials.
- **Empathy**: showing consumers that you care about them; individualized attention to customers.
- **Responsiveness**: the willingness to help customers and to provide prompt service.

When building any kind of customer satisfaction questionnaire, these five measures are foundation attributes.

Other factors affect the customer relationship. Propinquity, or how personable you are with your customers, will increase the emotional connection to your brand and company. Attunement, or your listening position, also impacts the relationship. Are you rushed and sharp with
your customers, or are you warm and friendly and genuinely interested in how well they are being taken care of?

In addition, two excellent feedback questions to ask customers are:

- “How would you rate your most recent experience with us?”
- “How inclined are you to recommend our services to others?”

These small differences become competitive advantages. Be sure you provide feedback to Engineering and Product Management, as needed.
Chapter 4 - Six Forms of Process Improvement

A simple value stream may have only one repeatable process, but more often a value stream is a chain of repeatable processes. In these scenarios, value flows like a river through multiple processes cascading from supplier processes to receiver processes. Each supplier process produces a deliverable that is received by a receiver process at a checkpoint, added to, and then handed off to a receiver process at the next checkpoint. There is always information flowing along with each deliverable. This information holds the key to process improvement.

Value Realization doesn’t happen until the final process finishes at the end of the value stream, and the external customer receives and pays for the final deliverable. This is an important concept. All the effort going into a value stream is either an investment or waste, depending on whether the customer receives the product and pays for it. If time and effort are put into a deliverable, and the customer never receives it, it was waste. If the customer does receive it, that time and effort was an investment.

The key to streamlining a process is to maximize those elements that are of most interest and value to the customer, and at the same time, minimize the time, effort, and cost it takes to deliver those elements. Improving performance over time is the key to lowering costs. Don’t spend too much time looking for lower-priced suppliers and lower priced materials. Your big savings will be found in improved processes.

We call the ability to deliver unimpeded value “Flow,” and we call the improvement efforts performed by everyone within the company to reduce time and costs involved “Lean.” In addition, non-conforming process issues will arise, and we’ll need to discover their root causes, triage them as fast as possible, and then mistake-proof them so they don’t recur.

After we are sure we are producing the right product for the customer and producing it correctly, we can work on producing it faster and cheaper by using better ideas, and by minimizing the Hidden Factory.

You can’t manage all of this correctly until you first systematize the process. How will you know the process is improved if you haven’t mapped it out or measured its previous cycle times? How will you preserve a better idea long-term if the improvement is not written down?
How will you save the time spent identifying the root cause for an asset failure going forward?

There are six different approaches to improving a process, all of which contribute to its overall capability. They are listed here:

1. Systematization: Process Modeling and Control
2. Measures: Defining Process Metrics and Service Levels
3. Flow: Identifying and Eliminating Bottlenecks
4. Lean: Identifying and Eliminating Waste
5. Resilience: Addressing Issues as They Happen
6. Durability: Mistake Proofing the Process

All these areas require innovation and sustained continual improvement, or Kaizen. We will examine each one below.

**1. Systematization: Process Modeling and Control**

A process must first be modeled, or systematized, before it can be sustainably improved. If an improvement is made on an un-systematized process, the improvement is only an anomaly that will soon be forgotten as soon as a new operator is in place. For this reason, processes must be systematized to be continually improved. Systematizing a process means to create a proper structure around it, and then to document this structure in a version-controlled repository. The knowledge created in this manner is called “Process Controls.”

The challenge in IT is that technology changes so rapidly, driving industry changes, that internal IT processes end up changing more often than most other industries. For this reason, documentation should be minimal. Use diagrams and drawings instead of long pages of text, where possible. When modeling your processes and associated process data, these attributes should be identified:

1. The process name.
2. All triggering events that start the process.
3. The process value propositions, or deliverables of value.
4. Which value-streams the process serves.
5. Which customers the process serves.
6. Which suppliers the process requires.
In addition, each process should be listed in the Process Controls section of the Configuration Management System governing your area, and should contain the following components:

1. A Standard Operating Procedure (SOP) preferably in diagram form.
3. A Process Recovery Model.
4. Links to Asset Recovery Models for the problematic assets within the process.
5. Scrap, rework, and cycle-time metrics.
6. A Service Level Goal or Agreement that can be measured.

If you don’t have a Configuration Management System, or Configuration Management Database, you’ll need to establish one. Think of it as a central repository for storing process and asset knowledge.

**Initial Setup**

The initial setup happens during Quality Planning, when the initial quality program is being established in an environment. During this initial setup, and periodically afterward, the following should happen:

1. Be sure all repeatable processes that need quality protection have been identified and added to the Process Controls.
2. Be sure all process SOP’s are up to date.
3. If necessary, diagram complex flows using flowcharts or sequence diagrams.
4. Add new risk items to process checklists.
5. Improve existing steps so there are fewer mistakes.
6. Innovate better ideas to save work and time.

It is convenient to store this Process Control information in the Configuration Management System (CMS). The CMS should be centralized and easy to access. Quality Planning should be revisited and brought up-to-date every six months or so.

To Systematize an existing process, we’ll use the Standardize-Do-Check-Act (SDCA) model. SDCA is based on the more popular Plan-Do-Check-Act (PDCA) model which Walter Shewhart created, and W. Edwards Deming wrote about in his books. While PDCA is a pattern for
continually improving a standardized process, SDCA is a pattern for standardizing a current process that has not yet been modeled.

SDCA was introduced in Masaaki Imai’s 1997 book *Gemba Kaizen*, as a tool for standardizing a process. The circular pattern works like this:

![Figure 9 - Standardize-Do-Check-Act](image)

1. Standardize – Create the process model.
2. Do – Execute the process (perform one Process Kata).
3. Check – Check the process model against the process steps for accuracy.
4. Act – If the model is not accurate, improve it.

This pattern repeats until the Operator is confident that the model reflects the repeatable parts of the process accurately.

According to Mike Rother’s book *Toyota Kata*, Toyota added “Go and See” to its PDCA and SDCA cycles because the automaker found this to be a critical step. Although it’s really part of “Check,” Toyota learned it is critical to go to the customer, or to the work location, and see the results in their natural state.
System Learning & Improvement

Once processes have been systematized, they must be kept up to date and improved over time. Defined processes are supposed to represent the best-known way to perform process actions. If a team comes up with a better way to perform a process, the improvement should be adopted, and the process model should be updated to represent the improvement. The whole point of standardizing a process is to preserve institutional knowledge while ensuring consistency and sustainable improvement.

In addition, process reviews should be scheduled every six months or maybe once a year as part of regular Quality Planning to be sure they are accurate.

To improve an existing process, we’ll use the Plan-Do-Check-Act model shown below.

Shewhart and Deming used this model to gain what they called “Profound Knowledge” about how their systems worked. A system of Profound Knowledge is a system that is understood by its operators.
For example, Starbucks Coffee owners have learned if they periodically have the sound of coffee grinding in the background, patrons in the store buy more coffee over time. This is a beneficial input to a predictable system they have come to understand. You could say, they have profound or insightful knowledge that this technique works because they have previously experimented with it.

Deming said operators should experiment with different theories until they have converted their theories into profound knowledge of how the system reacts. This empowers them to tune each step for increased efficiency within their processes.

Deming taught us that these four steps are key to understanding a system:

1. Thinking of all the moving parts as a system.
2. Understanding the amount and range of variation occurring within the system.
3. Knowledge of the causes of variation within the system.
4. Understanding the psychology of people working within the system.

He calls his model the SOPK, or System of Profound Knowledge.

The approach is to take a hypothesis like the coffee grinding idea mentioned above (Plan), try it at several locations (Do), and then measure the results (Check). If the results are consistently favorable, start grinding coffee every few minutes in all the store locations (Act).

![Figure 11 - System Learning Using PDCA](image)
Now, you could adjust the frequency of the coffee grinding over time and find the best tradeoff between grinding frequency and purchasing trends. In a case like this, you would really be performing the PDCA cycle multiple times and finally selecting the optimized results. This is called ‘System Learning” and the pattern will look like the previous diagram.

**Non-conforming Output**

If a defect escapes and is found by a customer, the organization is faced with three possibilities. A specific new external problem may have caused the defect to occur, and after a root cause analysis is conducted to determine the reason, the process can be protected against the cause by removing the vulnerability completely, or by adding a “watch out for” item to the process checklist. We call these “Special Causes.”

The second possibility is that natural variation within the process created an unexpected result. These types of problems usually require solution rework from the engineering group. We call these “Common Causes.”

The third possibility is that the operator did not fully understand an item on the checklist and checked the item off without completing it. In this case, the operator will require retraining with the process controls. This retraining is an opportunity to re-read the Standard Operating Procedure and to update it and the process checklist for clarity, if necessary. This re-training should be recorded in a historical training log.

**2. Measures: Defining Process Metrics & Service Levels**

If it’s not measured, you can’t improve it. At least, not sustainably. Not everything needs to be measured, but for processes in need of improvement, the starting point is measurement. Measurement is a doorway into reality and dispels bias. Measurement is a great way to make a persuasive argument and simplify disagreement.

Metrics should be a visible part of the organization’s culture and should be used to trigger brainstorming sessions with positive dialogue for better ways to improve existing practices.

Historically in Western culture, measurement has been a way of
conditionally evaluating employees. This has produced much anxiety around metrics in the workplace. Care must be taken to reinforce the message to the group that the purpose of measurement in a quality program is to connect us to reality. With accurate measurements, we can work together to brainstorm ways to get more work done faster with less effort and resources.

Process Metrics

One simple rule for measurement is that the results should be similar regardless of who performs the measuring.

There are several natural attributes that can be measured for almost any process. They are:

2. Customer satisfaction with the process results.
3. Rate of customer consumption.
4. End-to-end process cycle times.
5. End-to-end process costs.
6. Amount of scrap and rework over time.
7. Post-sale warrantee losses over time.
8. End-to-end process capability: Did it work every time?
10. Root causes of failures resulting in scrap or rework, using a Cause and Effect Diagram.
11. Frequency and priority of failure causes resulting in scrap or rework, using a Pareto diagram.
12. History of the triggered process, customer, or market events, using a Histogram.
13. External Impacts on process outcomes, using a Scatter Plot Diagram.

In some situations, measurements can be made more useful by combining two or more measurements. For example, measuring customer adoption rates can be insightful, but measuring customer
adoption surges after specific marketing campaigns can be much more insightful.

Reporting the number of bugs found is not as helpful as combining that number with the percentage of modules having accompanying test cases, and the percentage of test cases completed.

The various types of charts most helpful for each of these measures are described in the last chapter of this book titled “Data Collection and Analysis Tools.”

This is only a sampling of common process attributes that can be measured. Care must be taken to identify a business need for the measurement, and only spend time measuring process attributes that require optimization. In other words, don’t spend time measuring something that has no business value.

Service Levels

A Service Level is a desired measurable level of performance achieved by a process. It may be as simple as indicating if a process has completed successfully, or it may be something complex like measuring the uptime of a service during defined business hours per year.

If there is no formal contract with a customer, we call this target a Service Level Goal (SLG). If a formal contract with the customer exists specifying a level of performance, we call it a Service Level Agreement (SLA). In addition, an organization may create an internal Service Level Agreement for a service offered with a publicly promised standard.

3. Flow: Identifying and Removing Bottlenecks

Flow is the lack of resistance to movement throughout the value stream. Good process flow enables better predictability, less batching of work, and continual delivery to the customer.

Flow is important within a value stream so that customer orders do not get delayed. If customers are delayed too long, they will go elsewhere for business.
Often the workflow must be completely restructured to increase the flow of information and assets through an organization. Modern manufacturing facilities are designed from the shipping bay first and back to the materials input bay. People that were originally organized into groups of roles become re-organized into cross-functional teams containing each role necessary for the work to flow smoothly. This restructuring makes it easier for flow to move through the organization and is a huge competitive advantage.

Think of the old pin-ball games. You want value flowing through your system directly from supplier to customer, instead of hitting all the bumper stops and bouncing back and forth, from silo to silo. This direct flow requires a new focus on cross-functional teams, instead of traditional, functionally siloed departments.

Recently, I was talking with a group that was asking for better tools for Operations, and DevOps. I told them they needed better team structures, not just better tools.

Minimizing WIP with One-piece Flow

Also called “Continual-flow,” or “Single-piece flow,” One-piece flow is a work configuration accommodating the complete construction of one sellable item from start to finish with as little work-in-progress inventory accumulating between operation steps as possible.

Contrast this with batching multiple production items during construction. In a One-piece configuration, each single item is ready for sale much quicker than the entire batch will be. In contrast, items for sale in the batch must wait for every other item in the batch to be completed before any of the items can be sold. This batched clump we call Work-in-Progress (WIP), and is unwanted. Waiting on WIP ties up resources, capital and delays sales income. In IT this is true for Portfolio Management, Development, DevOps, and Implementation.

The Theory of Constraints

In 1992 Dr. Eliyahu Goldratt wrote a ground-breaking book called *The Goal*. In his book, he tells the story of a plant manager frantically trying
to make a failing plant become profitable again. He tries a lot of conventional ideas such as maximizing resource utilization and running double shifts but quickly finds out these techniques cause more problems than they solve.

He finally discovers the real “goal” is to examine the entire value stream and identify the slowest bottleneck in it and then fix it, continually repeating this cycle until work can be completed at the same rate of customer consumption.

He presents the idea that the cost of the bottleneck is the cost of the entire enterprise idly waiting for each work item to move through it. He calls this the “Theory of Constraints” (TOC) and provides a 5-step plan for improving workflow within a value stream.

Five Steps for dealing with Constraints:

1. **Identify the biggest constraint:** Think of this as strengthening the weakest link in the chain. A focus on other links is not as beneficial while this weaker link exists in its weakened state.

2. **Exploit the constraint:** This means to do everything possible to minimize the effects of the constraint, preferably using available resources and taking care to not spend more money in the beginning to solve the problem. Be sure the step containing the constraint has priority with work items and attention so that it is always executing, and its slowing effects are minimized. Also, be sure that as much quality control as possible happens before the constraint so that known defective items do not get sent through the expensive constrained step, only to be reworked later.

3. **Subordinate everything else to the constraint:** Line up all dependent activities and have them timed in production with the constraint, and no faster. Producing more upstream incoming work items than the constraint can handle creates waste because these intermediate components will have to wait their turn through the constraint.

4. **Elevate the constraint:** Once the capacity of the movement through the constraint has been maximized using the steps above, the next logical step is to invest additional resources into expanding the constraint. It may require more people, or tools, or money. Be careful not to jump from step 1 to this step, skipping
steps 2 and 3. A lot of lift can come from steps 2 and 3 before needing to spend additional money on the problem.

5. **Repeat steps 1-4:** Repeat these steps until your rate of production matches the rate at which your customers consume your products or services.

I remember one client explaining to me how his or her team of 50 programmers would wait as eight QA testers spent five days regression testing the entire system, every three weeks. Supposedly, the developers were busy with new work during each third week. In reality, their production was somewhere around one third of their true capability during this time, as they waited anxiously for the inevitable bug reports to be given back to them.

This was a classic TOC problem. Regression testing was the biggest constraint (step 1). One solution was to have the developers peer test each other’s work before it was sent to be QA Tested (step 3). This would eliminate a lot of the rework coming back from the testers. The second part of the solution was to elevate the constraint (step 4) and have all 50 developers become testers for 1 day (steps 2 & 4), reducing excess input (step 3), and shortening the regression testing process from five days to one.

### 4. Lean: Identifying and Removing Waste

Lean Manufacturing principles have been around for about half a century and are still being adopted aggressively by manufacturing facilities. Lean, in a sentence, is identifying and eliminating waste within a value stream. Every activity within a value stream either adds value to the finished product or it doesn’t. If it doesn’t then it contributes to the Hidden Factory of waste, called “Muda” in Japanese. There are eight types of activities that consume resources of some type but do not add value to the final product. Together they spell the acronym T.I.M.W.O.O.D.S.

Eight contributors to the Hidden Factory of waste (Muda):

1. **Transport:** Transport is the time delay waiting for an asset as it is moved from one location to another. In modern warehouses, this is easily identifiable by the elaborate conveyors spread out in some facilities, and the carting needed to move items around in them. In IT this can be physical transport encumbrances such as shipping a server to a customer site, but more often transport waste occurs
CHAPTER 4 - SIX FORMS OF PROCESS IMPROVEMENT

when unnecessary knowledge transfer is needed. Employee attrition, off-shore development, filling someone in on details because they were not part of a discussion, or transporting requirements onto paper to be read by a design team that was not at the discussion are all examples of transport waste in IT.

2. Inventory: Inventory costs money. It costs to rent the space to store excess material. Fires, theft, and revisions can destroy inventory. There is a natural law that says higher inventory levels conceal problems. If there is a large supply of needed parts, nobody will spend the time identifying just how many are required to meet actual demand. In this case, efficiency is lost. In IT, a heavy stack of requirements is inventory waste. They will rot away into irrelevance over time, as the customer’s needs change.

3. Motion: Any human motion not directly related to adding value is unproductive. A person walking or reaching beyond his or her workspace to obtain a tool is creating waste. For this reason, the more frequently a tool is used, the closer it should be to the workers reach. In a knowledge-working environment such as IT, think in terms of how easy information can be seen. What type of screen layouts do the team members have? I used to think writing code with dual monitors was a treat until I tried using eight monitors simultaneously. That changed my life.

4. Waiting: Waiting happens when the hands of the operator are idle such as in meetings, while waiting for decisions or pending approvals, or when needed information is uncollected or ambiguous. Waiting on requirement elicitation or clarification is a form of waste.

5. Overproduction: Overproduction means doing more work than is necessary right now. In IT, this is often called “gold plating” and can be described as adding features that no paying customer requested, or doing additional activities that are not required. This can also be making decisions without the proper representation of stakeholders present, which often ends up causing rework or scrapped work when the other decision participants become available and have different ideas.

6. Over Processing: Waste in processing can happen when processes take an unnecessarily long time to complete, or if two processes could happen in parallel, but for some reason don’t. Task
switching is time wasted moving between two activities and is a form of over processing.

7. **Defects**: Defects interrupt production and require rework or replacement.

8. **Skills**: Under-utilizing the capabilities of your people, lack of teamwork, or inadequate training all represent skill waste.

### 5. Resilience: Addressing Issues as They Happen

In addition to Flow, and Lean techniques, Process Issue Management is a critical part of building a robust process.

When problems occur, and they will, the operator needs to get the process back on track as soon as possible. In order to do this, we use Process Recovery Models, and Asset Recovery Models, as part of the process controls.

A Process Recovery Model is a list of known problems that have occurred in the past with the process. It contains a description of the problem, a root cause, and a solution to recover from the problem. This enables operators to quickly be able to determine if a problem that was just triggered has a known solution. This saves the time needed to diagnose and come up with a solution when someone in the past has already spent time doing that. Process Recovery Models should be easy to access and readily available.

Asset Recovery Models are the same as process recovery models except they are based on an asset within the process, which may be used by other processes.

An example would be an application server used by a call center. Occasionally a clustered server may failover to the secondary node, alerting the operators that something is wrong.

In a situation like this, the operator would be trained to go directly to the Configuration Management System, obtain the Asset Recovery Model for the server, look for this particular problem, and see what root causes were found in the past to explain the server’s state. Remediation steps should also be listed, to enable the operator to quickly fix the situation and get the server back on its primary node.
When a new problem is encountered, a root cause analysis would be conducted, and when the source of the new problem is identified its description and remediation steps would be added to its Process or Asset Recovery Model.

Root Cause Analysis techniques are covered in the Process Owner chapter.

**6. Durability: Mistake-Proofing a Process**

W. Edwards Deming said that management should stop blaming employees for process issues. Instead, the burden of mistake-proofing a process should fall on the management team. The management team should work with the operators to brainstorm ways to prevent the same mistakes from occurring in the future.

**Elimination**

Eliminating unnecessary or error-prone steps is a great way to reduce errors. Work with your supplier or select a different sequence that could result in a simpler set of required steps to produce the same results with less effort.

**Simplification**

The reduction of variation is always a good idea. Variation, by itself, creates overhead of some kind or another. Sometimes that overhead becomes an exponential resource drain.

Many companies have discovered if they can reduce internal variation, they can become more competitive. In IT, a simplified set of database platforms, programming languages, and architecture components means fewer skills are needed for development and support activities, for example.
Automation

In IT, we often automate tasks to remove manual redundancy or protect the process against human error. Many sophisticated and inexpensive automation tools exist today. Automation, where practical, is always a good idea.

Acceptance Screens

An Acceptance Screen is a mistake-proofing tool that prohibits an asset from proceeding through a system and onto the next step until it is structured correctly. Toyota called this “Poka-Yoke” which is a Japanese term that means “Mistake Proofing.” It was made popular by Shigeo Shingo and is a helper-mechanism made to reduce or eliminate possible mistakes.

The plastic shim on the inside half of a USB drive, for example, prevents a user from inserting the drive into the USB slot incorrectly. Many ATM machines require you to take your card back before they dispense cash.

Bank software applications use a check-digit technique to dramatically lower the probability of errors when typing in long account numbers. A check-digit is a number added to the end of an account number which is calculated from a simple math formula applied to the account number. Thus, the receiving computer having the same standardized formula can perform a check-digit calculation and compare the result to the digit at the end of the account number to determine if the entire number is a valid entry. All your credit cards have standardized check-digit formulas that are published.

Acceptance Screens can reduce the Hidden Factory without requiring much cost to implement. Shigeo Shingo, while working at Mitsubishi discovered that using Acceptance Screens where possible significantly reduced the need for statistical process control, which reduced the need for measurement and analysis.
Chapter 5 - The Components of a Quality Program

My friend John Lee wrote a book called *Rising Above it All* which won the Shingo Research and Professional Publication Award in 2016. In his book, John talks about two different approaches to achieving a high-quality process.

The first and most common approach is to hire competent people and then pressure them. At best you might achieve something like a 95% process capability. Process Capability means how many times a process is executed perfectly, with no rework, scrap, unwanted variation, or unexpected delays. A Process Capability of 100% means every step of the implementation is successful the first time through. This requires people with competence to get it right as much as humanly possible, and tough skin to take the scolding when things don’t go right. John calls this PBP, or Personify, Blame, then Punish. You will get pretty good results with this technique as you churn through good employees, but you will never be world class.

The second and more effective method is to create a systematized process to govern the work performed, and then have everyone use the process, and keep improving the process until your capability is more like “five 9s.” Five 9’s means your system is working without problems 99.999% of the time. This is how companies achieve world-class process capability, and world-class service provider status.

A systematized quality program is like a symbiotic helper. The employees and management work together to continually improve the process, and the process reciprocates by improving the performance of the employees and management.

*What is a Systematized Quality Program?*

It may help to first define quality. As a word, quality means the degree of excellence of something. Of course, the mark of excellence is open to interpretation. Deming taught us that quality must be defined by your customer. Therefore, a simple way to think about quality is to produce something that meets the needs of your customer exactly. Remember that you often have multiple customers, and that their needs constantly
evolve. Part of the challenge of defining quality is being sure that your understanding of your customers’ needs is kept current.

A quality program is a collection of structures created to enable imperfect people to sustainably produce quality products and services. A quality program manages input quality, throughput quality, and output quality. We call these three areas Incoming Quality Assessment (IQA), Quality Assurance (QA), and Quality Control (QC), respectively.

There are two other components called Quality Planning, and Sustained Continual Improvement. Quality Planning is where we set up the whole program, and Sustained Continual Improvement is a set of mechanisms we put in place to improve the products, services, and our ability to deliver them over time. We’ll discuss Quality Planning first.

---

**Figure 12 - Quality Management**

**Quality Planning**

Quality Planning is the systematization of a work environment. It starts with identifying the value propositions offered by a system, and then identifying all the services that makeup the value stream required to deliver those value propositions. It includes the initial identification of the Master Services offered by the environment, and the processes, triggers, inputs, outputs, customers, and suppliers needed to facilitate the services. In addition, the specific metrics that represent goals important to the organization and its customers are identified. All this information gets stored in a Configuration Management System (CMS).
If you don’t have A CMS, you’ll want to start by creating a central repository for storing all your process information. A good Confirmation Management System is version controlled and easy to navigate.

Quality Planning should be refreshed about every six months.

**Incoming Quality Assessment (Incoming Quality)**

An Incoming Quality Assessment, or IQA, is a much-overlooked component in Quality Management and is the first step in minimizing the Hidden Factory. An operator should examine incoming deliverables or information from the supplier and be sure it is complete and everything is available.

If something is missing, the supplier should be notified immediately. It may be that the operator can proceed with incomplete material or information, but both the supplier and the operator need to know that each understands there is an outstanding deficiency and the supplier still needs to provide additional inputs.

In some cases, the input should be rejected, or quarantined until the supplier delivers the outstanding material or information. With ongoing supplier relationships, Deming taught us to develop the relationship so that they are consistently giving you quality material or information that meets your system input standards.

**Quality Assurance (Process Quality)**

Quality Assurance is the proof that the process in question was followed correctly. It is the operators completing checklists during or at the end of their work and then passing them along with the completed deliverables or processes at strategic checkpoints. These checklists become historical logs. Data from these checklists is not used to punish operators for what they did wrong, but rather to provide a glimpse of reality to drive retrospective conversations and brainstorming sessions for how to improve performance going forward.

In time, these processes become improved, and institutional knowledge is preserved. The enabled organization can make better decisions over time,
resulting in fewer mistakes, less rework, and less scrap. This enables the organization to produce the same results with less effort. Having these elements in place ensures the organization is a learning organization. This systematization of process steps throughout the value stream provides a foundation for sustained improvement.

Quality Control (Outgoing, or Product Quality)

Quality control is also known as inspection. It is a way to ensure high product quality. When a candidate item is complete, and before it goes to a customer, it should be inspected to be sure it is ready. Quality control has two steps: verification and validation.

Verification means a completed deliverable is checked to confirm it meets the quality goals it was designed to meet, which would be found in the product or service requirements. If for some reason a completed deliverable fails this test we must rework or scrap it. We pay for this fix and more time and money is lost to the Hidden Factory.

Testing software is verification. When testers find a problem, the engineers are notified and must rework the code. In some instances, the entire feature, module, or even application may need to be scrapped and a new approach taken until verification is completed.

Once an item passes verification, meaning we are confident it has met all confirmed requirements, we show it to the customer for inspection and solicit final approval. This is called validation. If the customer accepts the item, we are finished creating it. If the customer wants something changed, a change request or change order is required to alter the item and the customer must pay for the rework.

In software we call this validation process “User Acceptance Testing,” or UAT.

<table>
<thead>
<tr>
<th>Does it match our requirements?</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation</td>
<td>Can the customer use it as intended?</td>
</tr>
</tbody>
</table>
The same quality control principles apply to any deliverable, regardless of whether the customer is the final business customer or the customer at the next step in the process.

Continual Improvement

The first goal of any organization is to deliver exactly what the customer wants. Measuring customer satisfaction levels and observing customers using a product or service provides immediate feedback on how well the organization is meeting its customers’ needs. The problem is that customers’ needs change over time.

Market trends evolve, technologies improve, competitors emerge, and many other dynamics impact a customer’s reality over time. For this reason, relentless surveying and observation is required to be sure the company is continually servicing its customers accurately.

Some improvements can be made by the suppliers to a value stream. Any suggested improvements to process inputs should be communicated upstream to the process suppliers. This requires a healthy dialog between a process operator and its process suppliers.

In addition, while companies are delivering value to their customers, they must also be reducing the Hidden Factory through relentless continual improvement. Organizations with the smallest Hidden Factory are naturally the most competitive.

Learning mechanism are needed to facilitate these continual improvement trends within an organization. Customer satisfaction surveys, observation, supplier dialogs, measurements, retrospectives, root-cause analysis, experimentation, failure, and learning from the past are all part of continual improvement, and all these activities produce knowledge that must be fed back into our process controls, making the systems more efficient and effective over time.

The Myth of Continual Improvement

The only way an organization can be a real learning institution is by preserving institutional knowledge. Without doing this, the same improvements will be continually forgotten, and then re-discovered by
different operators over time. Some lift is achieved by the discovery, but the information inside people’s minds that is not written down remains as tribal knowledge, and as soon as they leave the organization that knowledge leaves with them. The next operator must start all over again, re-discovering the same improvements, just at a later time.

Therefore, it’s important to distinguish between *sustained continual improvement*—which is the key to excellence, verses only *continual improvement*—which is a myth.

**Sustained Continual Improvement**

Sustained Continual Improvement must be systematized, which means organized into knowledge which gets stored and improved over time. Important information, lessons learned, and process improvements should be stored and updated in the Configuration Management System (CMS). We call this Process Control information.

The CMS must be stored in a centrally accessible and known location so that it can be easily accessed. Many companies augment their CMS with an internal WIKI.

**Training**

Quality programs require training on how the program works, and training on the complex assets the operators will be using while working within each process.

Training should be logged, and periodic retraining may be required every six months or perhaps annually. Employees new to the program should be trained, and anybody caught not following the quality program should repeat the training.

**Non-Conforming Product Management**

Another component in a formal quality program is a list of customer-reported issues with a product or service. In IT, this list is typically created by the Service Desk. The Service Desk should be separating
individual issues from repeating issues and bringing the repeating issues to the attention of the Engineering or Operations teams, as needed.

The Service Desk, if equipped properly, can become a central portal for existing customers to interact with your organization. A good Service Desk can make a poor product passable. A bad Service Desk can make a great product suffer.

While the Service Desk can take care of minor issues and configuration problems, more serious issues should be forwarded to the CCAPA Committee for triage. CCAPA is an acronym for Correction, Corrective Action, and Preventative Action. The CCAPA Committee decides how the identified problems are prioritized for rework and should be staffed by a representative from Engineering, Operations, Implementation, and the Service Desk.

The CCAPA Committee should meet regularly, and in some cases such as just after a major release, they should meet daily.

Incident/Incident Avoidance Register

The Incident/Incident Avoidance Register, often simply called “Incident Register” is a tool used to indicate how much Hidden Factory your environment has, and how much progress your team has made eliminating it. An example exists elsewhere in this book.

Measured in months or quarters, the first part of the register, titled “Incident List” shows any repeatable processes which had to be executed more than once because something didn’t go right the first time. This can be quantified into hours and monetized into an amount per month or quarter of Hidden Factory inherent in the environment.

The second part of the register, titled “Incidents Avoided,” shows every potential problem caught by the quality program, before it could continue down the value stream and escalate into an unscheduled problem.

Certainly, in addition to internal process problems, this list would be augmented by testing process failures and customer-facing incidents from the Service Desk that require Development rework. Using a tool like this, teams are able to measure the amount of environment chaos they are working within and how much benefit over time they have achieved.
by implementing a quality program.

**Measuring the Cost Savings of Quality**

There is, of course, an overhead cost incurred in time and money for performing all these quality activities. It takes time to create Standard Operating Procedures and the checklists, and then review them regularly. It takes time to meet with other busy employees, make appointments and meet with customers.

The alternative is an environment filled with unscheduled problems causing rework, scrap, fires to put out, angry customers, reputational damage, and lost revenue.

Quality Management is not about adding unneeded overhead to your daily work, it’s about trading predictable, pre-scheduled time for not having the downstream unpredictable, un-scheduled time burdens needed to recover from the never-ending circus of reactions.

Teach your people to respect and improve the system. Build a culture that allows for mistakes, reflection, innovation, new ideas, and group creativity for solving problems.

> **Quality Management is not about adding over head to your daily work, it’s about trading** **predictable pre-scheduled time** **for downstream** **unpredictable un-scheduled time** **needed to recover from the never-ending circus of mistakes.**

Celebrate victories together. Generate better ideas and performance improvements as a group. Challenge your team to improve continually. When mistakes are made, don’t blame the employee, instead challenge the individual or the team to come up with a way to prevent anyone in the future from repeating that same mistake.

**Won’t Heavy Process Slow Us Down?**

When W. Edwards Deming first approached American car manufacturers in the 1940’s and told them he had techniques that could double the
lifetime of their products, they dismissed him. They had considered other early industry attempts to manage quality by past failed companies, and rejected the idea that it should be a prominent focus.

In 1949, Deming was asked to help with the joint U.S-Japanese census efforts in Japan after WW II. While there he was able to share his ideas with Japanese manufacturing companies. He stayed in Japan for 25 years helping Sony, Toshiba, Toyota and many other companies rebuild their factories around his ideas. Together, they created a collection of guiding quality management principles we now call Lean.

In the end, they proved the American car manufacturers wrong. They demonstrated they could produce a higher quality product in less time and for less money, and they did it without expensive robotics and technology. The difference was that they had learned the correct way to manage quality.

The same argument surfaces today. Often, technical teams approached with a quality management proposition push back, thinking the process will slow everything down to a crawl.

It is critical to understand that a quality program presents a trade-off in time and resources. Some overhead in time is required to operate within a quality program—minimal if executed correctly—but the benefit of the trade-off is that your team will have much more time freed up going forward due to the minimization of unexpected rework, scrap, delays, and downtime.

You will get some short-lived pushback from newcomers when establishing a systematized quality program, but once they have a few positive experiences from their quality monitoring efforts they’ll catch on and become believers.

**Good Verses Bad Process**

A good process implementation does not provide much resistance to the natural flow of work through a system. Even though checklists are used, and surrendered at checkpoints, the workflow should never be held up waiting for an approval. The key is to make sure the same people making the decisions are the people with access to the information.
By contrast, bad process has a noticeable distance between the decision maker and the information. This is a common source of artificial delays.

One example of bad process occurred during Desert Storm. Some colleagues of mine worked in Iraq as government contractors during the war. They explained that for some approvals they had to suit up, get into hardened vehicles, drive from their base through several known kill zones, and into another base where the higher-ranking officer was available, only to obtain a signed piece of paper authorizing something. Then, the team had to drive back through the kill zones to be able to proceed with the next step in their process. Although this process needed improvement, these men were clearly dedicated.

A Quality Program is Optional

The US auto manufacturers were faced with this decision in the 1940s, and said no. As a result, the Japanese who embraced these practices put the US automakers into bankruptcy, requiring a bailout from the US taxpayers. Today the US auto manufacturers are still going bankrupt, just much more slowly.

A quality program is optional, but not if you want to stay competitive.

My First Experience with a Quality Program

One of the companies I worked for as a software development manager had over 200 sites using our client-server software. It was a complex system with a lot of moving parts, and consequently, we had a lot of unexpected problems continually appearing.

I remember after some time, we started budgeting future schedule bandwidth for unforeseen problems that we didn’t know about yet but were sure we’d have in the future, because we seemed always to have had them in the past. Then, when the next week rolled around, we would congratulate ourselves because we were right—a whole new slew of issues was now on the table to address. These were usually not coding issues; they were implementation problems.

One customer had an old version of Adobe Acrobat and that was messing
up something else. Another customer, somehow, was missing several tables in the database and we had no idea how our system was even working at that customer’s location. With more and more customers, there was an endless river of problems we had to deal with. Finally, our wise CEO said, “I’ve seen these kinds of problems before—we need a quality program.”

That same day he called some consultants who came and spent a year with our company. They went from department to department helping us understand what a quality program is and how to set one up. We identified all our repeatable processes, created Standard Operating Procedures, checklists, and checkpoints, and nothing could advance until it met all the criteria at each checkpoint.

We complained about the extra overhead and discipline required for this level of accountability. Some of our best people even fudged the checklists in the beginning. Our production was slightly slower, at first.

But then, something interesting happened. Our quality program started catching potential failures before our work mistakes ever made it to the customer. We started finding problems in things we would have otherwise skipped over as we rushed on to the next process step, or into production. Suddenly, we realized everyone needed to be upgraded to the latest version of Adobe. Our automated code-build process needed to have a few extra steps added to it. Databases needed to be checked for all tables every time they were upgraded. One by one our problems disappeared and didn’t return.

We started celebrating our victories. We made a list and announced to our entire department every time our quality program protected us from a future problem. We posted this list publicly and had conversations speculating on how much additional time each potential mistake would have cost us.

A short time later, after the effects of our Quality Program began to permeate down to our customer base, we were in a new reality where everything was calm, and we had no fires burning. I remember distinctly asking the others in the department during a meeting if they could think of any problems we were dealing with outside of our current project work efforts. Nobody could name a single problem. We now had much more time to devote to our commissioned project activities because we had no fires to put out. We were able to complete more project work each day.
going forward.

We had transformed our environment from a circus of reactions into a stable, progressive, learning organization.
Chapter 6 - Common Quality Programs

Several popular quality programs exist today. They have been around for some time. They have a fair amount of overlap yet contain some unique qualities. Here is a brief summary of each:

**ISO 9000x**

The International Organization for Standardization published the ISO 9000x series of quality standards for manufacturing in 1987. Since then, the organization has grown and produced many variations of its standard tailored for various specific industries.

At its core, these quality standards are comprised of Standard Operating Procedures, Checklists, Checklist Historical Logs, Non-Conforming Item Registries, and Formalized Training, all of which are tracked for all participating employees.

**Lean**

Lean is a collection of practices that prescribe identifying and eliminating waste within a system and keeping a workplace clean and organized. Taiichi Ohno, who was the chief developer of the Toyota Production System, is credited for the origin of many lean practices.


In his opening paragraph of his paper he wrote, “Assembly line workers perform not only production line tasks, but also quality control and preventive maintenance activities.”

In 2004, John Krafcik became the CEO of Hyundai. While there, he helped the company increase its quality levels to eventually win the 2015 best car brand for quality, according to J.D. Power ratings.
Six-Sigma

Six-Sigma is a set of tools based on Statistical Process Control targeted towards identifying ways to locate, measure, reduce, and eliminate variation within a business. Companies have learned they can save money by identifying points of variation within their value streams and then reduce that variation.

For example, if you can reduce the variation in lead time from a supplier, you can stock less inventory needed to fulfill the same number of orders over time, freeing up valuable warehouse space, and pre-sales capital.

In software, if you track production errors found per module lines of code, you’ll discover an ‘ideal lines of code’ size where production errors tend to be minimized. Going forward, your teams could create a guideline limiting module sizes to that recommended size, minimizing post build errors across the board.

Six Sigma was introduced by Motorola in 1986 and is often coupled with Lean practices. Today it has no central governing body.

ITIL

The Information Technology Infrastructure Library (ITIL) was originally developed by the UK Central Computer and Telecommunications Agency (CCTA), in the late 1980s. It is a set of practices designed for managing assets and the Service Desk in a typical IT environment. It contains reporting tools such as a Service Catalog—showing the status of all services—and a SLAM Chart showing the performance of a service against its agreed service level over time. ITIL also introduced the Asset Problem Recovery Model, a quick-check list of lessons learned from past failures with a particular asset that can be used by operators to quickly troubleshoot a production asset, minimizing downtime.

ITIL emphasizes the practical value of the Service Desk in an IT organization. The Service Desk is described as a central interaction point between customers and the organization and can be tapped to provide product improvement insight and customer satisfaction information.
CMMi

CMMi is the Capability Maturity Model integrated, a product of the taxpayer funded Software Engineering Institute (SEI) at Carnegie-Mellon University in Pennsylvania. It contains five levels of software development process maturity with 125 checkpoints for approval. The origins of CMMi date back to 1987.

Companies can work towards achieving these levels and then becoming certified as a level 5 CMMi software organization.

COBIT

COBIT is a set of generic processes for managing IT. The “Control OBjectives for Information and related Technologies” is a pre-packaged set of recommended best practices for aligning business with IT. Each identified process contains inputs, outputs, and performance measures. COBIT was first released in 1996 by an international professional association focused on IT governance called the Information Systems Audit and Control Association (ISACA).

Toyota Production System (TPS)

The Toyota Production System (TPS) is a set of practices pioneered at Toyota, who in the 80s and 90s impressed the world with its superior automobiles. TPS is really a set of philosophies about building quality into the production process, instead of only inspecting for issues at the end of the line. This is the basic ingredient to Toyota’s success. In the beginning workers check each component against success criteria before it gets added to the assembly line. After they have removed enough variation and the system becomes predictable, they only check one part every so often because it becomes a form of waste to check every single widget of a production line that has consistent precision.

Toyota also pioneered the Just-In-Time practice of requiring suppliers to deliver supplies frequently, just moments before they were needed on the assembly line. In this manner materials were moved directly from the trucks to the assembly lines, eliminating the need to temporarily store
them in a warehouse. The primary concepts in the Toyota Production System are what we call “Lean” in the West today.

**Hyundai/Kai Quality Model**

According to a 2015 JD Power consumer report, Hyundai is now the quality manufacturer of choice in the auto industry. Outperforming Toyota, Hyundai cars have the fewest reported defects over time, and at the time of this writing, are the best cars to own in terms of quality ratings.

Hyundai used the “Sandcone Model” to out-perform Toyota. Using all of Toyota’s very public practices, Hyundai added this one more practice. The Sandcone model is a portfolio management prioritizing guide.

Portfolio Management is the advanced practice some companies use to prioritize which projects are to be funded next.

The Sandcone model states that when deciding which type of project to spend money on, a company should prioritize spending in this order:

1. Customer feedback priorities
2. Product stability and robustness initiatives
3. Decreased production response times initiatives
4. Cost reduction initiatives

Different companies disagree over which order to prioritize those four interests, and there are many debates in academia over which is the best order. After trying several variations, Hyundai chose that one...always listening to its customers first, then stability, then decreased response times, and then cost cutting initiatives. Note that Hyundai was the first car manufacturer to provide USB ports for charging and playing MP3’s. Many other car manufacturers still have not caught on to this.

On a personal note, I was at an auto show in Salt Lake City some years ago and told a Hyundai representative her company had great cars, except the company should take a lesson from the Germans and put the emergency brake in the center console rather than the floor mounted push-button design, which Hyundai had at the time. German car manufacturers do this deliberately because it enables the driver to slow down quickly without activating the tail lights, should a police car
suddenly appear. This tool is part of the ultimate driving experience. Voila! Today, Hyundai has relocated the emergency brake to the center console. I like to think the company listened to me.

*The Stable Framework™*

W. Edwards Deming, in his book *Out of the Crisis*, wrote about a stable framework that companies should create to systematize and continually improve their repeatable processes over time. Such a framework would reduce ambiguity, errors, rework, and scrap, and enable an organization to produce more value with less investment over time.

Inspired by his term, this book explains how to implement the Stable Framework™, which is a light-weight, out-of-the-box quality program designed for easy implementation, while leveraging the best parts of the above programs. Follow the steps in this book to enable the smooth flow of value within your organization.
Chapter 7 - The Stable Framework™

The Stable Framework™ is a performance management platform designed to continually improve the tactical aspects of an IT environment by eliminating all unnecessary waste. While Agile practices exist to facilitate innovation, collaboration, and value-delivery, Stable was created to facilitate the operational components within them. Stable is applicable to Development, Operations, Implementation, DevOps, and any repeatable processes performed elsewhere. Stable can be used alongside an Agile development team, or it can be implemented as a complete stand-alone solution in a more Operations-based environment.

The Stable Framework™ is comprised of one Process Asset Library (PAL), two framework roles, three domains, four core meetings inside of a repeating timebox, five quality principles, six forms of process improvement, and seven steps of systematization. All of these are used to equip adopting organizations with the structure, visibility, and conscientiousness required to achieve Operational Excellence.

One Process Asset Library

A fundamental principle in any quality program is the need to transfer lessons learned from tribal knowledge into institutional knowledge, for future use. This works best when there exists an easy, centrally organized repository for such knowledge. Help Desk teams learned this lesson years ago. An organization cannot consider itself a learning organization without a central knowledge repository.

In Stable, we call this a Process Asset Library (PAL), and establish it to house important information about our processes and assets. Some people call this Configuration Management, or a Configuration Management System (CMS).

Essentially, a PAL is a single point of reference for information regarding any repeatable steps assigned to service offerings within the environment.

A PAL is organized by a Service Register which contains lists of different categories of services your group offers the organization. Each service can be thought of as a value proposition, and contains corresponding
process and asset information, sometimes called Process Controls, and Asset Controls, respectively. Services, process steps, and associated assets can all be linked together using a Process Asset Matrix (PAM), that is discussed by example in Chapter 8.

Standard Operating Procedures, and Lessons Learned for each process step can be created in various forms, and Kata Cards—or risk checklists—can be created for each repeatable step.

A good PAL is backed-up, version controlled, and has good search capabilities.

Two Framework Roles

The two framework roles in Stable include the Master Chief and one or more Process Owners. The group leader is the Master Chief, and the group members comprise one or more Process Owners.

Typically, the Development or Operations Manager is the Master Chief, and everyone else is a Process Owner. Under the Master Chief’s direction, the team can self-select a facilitator to assist with calling team meetings and representing the team when needed. This facilitator should be changed every so often, such as at the beginning of each Sprint or Master Cycle.

![Master Chief (MC) and Process Owner (PRO)](image)

**Figure 13 - The Two Roles in The Stable Framework™**

The Master Chief (MC)

The Master Chief is responsible for making sure the Stable Framework is set up and followed correctly. They are “Servant Champions” of their teams, responsible for leading their teams into Operational Excellence.
Master Chiefs work “on” the system. This means:

Environment Support:

1. Establishing the Stable environment (Quality Planning), including a Process Asset Library, the three domains, Incident Register, the performance console, and training for all involved.

2. Providing to the team a culture of empowerment, accountability, and sustained continual improvement.

3. Coordinating the whole set of value streams and their associated processes that collectively we call a System. This is done through facilitating the System Schedule and the Work Queues in the System Backlog.

4. Establishing a customer feedback and a supplier feedback channel. The MC doesn’t necessarily need to collect feedback metrics directly, but must put in place a feedback mechanism, via marketing or help desk surveys, or some other means.

Individual & Team Support:

5. Enabling the individual success of each Process Owner. This includes establishing the intent, boundaries, and constraints for each process, empowering the Process Owners, and coaching each Process Owner to continually improve the performance of his or her repeatable processes, called Katas.

6. Prioritizing the work objectives for the team of Process Owners in the Cycle Planning Meeting, and at the start of each day in the morning Kaizen Stand-up Meeting, as needed.

7. Presenting suggestions from the suggestion box to the group each morning.

8. Creating a Supplier Services SLAM Chart for reporting on the performance of shared Supplier Services into the System.

Individual & Team Accountability:

9. Verifying process accountability (Quality Assurance) at the end of each process value stream.

10. Collecting and storing Kata Card, Metric, and Training information
in Historical Logs.

11. Ensuring all team members are properly trained on their process kata’s, and on the Stable Framework™.

**Corporate Liaison:**

12. Providing information to the group about market conditions, CCAPA updates, as needed, and other relevant corporate news every morning.

13. Creating a Cumulative System Performance Chart and being sure to have Process Owners Update it at the end of every Master Cycle. This chart is used for quarterly reporting purposes.

**The Process Owner (PRO)**

A Stable environment requires one or more Process Owners (PROs). Each PRO is responsible for the results of each process they own. They work “in” the system, which includes:

**Customer Mastery:**

1. Ensuring that the current service levels represent their customer’s current expectations.

2. Ensuring the process results (Quality Control) meet the current service levels.

3. Ensuring frequent and insightful communication exists between the customers and the Process Owner to be certain of #1 and #2. A byproduct of this is a high Customer Satisfaction Rating.

4. Establishing a healthy positive emotional relationship with their customers and suppliers. A byproduct of this is a high Net Promotor Score.

**Supplier Mastery:**

5. Ensuring inputs meet expected quality levels (Incoming Quality Assessment).
6. Working with suppliers, indicating process and customer needs, to improve the material and information-based process inputs. A byproduct of this is improvements to process efficiency and effectiveness.

Process Mastery:

7. Ensuring the process, or Kata, is executed according to each Standard Operating Procedure and corresponding Kata Card. This is done by receiving the completed Kata Cards from the upstream process checkpoints, and handing them off, along with the newly completed Kata Card, to the downstream checkpoint, or to the Master Chief when at the end of the value stream.

8. Going straight to recovery models to recover systems and assets as quickly as possible. This minimizes downtime.

9. Asking coworkers for help, when needed. Brief Ad-hoc Kaizen Teams comprised of volunteers are extremely effective in overcoming sudden technical challenges.

10. Ensuring the Standard Operating Procedures, Kata Card templates, and recovery models remain updated and represent the best-known ways to perform and apply recovery steps to the process.

11. Ensuring newly discovered issues and recovery techniques are added to the Kata Card templates, or process or asset recovery models as new problems and root causes are encountered.

12. Adding incidents requiring repeated steps, and avoided incidents detected by the Kata Cards to the Incident Register.

13. Updating Service Level Attainment Monitoring (SLAM) Charts each day.

14. Updating Kata Cards and process descriptions as needed.

If a team member is stumped with a technical challenge they are encouraged to ask for help. An ad hoc team called a Kaizen Team can be voluntarily formed to assist. Kaizen Teams should be comprised of volunteers. Companies that adopt Kaizen practices have learned the volunteer aspect of a Kaizen team is critical. They have learned that
volunteers seem to generate the best ideas.

Companies that adopt Kaizen practices have learned the volunteer aspect of a Kaizen team is critical. Volunteers seem to generate the best ideas.

Kaizen improvement efforts can be performed on several levels. They can be triggered at the daily Stand-up when a team member needs help, as described above. These are called Kaizen Blitz’s. They can also be triggered in the Cycle Retrospective Meeting, when either the entire group identifies an item to improve, or one or more team members voluntarily identify an item in their work environment to improve. These are called Kaizen Projects.

*Three Domains*

The three domains are shared by each role and include the Future, the Present, and the Past Domains. We must plan our work ahead of time before we can measure the results of our work. For this reason, in the next three chapters we’ll discuss the Future, the Present, and the Past Domains, in that order.

![Figure 14 - The Three Domains of The Stable Framework™](image-url)
The Future Domain represents our relationship with the future. It contains information we’ll collect and use later. The Present Domain contains our workflow. The Past Domain contains our history, and metric results.

All three domains are tied together through a performance console, which is how senior management is kept aware of present status.

*Four Core Meetings within a Repeating Timebox*

In Stable, progress is measured over time using repetitive loops of 1-4 weeks, called time-boxes. If your group is developing products in an Agile environment, you can call this a Sprint. If not you can call this a Master Cycle.

This repetitive Master Cycle is how we measure progress over time. The cycle length should be consistent, and if you are not an Agile team, but coordinate with one, it’s best to synchronize your Master Cycle with your Agile team’s sprints. This way you can report together at the end of each cycle and sprint.

At the beginning of each cycle, a Sprint or Cycle Planning Meeting is held that can last up to several hours. In this meeting, the team establishes work to be done by the end of the cycle.

During the cycle, each member performs planned work they have selected at the beginning of the cycle, and any urgent work that has become important during the cycle.

At the end of each cycle, a Review Meeting, and a Retrospective Meeting are held. Combined, both meetings at the end of each sprint or cycle should last no longer than two hours.

Work activities are prioritized at the beginning of each cycle. Agile development teams performing Scrum will select their work from a product backlog, whereas non-development teams will select their work from several queues: Scheduled Activities, Customer Requests, Asset Maintenance, and the CCAPA Committee. Agile teams may naturally prefer a Sprint board, whereas non-Development teams should use a Kanban board, which allows the participants to stay focused on the highest priority tasks and avoid unnecessary task switching.
Repeatable process steps are called Katas and have Process Controls created for them in the Process Asset Library (PAL). Process Controls include Standard Operating Procedures, and corresponding Checklists, called Kata Cards. Performance against process service levels is reported using Service Level Attainment Monitor Charts, or SLAM Charts.

Every day within the cycle the group performs a daily Kaizen Stand-up Meeting where group members inform each other about the progress of their work activities and about any improvements they’ve made in their environment.

Kaizen started as a Japanese word that means “Ongoing improvement involving everybody, without spending much money.” We call it a Kaizen Meeting because as we perform our activities, we are always seeking ways to improve them. In 1993, Kaizen was added to the Oxford English Dictionary and became an official English word.

The Master Cycle

Figure 15 - The Stable Framework™ Master Cycle

It’s important for the team to communicate during each sprint, or cycle. Communication happens in two ways. Horizontal communication happens in the recurring meetings as mentioned previously. Vertical communication happens during normal business activities. Some
information is exchanged using horizontal communication, and some using vertical communication.

**Horizontal Communication** happens in pre-set, recurring meetings. **Vertical Communication** happens during normal business activities.

In summary, the four core meetings that comprise an Agile Sprint, or Master Cycle are the Cycle Planning Meeting, the Daily Kaizen Stand-up Meeting, the Cycle Review, and the Cycle Retrospective Meetings. These meetings are described in detail in Chapter 11.

**Five Stable Principles**

Stable is about insight, coordination, accountability, and improvement within a system. These four elements bring results.

A Stable team is first challenged with attaining the service levels expected by their customers. Once those levels of service are in place, and consistent, the team is then challenged with delivering those same levels of service for less internal cost and effort. This requires continual innovation and improvement.

To guide the team towards sustainable continual improvement, the following Five Stable Principles exist in the framework:

**Principle 1 - Anything Repeatable Can be Improved.**

Stable practitioners operate under the belief that anything repeatable can be improved. Again, and again, from Toyota to Ford to Hyundai and other companies that adopt Lean philosophies, these companies have realized that anything repeatable can be improved, and they operate with that mentality.

At first, only a few ideas for improvement may be identified. The nature of improvement is such that the constant focus and pursuit of these improvement ideas will uncover even more ideas over time. Sometimes
improvement is a massive breakthrough, sometimes it is a patient journey.

**Principle 2 - Improvement Requires Repetition, Reflection, and Change.**

Repetition alone brings systematized inertia. Add reflection and you get better insight. Better insight generates new ideas and those new ideas require change in something to bring improvement. Shewhart’s model for gaining system insight is the Plan-Do-Check-Act model. In this model, you make an improvement plan (Plan), then you act on the plan (Do), then you measure the results (Check), and if favorable, you update the Standard Operating Procedure and process Kata Cards to reflect the updated improvement (Act).

If the results are not favorable, of course, you’d skip the last step and try the next idea, working through the model. Toyota added “Go and See” to remind us of the importance of observation. Sometimes the model is known as OPDCA, or Observe-Plan-Do-Check-Act.

![Plan-Do-Check-Act Diagram](image-url)
Chapter 7 – The Stable Framework™

Principal 3 - Sustained Improvement Requires Systematization.

Continual improvement is not enough. You must have sustained continual improvement. Anyone can have a new idea for improvement, and implement it. But what happens when that person gets promoted, or takes a new job? The improvement vanishes with them. For this reason, sustainable continual improvement requires a system of record.

There must be a go-to repository containing lessons learned, standard operating procedures, Kata Card checklists, Recovery Models and other mechanisms to preserve each new improvement. Without this, you might be engaging in continual improvement, but not sustained continual improvement.

The financial loss to an organization that lacks the discipline to create such a repository is incalculable. In his popular book Kaizen, Masaaki Imai tells a story he heard from Toshiro Yamada, a Professor Emeritus of the Department of Engineering at Kyoto University. Yamada had visited an American steel manufacturing plant for a managerial discovery exchange when he was young, and then visited the same plant 25 years later.

He was shocked to see no significant improvements in how the plant was managed. In 25 years, nobody had come up with any significant sustained innovations. Don’t let your environment be like this plant!

Principal 4 - Improve Systems, Respect People.

Deming taught us that when problems occur, management must focus on improving the system and not on blaming the person. The old industrial model held employees accountable for problems and would often make them pay with their jobs. Deming taught us that the real question to ask is how the system can be improved so the problem does not occur again...both with the same employee and future employees working the same task.

This “safe culture” engineering fosters trust and assurance among employees and drives creativity when problems occur. When a problem does occur, the Master Chief should challenge the Process Owner to brainstorm permanent improvements to the process to prevent those same problems from happening again.
For difficult problems, other members from the team or the entire team can be asked by a Process Owner to volunteer solutions and work together through the problem. Working together to solve problems in a short amount of time is encouraged. We call this a Kaizen Blitz. Blitz teams work best when they are comprised of members who have volunteered to help.

** Principle 5 - Speak Using Data. **

Whenever possible, use data to express reality.

If you want to know how happy your customers are with your products or services, survey them. If you want to know how much time is spent reworking bad code that was rejected from the test department, measure it. The Stable Framework™ is all about improvement. You cannot conclusively improve something unless you first measure it.

Kata Cards presented at process handoff checkpoints, and SLAM Charts updated daily provide measured accountability data. In Stable we resist vague status updates like “we think we are on track,” or “we feel like we are moving along OK.” Comments like this are of little value.

From most workers perspectives, measuring has unfortunately been used in the past primarily for rewards or punishments. We need to think differently today. The proper application of measurement should be to provide a window into reality enabling employees and management to brainstorm ways to improve their systems. This is an empowering experience and means that measurement should be a recurring and vocal part of any team’s continual-improvement process.

As a group’s objectives are made known to them, they should identify which drivers will bring success, and then which measures would indicate progress along those drivers. Measuring performance in this manner will keep the team unbiased and much closer to reality, meaning they will make better decisions along the way, achieving success faster. Measurement approaches should be scrutinized and revised regularly.

The team should always consider better ways of measuring to improve visibility into the reality of the status and improvements of its processes and process goals. Measurement should be discussed often and should be an ever-present part of the work culture.
Be sure to keep efforts to measure information in check with the value of the information. One form of waste is collecting metrics nobody uses.

The proper application of measurement should be to provide a window into reality enabling employees and management to brainstorm ways to improve the system.

This is an empowering experience and means that measurement should be a recurring and vocal part of any team’s continuous-improvement process.

Some measures we’ll use may be components of larger measures and indicators of organizational progress. These are primary indicators and we call these measures Control Points.

### Six Ways to Improve a Process

Six forms of process improvement are covered theoretically in Chapter 4. Using the Stable Framework™, groups adopting the framework develop the visibility and conscientiousness required to put in place mechanisms that produce these six forms of process improvement.

Calculate the Hidden Factory, or percentage of lost effort within your organization. Now estimate how much money would be saved every month if your organization eliminated all of it.

### Seven Steps of Systematization

Systematization is the key to invoking Stable. Deming taught that it was the responsibility of management to implement a quality program. He reasoned that common employees don’t have the decision-making authority nor the awareness typically to construct a quality program by their own initiative.

Further, it’s management’s responsibility to create an environment where employees can succeed, and the best way to do that is with a quality program. So, if employees are wrestling with a volatile process,
generating significant Hidden Factory waste, it is really the management’s omissive failure because they are not using a quality program.

Below is a step-by-step guide to systematizing a process to be improved by the Stable Framework.

**Step 1 – Identify the Domain’s Value Propositions.**

Every group, team, department, or division is organized to perform certain functions for the organization. Although many steps are involved, there is always one or more final endpoints of value. These are called ‘Value Propositions.’

In a software development team, for example, these would be:

- Producing the next application release
- Providing install and upgrade tools for each release
- Backing up the source code
- Performing last-minute production changes
- Supporting the Help-desk with technical questions

Each of these value propositions have repeatable steps behind them that constitute the value stream needed to produce the product or service.

**Step 2 – Establish Expected Service Levels for each Value Proposition.**

Depending on the criticality of the value proposition, service levels and metrics can be applied to each value proposition.

If a level of service is contractually set with a customer, we call this a Service Level Agreement. If not, we can set an internal target and call it a Service Level Goal. Some of these will be numeric, some will be binary.

Service reliability is typically measured in overall percent of uptime for the year. Release stability might be measured by the number of annual releases not requiring hotfix patches immediately after production.

**Step 3 – Identify the Standard Workflow Steps (Kata’s) Within Each Value Stream.**
Look for the repeatable process steps within each value steam. This works best if you start at the final endpoint and move backwards. From this backward-looking perspective, it’s easier to see how sometimes a value stream converges and has multiple starting points.

Step 4 – Create Process & Asset Controls in a Central Process Asset Library for each Repeatable Step (Kata) within the Standard Workflow.

Inside the Process Asset Library (PAL) we are creating a location for every Standard Workflow Step where we can add pertinent framework information. We call this “Process Control” information and it includes the following:

- A Standard Operating Procedure (SOP) for that step. Be brief here. A sentence or two describing the task is sufficient to begin.
- A Kata Card, or checklist Template for that step.
- An entry in a Process/Asset Matrix if needed.
- A Process Step Recovery Model if needed.

Any assets identified along the way can be added to a list of Asset Controls and a Process/Asset Matrix (explained elsewhere in this book) with pertinent Asset Control information.

Step 5 – Establish a Standard Workflow Chain of Custody Checkpoint Sequence.

As PROs complete their Kata Card governing their repeatable process step, they will be handing their card and any upstream cards they will have already been given, down to the next PRO in the value stream. These handoff points are called checkpoints.

Step 6 – Perform Standard Work Steps Completing Kata Cards and Sending them Downstream at each Checkpoint.

Going forward, as work is completed by PRO’s, they use the latest copy of the Kata Card template for each step and complete it as they complete their work. The Master Chief eventually collects the stack of completed
Kata Cards and saves them in a Kata Card Historic Library.

**Step 7 – As New Problems Occur and Get Resolved, augment the appropriate Kata Card Template and Recovery Model.**

As new problems occur and get resolved, augment the appropriate Kata Card template with the newly discovered risk to prohibit the same problem from recurring.

*In Summary*

The Stable Framework™ equips organizations with a concise, working model for performance management and sustained continual improvement. These elements implemented together create an Operational Excellence program applicable to any organization.
Chapter 8 - The Future Domain

The Future Domain is your relationship with the future. It’s where information is stored for future use. This includes our System Configuration, our System Schedule, and our System Backlog.

Future Domain

### System Configuration

**Process Asset Library (PAL):**
- **Service Register**
  - Master Services, Pending Services, Retired Services, Supplier Services; Customer & Supplier contact info
- **Process Controls**
- **Asset Controls**

### System Schedule

- Annually (1/Year)
- Bi-Annually (2/Year)
- Quarterly (4/Year)
- Monthly (12/Year)
- Weekly (52/Year)
- Week Day (1/7 M-S)
- Working Day (1/5 M-F)
- Daily
- Day of Year (Date)
- Week of Year (Week)
- Month of Year (Month)

### System Backlog

+ Scheduled Activity Queue
+ Customer Request Queue
+ Asset Maintenance Queue
+ CCAPA Queue

Figure 17 - The Future Domain

### System Configuration

The System Configuration is our Process Asset Library (PAL) and represents the institutional knowledge developed and stored for the retention and improvement of the environment. Using a PAL, various Service Lists, Process Controls, and Asset Controls can be stored and improved over time.

A good PAL is version controlled, backed-up regularly, is centrally accessible, and is easy to navigate.

Following is a description of the Service Register which contain the important lists within a Process Asset Library:
Service Register

The Service Register contains various lists of services and any related system information that should be kept in the environment. At minimum, this includes a list of the present service offerings (Master Services), work being developed (Pending Services), retired services, and services your group relies on to perform their work (Supplier Services). In addition, any information about corresponding customers, suppliers, and lessons learned can be included or linked to this register.

<table>
<thead>
<tr>
<th>Service Register</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Master Service List</strong></td>
</tr>
<tr>
<td>... all current services provided</td>
</tr>
<tr>
<td><strong>Pending Service List</strong></td>
</tr>
<tr>
<td>... services in development</td>
</tr>
<tr>
<td><strong>Retired Service List</strong></td>
</tr>
<tr>
<td>... services not advertised but still supported</td>
</tr>
<tr>
<td><strong>Supplier Service List</strong></td>
</tr>
<tr>
<td>... incoming services required to deliver the above services</td>
</tr>
</tbody>
</table>

*Figure 18 - Service Register*

Value Streams and Master Services

The Master Services List is a collection of all the services offered by the environment. It should represent all the value propositions and corresponding supporting services. Think of this as the list of value streams the system provides.

Some of the services might be flagged as critical to the company’s success. We call their metrics Control Points, and they are usually part of the overall company Key Performance Indicators (KPI’s) that may roll up into the organizations Objectives and Key Results (OKR’s).

As part of Quality Planning, the Process Owners of each of the services should have an initial consultation with the major customers of each service and draft a specific service level for each service they are list.
After the Process Owners do this, they need to report their findings responsible for. The service level should be noted in the Master Services to the Master Chief and demonstrate that the appropriate service level expectation has been noted in the Master Services List.

The status of these services is reported using SLAM Charts and Cumulative System Performance Charts, discussed in Chapter 10 – The Past Domain.

Optionally, the Master Services List could include pending and retired services, which are discussed next.

### Pending Services

Pending Services are services that have not yet been deployed but are presently in development. This list is updated by the Master Chief from regular meetings with Engineering. It’s important to keep this list updated so that no last-minute surprises impact an Operations group.

Problems in Operations begin in Engineering. Therefore, so do solutions. It’s important the Ops, DevOps, Implementation, Support Desk, and even Sales, and Marketing leads should be attending a regular Engineering Status Meeting, or be receiving updates from the meeting.

Often in software development, a well-intentioned but forgetful Engineering team will spend a year developing a new system on a new...
platform just to slide it under the door for Operations to begin supporting it, without any previous warning. Depending on the situation, Operations may be unequipped with the skills required to service the new offering, and chaos ensues. Keeping this list up to date is healthy.

**Retired Services**

Retired services are those legacy services that were once offered but have now been retired. Some customers may still use these, and they should stay on this list until there are no more legacy customers using them and they are not needed going forward.

**Supplier Services**

Supplier Services are those incoming services required for your active services to function properly. These services are part of your supply chain.

If your Master Services require power, networking, a database backup, and an active internet connection, then you would list all of those in the Supplier Services List and report on their service level performance each day or at the end of your Master Cycle using a SLAM Chart. Obviously, your Master Services will be negatively impacted by your Supplier Service failures, so it is important to measure the performance of these also and present the finding to your suppliers if they are problematic.

<table>
<thead>
<tr>
<th>Value Proposition</th>
<th>Service Level Agreement/Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet</td>
<td>SLA: 99.999% Uptime during business hours</td>
</tr>
<tr>
<td></td>
<td>100 mb Up, 50 MB Down</td>
</tr>
<tr>
<td>Data Center Power</td>
<td>SLA: 100% During business hours</td>
</tr>
<tr>
<td>Data Center Cooling</td>
<td>SLA: 100% During Business hours 62 degrees in server room</td>
</tr>
</tbody>
</table>

**Figure 20 - Supplier Services List**
**Customer List**

The Customer List is where contact information for your primary customers reside. Customers should be associated with Master Services, listing primary customers and their contact information for each Master Service. It’s a great place to note how often your customers expect to be surveyed for satisfaction ratings or ideas for product or service improvements.

**Process Controls**

The Master Services List is a list of processes organized by Value Propositions. For each repeatable work process step, or Kata, a Standard Operating Procedure (SOP), a Kata Card template, a Processes Recovery Model, and a Lessons Learned document should be created where needed.

The SOP can be a brief description, a diagram, a SIPOC Model, a video, or any other means of communication acceptable to the MC (Master Chief).

The Kata Card is not a step-by-step instruction, but rather, a super checklist of lessons learned and risks to watch out for while performing the process. This Kata Card becomes a type of certificate indicating the step was performed correctly for the next Process-Owner recipient of the work product in the value-stream workflow.

As we read in *The Checklist Manifesto*, written by Atul Gawande, checklists should be short. Several short checklists are better than one long checklist. Some checklists are meant to be completed in a step-by-step manner as the work is being performed, and some checklists are meant to be completed as the final step after completing work items. The Process Owner determines which type of checklist to build during Quality Planning.

A Process Recovery Model is a tool used by Process Owners to quickly troubleshoot present problems based on past knowledge. As problems occur, Process Owners should be trained to immediately obtain the Process Recovery Model for that process and look at the list of past problems. If the problem is listed, they should look at the remediation
steps and quickly act to resolve the problem. If the problem encountered is on the list it should be added once the problem has been resolved, so that future problems can be quickly remedied with help from past knowledge learned.

---

### Asset Controls

The Configuration Management System includes an Asset Control section, listing all the primary assets required for the Process Owners services to function properly.

Assets should be linked to the services they support. This can be done using a Process Asset Matrix (PAM) and can be incorporated into the Service Register, or it can be a separate spreadsheet.

<table>
<thead>
<tr>
<th>Process Asset Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Master Services</strong></td>
</tr>
<tr>
<td>Servicing ERP</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Document Management</td>
</tr>
<tr>
<td>Servicing Backup</td>
</tr>
<tr>
<td>Code Library Backup</td>
</tr>
<tr>
<td>Daily Reconciliation</td>
</tr>
<tr>
<td>Data Warehouse ETL</td>
</tr>
</tbody>
</table>

**Figure 21 - Process Asset Matrix**

You may also want to track the last time the asset was serviced, or a recurring service schedule. For example, servers should be set to a recurring automated defragmentation schedule, and checked for CPU saturation, available drive space, swap-file to RAM ratio, and disaster
recovery planning every so often.

In software, the assets may be the database server, the file server, the application server, the web server, the internet connection, or any other standard asset the process needs to use to perform successfully. As depicted above, it’s common for an asset to be used by more than one process.

Each asset should include an Asset Recovery Model, or ARM Sheet. This sheet contains the known past problems experienced with this asset, and the steps taken to recover from the known problems. You can have link from the PAM to each ARM Sheet.

Like Process Recovery Models, Process Owners should be trained to go straight to this ARM when an asset fails to find out if the problem has a known cause already experienced in the past. They should be able to read the resolution quickly and get the asset back up and running.

It would also be useful to note any manufacturer warrantee information or support contact information for these assets, so that they can be quickly found during an emergency.

Any other Lessons Learned about these assets can be stored on a Lessons Learned document or field for each asset.

**System Schedule**

The System Schedule is a holding place for all recurring events impacting the system. The schedule is used to list future scheduled tasks which will then feed into the Present Domain’s work queue, during the Master Cycle Planning Meetings.

It is organized by Yearly, Quarterly, Monthly, Weekly, Day of Week, and Daily categories. You can add or adjust time selections as needed.

Looking at the following figure, the notes indicate scheduled recurring items. These recurring items should be added to the Scheduled Work queue as needed.

This tool will help the group remember frequent and infrequent maintenance tasks, meetings, performance goals, and other schedule specific activities.
System Schedule

- Annually (1/Year)
- Bi-Annually (2/Year)
- Quarterly (4/Year)
- Monthly (12/Year)
- Weekly (52/Year)
- Week Day (1/7 M-S)
- Working Day (1/5 M-F)
- Daily
- Day of Year (Date)
- Week of Year (Week)
- Month of Year (Month)

Figure 22 - Example System Schedule

System Backlog

The System Backlog represents all the incoming requests from various sources within and outside of the system. They are organized into the Scheduled Activity Queue, the Customer Request Queue, the Asset Maintenance Queue, and the CCAPA Queue. Each queue is described below:

Scheduled Activity Queue

Based on items listed in the System Schedule, upcoming work items that are time sensitive are placed in this queue to be worked on by the team in the Present Domain. The Master Chief is responsible for being sure time sensitive activities are added to the System Schedule and placed in the Scheduled Activity Queue as needed

Customer Request Queue

The Customer Request Queue is simply the bucket of urgent requests that materialize daily in any operations environment. In a small
organization, these get handed to the engineering team, interrupting its daily planned activities. These constant interruptions are usually unavoidable and slow down scheduled project activities. In larger organizations, often a single engineer or an entire engineering team is devoted to handling these daily requests. In Stable, they are queued here and prioritized in real-time by the Master Chief to be worked on by the team in the Present Domain.

**Asset Maintenance Queue**

The Asset Maintenance Queue is the selection of service objectives that are regularly needed for equipment maintenance. In an IT environment this can be checking available hard drive space or CPU saturation while running critical services, verifying backup media, organizing CAT-6 wiring, or any other needed housekeeping. Think of the assets in your environment as a fleet of tools which require occasional maintenance.

**CCAPA Queue**

CCAPA is an acronym meaning Correction, Corrective Action, and Preventative Action. These are the remediation steps involved when unexpected production issues occur.

The CCAPA Queue comes from the Service Desk. When a customer calls a Service Desk with a nonconforming experience, we call that an Issue. When multiple customers call in about the same issue, we call that a Problem. The Service Desk should prioritize the problems and present an escalated Problem List to Engineering regularly containing the most critical problems to be solved.

This can be done daily, in an early morning meeting with the Master Chief just before the team’s Kaizen Stand-up Meeting, or in a larger company moving slower, it can be done at a separate meeting once per week, per cycle, or during the Cycle Review Meeting.

For each incident reported, a Correction should occur for that customer, meaning a workaround, or replacement, or fix should happen. Next, the Process Owner should be tasked with identifying the root cause for the nonconforming product or service so that Corrective Action can be taken.
Corrective Action means that the Standard Operating Procedure, and Kata Card for the process that failed should be improved to prevent that problem from recurring.

Finally, Preventative Action means that any additional processes that might contain the same vulnerability should also be improved. The status of the nonconforming problems in the CCAPA Queue should be managed between the Service Desk and the Process Owners using Vertical Communication.
Chapter 9 - The Present Domain

The Present Domain is where work gets performed and tracked real-time. The team uses visual workflow tools to indicate what they are working on and the completion status of that work.

Present Domain

Team
- Daily Kaizen Standup
- 5S Housekeeping
- Adhoc Kaizen Teams
- Process Kata & Kata Cards
- Training/ReTraining

Committees
- Change Control Committee
- CCAPA Committee

Environment
- Alert System
- Production Change Log
- Suggestion Box

Figure 23 - The Present Domain

The Sprint Board and Kanban Board

A Stable team may use a Sprint board, or a Kanban board, or both.

A Sprint board is best used to track a team’s project. Work inside of an Agile Sprint is pre-selected, batched, and completed during the Sprint. Teams pre-select enough work to fill their time during the Sprint, and should complete all of it by the end of the Sprint. This technique is sometimes called a push system.

The example below shows a Sprint board with four columns. The “To Do” column contains User Stories and their respective task break-downs to be completed. “In Process” is the work currently being performed. “To Test” is self-explanatory, and “Done” indicates the work has been tested successfully, and is marked as complete.
The capacity in every environment is naturally limited by the number of people available to perform the work. A Kanban is an alternative visual tool for managing workflow that matches tasks with available people, thereby limiting allocated tasks to only the people available to work on those tasks.

This pull system limits work to a single-piece-flow to avoid overloaded workers. This expedites individual items faster, and keeps the task queue upstream, where it belongs.

Column titles can vary, but one common configuration of a Kanban board is to organize columns by “Objectives,” “Selected Tasks To Do,” “In Process,” and “Done.” The “Objectives” column is a list of items sourced from the System Backlog which is a combination of the Customer Request Queue, the Scheduled Activity Queue, the Asset Maintenance Queue, and the CCAPA Queue. The Master Chief owns the triage process and prioritization of items from these queues into the “Objectives” column. The team of Process Owners, as they finish past tasks and
become available for more work, will then self-select items from the “Objectives” column and move them into the “Selected Tasks To Do” column. As they receive requests from their customers during the day, they can add them to the “In Process” column and finally to the “Done” column when complete. When they are freed up for more work, they can go back to the Kanban board and select the next item from the top of the “Objectives” column.

We’ve learned in books like *The Phoenix Project*, and *The Goal*, that task switching creates an illusion of more work getting done but actually slows down the flow of value within a system. For this reason, the “In Process” column contains a work-in-progress limit. In the example below, it is 3. This means no more than three tasks should be in the “In Process” column. Protecting natural bandwidth limitations reduces Work-in-Progress (WIP) inventory and keeps us in One-piece flow mode. Otherwise, our bottleneck would start collecting WIP inventory which hurts the entire value stream.

Some groups add an “On Hold” column for blocked tasks that are delayed for substantial amounts of time. Items in this column might require escalation to higher authority to be expedited.

Development teams can use Sprint boards, and Operations groups can use a Kanban. Some environments it may be best to use both. Project activities are managed with the Sprint board, and real-time request can be managed using the Kanban.

Some groups create horizontal swim lanes on their boards and use one lane for each project. Some use different colored sticky notes for different projects.

No matter how your groups choose to uses these tools, they are both excellent ways to keep the flow of work visible over time.

**Daily Kaizen Stand-up**

Every morning, the Master Chief and the Process Owners meet together in a morning Kaizen Stand-up Meeting to hear new suggestions from the suggestion box, to get an optional business/market summary report from the Master Chief, and to discuss the daily progress and objectives for which the team is focused.
If needed, the Master Chief puts the highest priority objectives in the “Objectives” column of the Kanban board, in order of most to least urgent. The team can pick from there when loading new work into the “Selected Tasks To Do” column of the team’s Kanban flow.

Kanban Objectives and Sprint User Stories should be no larger in estimated effort than a single Sprint or Cycle time-box. They should break down into tasks that are two days or smaller.

Wherever possible, the Process Owners should be cross-trained to work on each other’s processes and tasks. In Lean, we say it’s the goal, not the role, that is important.

Wherever possible, the Process Owners should be cross-trained to work on each other’s processes and tasks. This eliminates the risk of nobody being able to perform the task because that one person who knows how to may not be in the office today. SOP’s and Kata Cards make this easier. In Lean, we say it is the goal, not the role, that is important.

**Measuring Team Velocity**

While team velocity is easily measurable using a Sprint board, a Kanban board requires a slightly different technique.

Work Items should be categorized and time stamped as they move from column to column on the Kanban board so the group can get accurate historical information about the quantity of and turnaround times that different types of work items take to move through the team’s Kanban. This way, empirical work completion times for each category of work items can be collected and tabulated over time. This gives teams better ways to measure how much work is being performed within each Master Cycle, and how long a typical item from a specific category will take to move through the cycle, under typical environmental conditions.

If your team performs both project and operations work, and the project work requires estimation and tracking over time, it could simply be
estimated up front using story points, and tracked for velocity per Master Cycle over time. Each objective could be time stamped for when it entered the Kanban, and then when it was completed. This helps with estimates of similar sized objectives in the future.

Operations tasks are easier to account for by categorizing them into types of requests, and then time-stamping their start-times and end-times through your system. Then you can report on how many of what types of requests have made it through your Kanban during each Master Cycle, and the average length of time required to complete each task. Tracking this kind of information becomes valuable for setting future customer expectations correctly.

**Process Katas and Kata Cards**

The key to keeping work flowing is to make it visual. That’s what a Kanban and a Sprint board are really all about. Toyota calls this “Visual Management.” Keeping work visual maintains status transparency and exposes delays and bottlenecks which hinder the flow of value.

At the process step level we use a different technique. The individual repeatable steps within a process we call a “Katas.” A kata is a repeatable set of activities performed over and over again for improvement. To help us improve these repeatable steps, we used Kata Cards.

Kata Cards are a kind of super-checklist. They are a risk-management tool ensuring everything was performed properly, and they may contain process data as needed. Think of completed Kata Cards as certificates that the process step work was performed correctly and completely.

![Kata Cards](image)

**Figure 26 – Process Kata Cards**
In medicine, there is a concept called a “Standard of Care.” This means you should be able to walk into any proper hospital presenting severe flu symptoms and get tested for COVID-19, and Pneumonia. This standard of care for this condition has been pre-set and standardized over time from past experience. Kata Cards provide a tool for creating and improving a standard of care for each repeatable process step.

A fully systematized process should have a Standard Operating Procedure created for that process describing each repeatable step, and a corresponding Kata Card Template created for each step or set of steps to be completed while the Process Owner executes them. The completed Kata Card, and any previously accumulated cards during that process run, are then turned in to the next internal downstream Process Owner at the next natural process checkpoint. If there is no next process step, then the Master Chief gets the final stack of Kata Cards for that process flow upon its completion, and that is the final checkpoint for that process value stream.

The Master Chief stores the accumulating stacks of Kata Cards in a Kata Historical Repository, where a specific completed past Kata Card can be retrieved if needed during an audit or a CCAPA event.

Think of these completed Kata Cards as certificates that the work was performed correctly and completely.

The full cycle of process execution involves knowing the customers, establishing a service level agreement with them, or just a service level goal with the Master Chief if a service level agreement is not appropriate, and working through the process using Kata Cards always looking for ways to improve the process steps. In some situations, an “As Is” and a “To Be” analysis may be created with improvement plans made to achieve the desired service levels.

Although completed Kata Cards at the end of a set of process steps may be handed to the Master Chief at any time, the most convenient time is during the morning Kaizen Meeting.
Ad-Hoc Kaizen Teams

Cooperation is encouraged in Stable. Occasionally, a Process Owner may get stuck on a challenging activity and need more ideas or different approaches. During a morning Kaizen Meeting, or anytime during the day, the Process Owner can request group help and form a Kaizen Team to help get past the challenge. Anyone with available time can volunteer to assist in this ad-hoc manner. We call this a Kaizen Blitz.

CCAPA Committee

The CCAPA Committee owns the CCAPA list and the committee meeting is attended by the Service Desk Director, the Development Director, and the Master Chief, at least. Any relevant team members can participate as well. This meeting can happen daily, before the morning Kaizen Stand-up Meeting, or once per cycle, preferably during the Cycle Review.

The purpose of the meeting is to be sure all parties are aware of the status of any new or outstanding non-conforming issues. New issues can be brought to the attention of the team by the Master Chief during the morning Kaizen Stand-up meeting.

This meeting can shift in frequency, happening daily after a major release and then less frequently after release risks subside.

Change Control Committee

The Change Control Committee is the whole Process Owner team and any other external members who are required to approve a change. This change control process can be daily, during the morning Kaizen, or once per cycle during the Cycle Review Meeting.

If changes created from this group need to flow downstream to a larger change control body, the committee can approve changes internally during morning Kaizen, then the approved change documents can be sent to the larger multi-department Change Control Meeting which would meet at regularly scheduled intervals.
Training, Retraining, and Cross-training

Formal quality programs must prove that all employees have received training and are aware of their work processes. This means there is a record of their having read the Standard Operating Procedures required to perform their jobs.

During Quality Planning, when the SOPs are created by the Process Owners, this represents their first completion of training and can be logged as such. In practice, reading an SOP is never popular, therefore retraining should be scheduled perhaps every six months, or annually, where the Process Owner in charge of each process reviews all the SOP’s governing his or her processes and makes sure they are up-to-date. The SOP’s should reflect the best-known ways to perform the work. These SOP’s can be brought up to date every six months or annually, during Quality Planning.

If the team finds a better way to perform a process, they can update the SOP at any time. The SOP, and all the process controls should be stored as successive versions under document change control.

A cross-trained team is a good idea in Stable. If a team member is absent another team member can perform a critical task for them. A Training Log should show that every Process Owner has been trained on every other Process Owner’s processes, where appropriate.

When a nonconforming issue is experienced by a customer, one of three things should happen. Either the process needs to be improved internally, or it needs to be protected against some outside factor that impacted it, or the Process Owner requires retraining to perform the process correctly going forward. Frankly, this last option means the operator didn’t follow the process, regardless of whether there was a record showing that they did or not. In situations like this, where the employee was confused, re-training and logging the re-training is appropriate. A training log should be kept by the Master Chief showing when everyone was trained, and when they are queued up for retraining. Scheduled training should be listed in the System Schedule.

5S Housekeeping

A clean, orderly environment is a critical part of value flow in a work-
place. Toyota discovered its employees could concentration better when they invested just a few minutes each day performing housekeeping activities. Remember, IT is an assembly line of decisions. A clean environment helps everyone make better decisions faster.

Every morning after the daily Kaizen Stand-up, the team returns to their work environment and performs 10 minutes of cleanup. This includes tidying up the immediate environment, and can also include updating any SOP’s, Kata Cards, customer lists, or other process controls.

An excellent technique for workplace housekeeping is a process called 5S. 5S is an organizational philosophy made famous by Just in Time manufacturing practices. It represents five organizational concepts that begin with the letter “S.” Adjusted for IT, they are:

1. **Sort**
   1. Dispose of any items not necessary in the workplace. This includes items on walls, bulletin boards, aisles, and stairways.
   2. Reducing the number of items in the workplace is always good. The goal is to have only those items needed in the workplace that are required to perform the work.
   3. When the physical area is clean, intellectual sorting is in order. Process Owners should clean expired email, directories, file folders, common shared folders, etc.
2. Set in Order

1. Arrange items so they can be found when needed, by creating a place for everything and then putting everything where it belongs.
2. Examples: Sticky Notes, Projector Remotes, Phone Numbers, Whiteboard Markers, etc.
3. After the physical areas is ordered, common information like files, code showcases, and automation comments should be groomed.

3. Shine

1. Clean your workplace daily. Be sure equipment, floors, walls, stairs, and surfaces are free from dirt and grime. Clean work areas promote better and faster decision making.
2. Be sure labels, signs, and so on are clear and obvious.
3. Be sure cleaning materials are easily accessible.
4. Be sure backups are in place and working.
5. Be sure all license keys are recorded and accounted for.

4. Standardize

1. Create and follow a set of regular procedures to keep the work area orderly and clean.

5. Sustain

1. Are the 5S rules being practiced daily, after each Kaizen Standup? Have all employees received 5S training?
2. Stay on top of things using 5S principles, so the environment doesn’t deteriorate.

Alert System

Part of any operational environment is some type of alert system that should provide real-time status updates regarding environmental assets and their attributes.

This can take many forms, and some readymade tools exist for this purpose. You can use a network monitoring tool or create your own alert system to send email alerts or text messages when something is not as it should be.

Scoreboards are becoming popular today, where a large screen is placed in a work area showing the real-time status of the environment.
These tools equip teams to be constantly aware of performance status and be quick to respond to problems.

*Production Change Log*

If applicable, some groups keep a Production Change Log. This makes it easy to determine what changed in the environment in the past on a certain date when troubleshooting production problems.

*Suggestion Box*

A suggestion box should be clearly visible and easy to access. Suggestions should be part of the team’s culture. Each morning the Master Chief should check the suggestion box and read the contents to the team. A twitter feed could also be used for this.

Suggestions should be taken seriously, and the Master Chief should provide an explanation for any suggestion not able to be implemented.

One indication of the health level of team morale is the number of suggestions that are submitted by the team over time. Disempowered teams don’t make suggestions. Empowered teams do.
Chapter 10 - The Past Domain

The Past Domain contains historical information regarding the most recent and trending status reports from the environment. Charts created and used for reporting here will be posted on the Performance Console.

There are three types of data in the Past Domain: Master Cycle Totals, Historical Logs, and a Cumulative System Performance chart. This latter chart is the quantitative summary of performance metrics that can be brought to and discussed in the organizations Quarterly Business Reviews (QBR’s), if needed. Some of these metrics may provide proof that Key Results towards a Department or Corporate Objective have been met.

Master Cycle Totals

Each of the Process Owners report on the daily performance of their Master Service and Supplier Service Value Streams using Service Level Attainment Monitoring (SLAM) Charts. In addition, a single Supplier Service SLAM Chart is updated by the Master Chief for all shared Supplier Services.

Past Domain

Master Cycle Totals
• Cumulative System Perf. (CSP) Chart
• Supplier Services SLAM Chart
• Cycle Totals
• Customer Satisfaction
• Customer Net Promoter Score
• SLAM Chart A, B, etc.

Historical Logs
• Incident Register
• Production Change Log
• Critical Decision Log
• Kata Card Historical Repository
• CSP Repository
• Training Log

Figure 28 - The Past Domain
In addition, Master Cycle metrics such as total and type of work items completed, customer satisfaction ratings, customer Net Promoter Score ratings and any other important metrics are reported. SLAM Charts should show target service levels vs. actual service levels attained.
A SLAM Chart may be as simple as indicating that each service level was met each day for all services. Or, they may contain other useful information such as average usage rates, average times, customer satisfaction rates, etc. The two preceding charts are examples of a simple, and then a more complex SLAM Chart, respectively. Notice the Statistical Process Control Data showing on the second chart. Also note the solid boxes in the right-most column. These boxes would be colored green, yellow, or red, depending on the status of the service so far, during the Master Cycle.

<table>
<thead>
<tr>
<th>Date</th>
<th>Incident</th>
<th>Hours Wasted</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-Jun</td>
<td>Backup failed</td>
<td>4</td>
<td>$296</td>
</tr>
<tr>
<td></td>
<td>Recode report</td>
<td>15</td>
<td>$1,125</td>
</tr>
<tr>
<td>Total</td>
<td>Burden</td>
<td></td>
<td>$1,421</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Incident Avoided</th>
<th>Hours Saved</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-Jun</td>
<td>Adobe Acrobat Upgrade Incompatibility</td>
<td>15</td>
<td>$1,125</td>
</tr>
<tr>
<td>8-Jun</td>
<td>Alternate Flows in Requirements</td>
<td>80</td>
<td>$6,000</td>
</tr>
<tr>
<td>10-Jun</td>
<td>Promote to Test</td>
<td>2</td>
<td>$150</td>
</tr>
<tr>
<td>Total Saved</td>
<td></td>
<td></td>
<td>$7,275</td>
</tr>
</tbody>
</table>

**Figure 31 - Incident Register**

The Incident Register should be updated daily as new incidents are found, and as the quality program detects and prevents potential incidents from occurring. This is an excellent real-time evaluation of how well the quality program is working.
Cycle totals include a profile of the types and amounts of tasks completed by the team during the past Master Cycle. Tasks can be organized into any group categories such as Critical Fix, Pre-Emptive Fix, Standard Work; or Internal Customer, External Customer; or perhaps by Sources of requests.

Easy ways to collect customer satisfaction ratings include surveys at the point of contact with each customer, Service Desk surveys, or random customer outreach surveys.

**Project Work**

Project status in Stable is reported on using a Master Project Schedule. This schedule shows the summary status of any projects being worked on in a Stable environment. For an example, please see Chapter 21 “Stable Portfolio Management” in this book.

If detailed status information is needed for a particular project, a Gantt Chart and a Burnup Chart, and sometimes a Work Breakdown Structure is used. Please see Chapter 15 “Stable Hybrid Project Management” for an example of these charts.

Stable uses a hybrid approach to managing projects that incorporates the best parts of traditional Plan-based and Agile project management. This gives the team the benefit of managing project dependencies using a Gantt Chart, and managing work items using a Kanban Board and a Burnup Chart. Please see Chapter 15 “Stable Hybrid Project Management” for more information.

**Non-Project Work (Sustainment)**

Non-project activities can be tracked several ways. If helpful, a chart showing the total number of tasks completed per cycle, by category, can be created.

If the individual tasks are timestamped upon arrival, and as they are completed within the Kanban, averages can be collected over time showing the typical historical timeframe for fulfilling certain kinds of requests.
This can help manage customers’ expectations better by providing a probable completion time for similar request types going forward.

Another helpful chart may be a Workload Analysis pareto chart showing the hours spent per request type, during the past cycle.
Another chart, called a Hastings Chart, contrasts project work against sustainment work. Named after John Hastings who developed this chart while working at Rackspace, this chart shows primary factors that compete with project performance per cycle.

By tracking this information, John was able to show how changes in the senior technical staff count had a cascading impact on sustainment efforts which then impacted project efforts dramatically.

With his permission we’ve added Hidden Factory tracking to it to better understand the dynamics of an environment.

![How Our Time Is Spent Chart](image)

Using a Hastings Chart like this helps everyone involved understand the dynamics involved with the workload in any specific environment.

**Cumulative System Performance**

As your group works through Master Cycles during the year, you will want to understand performance trends over time for your System. Most companies review performance quarterly. For this reason, we create a Cumulative System Performance Chart showing Master Cycle trends over time. This chart is created by the Master Chief but is updated during each Cycle Review Meeting by each Process Owner.
<table>
<thead>
<tr>
<th>Year</th>
<th>Quarter</th>
<th>Cumulative System Performance 2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018</td>
<td>Q1</td>
<td>04</td>
</tr>
<tr>
<td>2018</td>
<td>Q2</td>
<td>03</td>
</tr>
<tr>
<td>2018</td>
<td>Q3</td>
<td>02</td>
</tr>
<tr>
<td>2018</td>
<td>Q4</td>
<td>01</td>
</tr>
</tbody>
</table>

**Figure 35 - Cumulative System Performance Chart**

*NOTE: The chart details specific system performance metrics and trends over different quarters and years.*
Figure 35 shows an example of this chart. Note that it contains Executive Sponsor and Team Morale satisfaction scores, and then Master Services and Supplier Services metrics.

Note the trendlines along the right side, and the descriptions of deficiencies along the bottom. This data is collected to bring the group closer to reality, and to trigger constructive dialog on how the system can be improved going forward.

Remember the two System Vectors: If customer satisfaction is down, focus on improving that first. Once that is where it needs to be, then focus on reducing time and costs by reducing rework, scrap, and recovery time.

A Cumulative System Performance Chart would be brought by the Master Chief to the Quarterly Business Review meeting, where the Master Chief would report on trends in his or her area, and set goals and create initiatives for wide-scale improvement.

**Historical Logs**

Historical logs include an Incident Register, a Production Change Log, a Critical Decision Log, a Training Log, a past Kata Card Repository, a past SLAM Chart Repository, and a Cumulative System Performance Chart Repository.

Storing past Kata Cards and SLAM Charts are important for historical reference purposes when past quality information is needed.

A Production Change Log is an excellent tool uncovering root causes of issues in a production environment. When an anomaly is detected, it’s helpful to match the start of the anomaly to a past change entered in the log.

A Critical Decision Log can be helpful for tracking the history of important environment decisions, or process decisions.

The Cumulative System Performance Log is the summary of improvement within the system over time. This is the summary of value the System has produced for the organization.
The Performance Console: Tying All Three Domains Together

All these domains are tied together using a Performance Console. The Performance console is where you post all the visual information summarizing the value streams.

Figure 36 - The Performance Console

The performance console should be a wall space dedicated to tracking the group’s performance. In Lean, we call this technique Visible Management. In Agile, we call this space an Information Radiator. According to Alistair Cockburn, a good Information Radiator has the following characteristics:

- It’s large and easily visible to the casual, interested observer.
- It’s understood at a glance.
- It is easily kept up to date.
- Information on it changes periodically so that it is worth visiting.

The Performance Console is the meeting place for the team. Morning Kaizen Stand-up Meetings happen in front of the console. Virtual teams will need to use a webcam or special software tool to connect the team with this information.

The goal of the information posted on the Performance Console is to
present the overall performance of the department in a manner so clear that a person walking in off the street could see this information and know what is going on in the department. If a person walking in off the street can read and comprehend the information, then so can senior management.

This enables senior management to be freed up to spend their efforts growing the business instead of having to chase the group around probing for answers. We call this level of reporting Operational Excellence.

Pay attention to any other information senior management asks about and add it to the Performance Console. This increases your team’s value and contribution to the business-IT alignment within the organization.
Chapter 11 - The Four Core Meetings in Stable

The Stable Framework™ includes four core meetings. If you have an Agile development group, you may want to continue calling these “Sprint Planning,” etc. Other groups will want to call these “Cycle Planning,” etc., as described below:

1. The Cycle Planning Meeting which occurs initially at the start of the first Master Cycle, but then can occur as frequently as every day, or as needed during the Cycle when the workload queue is empty, or at the start of each new Master Cycle. The MC and the PRO’s together can determine what frequency is best.
2. The daily Kaizen Stand-up Meeting.
3. The Cycle Review Meeting where progress is reported to stakeholders.
4. The Cycle Retrospective Meeting where the team reflects on how to improve their environment.

These meetings repeat with each Master Cycle. In addition, two optional but recommended meetings are the Change Control Committee and the CCAPA Committee Meetings, which were introduced in Chapter 9 and are discussed more in Chapter 14.

Many organizations also hold a Quarterly Business Review meeting. The Master Chief may attend those meetings and present the Cumulative System Performance Chart.

The Master Chief must also be aware of the status of any soon-to-be completed Engineering projects. The Master Chief can be updated on this information by attending Development status meetings, or the Master Chief can collect this information through vertical communication.

The Master Cycle Planning Meeting

The Master Cycle Planning Meeting is called by the Master Chief and is initially held at the start of the first Master Cycle. In this meeting objectives are selected from each of the work queues as needed and added to the Objectives column of the Kanban board to be completed as
individuals on the team are freed up to work on more items.

The Master Chief orders the list of Objectives from top to bottom and the team self-selects which work objectives to do from the top of the list. Process Owners on the team self-select who gets the next item on the list.

During this meeting, objectives are broken down into individual tasks required to complete each objective. Tasks should be kept to one or two-day sized units of work. Expected one-or-two-day time frames are then assigned to each task by the person who selected the task to complete. Most objectives should be familiar work, so the breakdown should already be known.

Process Owners should add any time-triggered items they need to the System Schedule, to be done later during planned intervals.

This meeting can be held at the start of each new Master Cycle, or any time during the Cycle when the workload queue has emptied, or even daily if that works best.

For a development team, project work is covered in chapter 15. Some groups create horizontal swim lanes, or use different colored sticky notes to separate multiple projects being worked on the same Kanban Board.

The Daily Kaizen Stand-up Meeting

The Daily Kaizen is a daily morning Stand-up Meeting called by the Master Chief. The meeting is attended by the Master Chief and all Process Owners, or those on the shift.

The meeting should last no longer than 20 minutes. The meeting consists of the following:

- The Master Chief optionally provides a brief 5 minute or less market summary report which is a summary of any outstanding news from Senior Management or the Service Desk or any other part of the company. The Master Chief may want to meet regularly with the Service Desk (CCAPA Committee Meeting) for this purpose before the daily Kaizen each morning, or once a week.
• The Master Chief notifies the team of any new items or items that have received a higher priority.
• The Process Owners clarify any new work items they have added to their “Selected Tasks To Do” column.
• The team updates their SLAM Charts with yesterday’s information and reports on the following four questions:

1. What did I accomplish yesterday (including any change control items)?
2. What do I intend to accomplish today?
3. What roadblocks are in my way?
4. What did I improve in my environment since yesterday?

This Kaizen Meeting is a daily event where the whole team can discuss and inform each other concerning what each Process Owner has completed and what they will be working on next. If a new high-priority objective has appeared in a work queue the Master Chief can add it to the top of the “Objectives” column on the Kanban board and present it to the group and discuss it briefly.

During this meeting each team member is reporting to the team. Once the team starts reporting, the Master Chief should do what they can to not be the focus of the dialog. Two excellent techniques are to remind the group that this second part of the meeting is for the team to report to themselves, and then walk behind the circle breaking eye contact.

Immediately after the stand-up meeting has ended, the team returns to their workstations and performs 10 minutes of 5S improvement. A clean environment enables faster and better decision making, so this activity is performed daily.

The Master Cycle Review Meeting

At the end of the Master Cycle, the Master Chief calls the Master Cycle Review Meeting. In this meeting, the group meets to showcase their work performance during the past cycle and discuss recent customer feedback. Implemented suggestions are explained. Total work completed is reported, the Cumulative System Performance Chart is updated, and old SLAM charts are filed away historically.

This meeting is called by the Master Chief and attended by the team, and any customers invited by the Master Chief and Process Owners.
The Process Owners are tasked with collecting customer feedback to bring to these meetings. It’s important for Process Owners to be aware of their current customer satisfaction ratings. It may be useful for the Process Owners to set up a regular performance review survey with their primary customers and report on their findings. They can also work with the Service Desk for customer satisfaction information.

Kaizen Project results, which are described in the next section, can also be discussed in the Master Cycle Review Meeting, if appropriate.

The Master Cycle Retrospective Meeting

The Master Cycle Retrospective Meeting is the final meeting in the Master Cycle. It is called by the Master Chief and attended by the team, but not any customers. In this meeting the team discusses the following:

1. What went well this past cycle?
2. What could have gone better this past cycle?
3. What still feeds the Hidden Factory?

If no feedback or discussion is happening, it may be that the Master Chief needs to be the topic of discussion. The Master Chief may consider asking the team to self-select a scribe to write down comments the team can discuss and leave on the table for the Master Chief to pick up after the meeting has ended.

Kaizen Projects

In addition, the team should brainstorm and identify one or more target items to improve for the next cycle. The group can either brainstorm together and come up with an idea, or individuals can provide their own ideas to work on during the next cycle. These items should be something the Process Owners want to have improved.

Examples include automating an existing process, reorganizing an existing workflow, creating a customer education asset, or creating a new customer survey.

A team-based Root Cause Analysis can be performed using an Ishikawa
Diagram to identify problem areas collectively. The team can then use their conclusions to identify a group based or individual Kaizen Project. See the chapter titled “Data Collection and Analysis Tools” for details about using an Ishikawa Diagram for this.
Chapter 12 - The Process Owner

The Process Owner, or PRO, is an intelligent, capable decision maker, who is singularly responsible for the success of each process he or she performs and the satisfaction levels and Net Promoter Scores of the PRO’s customers. Capable means the PRO is empowered to make decisions, is aware of the organization’s intent, and can interpret reality correctly. Since the PRO is the single person responsible for the success of the processes for which he or she owns, it’s imperative that the PRO has a healthy relationship with his or her suppliers, understands all the components of value flow within his or her processes, and knows the expectations and needs of his or her customers.

Many of the PRO techniques discussed in this section are life skills, and in addition to improving workplace performance, can help the PRO improve many aspects of his or her own life. These techniques include identifying and engaging customers, making better decisions by seeking better data, understanding the root causes of problems, and progressing using sustained continual improvement techniques. All these skills will enable better long-term results in a Process Owner’s life both professionally, and personally.

Customer Engagement

Process Owners should regularly collect actional feedback from their customers. This should be qualitative information, not quantitative information. Process Owners should continually be asking, “Is there anything else you need?” or “Can our service be improved in any way?” This can happen every time a Process Owner engages a customer.

Customer satisfaction metrics, usually collected in quantitative form, should be collected by surveys, questionnaires, or direct contact from the Master Chief. Customer Satisfaction information should never be solicited by the person responsible for customer satisfaction. This will bring skewed data. I remember once being told by a restaurant host if I gave him 5 stars on his survey he had just handed me, I would receive a free milkshake. I now call this the “milkshake problem.”

In some cases, customer feedback can be coordinated through
marketing, the Service Desk, or a centralized customer feedback effort within the company.

It’s important for the Process Owner to understand he or she has more than one customer. In fact, the Process Owner has many customers. Four types of customers include:

1. The business consumer at the end of the value stream.
2. The worker at the very next step in the process who receives the completed work item.
3. Another worker skilled in the art who may take over the process as the company grows, or when staffing is low.
4. An auditor in a regulated industry who will want to see proof of process consistency, or an internal auditor checking for compliance.

To summarize, actionable feedback should come from the Process Owner, whereas customer satisfaction metrics should come from an automated survey, Marketing, Support Desk, or the Master Chief.

**Process Measures & Service Levels**

Part of the responsibility of owning a process is being able to report on its status and progress accurately. Only part of the responsibility of the Process Owner is to perform the work, the other part is to be able to express clearly how the work is progressing against established Service Levels.

At a minimum a Process Owner could report daily on whether the service was delivered at expected levels or not, using a SLAM Chart.

An advanced measurement to consider is calculating the cost per unit delivered. Simply add the direct fixed and variable costs required to perform a service, or to maintain a service over a measured amount of time and then subtract that amount from the revenue gained from the value stream during the same period.

For example, an on-site software installation could be itemized into the following categories:

- Travel, Hotel, and Meals for all involved on-site Process Owners.
• Total hours planning the install.
• Total hours away from the office implementing the install.

Summed up and subtracted from the price of implementing the contract, you’ll get the cost per unit delivered.

For ongoing services direct fixed and variable costs such as operator hours, and equipment configuration time can be combined with indirect costs such as utilities to come up with a cost per daily unit of service.

It may be helpful for your department to obtain a standard burden rate from your finance department for some of these factors.

The Process Kata

As a Process Owner works his or her repeatable processes over and over, we call each iteration a Process Kata. The more you iterate a process, the better you get at it. Eventually, you can perform the process almost without thinking about it.

A Process Kata consists of the following steps:

1. Clarifying what the customer expects.
2. Starting at the beginning of each process with a Kata Card dedicated to that process.
3. If the process has a supplier process, collecting the incoming upstream Kata Card from the supplier and making sure it is complete.
4. Performing the process, as described in the Standard Operating Procedure, completing the Kata Card along the way.
5. As unexpected challenges arise, immediately consulting the recovery model for that process or the problematic asset in that process, and then executing the recovery steps if available, or performing a workaround, and then a root-cause analysis to determine the source of the problem. Once known, the problematic incident description, root cause, and recovery steps should be added to the process or asset recovery model to assist with future problem recoveries.
6. When an internal process is complete, record any tracked data on the Kata Card, and then hand off the completed Kata Card to the
internal customer at the receiver process checkpoint, or the Master Chief if you are at the end of the value stream. It’s helpful to ask the customer if he or she is satisfied with the results and if there are any additional improvement requests. It may be helpful to have a survey ready during this exchange.

The Coaching Kata

A Coaching Kata is performed when an existing process must be upgraded to achieve a more desirable result. A Coaching Kata consists of discussing the following questions:

1. What is the current state compared to the desired state?
2. What needs to happen to reach the desired state?
3. Who could we ask, or what models exist that we could learn from?
4. What is the next step?
5. How soon can we learn from the results of the next step?

The target service level should be written in the Master Services List in the Configuration Management System and should also be specified on the associated process SLAM Chart.

It’s important when making improvements to any system that you implement the improvements one at a time, or at least independently. If you improve several dependent factors at the same time you won't know which factor produced which effect. The total effect may even be negative, and you won’t know which change caused the negative impact. For this reason, it’s important to make only one dependent improvement at a time.

Solving Process Problems

Process problems can occur from all sorts of causes. Some examples include ambiguous goals, insufficient system-wide understandings, flawed designs, sub-optimized goals and incentives, communication breakdowns, inadequate feedback, poor cooperation, and lack of accountability.
As problems occur—and they naturally will—the Process Owner adds an entry to that process Kata Card, to remind all future Process Owners to watch out for that problem going forward. This is sustained continual improvement.

Process Owners become expert problem solvers. They learn to harness the power of teams and their extended resources through Kaizen activities. They identify and learn from others who have succeeded before them or have other relevant experience that may be applicable.

**Eliminating Problems Using Kata Cards**

At the core of sustained continual improvement is the Kata Card. As items are discovered that can cause problems if undetected, they are added to the process Kata Card and possibly the Standard Operating Procedure. From that time forth, the process should be protected from that known problem as the Process Owner uses the updated Kata Card. This is the philosophy of continual improvement. Any problem can impact a process, but the same problem should never impact a process twice.

Effective Kata Cards have three sections: What to do before the event, what to do during the event, and what to do after the event.

The following figure is an example of a Kata Card. Notice it is organized by what to do before the event, during the event, and after the event. Notice also that this Kata Card encompasses two process steps, with signatures for each part. Be sure your Kata Cards have revision numbers or revision timestamps.

![This is the philosophy of continual improvement. Any problem can impact a process, but the same problem should never impact a process twice.](image)

It’s also critical that Kata Cards get collected and archived. Therefore, Kata Cards should get handed off to the receiver process at each checkpoint until the final Value Stream checkpoint, where the Master Chief receives the stack of Kata Cards and archives them.
Improving Processes Through Critical Thinking

Critical thinking is an important skill to develop as a Process Owner.
Critical thinking means that you have considered every possible angle to a situation, both popular and unpopular. It means you have reasonably researched and considered all probable and alternative information.

**A Simple Model for Critical-Thinking**

One simple model for critical thinking is: Thesis > Antithesis > Synthesis. Start with what you assume (Thesis), examine any evidence to the contrary (Antithesis), and then you should have a new understanding of the situation (Synthesis). This simple critical thinking model will help you make better decisions over time.

Critical thinking is a life skill, and an essential skill in problem-solving. Sometimes it requires courage, such as when you must challenge an assumption with a customer or with your manager.

The opposite of critical thinking is biased thinking. While the list of cognitive biases is long and interesting, the most common biases that inhibit critical thinking are listed here:

1. **Naivety Bias**: Inexperience leads to poor decision making. Believing everything without checking for supporting or conflicting facts leads to errors in judgment.
2. **Groupthink Bias**: The tendency to do or believe something because many other people do.
3. **Confirmation Bias**: The tendency to search for or interpret information in a way that confirms your own preconceptions.
4. **Extreme Aversion**: The tendency to not pursue the best choice because it is less convenient than a more convenient lesser choice.
5. **Planning Fallacy**: The tendency to underestimate task-completion times.
6. **Overconfidence Fallacy**: The tendency to overstate the team’s ability to perform.
7. **Reactance Bias**: The urge to do the opposite of what someone wants you to do out of a need to resist a perceived attempt by the other party to constrain your freedom of choice.
8. **Status Quo Bias**: The tendency for people to like things to stay relatively the same.
9. **Anchoring Bias**: The tendency to interpret a scenario based on past personal experience, which may not be the same for others.
10. **First Impression Bias**: Disproportional weight is given to the first information received.
11. **Sunk-cost Bias**: Bad ideas are considered still valid because a lot of money has been spent on them.
12. **Framing Bias**: A situation is inaccurately explained second-hand, steering the listeners towards a pre-set conclusion.
13. **Not Invented Here**: The notion that we can build it better than anyone else. If we have it built somewhere else, it will be inferior.

The best defense against all these biases is awareness of reality. Therefore, measurement is a critical tool. When done correctly, measurement gets you closer to reality and steers you through all these biases so that decisions made are better over time. The better you are at making decisions, the more competitive you will become.

### Improving Processes Through Nominal Group Technique

Nominal Group Technique (NGT) is a group brainstorming activity which is balanced to get feedback from everyone, even the quiet participants, and throttle the feedback from the louder participants.

In this model, a facilitator explains the problem or challenge at hand, and then the individuals in each group proceed to write down ideas individually on paper or sticky-notes. The ideas are collected and listed on a board. The group ranks the ideas and the best idea becomes apparent because the worst ideas get voted off of the board, one-by-one.

If a large group is present, it can be split up into smaller groups who perform this exercise individually, bringing the best idea from each group back to be ranked finally by all.

This technique is particularly useful after a focus group has been conducted to discuss a problem.

### Identifying Problem Sources Through a Root Cause Analysis

A Root Cause Analysis is a popular problem-solving process that can be performed using several different techniques. The idea is to find out the real reason why a problem has occurred.

Often an immediate reason given for a problem might just be hiding the real cause of the problem. At other times, there may be no obvious
reason readily identifiable, so we must investigate. Following are some techniques that are useful in performing a root cause analysis.

**Performing a Root Cause Analysis Using The 5 Why’s Technique**

One simple but effective technique for root cause analysis is called 5 Why’s. Using this technique, the Process Owner repeatedly asks, “Why did this happen?” until the actual source of the problem is identified.

For example, when the famous Hubble Telescope was blasted up into space in 1990, its large mirror was found to have been shaped wrong, crippling the telescope’s existing optics.

In this scenario, the Process Owner interacting with the team would ask questions in this order:

1. “Why were the optics crippled?”
   Answer: “Because the mirror was shaped wrong.”
2. “Why was the mirror shaped wrong?”
   Answer “Because it was ground to the wrong specifications.”
3. “Why were the specifications wrong?”
   Answer “Because we failed to account for the gravitational differences between Earth and space on the large glass mirror.”
4. “Why did we fail to account for those difference?”
   Answer: “Because we could not think of a way to test the impacts of weightlessness on the glass mirror while it was still on the ground.”

So, you see that within 5 questions, the root cause was uncovered. This was the actual explanation provided by the mirror manufacture. The
optics in the Hubble telescope were corrected three years later in a subsequent shuttle flight.

In rare cases, it may take you even more than five questions to find the root cause, but you get the idea.

**Performing A Root Cause Analysis Using an Ishikawa Diagram**

Another root cause analysis tool that builds upon the 5 Why’s technique is the Cause and Effect, or Ishikawa Diagram.

![Ishikawa Diagram](image)

**Figure 38 - Cause and Effect, or Ishikawa Diagram**

This diagram is named after Kaoru Ishikawa, it’s inventor. Ishikawa was a Japanese shipyard quality manager who also developed the Kaizen circle, where shift-workers standing in a circle at the end of each shift discuss what they have improved each day. Agile practitioners will realize this is the forerunner to the daily Scrum standup meeting.

Using this Cause-and-Effect diagram, combined with the 5 Why’s technique, a group of people familiar with the problem can collectively brainstorm a root cause for a known problem.

Two kinds of Ishikawa diagrams exist. The example above is a Process-based Ishikawa diagram. Your team can also create a Theme-based Ishikawa diagram. The former is organized by process steps, while the latter is organized by some variation of Cost, Culture, Context, People,
Process, Policy, Platform, and Proximity.

It can be helpful to refer to or create a process flow model while identifying items of interest for the Ishikawa diagram. In addition, creating an Interrelationship Diagram is helpful while populating an Ishikawa diagram. With the benefit of a group discussion, several potential causes can be identified and prioritized in terms of what steps to take in what order to improve the system. Afterwards, a Process Decision Program Chart (PDPC) can be created to address each specific problem.

**Performing a Root Cause Analysis Using a Focus Group**

Focus Groups are meetings called by a Process Owner who invites others to the meeting who have experienced a problem or talked with customers who have experienced a problem.

The purpose of a Focus Group Meeting is to understand a problem better by asking questions to those closest to the problem. As the PRO facilitates, the invited guests are asked to describe what they experienced. The participants search for patterns that may provide clues to eventually identify what went wrong.

The meeting is for information gathering purposes, and not specifically to solve a problem right there. One of the other techniques listed in this section could be conducted after a Focus Group, such as the 5 whys, or an After-Action Review, to get to the root cause. You don’t really want to expend customers’ time on technical troubleshooting if it can be avoided.

**Performing a Root Cause Analysis using the A3 Problem Solving Approach**

An elaborate technique for identifying and solving a problem is the A3 Problem Solving Approach. With this approach, the team uses a single sheet of A3 sized paper (11” x 17”) to show all the details about a problem, including remediation plans.

Various templates exist on the internet containing a simple logical flow of information gathering and analysis steps including a problem statement, current condition, target condition, root cause analysis, corrective actions, preventative actions, results summary, and follow-up actions.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A3 No, Name, Date

Process Owner

Team Members

Figure 39 - A3 Problem Solving Template
Most of the fields in the example template on the previous page are self-explanatory. Remember that Corrective Actions are what needs to change within the target system, while Preventative Actions are what changes can be made to additional systems to achieve similar improvements elsewhere. The A3 format is an excellent template for Kaizen Projects.

**Improving Customer Engagement using After Action Reviews**

An After-Action Review (AAR) is a retrospective meeting held by the team directly after an engagement with a customer. They can also be called for other purposes as well, such as Root Cause Analysis, or after a major troubling event.

In an After-Action Review, the team shares observations, questions, and lessons learned about what went well, and what could have gone better. It is targeted at a specific event and should be conducted directly after the event when the team’s observations are fresh in their memories.

Be sure any lessons learned are recorded in the Process or Asset Controls.

**Improving Customer Engagement using a Facilitated Workshop**

A Facilitated Workshop can be called for many purposes. These meetings are excellent tools for brainstorming, knowledge transfers, walkthroughs, requirements elicitation, and many other purposes.

A Process Owner calls a facilitated workshop and invites customers and technical engineers, if needed, to the meeting. These meetings tend to be longer and some can span multiple days.

**Risk Management using a Failure Mode and Effects Analysis (FMEA)**

A thorough technique used in military-grade development and operations planning is a tool called a Failure Modes and Effects Analysis (FMEA).

In this exercise, a spreadsheet is created listing every component of a...
system or process, and the next column over lists every potential way it
could fail. Then, the impacts of each failure are listed in the next column,
and each impact is assigned a probability and severity rating.

Detection techniques and mitigation steps are listed, and the analysis
becomes a reference sheet for designers of the system and Process
Owners going forward.

After an FMEA is conducted, a Process Decision Program Chart (PDPC)
can be created, identifying risk mitigation strategies for specific high-
risk areas. See Chapter 23 – Data Collection and Analysis Tools for a
description and an example of a PDPC.

The Future Domain and the Process Owner

The Process Owner is the closest to the work and is responsible for
creating and maintaining the Standard Operating Procedures (SOPs) if
they don’t already exist. In my experience, SOPs rarely get read by
employees beyond required training. The practical purpose for an SOP is
to ensure consistency. They should be read by new employees during
initial employee onboarding and then be reviewed once or twice a year
and brought up to the current standard, as needed.

### Future Domain

<table>
<thead>
<tr>
<th>System Configuration</th>
<th>System Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Asset Library (PAL):</td>
<td></td>
</tr>
<tr>
<td>- <strong>Service Register</strong></td>
<td></td>
</tr>
<tr>
<td>Master Services, Pending Services, Retired Services, Supplier Services; Customer &amp; Supplier contact info</td>
<td></td>
</tr>
<tr>
<td>- <strong>Process Controls</strong></td>
<td></td>
</tr>
<tr>
<td>- <strong>Asset Controls</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System Backlog</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Scheduled Activity Queue</td>
</tr>
<tr>
<td>- Customer Request Queue</td>
</tr>
<tr>
<td>- Asset Maintenance Queue</td>
</tr>
<tr>
<td>- CCAPA Queue</td>
</tr>
</tbody>
</table>

**Figure 40 - The Future Domain**
An SOP should always represent the best-known way to execute a process. The process Kata Card should be filled out with every Stable Process Kata.

The Kata Cards should be updated regularly as any new failures are experienced and root causes are identified during normal work processes. All training should be recorded in the historical training log by the Master Chief.

For groups just adopting Stable, the Quality Planning phase is where all of this gets created. Here the PRO’s create their SOP’s and Kata Cards using the Standardize-Do-Check-Act model, based on the best known-method for performing each process.

A list of primary customers for their processes and their contact information should be made by the Process Owner and kept in a Customer List in the Configuration Management System. Initially, the PRO should meet with all or a representative sampling of the customers and establish the relationship and confirm expectations.

An agreement should be made between the MC and the PRO on how the PRO will get regular customer feedback and get any new feature requests. Depending on circumstances this can be a regular survey, phone call, visit, golf game, lunch, hallway conversation, or whatever seems appropriate. Whatever form it takes, what’s important is that it is done regularly. A good rule of thumb is to confirm expectations have been met at every customer point of contact.

Deming would help employees connect with their customers regularly. He would shut down entire assembly lines and have all the workers listen in to a conference call with random customers.

He would ask the customers how the company’s products and services helped the customers and how they could be improved. This is important because it kept the employees emotionally bonded with the customers, which improved what Deming called “Pride in workmanship.”

It’s healthy for the PRO’s to meet the people they are helping, and it helps the customers feel emotionally bonded to your organization. This a critical success factor in raising your Net Promoter Score among your customers.
Customer Satisfaction and Net Promoter Score

It turns out that Customer Satisfaction ratings are important, but not the most important customer metric to a business. Surprisingly, how well a customer has emotionally bonded to your company is a better indicator of future business than only having good customer satisfaction scores.

Customers might give you high satisfaction ratings, but not necessarily refer anyone to your business. It requires an emotional connection for them to take the initiative and refer their associates to you.

The probability for a customer to recommend your services to someone else is called a Net Promoter Score (NPS). The Net Promoter Score can be calculated by asking your customers in an anonymous survey, on a scale of 0 to 10, how likely they are to recommend your company to another colleague. Depending on how they respond, they fall into several categories:

- 0-6 detractors
- 7-8 passives
- 9-10 promoters

Then, to calculate the score, use this formula: \( \text{NPS} = \% \text{ promoters} - \% \text{ detractors} \). The larger the positive number, the better.

So how do you get your customers to emotionally bond with you? Emotional bonding happens when customers are addressed and respected as the individuals that they are. Surveys and emails should be personalized. Feedback from them should be answered personally, and quickly.

Your customers should get a consistent message that you are always aware of them and care about their needs and their successes. Teddy Roosevelt was famous for saying, “People don’t care how much you know until they know how much you care.” Ask your customers about their individual needs and then discuss how your services could better address them, rather than talking only to your customers about your services.
Ask your customers about their individual needs and then discuss how your service could better address them, rather than talking only about your services to your customers.

The best brands have figured out that they are really selling feelings, not features. Think about Nike and you can imagine yourself as someone who is fit and exercises every day. Think about any of the prominent insurance companies and you’ll get a feeling of protection. Think about driving a Volkswagen and you’ll suddenly be thinking of yourself as a young urban professional with a lot of friends.

There is an entire industry devoted to helping create an emotional connection with your customers. You will grow your customer base by learning from it.

The PRO is also responsible for compiling a Process Asset Matrix linking the Process Owners Processes to required Assets, and also an Assets List containing all the primary assets responsible for performing the PRO’s repeatable processes. For each asset, an asset recovery model and maintenance plan should be created, if needed.

The Present Domain and the Process Owner

For operations work, the Master Chief is responsible for populating items from the System Schedule and System Backlog into the “Objectives” column of the Kanban board during the Cycle Planning Meeting. As Process Owners complete existing work items and are freed up for more work, they can grab items from the top of the “Objectives” column. During the Master Cycle, they will naturally have additional process work items emerge that they need to work on. They should note these items on the Kanban so that they can be accounted for at the end of the cycle.

As Process Owners finish existing items, they can grab the next item they are qualified to perform from the “Objectives” column.

As they work the items, they move them from the “Selected Tasks To Do” column to the “In Process” column and finally to the “Completed” column. Some groups have a “On Hold” column to deal with larger objectives that get held up or require senior management to expedite.
Items should be time stamped as they move from the “Selected Tasks To Do” to the “In Process” and finally the “Done” columns. Kanban Boards allow us to measure how many of each type of work items that were completed during the Master Cycle, and the average time required to complete each type of task. This provides historical data that can be used for forecasting future performance for customers.

The team should be encouraged to complete what they are working on before turning their attention to something new. Task switching forces us to move our mental assets around in our minds and is a form of inventory, motion, and transport waste.

### Present Domain

**Team**
- Daily Kaizen Standup
- 5S Housekeeping
- Adhoc Kaizen Teams
- Process Kata & Kata Cards
- Training/ReTraining

**Committees**
- Change Control Committee
- CCAPA Committee

**Environment**
- Alert System
- Production Change Log
- Suggestion Box

**Figure 41 - The Present Domain**

When problems occur while working a process, the Process Owner should identify the process or asset containing the problem and go directly to its respective Process or Asset Recovery Model (PRM, or ARM Sheet). This is a list stored in the Process Controls or Asset Controls portion of the Configuration Management System and contains descriptions, root causes, and solutions to past problems associated with the process or asset.

When new problems occur, their root causes, and steps to fix them get added to the PRM or ARM sheet. By consulting these Recovery Models, repeated problems get resolved quickly.
The Process Owner is responsible for communicating the status of all the processes they are responsible for, during, and especially at the end of each Master Cycle.

### Past Domain

**Master Cycle Totals**
- Cumulative System Perf. (CSP) Chart
- Supplier Services SLAM Chart
- Cycle Totals
- Customer Satisfaction
- Customer Net Promoter Score
- SLAM Chart A, B, etc.

**Historical Logs**
- Incident Register
- Production Change Log
- Critical Decision Log
- Kata Card Historical Repository
- CSP Repository
- Training Log

This is done through measurement. Measurement is the essence of accountability and helps us understand what is going on in our environment.

The process measures described previously in this chapter get recorded and displayed visually on the Performance Console. This type of visual management is critical to achieving Operational Excellence. We want to be transparent about the health of our processes, customer relationships, service levels, and team morale so that our leaders whom we report to are free to spend their time growing the business and not wondering about what’s happening in our areas.

We are trying to express this information so clearly that a visitor walking through our area could see our Performance Console and know how our department is doing.
Two Types of Measurement

There are two types of measurements that are important. One is measuring how close the results were to the target. The other is measuring how close the target is to what the customers want.

For example, the first type of measuring is detecting how close we can get the room temperature to 72 degrees if that was our target. The second is measuring to see if 72 is the most desired temperature for the people in the room. You want to see how accurate you were to the target, but you also want to know if the target is currently accurate.

In summary, we want to use data to determine if we are reaching our goals, and what the current success criteria is for those goals.

Visual Communication

Visual aids are most helpful. It’s always better to use graphics and as few words as possible, but enough to be clear about what the graphics indicate.

Figure 43 - The Performance Console
All these graphs and charts are displayed on the Performance Console, which should look like the previous figure.

Graphs and charts are excellent tools for showing performance over time. The final chapter in this book is called “Data Collection and Analysis Tools” and details a set of tools for collecting, analyzing, and reporting in various ways.

While creating your charts be sure that your graphics are not too distracting. Every single pixel competes with your message, so when reporting with graphics be sure to favor simple graphics over fancy graphics.

As a professional, every time you report on status you should be interested in two factors. One is the current status, the other is the trend. The trend is almost as important as the current status. Is the reported performance improving or degrading? You’ll want to report on this, as well.

The Incident Register should be added to regularly by Process Owners, for both unexpected problems (Hidden Factory), and potential incidents detected and prevented by the quality program. Service Level Attainment Monitors, or SLAM Charts should also be created and updated.

<table>
<thead>
<tr>
<th>Critical Systems</th>
<th>Overall Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Day of Month Performance</td>
</tr>
<tr>
<td>Database backup</td>
<td>01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31</td>
</tr>
<tr>
<td>Home Page Load Time: Target: &lt; 4 Seconds, Mean: 2.86</td>
<td>4 5</td>
</tr>
<tr>
<td>Nightly Process Run Time: Target: &lt; 8 Hours, Mean: 7.16</td>
<td>6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31</td>
</tr>
<tr>
<td>ETL 1</td>
<td>1 2</td>
</tr>
<tr>
<td>ETL 2</td>
<td>3 4</td>
</tr>
<tr>
<td>Corporate Website Up 99.999</td>
<td>5 6</td>
</tr>
<tr>
<td>Help Desk Up 99.999</td>
<td>7 8</td>
</tr>
<tr>
<td>Network Support 99.999</td>
<td>9 10</td>
</tr>
<tr>
<td>Internet Available 9.9</td>
<td>11 12</td>
</tr>
<tr>
<td>10 Minutes/day SS Housekeeping</td>
<td>13 14</td>
</tr>
</tbody>
</table>

*Exception Notes:* 1 - Backup Abended 2 - Ran out of HD Space 3 - Ran out of HD Space 4 - Second Mirror getting hardware upgrade 5 - Second Mirror getting OS upgrade 6 - Final patch being applied to second mirror node

**Figure 44 - Service Level Attainment Monitor (SLAM) Chart**

The number of tasks completed can be categorized by type or urgency and listed over time in a bar graph, or a line chart.

In addition, any of the quality charts mentioned in the last chapter of
this book can be used to report on specific information important to the organization.

**Master Cycle Review Meeting**

Status Charts are viewed with internal customers and senior management at the Master Cycle Review Meeting, which occurs at the end of each Master Cycle. During each Cycle Review Meeting, each team member reports on the service level performance of their processes and explains any outstanding challenges remaining. Customer Satisfaction results are also presented at the meeting, by each Process Owner.

**CCAPA and Change Control**

The CCAPA List status is presented by the Master Chief, as well as any outstanding Change Control items that need to come before the group if change control is handled at this meeting.

**Kaizen Project Initiatives**

Optionally, individuals can be challenged at each Master Cycle Retrospective to identify a Kaizen Project to improve something in one of their processes, or in their workplace. These should be small improvement efforts that take from a few minutes to few days of effort. Then, during the Master Cycle Review Meeting they should be tasked with reporting on the improved objective and explaining the benefit to the organization.

**Master Cycle Retrospective Meeting**

The Master Cycle Retrospective Meeting happens just after the Master Cycle Review Meeting. This meeting should last no longer than one hour. Two hours, perhaps, if the team engages in a Kaizen Project. Only the team of Process Owners and, optionally, the Master Chief, will attend. No customers should be present.
The team discusses these three questions:

1. What went well this past cycle?
2. What could have gone better this past cycle?
3. What still feeds the hidden factory?

These questions can lead to discussions and ideas that trigger new Kaizen Projects.
Chapter 13 - The Master Chief

The Master Chief, or MC, is responsible for the overall success of the value system. MCs are responsible for ensuring that the qualitative work efforts expended within the environment get converted into value, with as little byproduct as possible. Their impact is measured by quantitative information that gets reported at the end of each Master Cycle and then aggregated into quarterly trend reports. The MCs are the stewards of the value system. This means they are responsible for organizing all the value streams and then exacting accountability among all the Process Owners to achieve the desired results for each value proposition within these streams. In addition, MCs are responsible for any necessary coordination between interactive value streams. Simply put, the MC’s are responsible to ensure that the Stable Framework™ is understood, implemented, and executed properly within their environments.

Ineffective leadership looks like lots of people doing unrelated busy work producing nothing of value over time. Effective leadership is getting everyone involved to do the right things.

The key to producing support for Stable among the Process Owners is by creating a sense of ownership through involvement. The Master Chief must ensure the Process Owners are involved in the systematization, customer engagement, measurements, and improvement efforts of their systems. If the MC’s cannot create ownership, they will have a difficult time generating support for Stable. It’s also important to understand that transitioning a group to Stable requires much more effort than maintaining an existing Stable environment. Be prepared for resistance by the uninformed. What they won’t know is that groups who accept and work with formal quality frameworks make fewer mistakes, get more done over time, and report about a 20% improvement in employee satisfaction after the framework is in place and working.

The Master Chief must be flexible enough to stay informed to help and assist anyone, anywhere when needed. This is a challenge that requires a lot of responsibility, competence, and conscientiousness. Coaching, connecting, and communication are the primary tools of the Master Chief. Plan on devoting time to improving your skills in these areas.
Finally, the Master Chief acts as a corporate liaison, providing information to the teams about what is happening elsewhere in the company and with the external customer base. This information is collected from other groups within the organization, and from intra-department committees they should attend such as the support-desk CCAPA committee.

Since Stable is applicable anywhere there is a team, there may be one Master Chief, or there may be several Master Chiefs in a larger functional unit working in coordination with each other under a Senior Master Chief. A Master Chief presides over an empowered team of Process Owners, with each Process Owner separately being responsible for the execution and accuracy of the repeatable processes they own. The Master Chief is responsible for facilitating the coordination of work between these processes.

Master Chiefs are responsible for the success of the entire system, and they achieve this by informing, equipping, empowering, coaching, and holding their Process Owners accountable for the performance of their processes. All of this requires leadership.

Although the individual Process Owners are responsible for collecting actionable qualitative information about how their process deliverables can better meet their customers’ needs, the Master Chief is responsible for collecting quantitative customer satisfaction, and Net Promoter Score metrics. If they don’t collect it themselves, they need to be sure Marketing, or the Service Desk, or someone other than the Process Owners are collecting these metrics. Customer satisfaction metrics should never be collected by the same people responsible for the scores.

**Leadership**

Leadership is one of the most written about topics in history, and for an astute professional, should remain as a lifelong topic of study. Some people are born natural leaders, but the rest of us must learn effective leadership patterns so that we can succeed at completing goals with teams of people.

I remember several college programming projects where I was paired with students who were unskilled or new to software development. After sizing up the lack of technical skills among the team members I happily
volunteered to do the entire programming component of each project. In every case our group received an A, and the team members thanked me and felt fortunate to have done almost nothing for the grade.

At the end of my college experience I was talking with the Dean of the Information Systems department and mentioned to her my role in doing all of the programming work on the group projects. Instead of being impressed, she scolded me for being unable to work effectively with a team. She was right, and her comment changed my perspective permanently. What I did was the opposite of leadership.

In James Pizzola’s book, *Managerial Leadership in the Workplace*, the author describes leadership as “…directing, influencing, and inspiring people to work together towards a common goal.” Sometimes this is called Coaching.

**Leadership Coaching Syles**

There are two approaches to coaching a team. Subject-matter coaching, and accountability coaching. If you are hired to teach a team a new set of skills in which you have expertise, you are a subject-matter coach. Sometimes this is called “Command-and-control” leadership. You are literally telling the team what to do and how to do it. This approach works best in temporary situations and for non-thought-worker activities.

The other style, Accountability coaching, is best applied when the team already has the expertise to complete the work. In this situation, your role is to use curiosity and consensus to move the team towards their goals.

Asking strategic questions such as “Whom could we ask for insights who have already succeeded at this?” or “What are the major steps to get us there?” or “What are the major risks along the way?” or “Who will do this part of the work?” and then confirming a future date and time to regroup and assess the team’s progress. In this manner an accountability coach is really acting as a consultant. Good consultants know the right questions to ask.
Leadership vs. Management

Leadership is often compared and contrasted with management. You will get almost as many different perspectives on this comparison as people you ask, typically with leadership presented as a positive move and management presented as something negative.

After engaging with literally hundreds of professional on this subject the clearest comparison came from a group of US ARMY project-managers working on the South Korean peninsula. They told me leadership is about people, whereas management is about things.

We don’t lead staplers and we don’t manage people. We manage people’s time and skills, but we lead them.

In other words, leadership is about direction and management is about efficiency. Direction and efficiency are the two components of Operational Excellence, and are the substance of The Stable Framework™. Therefore, Operational Excellence requires both leadership and management. You need them both.

Leadership as Influence

Like John C. Maxwell says, “Everything rises and falls on leadership.” Leadership is influence. Therefore, to become a leader you must become a person of influence. To become a person of influence, you must become a person of value to those of whom you report, and to your Process Owners. You must develop knowledge, connections, trust, past experience, competence, character, and good judgment. You must learn to express yourself clearly, persuade others, and defend yourself and your team when required. Where you fall short, good mentors and good team support will compensate.

Ineffective leadership looks like lots of people doing unrelated busy work producing nothing of value over time. Effective leadership is getting everyone involved to do the right things.

Let’s discuss the dynamics of team leadership.
The Leader’s Two Teams

Every leader is given a primary team which they have stewardship over and for which they hold responsibility for their results.

Leaders will magnify their effectiveness in many areas if they seek out a secondary team of peers, coaches, and mentors. As a leader your circle of influence will grow as you tap this second team.

Stay positive and encouraging with your primary team, but rely on your secondary team for expressing frustrations and collecting ideas for dealing with difficult team problems that you can’t solve by yourself.

The Three Concerns of Team Leadership

As a Master Chief, you are ultimately responsible for the performance of the team. There is a lot to learn to do this correctly. To succeed, you must concern yourself with three areas:

1. Team production results
2. Team moral and cohesion
3. The individual needs of the individuals on your team

Team Production Results

Management guru Peter Drucker made popular a concept called “Management by Objectives” which was later revised by Intel’s Andy Grove into a performance management pattern today called “Objectives and Key Results,” or OKR’s.

Objectives are goal-related milestones which can be decomposed into one or more self-evident actions called Key Results. From the top of the organization down to the bottom, Objectives are identified by each respective manager, their Key Results specified, and the teams work together on these Key Results until they are all met, which completes the objectives.

In a well-run company, from the CEO on down, all of these OKR’s are
posted publicly, monitored for progress, and align under a central “North Star” vision which guides the company forward.

Accountability for these OKR’s happens at different intervals, depending on the level of the objective. Stable Team OKR’s are identified at the start of each Master Cycle, or Sprint, and the status for each is accounted for at the end.

With Development teams, Objectives can be thought of as the Sprint Goal for each Sprint, and completing the work that makes up each Sprint Goal at the end of each Sprint can be thought of as the Key Results.

With Operations teams, Objectives can be thought of as meeting Service Level Agreements and Service Level Goals for that Master Cycle.

At a higher level, OKR’s can be established for Application Development feature-set completion, or larger projects inside of Operations (ie: “Upgrade the telecom system.”). These larger OKR’s might be reported on quarterly, at the intra-departmental Quarterly Business Review meeting.

Using an OKR framework enables everyone involved to synergize and focus on what’s most urgent for the organization. An excellent book on this topic is *Measure What Matters*, by John Doerr.

It’s the leader’s job to establish these Objectives, along with clarifying any constraints and roles involved. Together the MC and PRO’s can establish the Key Results. If any of these are not clear, the leader has not yet done their job.

Once the Objectives are clear, it’s the leader’s job to assess whether the team members know enough to get the work done, or if they need assistance. If they need assistance, the leader must arrange it, or challenge them to find or develop it themselves.

**Team Moral and Cohesion**

Team Moral is improved by purpose, involvement, transparency, communication and recognition. Team members involved in decision making will own their performance outcomes and work harder to make
Teams with healthy recognition practices enjoy 31% less turnover and double the employee moral ratings, according to a study at JetBlue. An excellent book on recognition in the workplace is *Influencer: The Power to Change Anything*, by VitalSmarts™. This book details team performance improvements from public recognition and how moral can be magnified through specific, timely, and personal positive comments.

Team Cohesion is a function of team spirit, which is developed experience by experience. In the book *The Carrot Principle*, by Adrian Gostick and Chester Elton, the authors clearly show how manufacturing positive memories together creates strong team spirit over time.

### The Individual Needs of the Individuals on your Team

Conduct short one-on-one meetings regularly with each team member. Weekly or every-other-week is a good cadence. Spend 15 minutes or longer together discussing their goals, challenges, and roadblocks to success.

The meeting agenda should be set by the subordinate, with any appropriate feedback delivered by the manager. Andy Grove found that a little bit of a manager’s personal attention enhanced the quality of a subordinate’s work for the next two weeks.

These meetings are a great forum for personal goal-setting and progress updates, coaching and mentoring (long term coaching), performance reviews, and career growth.

### Production Velocity vs. Individual Needs

In 1964 Robert Blake and Jane Mouton developed their Managerial Grid model to articulate approaches for managing individual needs versus team needs. They organized their grid into two axes: Concern for People, and Concern or Production. Their model contains five approaches to balancing these concerns. Each approach is explained below.
Indifferent leaders tend to hide and evade from the team as much as possible. By protecting themselves from getting involved and potentially being part of mistakes, they think they are preserving their seniority and job security. Innovation and conviction suffer in these environments.

Accommodating leaders demonstrate a high concern for people at the expense of production. These leaders pay close attention to security and comfort of their employees, but generally do not lead strongly productive environments.

Dictatorial leaders find employee needs mostly unimportant, and lead transaction-based environments. Rewards and penalties are emphasized, and employees are expected to do their jobs with minimal personal conflicts. This can be an effective balance during a crisis or in highly competitive production environments.

Status Quo leaders attempt to balance employee needs with production goals, making compromises in each area as needed. This is a reactive mentality and minimal leadership is devoted to making up the difference.
Sound leaders are transformational. When conflicts arise between concerns for people and production, sound leaders innovate to find ways to cater to employees while keeping commitments and meeting goals along the way. Sound leadership requires ingenuity, creativity, and innovation.

**Production Velocity vs. Team Spirit**

Kurt Lewin, the founder of early social psychology who coined the term “The whole is greater than the sum of its parts,” helped us understand that within any team there needs to be two types of activities--those needed to achieve the team’s objectives and those needed to develop and maintain team spirit.

Dr. Eddie O’Connor expounded on these ideas by identifying four components needed to achieve excellent team spirit: Cohesion, Cooperation, Role Relationships, and Leadership.

**Cohesion** is how willing a team is to take care of each other and work together particularly during adversity. Bonding with a group is satisfying and meets several basic human needs. Cohesion can be based on different things and usually evolves over time, making teambuilding activities an ongoing requirement for long term teams.

**Cooperation** requires the team member to emphasis the team above him or herself, which is essential for effective teamwork. To make this easier, it helps to have a common set of team values and goals. We’ll discuss creating a team charter in an upcoming section about creating a team.

**Role Relationships** are important to establish and clarify. Ambiguity is the enemy of progress and will drain physical and mental effort in team members as they feel a sense of urgency, but don’t know what to do next. This leads to confusion and then a lack of cohesion resulting in reduced performance. Be sure to clearly define roles and decision-making power as you interact with your teams.

**Leadership** styles have an immense impact on team spirit and velocity. The next section is an overview of different approaches to leading teams. The differences are important to understand because different leadership styles affect the way team members feel about themselves. Be sure your style and words build confidence. Catch members doing something right
and compliment them for it. Celebrate achievements and make it apparent often that you believe in your people.

**Leadership Styles**

In the past, leadership was thought of as an attribute that only some people were born with and the rest don’t have. Today, the thinking has changed, and leadership has become more of a science. Here is a brief description of the more common leadership styles identified today:

**Transformational Leadership** is thought of as the most effective style to use. Transformational leaders inspire their team members because they expect the best from everyone, and hold everyone including themselves accountable for their actions. They demonstrate high integrity, emotional intelligence, a shared vision of the future, and they communicate well. They are self-aware, authentic, empathetic, and humble. They set clear goals and are good at resolving conflicts.

**Transactional Leadership** is by mandate. When an employee accepts a job, they agree to follow a leader. Transactional leaders enjoy influence due to position or title, and have the right to apply penalties if work standards are not met. This leadership style is ideal for employees seeking performance-based rewards within a structured environment. The high structure and rigid expectations, however, can curb creativity and innovation.

**Autocratic Leadership** is like transformational leadership except that the focus is on the approach the leader wants to take to achieve the team goals, instead of the approach the team wants to take. Although sometimes appropriate for short term tasks, this approach compromises team buy-in and is demoralizing during long term projects. This is sometimes called Charismatic Leadership.

**Democratic Leadership** is like autocratic leadership except that the team is solicited for ideas and feedback before a final decision is made by the leader.

**Bureaucratic Leadership** is present in high-risk, volatile environments. Work performed with toxic waste, expensive manufacturing, or the handling of large sums of money requires checklists and checkpoints to protect against unnecessary mistakes or misfortune. This type of
leadership must be applied carefully so as not to restrict innovation and creativity where needed.

Laissez-Faire Leadership is where a leader offers little or no guidance to group members and leaves decision making completely up to the group. This approach works well with a team of highly-qualified experts, but can also lead to low productivity and high-maintenance team members.

Servant Leadership is a facilitative-based leadership style where the leader clarifies objectives, constraints, roles, and expectations, and then challenges the team to come up with the plan to achieve the objectives. Then, the servant leader uses his or her authority to empower the team and remove roadblocks from their path.

Based on the situation, some styles are more appropriate than others. Like having a set of golf clubs, a leader will be most effective if they learn to use the right club at the right time.

**The Stable Framework™ “Servant Champion” Leadership Model**

No matter which leadership style is applied, Master Chiefs must develop the qualities shown in the Stable Framework™ “Servant Champion” Leadership Model below, and then use them to succeed within their teams.

In a nutshell, the model presents qualities that describe a “Servant Champion.” A Servant Champion is a leader who is committed to a goal, can express why it’s in everyone’s best interest, is competent enough to stay focused, and works to enable others to achieve the goal. All of this instills trust.

Any gardener will tell you; it’s not just about planting flowers--you must also pull weeds. To balance positive leadership qualities, we need to embrace teeth and backbone traits necessary for any leader to be successful. These are recessive traits, but nonetheless important, and are shown in (parenthesis) in the leadership model.

Warren Buffet said, “A leader is someone who can get things done through other people.” To understand leadership better, it’s useful to understand why people follow leaders.

Gallop Research interviewed 10,000 followers across a broad spectrum of
society and asked them what they experienced from the most influential leaders in their past. The four most common words were: Stability, Hope, Trust, and Compassion.

As a Master Chief, remember that your behavior, focus, and communication must create these four feelings within your people. The Stable Leadership Model will help you with this.

Adaptably Committed

The first step in leadership is being committed to a purpose. Leadership is about moving people from one state to another, and it’s easiest to do
that if your people understand why a change is needed. A leader must have a vision of the future and believe that the future is in the group’s best interest and be able to express why it’s in the group’s best interest.

Deming talked about the importance of maintaining a constancy of purpose with your efforts towards your vision. Relentless focus while working towards that vision is how great things come about.

Deming emphasized the importance of connecting workers with the customers their labors are helping. He said it is critical for them to understand the benefits their labors are contributing to the lives of the customers. This instills pride of workmanship and motivation in the teams.

As the team moves towards their goals, the landscape can change. New challenges arise, team members shift, markets change. It’s critical the Master Chief be adaptable to new approaches proposed by the team to reach their goals.

As a Master Chief, be sure your team understands your conviction for consistently improving customer understanding, customer experience, and system improvements using Stable principles. Be sure they understand it is a never-ending cycle. Your conviction will catch on as you transfer your energy into the team.

(Rejector of Distractions)

A Rejector of Distractions can say “No” to distractions. This is a life skill, and an important leadership trait, and represents one of the teeth and backbone traits required to be an effective leader.

Organizations focus on strategy by saying no to anything else. Teams focus on the work at hand by pushing non-priority work and conversations away.

A simple example of this ability is reminding a team in a Scrum or Kaizen Standup to move a non-essential conversation to the “Parking Lot.”

Trusted

A leader must be trusted. Your people must feel safe in your presence
and in the culture that you create so that they can be transparent and emotionally vulnerable with you and with each other. Emotional vulnerability is one of the primary elements of high-performance teams because it connects people to each other and reduces the noisy emotion and politics that slow down group trust and progress.

Trust is generated by a leader who demonstrates competence and character over time. Competence is the ability to read a situation accurately and apply the correct agenda, while Character is the assurance that a leader will consistently do the right thing, whether that team member is around or not. A history of observations by your team members that you are competent and have character will produce trust by your team in you.

It's also important to engage in team antics. Be a part of all the goofy challenges and group norms your team comes up with. You will lose credibility and respect if you are always on the outside looking in.

As a Master Chief, your past personal experience in your business domain and skills will add to your competence. Be sure you are a person of character. The combination of the two will enable people to trust you.

As a trusted leader your Process Owners will feel safe to work together in the culture you create and support.

(Rejector of Patterns of Mistrust)

As a leader you will be faced continually with opportunities to conceal information, or set a bad example, or take advantage of someone else’s efforts or assets. Your team is watching. Have the courage, teeth, backbone, and ethics to call out such options when they are suggested and move the team past them. Transparency and trust go hand-in-hand, and principles espoused from the top permeate down into the teams. Unfortunately, so does fraud, corruption, and dishonesty. As a leader, your teams will treat you and others the way they see you treat your superiors, peers, and customers.

Persuasive

A leader must have followers, or they are not a leader. In order to have
follows a leader must be *persuasive*. Persuasion is about communication and psychology. A leader must be able to communicate his or her vision and understand how people think and reason in order to put the vision message in the right context.

A good leader can express his or her message so that it calls people to action. This is done by validating the challenges and pain points of the leader’s team or customers and then stating an improved future vision that the leader believes is attainable. Depending on the situation, the leader can then provide a solution (subject-matter coaching) or challenge the team to create a solution (accountability coaching) that will bring the team from the current state to the desired state. This is a basic model for sales and is the basic formula for persuasion.

As a Master Chief helping your team succeed through Stable principles, you’ll often need to persuade your people to stay on track. You’ll need to help them understand the value of spending a little more time with their customers which will emotionally connect their customers to their work. You’ll need to persuade them to understand that a little more time spent doing rudimentary quality tasks such as completing checklists is much better than spending unscheduled time putting out fires and reworking inaccurate efforts for a second or third time.

*(Formidable When Necessary)*

At times you may need to defend your ideas, or correct a senior colleague giving you bad advice. This takes courage and a backbone.

I remember a long time ago having a network admin who insisted on installing all .DLL files to a central network location, and not on any local workstations. It was a mystery why our $100,000 database ran so slow across the Enterprise. I was a junior programmer and lacked the courage to suggest that maybe the two issues were related, and that we should conduct some multi-scenario speed tests.

Many years later a CIO insisted I allow a less-than qualified, but favored DBA to participate in a high-profile server upgrade. My inability to say no led to myself and the CIO both losing our jobs over the incident. I wish I knew then what I know now.
An emphasized leadership style in the Stable Framework™ leadership model is Servant Leadership. Servant Leadership is a concept put forth by Robert K. Greenleaf in a book by the same name written in 1977. In his book, Greenleaf suggests that the best leaders empower their peers and direct reports and help them become leaders themselves. They give them objectives and constraints but challenge them to solve their own problems. Servant Leaders use their authority to remove roadblocks from their team’s paths instead of commanding them in every matter. Indeed, they are accountability coaches for their though-worker teams.

Servant Leadership requires respect for people and facilitation skills. As a Master Chief, you must believe and trust that your people can generate excellent ideas and solutions to problems. Avoid the impulse to provide the answers, and instead use curiosity and involvement as tools to get your people bought in. Get them to come up with options by asking questions like:

- What’s the best way to get started?
- What are the major steps along the way?
- What kinds of risks might we encounter if we did that?
- What other options can you think of?
- Who else could you ask?
- How would we know if it’s the best decision?
- Is there an option that could get us there faster?
- Could we achieve the same result some other way without spending any money?
- Who wants to take the first step?

Jack Welch, the retired CEO of General Electric, wrote about this technique in his book, Winning. He called it the “Dumbest guy in the room.” technique. He said you ask all the questions and soon your team has come up with the entire plan. If you do it right when you leave, they will say, “What did we need our leader for? We came up with the whole plan by ourselves.”

Facilitation is an important part of servant leadership. A facilitator deals with the meta aspects of interpersonal behavior, initiates conversations and meetings, creates appropriate space for dialogue and discussion, and
uses gatekeeping skills to encourage quieter people to share more while throttling back the continuous energy of the more outspoken participants.

A facilitator also works behind the scenes to prepare for these events, and to remove roadblocks from his or her team’s path when possible. As a Master Chief, you will constantly be facilitating for your teams.

(Protector When Necessary)

At times you may need to protect your team. When attacked, have the humility to listen to understand the other party’s concerns. Restating them is an excellent communication technique in these cases.

As a leader, have the backbone to defend their hard labors, and difficult changing environment they must work within. Remind your concerned aggressor that your team members are Dad’s and Mom’s and somebody’s sister. They have a world outside of work they must succeed in also and they are doing their best to make the right decisions.

Continual Self-Improvement

At the heart of a successful Master Chief is continual self-improvement. A Master Chief is always learning, and always favoring improved ways to deliver value over time. Master Chiefs must also encourage sustained continual improvement in their environment and within their people. Master Chiefs will be unable to do this well unless they are always improving themselves along the way.

Deming talked about personal learning being a foundation of leadership in a group setting. Constantly share with your team what you are studying and learning about. Encourage book clubs and brown-bag lunch sessions where your Process Owners are instructing each other on better ways to live life. Make the workplace an asset in your team members’ lives.

In Jim Collins book, *Good to Great*, his team of researchers analyzed leaders of top performing companies and discovered that most of these effective leaders don’t profile as magnanimous giants of industry. In fact,
quite the opposite is true. Most of the effective leaders his group examined were humble, down-to-earth performers, who had constructed effective teams. This is a worth noting.

Adolescent leaders try to do everything alone and not tap into their teams. In some cases, they even wastefully compete with their own teams. As leaders mature, they learn to harness the power of their teams by empowering the teams to make the important decisions and showing them that they trust their team’s judgment.

Deming is famous for his fourteen points of leadership, which he writes about in his book, Out of the Crisis. Every Master Chief and Process Owner should familiarize themselves with these points. They make an excellent set of topics for a book club or regular discussion.

Peter Drucker, the famous management educator and economist, described the presence a leader needs to have. He said, first, that leadership is not a position, but rather it is a means to get some future objectives accomplished. Leaders should see leadership as a responsibility, and not rank and privilege. When things go wrong—and they will at times—the leader must own the situation and blame themselves. Be prepared for things to go wrong occasionally—and be prepared to own the failure and blame yourself, not your team. Failure can be an awesome learning process, but if the professional will not own the failure, they will have learned nothing and, given the same scenario in the future, will repeat the same mistakes.

In addition, if you own the mistakes of your team, you will experience great joy and exhilaration in their successes. Be sure to always give credit for successes to your team. This protects you from competing with them as their leader.

A leader must be visible. They must be front and center. They must own
the airspace. If they don’t own the airspace, it will be filled with other things from other sources, which will confuse the team.

A leader must be right. Leaders that are wrong quickly erode the patience of the team. Some of the greatest leaders in history had no charisma but were right. Look at Albert Einstein during World War II who got the attention of President Roosevelt to warn him that science had advanced enough that the Axis powers could be working on the development of the first Atomic Bomb. He was right.

A leader must also understand that the domain size of his or her own areas of connection and influence exceeds the domain size of any of their direct reports. While direct reports have their own workplace environments, the leader’s environment is always larger. This gives the leader access to people, information, and resources the direct reports don’t have. Respect this difference and own it. This additional reach empowers you to help your team succeed, if you use that reach wisely.

In addition, be aware that as people we are all equals, but as roles we are not. Inside the building, your word has greater application and impact than theirs, but outside in the parking lot you are both equal human beings.

Find out about their career goals and life goals and help them any way you can. Remember that you are working with someone’s sister, or someone’s dad, or someone’s husband. Be sure your people know that you know they have families outside of the workplace that need them, too. It’s okay to learn about people’s personal lives and provide support. Help them understand you want them to succeed in life just as much as in the workplace.

John C. Maxwell talks about how people will follow a leader because of his or her title for a short time, but they are really watching the leader to determine if the leader wants them to succeed. If they believe their leader wants them to succeed, they will follow that leader for many reasons.

**Personal Leadership**

Four elements are fundamental to career success:

1. Communication and networking skills
2. A strong work ethic
3. Good decision-making skills
4. The ability to deliver results with your team

The challenge is, if you focus on any one of these too much, the others will suffer. As a leader, you must find quiet time to reflect on your goals in these areas and be sure you are not neglecting any of them.

Periodically ask yourself if you have any working relationship that need attention. Are there any skills that you need to develop or polish up on? Is there a problem developing that you can address now? Is your current performance measurement system addressing all the questions that your leaders ask? Stay focused on these questions as you work with the PRO’s coordinating the Value Stream results in your area.

Building the Right Culture

Culture is defined as the anticipated reactions between two or more people, and it varies from place to place.

In America, for example, when somebody says, “What’s up?” the only proper reply is the exact same expression, whereas in another country the question would certainly be confusing.

Just as international cultures differ, cultures exist and differ from one work environment to the next. Culture is a combination of group norms, group history, elements in the environment, and the personalities that each team member possesses in the group.

In a workplace environment, culture is often categorized along two continuums: Independent vs. Team Focused, and Flexible vs. Rigid. Your organization can be anywhere within these two continuums.

Independent cultures tend to be more competitive whereas team focused environments tend to be more supportive. Rigid cultures tend to have more inertia and be less tolerant of failure, whereas flexible environments allow for more innovation, creativity, and new ideas to flourish.

Independent cultures tend to produce more work over time, whereas team focused cultures tend to make safer decisions.

The beauty of the Stable Framework™ is that it offers stability in the
right places, allowing for innovation and flexibility towards improvement over time.

Independent cultures tend to produce more work over time, whereas team focused cultures tend to make safer decisions.

It also offers the team a model for working independently yet harnessing the best minds in the group for Kaizen Blitz problem solving, and Kaizen Improvement Projects. In other words, it provides mechanisms that support all four extremes.

Changing a Workplace Culture

You may be starting with a group that is not used to teamwork, or not used to flexibility. One of my clients was appointed senior project manager over a team of experienced project managers. She was surprised that her new team wanted assistance making decisions about every detail in their projects. My client reported to me that the team members acted confused when asked to make a decision. I asked my client how the team described the previous manager and I was told “an extreme micromanager.” I pointed out to her that the team was used to that culture and was not yet ready to be given the flexibility to make their own decisions but would require her leadership to shift into that paradigm. She was experiencing a culture clash.

Changing culture is historically challenging. Many corporate transformations failed after the employees were trained properly, and after the new tools were put in place, simply because the culture had not been addressed.

Transformations usually require a change in culture to be successful. Therefore, as a Master Chief you must know how to do this. Fortunately, there is a powerful set of tools for doing this.

First and most powerful is the creation of team memories. Memories manufactured over time, with the same group, create a sense of a team cohesion. It doesn’t matter what the memories are, it’s just important that they are frequent, and hopefully positive.
Eat together. Have a piñata activity after each significant victory. Get out of the building and have a movie night for team members and their spouses. People will tell you things outside of the building that they will never say inside of it.

Next, talk about desired outcomes. Culture is also shaped by what people talk about. Own the airspace. State the obvious. Become a broken record by repeating the desired outcomes. Say it so many times that your team members start to expect to hear it from you.

Pick your team members carefully. Jim Collins famously talked about having the right people on the bus. Look for team members who champion where you want to go and give them prominence. Draw attention to their victories implementing the new ideas.

Remind the teams that you are going through a transformation and it’s healthy to talk about what’s different now, and why it was needed. Encourage them to rely on each other for help.

Finally, in some cases, you may have to move people and furniture to help support the team.

Establishing team spirit and changing culture is empowering. These principles work not just in the workplace, but in family dynamics, neighborhood relations, and beyond.

Creating a Team

TOGETHER
EVERYONE
ACHIEVES
MORE

A great beginning with any team is a team branding ceremony. Make this a memorable event. Call a meeting specifically for drafting a team charter. A team charter consists of an identity, vision, mission, values, and goals statements.
Of course, the objectives of the team are pre-set by whoever organized the team, but the team is free to decide how they are going to operate along the way.

An Identity Statement is an expression of who the team members are. What do they identify with? Are they the best developers of their codebase in their area? Are they video-gamers turned coders? Are they athletes with programming jobs? Who do they identity as?

A Vision Statement is a description of a future state the team wants to bring to pass. For example, the team might agree that they want their sponsors to think they are the best team they’ve ever had, or that they are all great friends a year from now, or that they will all be masters of certain aspects of their trades by next year. Some groups think of their vision statement as their “North Star.”

A Mission Statement is a description of how the team will accomplish their objectives and their vision. For example, they might say “We will continually learn about our customers, innovate, and improve our processes to make our services better, and we will take care of each other along the way to make our workplace relationships better over time.”

**Our Team Charter**

- Identity Statement
- Vision Statement
- Mission Statement
- Team Values
- Team Goals (Short Term & Long Term)

A Value Statement is a short list of prominent values the team agrees will govern their decision making and their behavior. Values like “respect for people,” “assume positive intent,” and “punctuality” are good examples of team values.

Finally, a Goal Statement might be a combination of long-term and short-term goals. These might be to raise customer satisfaction ratings monthly or to have all the critical processes listed on the performance console.
Have everybody sign and date the team charter. Frame it and hang it on the wall in a public place. As new employees join the team, take the charter down and have the team re-evaluate the charter together with the new employee. It will be a great memory supporting the new team structure and a welcoming experience for the new employee.

One basic component of successful teamwork is the “one for all and all for one” principle. Help the team understand they are each an integral part of a larger entity and they must do what they can to support the team, and support each other.

A second basic component of successful teamwork is the willingness for every team member to give and take direction when appropriate. If even one team member struggles in this area the whole team will suffer.

An excellent book titled *Collaboration Explained*, by Jean Tabaka, contains many techniques and great advice for working with small and large IT teams.

```
Demand their best work by allowing for errors, but not failure.
Errors not corrected become failures.
```

Be resourceful and allow talent to thrive. Create a safe but challenging environment where people are free to try new things and make mistakes. Demand their best work by allowing for errors, but not failure. Errors not corrected become failures. Process Owners should be tasked to learn what they can from their errors and share it with the group. They should be challenged to work hard and always be stretching to improve.

Teamwork makes the Dream Work!
Be sure to encourage their hard work and not their intelligence. Intelligence and hard work are not the same thing. Process Owners need to know they can continually work hard on a challenging problem and eventually overcome it. People will give up if they think the problem is too difficult for their intelligence level.

One of the greatest leaders in modern times was Vince Lombardi. As the newly appointed head coach for the Green Bay Packers in the late sixties, Vince led a team of mediocre unproven athletes to win five out of the next seven Super Bowls. There has never been a coach with that kind of record before or since.

Vince never wrote a book about his leadership style but his son did. How to be Number One, by Vince Lombardi Jr., exposes one of Vince’s great secrets to motivating his team. Vince understood that people are different and respond to different forms of motivation. Some people like public praise. Others like a private thank you. Some people like pressure, others like hallway conversations. Vince kept a record of what his people responded to and used it to keep them motivated. Learn to read your people and understand how each one of them is motivated.

The Coaching Kata

A Master Chief must be more than a problem solver, they must build Process Owners who can solve problems.

When a Process Owner or a Kaizen Blitz team is stuck on a challenging problem, a Coaching Kata exists for the Master Chief to encourage their progress. As a coach, you want to keep them focused. Toyota had a lot of success using Coaching Katas. The Stable Coaching Kata has these questions Master Chief’s use while interacting with others in these scenarios:

1. What is the current situation vs. the desired situation?
2. What needs to happen to reach the desired situation?
3. Who could we ask, or what models exist that we could learn from?
4. What is the next step?
5. How soon can we learn from the results of the next step?

Coaching is like a marriage. You need to maintain a 5:1 ratio of encouragement verses scrutiny. You must be sincerely interested in your
Process Owners ideas, growth, and victories.

**Group Problem Solving and Conflict Resolution**

Although it’s natural for participants in a team to be primarily interested in their own ideas and accomplishments, one of the non-intuitive realities of teamwork is that the best way to maximize an individual’s progress is to harness the power of the team.

It’s important for team members to be accountable to each other. Good teamwork creates a sense of safety that lowers stress, increases focus, and speeds up everyone’s efforts. It increases communication which brings faster innovation and resolution to problems. Turnover diminishes and productivity goes up, producing better overall results.

Encourage debate, involvement, and critical thinking among the team. Help them understand the best decisions are scrutinized by all sides so that they are made with the least bias. Also, help them understand that measurement removes bias and is an excellent foundation for knowing where we are now and contrasting it with where we want to be.

**The Team Development Model**

Bruce Tuckman introduced the Team Development model in the mid-sixties which has become a standard in understanding team behavior.
When you are starting with a new group of people you don’t have a team yet—you have a group of people. You, as the leader, must develop your group of people into an empowered team.

Several stages of evolution transpire as the dynamics within the group evolve from a collection of self-aware people to a single group with a team identity. Bruce Tuckman identified these stages as Forming, Storming, Norming, Performing, and Adjourning.

The dynamics of each stage follow:

**Forming:** At this stage, a group of people meet each other and become acquainted. Introductions are made, and first impressions are generated.

**Storming:** At this stage, the group has been interacting a little and sampling each other’s personality characteristics. Loud people are favored in these situations. It’s important for you to be heard, and to use relevant industry vocabulary. Experts identify each other within groups by their use of correct vocabulary. Natural leaders, class clowns, quiet thoughtful answer holders are all identified in this stage.

**Norming:** This stage is where the individuals accept each other, agree to trust each other, and proceed to the next step to start performing as a team. We’ll see in the next step how problems in this step slow down performance in the next step.

**Performing:** This is where teamwork gets done. Depending on how well the Storming and Norming transitions happened, in this stage the team either performs with full speed, or they limp along, crippled. Effective team members can give and take direction. If even one member can’t do this, groups won’t norm properly, creating unnecessary tension and a lack of trust within the team.

In extreme cases called “Apollo Teams,” there are too many Type-A (extremely assertive) people and the team reaches gridlock and cannot seem to get any work done. The key here is vulnerability. A vulnerable team trusts each other and can collaborate better. Vulnerability means each team member feels safe expressing themselves and that they will experience no negative repercussions for it.

**Adjourning:** This stage is, of course, when the team has completed their work and is disbanding. If it was a long-term engagement, you really should have a party or fun memorable event to end the experience.
properly.

Experts identify each other within groups by their use of correct vocabulary.

Apollo Teams

In Apollo Team situations, you may need to shuffle some people to other teams or be a 3rd party facilitator when the team needs to make important decisions.

One group I am aware of incorporated a "survivor" theme within their multi-team environment. Team members were empowered to vote each other off of their teams if they contributed to significant problems slowing down the team. The manager reported a single team of “outcasts” emerged which in short time became one of the highest performing teams in the organizations.

Repairing Team Rivalries

Sometimes in coordinating various teams that must work together, you may encounter two teams who don’t work well together. Research was conducted on how to solve this problem in a book called, The Human Use of Human Resources. The authors concluded the way to get two conflicted teams to cooperate is to create a series of situations where members on both teams are needed to get the project done. Several scenarios where this approach succeeded are detailed in this book.

The Team Empowerment Model

The Team Empowerment Model is used to develop a foundation for Servant Leadership. Servant Leadership is a leadership philosophy that espouses creating clear objectives, constraints, and intentions, abdicating decision power to the team, and developing people within the team so that they can perform as well as possible. The legitimate leader is then tasked with using his or her authority to remove roadblocks the
As a servant leader, the Master Chief begins by providing the objectives, intentions, constraints, and boundaries for the Process Owners, but then focuses on empowering the Process Owners to be self-directed, champion owners of their processes.

This means the Master Chief pushes the decision making to those closest to the information, letting their team members make the decisions if the decisions don’t require extra money or jeopardize anybody’s safety or violate any constraints.

Once the team is moving forward, the Master Chief must learn to delegate the workload to the team. This enables the MC to be agile enough to move anywhere within the environment to deal with exceptions as they arise. In other words, the MC delegates the workload and then deals with exceptions as they emerge.

The steps forming the Team Empowerment Model are as follows:

1. **Bring Objectives, Constraints, and Dependencies to the Team**

   The first rule in leading a team is to understand that the objectives for which the team was organized come from outside of the team and should be delivered by the leader to the team. The same is true for any governing constraints. If a team is floundering, not aware of their objectives or constraints, the leader has not yet done his or her job. Some people call this “intent-based leading,” or “communicating intent.”

2. **Challenge the Team to Come Up with the Solution**

   The newly formed team should then be free to identify how they will achieve their goals. This can be difficult for an inexperienced leader, who hasn’t yet learned that ideas multiply when the team is having a healthy dialog. This is a way to push the decision making to those closest to the information.

   Be sure to remind and encourage the team to learn what they can from past experience. Past project histories, and feedback from engineers and executives who were part of similar past projects can be invaluable.
Depending on the scenario, the MC can play the role of a subject-matter coach, or an accountability coach. If the Master Chief happens to be a subject matter expert having experience with an objective and is instructing a new Process Owner, they can take on the role of a subject-matter coach. This kind of coaching helps the trainee to perform better by providing them with pre-existing knowledge and expertise concerning a work item. This obviously adds value to the learning Process Owner.

If the Master Chief has no previous experience with the work item, they then take the place of an accountability coach. An accountability coach is different from a subject-matter coach in that an accountability coach is an “accountability partner.” This person does not necessarily have knowledge about the objective but acts as a sounding board, a third-person looking in, and a dependable agent who will challenge the person being coached to stretch and grow, and to be accountable for commitments made. Much of the work of developing people is accountability coaching.

No matter which coaching role is used, the Master Chief must play the part of a shaper, challenging the team and its individuals to improve. Master Chiefs do this by continually challenging the Process Owners to understand and report on the reality of their environments and then brainstorm ways to improve those environments. The clearer everyone involved understands reality, the better decisions they will make. The better the decisions that get made, the more successful the organization will become.

3. Provide High-Focus Engagement and Support and be Sure the Team is Working Towards the Goal

This is the high-touch stage of leadership. Be present and available to your team. Using the techniques of observation and curiosity, be sure your people have a plan and are working it. Your objective is to build a team of problem solvers, and not be the person solving all the problems.

Don’t comment on the validity of their plan. If you have concerns express them in question form. Ask if they are certain it is the best way to proceed? Is it the quickest? Lowest risk? Least expensive? Have they discussed it with others? Is there someone who has already done something similar whom they could ask for advice? What other ideas
have they considered? How will they know when they have succeeded?

4. Equip the Team for Success

The fourth responsibility a leader has is to make sure your team has all the equipment and resources needed to succeed. The best way to do this is to ask the team members what else they need to be successful and then act on what they say.

Be careful to separate their needs from their wants. Needs should be met as soon as possible and wants can be bargained with. Be sure they are familiar with all the locations for Stable artifacts: The Configuration Management System, Standard Operating Procedures, Kata Card Templates, Asset Recovery Models, etc.

If you are unable to provide them with a necessity, explain why and challenge the group to brainstorm a workaround.

5. Delegate, Delegate, Delegate

The fifth responsibility of a leader, once the leader is convinced his or her team knows the objectives and constraints, is equipped for success, and is moving in the right direction, is to delegate the workload to everyone on the team.

There are different schools of thought on this. If you are a Master Chief on a small team then you probably should be busy helping the team do the work. If you have a sizeable team—8-10 or more—you should delegate all the work to your Process Owners and you then stand back and handle exceptions as they occur. This is called “Management by Exception.”

If you are busy working on the details every day with a large group, you will miss problem flags and then as these problems become critical issues everyone’s work will suffer.

Learn to delegate effectively. You will become a better leader and your people will feel more trusted as you delegate and follow-up.
6. Use Positive Psychology

The most important motivating factor an employee can have is to know they are helping another human being. Deming called this “Pride in workmanship.” He emphasized this heavily and believed every other form of motivation was a distraction. He was against short-term rewards and motivational slogans. He was even against paying performance bonuses. He emphasized the need to establish and elevate the connection between employee and final customer, and said once that was on place, no other incentive was required.

Without exception, when managers have complained to me that their employees don’t seem to care about their work, they also report that the same employees are disconnected from the end-users their efforts are supposed to help. This is a missing critical success factor in establishing pride in workmanship within a team, and it’s the leader’s job to ensure this connection is constantly in place. Employees must know who they are helping and have a clear understanding of their customers’ everchanging needs.

**Feedback**

Consistent one-on-one meetings are an excellent place for feedback. When providing feedback to subordinates, an excellent model developed by Kim Scott has become popular, called “Radical Candor.”

She explains the boundaries you must stay inside while delivering feedback. You can be too empathetic and enable an underperforming employee’s less effective behavior, or you can be a bully and harass your subordinate for underperforming. (As a side comment, a 2007 survey conducted by WBI-Zogby found that 72% of bullies in the workplace were the employee’s manager!) You can read more about this method in Kim Scott’s book, *Radical Candor*.

**Redirection**

In the book, *Whale Done*, Ken Blanchard describes a simple but effective way to train killer whales that works just as well with people. The pattern is called “A-B-C.” “A” is the Activator, or goal that a person is challenged to accomplish. “B” is the Behavior that must be observed. “C” is the
Conclusion of their actions, being correct or incorrect.

If they are incorrect, the leader performs a technique called “redirection” where the leader steps in and says, “I must not have been clear in explaining my intent. You'll need to do ...” and then restates the goal for the employee. This way the employee feels safe to make a mistake, knowing the leader will accept responsibility for the blame.

Ken Schwaber, the co-inventor of Scrum, taught me that when a student in a class doesn’t understand something, the instructor should always assume the blame for the lack of understanding. “I’m sorry I confused you,” or “I’m sorry I didn’t explain that very clearly,” and then covering the topic again is an excellent teaching technique.

These techniques breed a culture of safety. People feel like the leader respects them and wants them to succeed.

When complimenting your team, one good technique is to carry around a 3x5 card with all your PROs names on it. Keep track of whom you’ve complimented recently by putting a mark by each PRO’s name when you compliment them. This way you can make sure your comments are distributed throughout the group so that nobody is neglected.

Although you are trying to keep your compliments stratified, be sure to keep them honest. People can detect fake or shallow compliments.

Wherever you can, compliment your people in front of others whom they respect. This makes a compliment a thousand times more powerful.

**Dealing with a Needy Employee**

Some employees have learned to take problems that they should be able to solve themselves to their leaders for assistance. If you have an employee like this, simply say to them, “If it were me, I’d research ___ on Google,” or “I’d ask ____ how to do that.”

Don’t do it for them, or you’ll disempower them. Remember, one role of a Master Chief is to develop the Process Owners into empowered problem solvers. Jack Welch, the retired CEO of General Electric, wrote in his book *Winning*, that developing people should be about 40% of a leader’s efforts, not 4%.
**Dealing with an Irresponsible Employee**

Sometimes you may have to deal with employees who blame other people for their inability to get their work done. There is a model called “The Victim Triangle,” which is comprised of a victim, a persecutor, and a rescuer.

In the model, the employee presents his or her excuse for failure by blaming someone else or some other factor. This other person, being blamed becomes the persecutor. Your employee is hoping you will become the rescuer by telling them you understand the story and empathize, and that the failure is dismissible this time, but next time the employee should just try harder.

This is dysfunctional. This person learned somewhere in their past that he or she can get rescued from responsibility by having a good enough victim story. A chain is only as strong as its weakest link, and this behavior impacts the whole team negatively.

The way to help mature a person out of this recurring distortion is to challenge them out of this behavior. Point out that the employee’s failure hurt the entire team and ask what the employee could have done differently in that same situation to produce a successful result. Challenge the employee to do that in the future if a similar scenario arises. A self-proclaimed victim should be challenged out of any disabling behavior instead of consoled into repeating it.

**Dealing with a Challenging Employee**

Strange as it seems, you will experience employees who will challenge your authority. This is one of the most difficult behaviors to deal with and is most often experienced when managers are new.

If they ask you a challenging question, it’s important that you answer it. You have a lot of options, though. You can answer it yourself. You can solicit the team for an answer to the question. You can ask the team to suggest a source for the answer. You can also mention it sounds like the employee has a better idea and you’d like to hear it.

Be sure to express yourself from the perspective of what’s best for the customer, the sponsor, the work product, or the team. These
perspectives are always safe territory.

If your judgment is challenged make a firm effort to explain clearly why you made the decision. If the pattern becomes recurring and toxic, read on.

**Dealing with a Belligerent Employee**

Sometimes in rare situations, an employee may demonstrate unacceptable belligerent behavior. If they do this publicly, simply announce to them that you will need to see them right after the meeting, or that they will need to come and talk with you privately.

Meet with them. Highlight their value to the team as an individual contributor, but point out their behavior is eroding the safety and trust of the team, which hinders team performance. Never make it an issue between them and you, focus on the negative impact their behavior is having on the team’s performance. Insist that something must be distracting them, triggering this behavior, and challenge them to be open with you.

Be sure they are heard. Listen for whether or not they are being vulnerable. If not, they are probably not sharing the root cause of their distress.

Be sure you are not being an unreasonable leader. Unreasonable leaders make unreasonable demands on their employees. Your better employees will attempt the unreasonable demands, but your belligerent employees will publicly laugh at you—and they’ll be right. One way to succeed with belligerent employees is to not issue unreasonable demands. If the team is in a tight situation, ask them for creative resolutions, or ask for their voluntary help with the situation. Be sure you will be there with them, participating in the unreasonable request.

You may be missing a leadership pattern within your environment. Be sure you are regularly performing brief one-on-one’s regularly with everyone, and try to get each individual out to lunch once a quarter. These short meetings are excellent opportunities to demonstrate your concern about the success of each of your team members as a contributor and as an individual.

Challenge you employee to channel their frustrations into positive results
within the environment.

My friend Christian Moore, who has made a career out of speaking to rooms full of educators and prison administrators, has pioneered the channeling of negative energy for constructive use in his book, *The Resilience Breakthrough*. He explains that people’s emotions contain both positive and negative energy. Positive energy is easy enough to channel, but it’s what people do with their negative energy that separates geniuses from felons.

When Steve Jobs was fired from Apple, the company he founded, instead of acting out publicly he took his negative energy and started NeXT Computers, and Pixar. He had so much success, Apple asked him to come back.

If their behavior improves, be sure to thank them and encourage the positive changes. If you don’t, they will be confused and think you didn’t notice.

If the unwanted behavior continues, you may need to work with your leaders and put the employee on a 90-day progressive improvement plan. Council with your Human Resources department for guidance with these situations.

If nothing else works you may have to be firm with them and tell them their behavior indicates they are in the wrong place and if they don’t manage their career you will have to do it for them.

**Dealing with Team Conflict**

Whenever there is conflict within the team, always focus on the problems, and not the people. You may have to remind your teams to keep that focus.

Teach your team to respect conflict, and see it as a portal to a better solution.

In addition, teach your team how to express themselves from the perspective of what would help the customers, or the project, or the company, or the team the best. If they can learn to state their opinions from those perspectives, everyone will be on safe ground.
7. Lead Them to Reality

Measurements reflect reality, and part of a Master Chief’s role is to challenge the team to create effective measurements to indicate their progress. Measuring should not be used to reward or punish, but rather to show reality.

After measurements are available, the team can brainstorm better ways to improve with even better types of measures. Measuring is a window into reality and the secret to accountability and improvement. What gets measured gets done, and can be improved.

8. Remind the Teams to Systematize so that they can Learn from Their Past

Your teams will be busy brainstorming, interfacing with customers, completing Kata Cards and SLAM Charts, performing process work, and getting ready for the next Master Cycle. They will need you to continually remind them of sustainable continual improvement activities until the culture shifts enough that it becomes second nature for them, and they are performing systematized continual improvement activities by themselves.

9. Laugh and Have Fun

Before people studied workforce behavior, laughter was thought of as wasting time. Management projected a facade of somberness and stern personalities to keep workers focused.

Companies are discovering laughter has amazing benefits for the workplace. Among other things it reduces stress, reduces resistance to change, increases morale, disarms anger, improves attitudes, breaks the ice, increases trust, and improves team cohesion.

Have fun together and create positive memories as a team. It will make the workplace a welcoming environment.
Three Models of Excellent Teams

Three models of excellent teams every Master Chief needs to be familiar with are published in three different books.

In Jim Collins’ book *Good to Great*, he talks about the Hedgehog Concept. This concept is where a team is characterized by having three consistent qualities in alignment with their work. These consistencies were found in almost every world-class company Collins’ team of researchers studied. The questions that identified these qualities were:

1. What are you passionate about?
2. What drives your economic engine?
3. What can you be best at in the world?

In most cases, every world-class company felt passionate about its work and felt it could be world-class at it with the right management support.

In Marcus Buckingham’s book *First Break All the Rules: What the World’s Greatest Managers Do Differently*, Buckingham’s Gallop Poll team discovered that top performing companies had employees who for the most part could answer yes to the following twelve critical questions. Ask your team these questions and ask for feedback on how to bridge any gaps.

The questions are quoted from his book:

```
1. Do I know what is expected of me at work?
2. Do I have the materials and equipment I need to do my work right?
3. Do I have the opportunity to do what I do best every day?
4. In the last 7 days, have I received recognition or praise for doing good work?
5. Does my supervisor or someone at work seem to care about me as a person?
6. Is there someone at work who encourages my development?
7. At work, do my opinions seem to count?
8. Does the mission or purpose of my company make me feel my job is important?
9. Are my co-workers committed to doing quality work?
10. Do I have a best friend at work?
```
11. In the last 6 months, has someone talked to me about my progress?
12. This last year, have I had the opportunity at work to learn and grow?

“Finally, Google spent two years studying what makes a great team and concluded the following five factors are critical:

1. Psychological Safety – The ability to be honest, and even vulnerable with other team members without repercussions.
2. Dependability – The team could depend on each other and had confidence that work items would get completed by their teammates when expected.
3. Structure and Clarity – The teams had clear goals and constraints.
4. Meaning – The work had personal significance for each member.
5. Impact – The group believed their work was purposeful and that it positively impacted the greater good.

Are you beginning to see the pattern? Employees in all these groups work in an environment of respect, trust, vulnerability, purpose, empowerment, structure, respect, visibility, and success.

The Future Domain and the Master Chief

A Master Chief starts the Quality Planning process when the Stable Framework™ is first implemented by establishing the structural components within the environment. Every six months or so the Quality Planning process should be revisited.

During Quality Planning, The Master Chief is responsible for establishing the Configuration Management System and all the structures inside of it. Next, the Master Chief creates the System Schedule and begins populating everything that must go on it.

Finally, the Master Chief creates the Customer Request Queue, Scheduled Activity Queue, Equipment Maintenance Queue, and the CCAPA Queue.

In addition, the Master Chief must establish the Performance Console
Future Domain

System Configuration

Process Asset Library (PAL):
- **Service Register**
  Master Services, Pending Services, Retired Services, Supplier Services; Customer & Supplier contact info
- **Process Controls**
- **Asset Controls**

System Schedule

- Annually (1/Year)
- Bi-Annually (2/Year)
- Quarterly (4/Year)
- Monthly (12/Year)
- Weekly (52/Year)
- Week Day (1/7 M-S)
- Working Day (1/5 M-F)
- Daily
- Day of Year [Date]
- Week of Year [Week]
- Month of Year [Month]

System Backlog

- Scheduled Activity Queue
- Customer Request Queue
- Asset Maintenance Queue
- CCAPA Queue

Figure 48 - The Future Domain

and the historical logs for retaining past information over time.

The Master Chief assumes the role of the champion of process innovation in the Future Domain. Master Chiefs are always challenging the Process Owners to improve the results and efficiencies of their processes. They own the System Schedule but work with the PROs to put their scheduled events on it. This is done in the Master Cycle Planning Meetings.

Once the team is up and running, as the Process Owners receive customer requests, they place them in the Customer Request Queue, or in the Objectives column on the Kanban board if they are urgent.

As activities listed in the System Schedule become imminent, the Master Chief puts them into the Scheduled Activity Queue. This should happen at the beginning of each Master Cycle Planning Meeting.

All items added to the queues should be categorized and then timestamped. They should be timestamped repeatedly as they move through each column on the Kanban. This provides empirical information for estimating lead times for future similar activity requests.

Nonconforming product or service problems that are experienced by customers go into the CCAPA Queue for review and action. These problems are brought to the team usually from the Service Desk. The
Service Desk should pool common problems and bring them to Engineering or Operations.

The CCAPA Committee is a combination of representatives from the Service Desk, Engineering, and Operations who receive non-conforming problems experienced by customers and then prioritize them for remediation based on impact. Some groups hold CCAPA Committee meetings daily, and some hold them weekly or once per cycle. You might consider holding daily meetings after a major release and then weekly or once per cycle until the next major release.

*The Present Domain and the Master Chief*

The Master Chief sets up the Sprint board and Kanban board, as needed, and the recurring schedule for the CCAPA and Change Control committees during Quality Planning.

They do this, of course, with input from the other committee members, and the meetings can change as needs change.

**Present Domain**

**Team**
- Daily Kaizen Standup
- 5S Housekeeping
- Adhoc Kaizen Teams
- Process Kata & Kata Cards
- Training/ReTraining

**Committees**
- Change Control Committee
- CCAPA Committee

**Environment**
- Alert System
- Production Change Log
- Suggestion Box

**Figure 49 - The Present Domain**

The CCAPA committee, particularly, can be held once a week, or once each Master Cycle, but may happen daily for a few days right after a
release, and then return to its normal scheduled cycle. This way the organization can be ready to recover from a bug that made its way through testing and gets reported early by many customers.

In addition, Master Chiefs coordinate the Master Cycle Planning Meeting, the Daily Kaizen Stand-up meeting, and the Cycle Review and Cycle Retrospective Meetings.

**Master Cycle Planning Meeting**

At the beginning of each Master Cycle, the Master Chief calls the Cycle Planning Meeting, which he or she attends with the Process Owners. In this meeting, they examine the System Schedule and the Work Queues in the System Backlog and load any urgent items from the work queues into the Objectives column of the Kanban board. The MC moves the most critical items to the top. This enables the team to visualize their workload. Visualizing workflow in this manner exposes unexpected slowdowns and unnecessary distracting side work.

Depending on the environment, the MC can meet with a representative group from the Service Desk to discuss any new nonconforming issues that were reported from the customer base since the last meeting. Some groups do this every morning before the daily Kaizen Stand-up Meeting, and some groups just do this as an agenda item during the Master Cycle Review Meeting.

The MC should also be getting updates from Engineering regularly so that the MC is aware of any new additions or changes to products or services. The Master Chief could attend the Engineering Status Meetings or invite an Engineering representative to their Cycle Review Meetings.

If there is a project team involved, they can come prepared with a Gantt Chart, if needed, and a Product Backlog with activities to load into the “Selected Tasks To Do” column.

**Daily Kaizen Stand-up Meeting**

The Master Chief calls the daily Kaizen Stand-up Meeting. During this meeting, the MC spends five minutes updating the group about changes in the marketplace and organizational news, if they have changes to
The MC is a communication conduit from Management to the team.

The team meets around the Performance Console for their Stand-up Meeting. The performance console is the centerpiece for collaboration and communication.

The Performance Console must be highly visible, and information on it should be updated regularly and be easy to read. Feel free to add additional components to it such as a team morale report, employee of the week, personas, or daily comics.

The MC reads any new suggestions from the suggestion box. Innovation and improvement should be encouraged and should be a daily conversation.

The MC should take suggestions seriously. Taking quick action and demonstrating follow-through on suggestions helps grow trust, and the absence of this action erodes it. If a suggestion cannot be implemented the MC needs to explain why to the team.

---

**Figure 50 - The Performance Console**

The MC is free between or during the daily Kaizen Stand-up Meeting to reprioritize items in the “Objectives” column of the Kanban board. Team
members are also expected to add new items to the Customer Request Queue, or the “Objectives,” or “To-Do” column as needed. After the MC has prioritized the items to be completed, the team then self-selects the new, more urgent items. In some cases, it may be necessary to place items from the “Objectives” column back into the work queues.

Process Owners on the team update the status of their work items on the Kanban Board and choose from the “Objectives” column a new objective and activities. If the objective has not already been broken down into tasks, the Process Owner breaks it down into a list of two day or smaller tasks and populates the “Selected Tasks To Do” column with the decomposed list of tasks.

Change control can happen in the daily Kaizen Stand-up Meeting, as everyone on the team should be present for signatures, and to express concerns. If a long conversation is needed to discuss a concern, the people involved can postpone the conversation until after the meeting is finished. In some situations, you may want to just have change control during the Cycle Review Meeting.

The Master Chief also makes sure that the Process Owners add every unexpected rework incident and every avoided incident to the Incident Register, and that every production change made is recorded in the Production Change Log. This log becomes priceless when searching for root causes of a production problem.

**The Past Domain and the Master Chief**

The Master Chief is responsible for ensuring the PROs update their SLAM Charts at the Daily Kaizen Stand-up Meeting, and the Cumulative System Performance Chart at the end of each Master Cycle.

They are also responsible for populating the Supplier Services SLAM Chart daily, and updating the Cumulative System Performance Chart with that data at the end of each Master Cycle.

This helps the tracking of long-term process capability trends within the System. This gives the department sponsors a timeline view of how the value streams in the system are performing, which indicates the health of the overall system.
This information can be presented by the Master Chief at a Quarterly Business Review meeting, if they participate in one.

Process Owners should always be challenged to improve their processes. Two starting points for improvement are increasing the target customers satisfaction levels, and then improving the ability of the company to deliver its services more efficiently by making fewer mistakes and consuming less time, money, and other resources.
Chapter 14 - Committees, Groups, and Kaizen Events

Of course, organizations are made up of many functional units. Whether you are in Operations, Implementation, DevOps, or Development you’re sure to be collaborating with the other functional units within your organization.

In addition to the four core meetings explained in Chapter 11, The Stable Framework™ includes the following additional meetings.

**Change Control Committee**

As production changes are proposed, they may impact other processes going on around them. If these requests are standard recurring production changes, typically the change has been systematized and requires only a notification, not an approval. However, new changes to production or even some systematized changes must be understood by all concerned and approved to go forward.

In Stable, your whole team is the change control committee and is made aware of any pending changes during the morning Kaizen Stand-up Meeting. They are free to put a hold on a pending change if needed until they can adjust their own environment and approve the change.

If your team interacts with other IT groups, you may have a weekly multi-department Change Control Meeting. You simply queue up your changes every day for that weekly meeting.

If your whole department is practicing Stable and you are having a set of escalated morning Kaizen Stand-ups (Kaizen of Kaizens) then you can push changes daily by bringing the approved changes in the early Kaizen Stand-up Meetings to the Master Kaizen Meeting.

**CCAPA Committee**

CCAPA is an acronym that means Correction, Corrective Action, and Preventative Action. These three actions are what happens when a Nonconforming Incident is reported by a customer.
A Nonconforming Incident is any incident where the customer experiences something that wasn’t as it should be. You can think of your Service Desk receiving calls and logging a problem as receiving a report of a nonconforming incident.

These incidents, if recurring or significant, get passed to the CCAPA committee. The CCAPA Committee is then tasked with the following:

- **Correction** - Making sure the situation is corrected for the specific customer or customers that reported it, if the Service Desk has not already done so.

- **Corrective Action** - Identifying what should be adjusted in the current Process Controls (SOP & Kata Cards) to prevent this type of situation from happening in the future, and then making sure those adjustments are made.

- **Preventative Action** - Identifying any other processes in the environment that could benefit from a similar adjustment, and them being sure those adjustments are made.

If you are operating in a controlled environment, such as an FDA regulated industry, you’ll need to keep a record of all these CCAPA issues and resolutions.

**Quarterly Business Review**

This is common meeting in larger organizations attended by functional managers of various departments. If the Master Chief is a functional manager, they may be attending this meeting. If not, information collected in the Cumulative System Performance Chart, which represents a trend history of performance progress for the department, can be taken to that meeting by a superior.

**Kaizen Teams**

Kaizen Teams are groups of team members organized to solve problems.
There are two reasons to form a Kaizen Team. A team may be formed to work on a Kaizen Project during a Master Cycle, and a team may be formed ad-hoc at any time to assist another Process Owner or a Master Chief with an urgent problem. We call this situation a Kaizen Blitz.

Toyota discovered that volunteer team members perform much better than assigned members. Toyota also learned that encouraging hard work instead of intelligence for problem solving brought better results. If intelligence was encouraged, people would sometimes think a problem was too difficult intellectually to solve and they would give up. However, if they were encouraged to work hard on a problem they would usually eventually succeed. Use volunteers whenever possible and be sure to encourage hard work and persistence over intelligence.

**Kaizen Project**

During each Master Cycle the team should be encouraged to pick an item in their environment that needs to be improved, and work individually on their own items, or together on the same item. Each challenge we call a Kaizen Project. Examples of Kaizen Projects include creating an automation script, a customer survey, a useful web post, or maybe organizing information better.

**Kaizen Blitz, or Ad-hoc Kaizen Team**

A Kaizen Blitz is an ad-hoc team formed when a Master Chief or a Process Owner requests volunteer assistance with a problem. This can happen any time during a Master Cycle. Team members should be encouraged to ask each other for assistance when puzzled.

**Master Kaizen Meeting, or Kaizen of Kaizens**

When implementing Stable in a large environment, you will have several Stable teams, all requiring their own Kaizen Stand-up Meeting. This works fine as long as no one person is on multiple teams. If this is the case, however, you can stagger the meetings from, for example, 9:00 to 9:30 to 10:00, etc.
Alternatively, you can have a set of Kaizen Meetings at 9:00, and then a Master Kaizen Meeting at 9:30 where a representative from each team reports at the later meeting. This Master Kaizen Meeting is called a Kaizen of Kaizens.

*Gemba Walk*

“Gemba” in Japanese means the location where the work gets done. That means a team has their own Gemba, and the customers have their own Gemba.

Sometimes the best way to understand a problem, or get clarity on a requirement, is to visit customers in their own Gemba. We call this a Gemba Walk. It ensures a deeper understanding of the work or issue resulting in a better decision than can be had from just staying inside a conference room.
Chapter 15 – Stable Hybrid Project Management

Hybrid projects are typically Operations or Implementations efforts and tend to have many repeatable characteristics. In addition, these types of projects often have known dependencies that must be identified during planning. For this reason, Hybrid projects combine the best of Plan-based and Agile project management approaches into what is commonly called Hybrid Project Management. Due to the repetitive nature of the typical Hybrid Project, Stable principles can be applied to achieve the best results.

Challenges that require management as a project emerge in two categories of disorder: Disorganized Disorder, and Chaotic Disorder.

Disorganized Disorder is a situation where a team is tasked with building something similar to what has been built somewhere in the past. Precedents exist, the team just needs to build something similar. Projects like real-estate construction, road construction, copying competitive products, or even reproducing a legacy software application on a different technology platform are all examples of disorganized challenges. Plan-based project management works best for these types of challenges because the team is dealing with relatively well-known steps and outcomes, and typically has a lot of past examples to refer to as needed.

Chaotic Disorder is a situation where the project team must build something that has never been built before. The solution will be original because there is nothing like it previously existing. The team doesn’t have all of the answers yet and must learn as they go. Agile project management approaches work best for these situations.

In summary, Disorganized Disorder is best addressed with a Plan-based approach, while Chaotic Disorder is best addressed with an Agile approach. Let’s take a look at the differences between Plan-based, and Agile approaches.

Plan-Based Project Management

It is common in the industry to refer to plan-based projects as “waterfall” projects.
Plan-based project management as we know it today started at the beginning of World War I when the US Army hired management consultant Henry Laurence Gantt to help with supply line logistics.

He proposed a tool with which a project team could identify deliverables, list all known activities required to complete each deliverable, estimate durations required to complete each activity, and then identify dependencies between activities so that the plan could be carried out efficiently.

Years later, in 1969, the Project Management Institute® was created and grew to become the primary authority on plan-based project management. Today they offer the PMP© and CAPM© project certifications, which are excellent sources for learning about executing projects where the result is an output that can be easily modeled after something that already exists. This allows a skilled team to estimate the overall schedule and cost of a project by understanding work items based on past similar projects.

Plan-based projects require decomposing the entire known workload into a set of distinguishable deliverables called a Work Breakdown Structure, and then creating a Project Schedule and Gantt Chart listing the activities required to complete each deliverable. All known activity dependencies can be identified to ensure the project can be executed in the correct order. This approach to project management is traditional in many companies and contains the reporting components that senior management is often used to seeing.

The drawback to this approach is that if your team is tasked with producing a solution based on a chaotic challenge, the plan will change so often that it will age into irrelevance quickly. After many failed software projects, the industry responded with Agile.

**Agile Project Management**

To facilitate hyper-changing environments and unknown situations, Agile Project Management was envisioned in 2001. Agile emphasizes short delivery cycles of high-focused teamwork producing deliverables of maximum value to the customer for a minimum effort on the team’s part. Regular customer demos at the end of these cycles—called Sprints—ensures the target deliverables stay relevant as they are being developed.
Instead of creating a thorough plan up front, objectives are kept high-level on a list called a Product Backlog, and team members select the highest priority objectives to complete during each Sprint. At the end of the Sprint, the team re-groups, re-evaluates the priorities, and selects the next group of highest priority objectives to work on during the next Sprint. These techniques allow the team the flexibility to react almost real-time to changes within the environment, to learn as they go, and to produce direct value as quickly as possible for the customer.

We found in Chapter 2 that after Agile was introduced into the software industry in 2001, the improvements in the years following that indicate how project success rates have gone up about nine percent, and the rate of failed projects have gone down about nine percent, without really impacting the late project statistic.

While failed projects tend to happen because the result was not a good fit for the customer base, late projects tend to happen because of the Hidden Factory. So, as an industry Agile has transformed our failed projects into greater successes, but hasn’t really addressed the Hidden Factory in software. In addition, the drawback of Agile project management is that it can be challenging to identify all of the components, dependences, and the total schedule or cost requirements for a project up front. This is where Stable helps.

**The Hybrid Project Management Approach**

Hybrid Project Management combines the best of Plan-based and Agile project management approaches to produce an outward facing project structure resembling a traditional project, but an inward facing Agile engine which is most efficient for the team.

Stable can work in harmony with your Hybrid project and your current role structure. Existing Project Managers or Agile Product Owners can keep their titles and simply wear the hat of a Process Owner as they work their Hybrid projects. The team can work in a traditional Sprint, or use the Master Cycle approach with a Kanban board to track progress.

In a Hybrid project, we use a Project Schedule & Gantt Chart tool as our Product Backlog. This allows us to manage dependencies and to use the additional columns to add helpful traceability data to each item such as what project goal the item supports, which component it will be found in,
who is the subject matter expert on the item, and in which test case it was confirmed as working properly.

Items in the Product Backlog represent deliverable components and all of the activities required to complete each component. It’s best to show activities rolling up into components. Additional scope can be added to the Gantt Chart as needed and then sized and prioritized accordingly.

The Project Manager continually moves the highest prioritized items from the Backlog onto the top of the “Objectives” column of the Kanban Board where the team self-selects who will complete the next most urgent objective. As the objectives are completed, progress is updated on the Gantt Chart. This way dependencies are respected, sponsors can see project progress, and the team is insulated from dependent concerns. A Burn-up Chart with a total-scope series line provides further transparency on team velocity, total scope identified, and work completion trends. Following is a detailed look at the Hybrid Project approach.

Starting a Hybrid Project

Hybrid projects start with a process called Chartering, where the Project Manager captures important summary information about the project in a one-page Project Charter. Typical pieces of information in a Project Charter include the following:

- The project sponsor
- The project manager
- The project problem needing to be solved
- Primary goals and success criteria
- Any obvious objectives that comprise the primary goals
- Primary stakeholders and roles in the project community
- Any noteworthy risks, constraints, dependencies, or exclusions
- Approval via sponsor signature

If a more formal project plan is needed, such as when a group is working in a regulated industry, a Standard Operating Procedure can be created explaining how project work is performed, and this SOP can be referred to in the Project Charter. This chapter may serve as the foundation for an SOP.
Identifying the Project Scope

If you have no previous project template to model your project after, your team may need to first create a Work Breakdown Structure, and then transfer the resulting work packages as entries into your Product Backlog.

An excellent way to generate a Work Breakdown Structure is to use an Affinity Diagraming technique. A Define Scope meeting is called, comprising the project sponsor, the team, and one or more representatives from the user-base. Participants take 3x5 cards and write one deliverable or feature per card. After everyone has finished writing, the cards are collected. A facilitator is chosen to organize the cards into similar groups. Duplicates are eliminated and the result is a Work Breakdown Structure.

Next, deliverables are transferred from the Work Breakdown Structure into the Product Backlog (Schedule and Gantt Chart). Working together as a team, all dependencies between deliverables are identified, and then all the activities required to produce each deliverable are decomposed into two day or smaller sizes. Each day represents a story point, or nearly a daily unit of effort. Add up all of the story points and the team has the total amount of story points of effort estimated for each objective. Collectively, they comprise the total story points of effort required for the entire project.

If the project scope is large the Project Manager and team may need to spend time interacting with individual project stakeholders collecting
requirement information about what they need. Storyboarding or Use Case generation activities are excellent techniques for identifying user roles, and their needs. In addition, the team should perform enough of the design work to determine the architectural components required to fulfill all of the identified needs.

The information collected would then be taken to the sponsor so they can decide how much of the requested scope they are willing to fund. End users will give you a long wish list and the sponsor may not want to pay for everything. Be sure to separate the needs from the wants when identifying requirements.

To save time, requirements can be written in a testable manner, so that they double as test plans, eliminating much of the work needed to create tests for the project. For ambiguous requirements, a standardized sentence has emerged in the industry which is useful for capturing periphery information about a requirement. This can be used when needed, for clarification. It is:

“As a [user type] I need to [do something] for the purpose of [stated objective].”

We learned from Agile to keep requirements to a minimum in the beginning, and then elaborate them as we go. It’s important to identify the user roles, their needs, and any needed architecture components, but the specific details of the requirements of each item can be identified later during development unless it has unusual characteristics. Anything
ambiguous or complex should be scoped and sized up front. The remaining items ought to be reasonably estimated by the team based on past experience.

Later, you can track the cycle velocity of each team by counting how many story points of effort get completed per Master Cycle, and you can estimate the total length of time required to complete the project, based on past cycle velocity.

Executing the Project Work

A hybrid project team may execute within a traditional Sprint, using the Gantt chart as the product backlog, or they may manage their workflow suing a Kanban process, as described next.

At the beginning of each Master Cycle, the Project Manager gets with the team and puts the highest priority objectives from the Product Backlog in to the top of the “Objectives” column of the Kanban Board. The “Objectives” column of the Kanban Board can be thought of as the top of the Product Backlog.

<table>
<thead>
<tr>
<th>Kanban Board</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Objectives</strong></td>
</tr>
<tr>
<td><img src="image" alt="Kanban Board Image" /></td>
</tr>
</tbody>
</table>

The team then starts off individually selecting which item from the top of the “Objectives” column they want to accomplish first. Unlike Agile, no commitments are made for completion, but instead, the team understands that when they have completed these items, they are to select the next items from the top of the “Objectives” column.
As the “Objective” column decreases in size, the Project Manager can add to it at any time, keeping it queued. New items can be discussed during the morning Kaizen Standup meeting, as needed.

As daily production needs interrupt scheduled work, the team can add the new items to the “Selected Tasks To Do” column, and then work to complete them. When finished, they can return to the next planned work item in the “Objectives” column.

Towards the end of the Master Cycle, instead of selecting new items that may not be completed within the cycle, team members may assist other members to get “In Process” items completed by the end of the cycle.

At the Master Cycle Review meeting held when the Master Cycle is over, the team demonstrates what they have produced to the Sponsor and any additional stakeholders invited to the meeting.

**Reporting on Project Progress**

Many companies have developed a standard report for showing the health of all ongoing projects in a Red-Amber-Green stoplight format. In Stable we call this the Master Project Schedule. An example of this chart is shown in the “Stable Portfolio Management” chapter in this book.
Gantt Chart and a Burnup Chart. The Gantt Chart shows a snapshot in time. The Burnup Chart represents the history and work performance trends. The cycle velocity, or amount of story points completed each cycle, will form a trend where the project completion date can be estimated using projection.

If needed, it may be helpful to create a pictorial representation of the deliverables as a Work Break-down Structure and use it to report on progress, division of labor, or cost breakdowns.

Risk Management

In an Agile-like manner, Stable is able to handle risks naturally, as they occur. In addition, Kata Cards decrease the number of problematic events over time as defensive institutional knowledge grows within the project environment. As we work on projects, and within Operations, we have the added benefit of being able to anticipate possible future risks and plan accordingly.

Any risks identified during the Chartering session, or by examining past similar project histories, or throughout the duration of the project, should be listed on a Risk Register and posted somewhere publicly, preferably on the performance console. Risks should be revisited and discussed at each Master Cycle Planning meeting.

The Risk Register should be one of the final documents stored with the final project report. This way, the list of risks can inspire future Project Managers executing projects in a Stable environment.
Improving Team Performance with Retrospectives and Kaizen Projects

After the Master Cycle Review meeting, the team excuses the non-team stakeholders, thanking them for their time. The team now begins their Master Cycle Retrospective meeting. The Retrospective Meeting is called by the team facilitator.

In this meeting, the team asks themselves these three questions:

1. What went well this past cycle?
2. What could have gone better this past cycle?
3. What still feeds the hidden factory?

A discussion is had, and takeaways are used to improve the environment. As Process Owners, they can identify a small Kaizen project they can work on together, or individually, to improve their efficiency.

Closing a Project

As a project is wrapped up, it is important to preserve important information collected during the experience for the benefit of future projects.

A Final Project Report containing the following information should be filed away with the project in a location easy to access for future Project Managers. The report should contain:

- Comments from primary stakeholders – What went well? What could have been better?
- Lessons learned.
- Risks identified and mitigation plans.
- Estimates vs. actuals vs. actuals adjusted for overtime.
- Amount of additional scope that was added during the project. Different sponsors have different co-efficiencies in this area.

Sustained Continual Improvement with The Stable Framework™

Finally, it’s important to use Stable Framework components to improve your project efforts over time.

For a Hybrid project the Kata Checkpoints would include every repeatable step or cluster of steps that seem reasonable to systematize within the project scope.

*Improved Project Estimation with The Stable Framework™*

An important tool to create and improve as each project completes is a department “Estimation” Standard Operating Procedure and Kata Card. Start simple, but use it each time a new project begins, and improve it every time a project is completed. Examine the estimates verses actual results. Be sure to correct for overtime. Have a Project Retrospective at the conclusion of each project where your team brainstorms how the Estimation Procedure and Kata Card can be improved.
Chapter 16 - Stable Operations

Operations is the daily support of existing products or services. In IT, Operations is considered anything from network support to running batch jobs at night, to monitoring large websites, to the administration of layers of middleware and databases. Backups and antivirus and disaster recovery fall under this area, too.

Operations tasks are thought of as recurring on a scheduled basis. A typical Operations model is the Schedule-Do-Check-Adjust Cycle, shown in the figure below. In this model activities identified to be completed are scheduled for the end-of-business-day, or end-of-month, or whenever needed, and then started. This is the first step, “Schedule.”

It’s common for an Operations group to have a combination of automated and manual tasks. The automated tasks should have a historical log indicating they completed successfully, and an alert mechanism when they don’t. The operators will know if the manual activities completed successfully or not because they are performing them in real-time. This is the “Do” step.

If no alert mechanism is in place, the operator must check to be sure the actions completed successfully, which is the “Check” step. If the activity abended, it is examined and re-run, usually with some minor adjustment.

If everything worked as planned, the cycle repeats itself. If something needs an adjustment, the process is adjusted in the “Adjust” step.

This process repeats itself and evolves as the organization’s IT needs grow. Two areas that often get neglected in Operations are scheduling regular Disaster Recovery tests and scheduling regular maintenance on Operational assets.

Don’t be that shop where you have a Disaster Recovery plan, in theory, but it never gets tested until a real emergency. Test it regularly, perhaps every six months over a long weekend.

Also, learn to think of your assets as a fleet, and schedule routine maintenance for them. Every so often, defrag all the server drives, check average CPU saturation over time, check swap file size to RAM size ratios, check all firewalls, and get rid of any unnecessary apps or fancy
background desktop graphics that make servers run slower. Over time, with a lot of operators, people get in and mess with stuff. Be proactive.

Operations Model

![Operations Model Diagram]

**Operations Using the Stable Framework™**

For an Operations group, a quality program is critical. Repeating processes must work the first time, every time. A Systematized Framework like the Stable Framework™ fulfills this need.

We start by identifying all value propositions, and then all repeatable processes within their value streams, scheduling them on the System Schedule, and adding their information to the Master Lists and Process Controls in the Configuration Management System.
Next, we identify our appropriate Master Cycle length which we will use for short-term planning and reporting. A Master Cycle should span anywhere from one to four weeks and should match an Agile Sprint, if other teams in the organization are working in an Agile environment. This ensures reporting and planning happen together, and inter-department change control can be managed easier.

Then, during the next Master Cycle Planning Meeting, we identify everything that needs to happen during that next cycle, as listed in the System Schedule.

The Master Chief puts the most urgent items into the “Objectives” column of the Kanban board, ordering them by most to least urgent, and the team self-selects which items they will work on next. The team then decomposes the items into two day or smaller tasks, moving them into the “Selected Tasks To Do” column of the Kanban board.

The team will naturally have their own requests to fulfill during the work day. They should indicate these in the “Selected Tasks To Do” column as they begin working on them. As they finish their own personal backlogs in the “Selected Tasks To Do” column, they can pick a new item from the “Objectives” column.

Repeatable processes that are performed manually require Kata Cards. Kata Cards can be thought of as super-checklists. They contain any steps needed to prevent problems from occurring within the process and are based on knowledge gained from past problems that have occurred. They may also collect information, if needed. When the process is complete, these Kata Cards are passed down to the next Process Owner receiving the work product, as a certificate the work was completed correctly. The final Process Owner gives the stack of collected Kata Cards to the Master Chief for storage in a historical log.

As the team is busy performing their tasks during the cycle, team members meet every morning for a daily Kaizen Stand-up Meeting to review the work at hand and anything new they or the Master Chief may need to add to the list of prioritized objectives.

Optionally, the Master Chief spends five minutes giving the team a “State of the Market” report, and team members spend 20 minutes communicating with each other what they completed yesterday (including change control items), what they intend to accomplish today, if they have any bottlenecks, and what improvements they have made in
their workplace. The change control items can be signed at that time, if necessary.

Actual process work being performed during Operations comes from the System Backlog, which is a combination of the Scheduled Activity Queue, the Customer Requests Queue, the Asset Maintenance Queue, and the nonconforming production issues that become Corrections, Corrective Actions, and Preventative Action (CCAPA) items. All of these can be tracked using the Kanban Board, and may require the use of Kata Cards for error-proofing and accountability.

Operations Model + Stable Framework™

- Continuous Improvement
- Value Stream Mapping
- Root Cause Analysis
- Kaizen Teams
- System Schedule
- System Backlog
- Kanban Flow
- Daily Kaizen
- Service Levels
- Process Kata Cards
- Process Kata Checkpoints
- Key Performance Indicators
- Process Katas
- Process SOPs
- Process Recovery Models
- Asset Recovery Models

Figure 58 - Operations Model + Stable Framework™

Operations should have agreed-upon levels of performance, known as Service Level Agreements (SLA’s). If no contract stipulates an SLA, then a Service Level Goal (SLG) can be created with the business sponsors. These are pre-established targets for performance set by the Master Chief or Process Owner while consulting with the customers of the service.

We demonstrate our ability to meet these service levels by showing a Service Level Attainment Monitor (SLAM) Chart in a visible location called a Performance Console.

These SLAM Charts can be updated daily and reported on at regular intervals called Master Cycle Review Meetings.
The Kanban Board is a workflow management tool that maximizes the potential throughput of work in process (WIP) by protecting the process from activity overload and the ensuing task-switching that is generated as a result. Task switching is also known as change-over time and is one form of waste.

The fastest way to deliver value through a system is to address one item at a time. The is sometimes called “Single-piece Flow.” As items are completed one at a time, they can be deployed and start generating value sooner than later. Having work-in-progress limits on the Kanban Board enables One-piece flow.

![Figure 59 - SLAM Chart](image)

![Figure 60 - Kanban Board](image)

Also, as problems are experienced, Kata Cards are updated and in some
cases Process Recovery Models and Asset Recovery Models are updated with root cause information and remediation steps so that future repeated problems can be recovered more quickly.

Notice the overarching principle at play here: A quality program empowers an organization so that if a problem occurs once the same problem should never happen again.

A quality program protects an organization so that if a problem occurs once the same problem should never happen again.
Chapter 17 - Stable Implementations

Implementation is the deployment and training part of a product development project, which typically is the final set of the project’s activities.

During Implementation, a tested and completed solution is deployed to a production site, typically at a customer location. Although each customer is unique, usually the same milestones must be reached to deploy a solution successfully.

A typical Implementation process contains at least these five steps: Site Discovery, Preparation, Initial Implementation, Full Implementation, and Sustainment.

**Implementation Model**

![Implementation Model Diagram]

- **Discovery/Exploration**
  - Confirm goals, objectives, deliverables, fit, & feasibility
  - Assess impacts, vulnerabilities, risks, constraints, & assumptions

- **Preparation**
  - Establish communication channels, schedule, checkpoints, & team
  - Secure equipment, tools, etc.

- **Initial Implementation**
  - Identify small low-risk target environment
  - Spin up group collaboration & progress status tools
  - Install product, perform smoke tests, identify remaining problems

- **Full Implementation**
  - Standardize Initial Implementation process
  - Roll out to full customer base in staged increments
  - On-site training completed, on-site operators trained

- **Sustainment**
  - On-going training/help desk available
  - Financial information in place
  - Information logged, customer final approval

*Figure 61 - Implementation Model*

**Implementation Using the Stable Framework™**

Implementation activities are usually project-based. The project steps should be identified and processed through the Kanban. Each step or set of steps should be thought of as a repeatable process and should be identified, modeled into a process map, and have an SOP and a Kata Card created for it.
Figure 62 - Implementation Model using The Stable Framework™

- Implementation Model
- Customer approval
- Financial information
- On-going training/help desk available
- On-site training completed, on-site operators trained
- Roll out to full customer base in staged increments
- Standardize initial implementation process
- Failure: More planning or better models needed
- Installation of new product, product smoke test, identity retaining problems
- Spin up group collaboration tools, progress status tools
- Identify small low-risk target environment
- Failure: More planning needed
- Security equipment, tools, etc.
- Establish communication channels, schedule, checkpoints, & team
- Assess impacts, uncertainties, risks, constraints, & assumptions
- Confirm goals, objectives, deliverables, & feasibility
- Discovery/Exploration
- Preparation
- Initial implementation
- Full implementation
- Sustainment

- Customer billing in place?
- Help Desk Info in place?
- Sustainment KPI Card
- Full implementation KPI Card
- Initial implementation KPI Card
- Discovery KPI Card
- Preparation KPI Card
- Project plan created?
- Infrastructure in place?
As the implementation Process Owner performs the steps in the process, the PRO submits a Kata Card showing that each item has been checked off and the process was completed successfully. Think of a completed Kata Card as a certificate authenticating that the work was completed successfully. The Kata Card is collected at the next checkpoint by the recipient PRO, or given to the Master Chief if it’s the final step in the value stream.

Kata Cards for each step add value by identifying past problems that future implementers can be protected from.

For example, in the Discovery step, after sending someone out to a rural customer site who may still be on a slow T100 network connection when 1GB is required, the implementation team learns in the future to run a network speed test before all future site discovery visits. They would then add this step to their “Site Discovery” Kata Card so that future installs don’t experience this problem.

For preparation, the team may have learned to review the IP addresses at the facility with the local IT Director and have their hardware preconfigured to save time when they arrive. This would be added as a step to the “Preparation for Install” Kata Card.

For the Initial Implementation step, the group would eventually learn to perform a complete smoke test on site after the system has been installed. A smoke test is a test designed to check a single feature from every screen or all major screens after installation to be certain everything is present and working without spending all the time required to do a full test at a location. This would be added to the “Initial Implementation” Kata Card.

As another example, after discovering that the less expensive WIFI repeaters really didn’t work as planned, the installers can add a step on the Kata Card to use a WIFI signal strength meter and walk through all boundaries of intended use to prevent this problem from happening at all future install locations.

If the initial implementation went well, the team would then do a full deployment to all relevant workstations at the location using a “Full Implementation” Kata Card. Here they could also collect information such as the number of licenses deployed.

Finally, a “Sustainment” Kata Card could handle the remaining items
such as training expectations, ongoing support payment methods, and so on.

Over time these repeated processes will benefit as more good ideas and precautions are identified and added to the Kata Card for each step. This reduces errors, rework, and mistakes, making the implementation process faster and less expensive over time.
Chapter 18 - Stable DevOps

DevOps is the technical skill support group tasked with performing a critical part of the software value stream: deployment. Agile development techniques enabled teams to create releasable software more frequently, creating the need to coordinate multiple product releases, and in many cases dependent releases. The orchestration and progress of these activities is often called Continuous Integration/Continuous Delivery, or the CI/CD Pipeline. These efforts are independent of the Operations group present in IT environments. This DevOps group is specifically concerned with release activities.

In addition, if your organization is offering Software as a Service (SaaS) or Platform as a Service (PaaS) products, the need for excellence in DevOps is paramount.

Sometimes DevOps is expressed as a figure-eight, with Development process steps on one side and DevOps process steps on the other. For our purposes, we’ll just focus on the DevOps process steps which are shown linearly in Figure 63.

According to Gene Kim, co-author of The DevOps Handbook, and The Phoenix Project, there are Three Ways that direct DevOps efforts:

1. The First Way emphasizes the performance of the entire system – the value stream.
2. The Second Way is about shortening and amplifying feedback loops.
3. The Third Way is about creating a culture that fosters continual learning and understanding.

The Stable Framework™ enables these Three Ways in DevOps, whether your team is Agile or Plan-based. Your model may differ, but in this model, coding is completed, the programmers have completed Test Driven Development (TDD) testing and confirmed the application works correctly with the end users before checking in the code.

Traditionally, User Acceptance Testing (UAT) is one of the final steps in this process, but because we are trying to streamline the time it takes the team to complete these steps in this model it makes sense for the end user to approve the final functionality before the code goes through all
the downstream component testing, regression testing, and load testing steps. This is the second principle in the *Theory of Constraints: Exploit the Constraint*. In other words, why do all that downstream testing if you’re not sure the user will accept it as designed? Obtain user approval before performing all that time-consuming, expensive work.

Once the code is checked in, the DevOps team updates the automated test scripts and any deployment scripts that will have changed with the new features. You may have a group of testers who component test and System Test (Regression Test) the whole build at this point.

**DevOps Model**

If automated regression testing is engineered correctly it is structured like a main starting point sliced into multiple automated test scripts that all start and end at the same location. This way they can be daisy-chained together and only the scripts impacted by the new requirements in the candidate release need to be created or adjusted. A well-trained group can recreate updated automation slices for the specific changed portions of the new build while testing it. This way they minimize the effort and time required to test what changed and then regression test.
the whole build.

Assuming no issues, they then load test and stress test the new build. Load testing confirms the build meets any established Service Level Agreements (SLA’s), and Stress Testing gives the engineering team an idea of how far beyond the SLA the new release will function reasonably.

Finally, if everything passed successfully, they run the standard anti-virus check, and then the automated deployment scripts to push the production-ready build to its proper production destination.

If anything should fail along the way, a feedback loop exists to promptly alert an operator at a previous step that something failed, and adjustments to automation scripts or bug fixes would need to be made before the next attempt.

DevOps Using the Stable Framework™

Each value proposition a company offers has its own unique set of release logistics. Therefore, a company may have one or more CI/CD Pipelines, depending on the number of products or services it offers.

Due to the many handoffs and the complexity of the last few steps in a value stream, the team can easily create a lot of waste by repeatedly making the same mistakes. Using the Stable Framework™, Process Owners would have tight feedback loops which lower the probability of making mistakes as they learn from their past, resulting in a faster and leaner stream of value to the customer.

With Stable, the processes would be defined for each step in the diagram below. Kata Cards would be created for each step: A “Development” Kata Card would be used for completing the code and performing the Architectural, TDD, and UAT Reviews.

Then, a “Continuous Integration” Kata Card, a “Regression Testing” Kata Card, and an “Automated Deployment” Kata Card would be created.

Many of the alerts and customer communications can be automated, as well, and can appear as checkbox items on the Kata Cards.

With deployment, often some things must be temporarily turned off, for the new build to be deployed. A Kara Card ensures they are turned back
THE STABLE FRAMEWORK™

- Dev Kata Card
  - Architecture review?
  - Component Design review?
  - TDD/Code review?
  - UAT review?

- Continuous Integration Kata Card
  - Release Notes updated?
  - Build Number updated?

- CI: Scripts Kata Card
  - Install/Upgrade Scripts updated?
  - Regression Test Scripts updated?
  - Load Test Script updated?
  - Anti-Virus Script updated?
  - Push to Prod Script updated?

- Testing Kata Card
  - Regression Test successful?
  - Load Test successful?
  - Stress Test performed?
  - Anti-Virus check successful?

- Automated Deployment Kata Card
  - Automated deployment successful?
  - Updated Release Number confirmed?
  - Smoke Test successful?

---

**DevOps Using The Stable Framework™**

![Diagram of DevOps process]

Figure 64 - DevOps Using The Stable Framework™

- Send out notifications
- Automatically deploy build to production
- Anti-Virus test candidate build
- If needed, automatically deploy build
- Continuous Integration Incerement
- Confirm updated Version Number
- User Acceptance Testing (UAT)
- Test Driven Development (TDD)
- Coding

- Continuous Integration Kata Card
  - Automated Deployment successful?
  - Updated Release Number confirmed?
  - Smoke Test successful?

- CI: Scripts Kata Card
  - Push to Prod Script updated?
  - Anti-Virus Script updated?
  - Load Test Script updated?
  - Regression Test Scripts updated?
  - Install/Upgrade Scripts updated?

- Testing Kata Card
  - Regression Test successful?
  - Load Test successful?
  - Stress Test performed?
  - Anti-Virus check successful?

- Automated Deployment Kata Card
  - Automated deployment successful?
  - Updated Release Number confirmed?
  - Smoke Test successful?
on afterwards, every time.

Process Recovery Models and Asset Recovery Models would be on hand to quickly bring failed systems back online with the guidance of lessons learned from past failures. This way, no steps get skipped, and everything happens with minimum error. In addition, any production change request can be handled using the Customer Request Queue and the Kanban Board. Requests can be time stamped when work begins, and then time stamped when they are completed.

Categorizing these requests and tracking the amount of time for their completion can provide customers with accurate estimates regarding how long they need to wait for their requests to be fulfilled.
Chapter 19 - Stable DevSecOps

DevSecOps is the application of security protections within Development, Operations and all other IT areas. It includes data collection, transmission, storage, and access stewardship with respect to security, availability, processing integrity, confidentiality, and privacy. Industry guidance is available from multiple sources: HIPAA, PCI, SOC 2, FDA 21 CFR Part 11, and others.

Just like quality efforts, security efforts should not be an afterthought dealt with after the product is developed. Security, like quality, should be designed and addressed throughout the environment. Sometimes this proactive concept of developing these from the start is called "Shift-Left" thinking. The entire Stable Framework™ is based on Shift-Left thinking.

![Diagram of Security, Development, and Operations](image)

Security requirements should be identified at every repeatable process within an environment, and should be revisited and updated whenever a change occurs within it. It is an integral part of Systems-Thinking.

In addition, Process Owners should be champions of security, within their areas of practice. This requires training for everyone involved. Advanced tools exist to scan environments and detect known threats, but in many cases these tools are not enough.
Fortunately, the Stable Framework™ provides an excellent platform to ensure training and security needs are met for every security surface at risk. Because the framework is self-learning and sustainable, as new risks are discovered, the framework can easily adapt and grow to counter those new risks. The Training Log, the Kata Card, the Standard Operating Procedures, empirical learning, and the certificate-based workflow accountability process which ends at the Master Chief are all the tools necessary to implement adequate workflow security in most environments. Indeed, the web security organization OWASP produced results of a 2015 survey that indicated manual code reviews were the most effective ways to find general security vulnerabilities, logic bugs, and compliance issues in software.

DevSecOps concerns impact every environment. Below is a sample list of example Kata Card additions that could be added to each respective environment. These could be integrated into the Kata Cards for each step, or they could be stand-alone, but required for each step, along with that repeatable step’s natural Kata Card.

**Example DevSecOps Training Kata Card**

- Developers trained on password weak/strong principles.
- Developers trained on password hashing & salting storage.
- Developers trained on digital password best practices.
- Developers trained on encryption types/purposes (DES, AES, RSA, etc.)
- Developers trained on memory encryption best practices.
- Developers trained on web encryption (SSL/TLS/HTTPS).
- Developers trained on sources of industry encryption standards.
- Developers trained on blacklisting vs. whitelisting access control.
- Developers trained on Google dorking.
- Developers trained on OWASP’s Top 10 Security Risks.
- Developers trained on OWASP’s Top 10 Mobile Risks.
- Developers made aware of security penetration testing apps such as SonarQube, Coverity, RIPS, etc.
- Developers trained on SQL injection.
- Developers trained on cross-site scripting (XSS).
- Developers trained on IDOR/BOLA/RCE authorization spoofing.
- Developers trained on Unvalidated URL Redirects.
- Developers trained on website & webservice fuzzing.
- Developers trained on .NET Reflection (Reverse compiling .NET apps).
- Developers trained on reverse compiling Android .APK files.
Example DevSecOps Requirements Kata Card

- Secure data identified.
- Security level roles/privileges identified in User Stories/requirements.

Example DevSecOps Architecture Kata Card

- Secure (HTTPS) websites/webservices.
- Passwords stored hashed and salted in DBs.
- Security layer created with roles/privilege admin console.
- DB application users/passwords in place instead of using DB Admin.
- Password reset feature in place (and password not sent via email).

Example DevSecOps Component Design/Code Review Kata Card

- Password fields in place (showing *** instead of password).
- Unauthenticated webforms return to home screen.
- All timestamps sourced from DB Server, or network service, to avoid local PC clock spoofing.
- No sensitive data stored in the windows registry.
- No sensitive data stored in XML files, or other unencrypted files.
- No SQL injection friendly code techniques used to access DB data.
- Password must conform to strength principles.
- No shared usernames/passwords in source code source control.

Example DevSecOps Network Security Kata Card

- WIFI passwords must be strong.
- WIFI security must conform to latest available.

Example DevSecOps Testing DevOps Kata Card

- No unscrambled production (real) user data available to testers.
Example DevSecOps DevOps Kata Card

- Staging (test) and production passwords meet strong requirements.
- No shared usernames/passwords.

You may have noticed there are no chapters in this book for using the Stable Framework™ in Testing. That’s because each test performed at each particular level (component, interface, integration, system, user acceptance) is already a type of Kata Card, and should be continually improved naturally.

These examples above are a starting point for integrating DevSecOps into your environment.
Chapter 20 - Stable Support Desk

A good support desk uses software to create, monitor, and closeout support requests. Support can be thought of as External, and Internal.

**External Support Using the Stable Framework**

External support services external customers who call in with questions or issues regarding the organization’s products or services. A simple Kata Card can be created and embedded within the support software to provide a helpful and professional touch to customer contacts. Here’s a simple example:

Initial Contact:

  - Greeting
  - How can I help you today?

Engagement:

  - Current Version of software ____________.
  - Was problem found in existing Support Desk Historical DB system?
  - Attempted to reproduce problem, if appropriate.
  - Helped user with workaround, if appropriate.
  - Helped user with education issue, if appropriate.
  - Closed with “It was my pleasure to assist you” instead of “No Problem.”

Post Engagement:

  - Problem entered into Support Desk Historical DB system.

**Internal Support Using the Stable Framework**

Internal Support works much like external support, except along with requests from frustrated internal employees, internal service desk Process Owners receive standard work requests. These are preset administrative items such as issuing new employee security badges, or invalidating and replacing lost security badges.
Each standard work request should have a Kata Card created for it, and should be part of an internal ticketing system.
Chapter 21 - Stable Product Development

Let’s look at how a quality program can apply to a product development environment. A well-known approach to developing a new product or service is called the V-Model, because it looks like a V. It contains the initial development steps from concept to completion down the left side of the model, and the corresponding quality control steps up from the bottom to the top on the right side of the V.

The Product Development “V- Model”

![Product Development V-Model Diagram]

The development steps include the Initial Concept, Requirements Analysis, Architectural Design, and Coding.

The quality activities on the right represent Quality Control. These are the activities performed after a component has been built but before the customer receives it. You can think of these as inspection points.

The Smoke Test is important to make sure all the components have installed correctly. Component Testing is inspecting each part of a complex deliverable, such as individual forms, and reports. System
Testing is inspecting the entire deliverable, particularly data flow from one component to another within the whole system. User Acceptance Testing is showing the system to a customer and getting customer approval on the usefulness and accuracy of the new system compared to what was expected.

We call the inspections mentioned above Quality Control, because these inspections happen after the component has been built, but before the customer receives it. Quality Assurance, however, happens during the development of the component.

**Product Development Using the Stable Framework™**

Quality Control is product quality. It can be thought of as inspecting a candidate deliverable before the customer receives it.

Quality Assurance is process quality. It is your proof that a process step was followed correctly. Quality Assurance is the record of accountability that indicates each step was performed according to the best-known methodology available at that time. We call that methodology the Standard Operating Procedure (SOP). The SOP instructs the employee how to perform the pre-identified, repeatable task, and should include a checklist, called a Kata Card, of risks to watch out for.

This Kata Card is what gets signed, date-stamped, and handed to the receiving process owner when the work is complete. The actual SOP should be read by each practicing employee during initial onboarding, during periodic training, or when re-training is necessary. Retraining is necessary when a step is performed contrary to the SOP, or a completed Kata Card result proves to be inaccurate. This will become apparent when the root-cause of a nonconforming item was found to be the result of an employee not completing a Kata Card properly.

In product development, Quality Assurance activities include a Concept Review, Requirements Review, Design Review, and a Code Review. If you are using Stable outside of software, you can think of Code and Code Reviews as Construction and Construction Reviews.

A review works as follows. A Kata Card is created to govern that type of review. The Kata Card contains known risks and past problems found
CHAPTER 21 - STABLE PRODUCT DEVELOPMENT

The Product Development "V-Model + Stable Framework™

Figure 67 - Product Development V-Model Using The Stable Framework™
during that step. The reviewer can be a single person or a committee, and they go through the Kata Card with the author, indicate their name, date, and time, and submit the Kata Card as proof the review was conducted.

It is important that the completed Kata Card be collected by the Process Owner of the next step. If the next step is the end of the value stream, the final Kata Card and all preceding Kata Cards go to the Master Chief, who puts them in the Kata Card historical log. When the project is completed, the review records become part of the project historical documentation.

A Concept Review is the most elaborate review. We call this a Business Case Analysis and leave it to the strategists and portfolio managers to vet an idea and commission it to the design team.

A Requirements Review can be a small committee of Business Analysts and Quality Engineers who scrutinize the requirements for any areas of unreasonable ambiguity or past requirement mishaps.

In a situation where a new requirements risk is discovered, the risk should be added to the Requirements Review Kata Card so that future projects will benefit from that knowledge.

An architecture review works the same way. An “Architecture Review” Kata Card is created and added as the company learns about new risks. It may begin with commonly known architectural best practices and get expanded as future problems are experienced.

For example, while designing an enterprise software solution, it’s important to source the date and time used by the software from the local database server. This way, all the workstations are synchronized and logging is consistent. Another design consideration might be to ensure the consistency of all Error Messages, Reports, Screens, and Screen Menus, so it looks like one person created the whole system instead of multiple people creating different parts of it.

In time, the group can add more items of design interest to their “Design Review” Kata Cards such as security items, and then whoever conducts the design review must inspect the design for each entry on the Kata Card, sign and date the “Design Review” Kata Card, and submit it to the next Process Owner downstream, repeating this process until the Master Chief receives the final stack of Kata Cards for that release and enters
them into the permanent project history.

A code review can be conducted by another coder. The team would have a common “Code Review” Kata Card, which lists coding habits that are not permitted. For example, the code must have a healthy ratio of comments per lines of code. Or, the code must contain readable variable names, etc.

As time goes on, this Kata Card can be added to as new issues are found. Again, the reviewer records his or her name, the author’s name and work, and time stamps the kata card, and then it becomes part of what is handed off to the owner of the next step in the process.

This is the basic work of Quality Assurance. It can be expanded to include a review of the right side of the V-Model. If testing is not producing good results, each QC task can be QA’d. A “Component Test” Kata Card can be created and used for a Component Test Review. An “Integration Test” Kata Card can be created for each Integration Test. A “System Test” Kata Card can be created for conducting System Test Reviews, and a “User Acceptance Test” Kata Card can be created for conducting a User Acceptance Review. As a company experiences natural problems and shortcomings, it can add the root causes as items to check for on the respective Kata Cards to ensure that the past problems never happen again. This is what it means to be a learning organization.

For some projects, you may even consider adding a decision log to the project, where critical decisions are recorded, timestamped, and key information is noted about them such as who came up with the idea, and who agreed to make the decision.
Chapter 22 - Stable Agile Development

Agile development originated in the software industry and is an excellent model for delivering value quickly when there are a lot of unknowns in the project environment.

Scrum, the most popular implementation of Agile, was invented in 1993 by Ken Schwaber and Jeff Sutherland. Many implementations of Scrum exist today, including SAFe®, LeSS, and Disciplined Agile. I’m displaying the PMI-ACP® Scrum model because I think it is a simple but well-developed implementation. It is the core Scrum Framework with some added accessories that enable it to scale well for large projects.

![PMI-ACP® Scrum Model Diagram]

The PMI-ACP® model starts with a Vision Meeting to charter the project, which means establishing the project objectives and success criteria, clarifying what the project should produce, who the intended customers are, and why the project is being executed.

Next is a Release Planning Meeting where large projects can have analysis, design, and proper estimation performed for them, giving the stakeholders a reasonable idea of how long the project will take, and how much the project will cost.
Agile Product Development Using the Stable Framework™

Stable can be adopted by an Agile team as an add-on set of practices. Instead of a “Master Cycle,” the team can retain the Sprint Board, and “Sprint” vocabulary, which is more accurate as the team is racing to maximize value to the customer. The team would simply add the other Stable trimmings as discussed in Chapter Seven. A useful technique is to use a Sprint board alongside a Kanban Board with the former tracking project work while the latter tracks accumulating production change request which are typical in a Development environment.

In a small shop, the Development Manager can assume the role as the Master Chief and all team members become Process Owners. In a larger multi-team environment, each Product Owner can assume the role as Master Chief of their teams, with the Development Manager as the Senior Development Master Chief.

In the Vision Meeting, where the project is chartered, a “Chartering” Kata Card can ensure all common objectives, constraints, risks, industry regulatory restrictions, competitive products, and end-users have been considered.

While performing Scrum, in the Release Planning Meeting, a “Design” Kata Card can ensure that proper design patterns, scalability concerns, security issues, and any microservices or Service Oriented Architecture needs are met.

For development, “Code Review,” “User Experience” and “User Acceptance” Kata Cards should be completed before a developer can integrate any code.

All the “Quality Assurance” Kata Cards apply to the testers on the team. “Smoke Test” Kata Cards, “Component Test” Kata Cards, “System Test” Kata Cards, “Interface Test” Kata Cards, “Load Test” Kata Cards and finally “User Acceptance Test” Kata Cards can be used to be sure the tests are capturing everything desired. Perhaps they can be combined into one document.
Chapter 23 - Stable Portfolio Management

Portfolio Management is a critical component for an organization that wants to stay competitive. As companies grow, they want to apply their resources to those investments that are in their best interests. Portfolio Management is about choosing just those project opportunities that best serve the objectives of the organization, and saying no to the rest.

Portfolio Management

Organizational Influences

Market Influences

Regulatory Influences

1. Projects get proposed by senior executives as newer ideas emerge.
2. Teams get tasked with even more work.
3. Every other project gets delayed as additional work is given to the already busy team members.
4. Projects are released sometimes delivering substandard value with no plans to recover.
5. Executive with the loudest voice gets funding. Dysfunction ensues.

New Products
New Features
Diversification
Marketing/Sales
New/Improved Features
Improved Durability
Service Desk
Knowledge Base
Process Improvement
Ops. Improvements
Shorter Time to Market
Technology Advances
Compliance

Unpredictable Impacts

No Accountability for ROI

Most organizations have no defined Portfolio Management process, and many have no idea what the term even means. In scenarios like this, projects simply get recommended by senior executives in various business areas. Little care is taken to throttle the workload placed on the teams, causing overburdens leading to frustration, thrashing, and delays everywhere. Post project analyses are rarely performed, leaving lots of lost value and revenue on the table for competitors. Poor portfolio
visibility leads to a constant imbalance of priorities for the company making their written strategy little more than talk.

**Portfolio Management Using The Stable Framework™**

An organization is a value system, complete with inputs, outputs, and throughputs. The outputs of the system are the value propositions that attract customers. We call that Customer Value. The inputs to the system are market forces, and various forms of capital. We call that Opportunity. The sustainment and growth a company experiences as a result of exploiting opportunities to generate customer value, we call Business Value. Whatever will get an organization to where it wants to be next has high business value. Business Value is best expressed using Key Performance Indicators (KPI’s). Well run organizations direct this business growth through identifying Objectives and Key Results (OKR’s) at all levels throughout the organization, and ensuring transparency and good communication so that they all align and their pursuit efforts can be maximized through synergy.

Think of Portfolio Management as a mechanism to direct your organization’s value system. Specific inputs to this system are market reports, product and service performance summaries, competitive analysis, regulatory requirements, and organizational influences. Outputs are product and service performance, market share growth rates, and internal quarterly business assessments. Throughput factors within the system are leadership, culture, skills, and knowledge proficiency. In other words, how well the organization executes.

A good Portfolio Management process ensures the following:

- Corporate strategy is defined and clear.
- Market influences and competitor trends are monitored.
- Project teams are not overburdened.
- Projects are prioritized to reflect the current corporate strategy.
- High and low risk is balanced during project selection, as needed.
- Low value projects get cancelled when necessary.
- Communication is improved between business and IT.
- Project value yield is monitored for some time after completion.
- Transparency and feedback are improved as all major project efforts are visible to the executive team.
CHAPTER 23 - STABLE PORTFOLIO MANAGEMENT

1. Identify Corp Strategic Core (Problem, Vision, Mission, Values, Anti-values, Goals)
2. Review Organizational, Market, Regulatory Influences
3. Identify Business Needs (Add, Protect, Lower)
4. Evaluate Progress of Current Portfolio & Completed Projects
5. Evaluate New Opportunities (Business Cases), Managing Risk
6. Scrutinize, Prioritize, and Select New Options
7. Make Changes to Internal Processes as needed

Cost to make $1 x (Market Opportunity/Penetration)

PM SYSTEM

Inputs
Organizational Influences
Regulatory Influences
Market Influences

Outputs

Portfolio Management + Stable Framework

PM SYSTEM

Product/Service ROI, Payback Period, NPV, Market Size, Market Share, etc.

Figure 71 - Portfolio Management Using the Stable Framework™
Typically, a strategy review happens annually. Quarterly Business Reviews are conducted to regularly assess market, product and service performance. The Portfolio Management Committee meets every two weeks to prioritize project approvals based on the latest information and opportunities at hand.

The committee is comprised of the senior executives within the organization, and typically the VP’s and Directors of all of the functional units.

Following is a description of a basic Portfolio Management process:

1. Identify the organizations Strategic Core (Vision, Mission, Values, Goals)

   The Strategy Review meeting happens annually, or sometimes every six months. This is where the CEO and board of directors and senior executives decide the corporate strategy for the organization going forward. It’s important to understand that strategy is not simply what the group decides, but rather how they spend the organizations money towards that decision. In a well-run portfolio management process, the majority of the projects that get funded reflect the decided upon strategy.

   As part of the strategy review, if the following core is not already in place, the executive committee should clarify the Corporate Vison, Mission, and Values.

   A Vision Statement is a description of a future state, which directs the company’s overall purpose of existing. A Mission Statement indicates how they intend to achieve the corporate vision. Value statements help guide decision making and prioritization.

   Once these are in place, the organization can set its present strategy, which means its long-term goals. Long term goals drive large strategic initiatives, and should reflect the Mission Statement and Values of the organization.

   Short term goals are subsets of long-term goals and drive department scorecards which get reported on quarterly during the Quarterly Business Review (QBR) meetings.

   Be sure everyone participating in the Portfolio Management meeting is aware of and clear about the organizations current Vision,
Mission, Values, and Goals. It may be appropriate to create a performance console on the wall in the Portfolio Management meeting room, and have all of this information posted.

2. Review Organizational, Market, and Regulatory Influences

Quarterly it is beneficial to have marketing prepare a report of marketplace trends, competitor trends, and any regulatory changes that are pending or have emerged. It’s important to be aware of what your largest or fastest moving competitors are doing.

In addition, it’s important to include trending keywords in your products and services where applicable, to stay relevant in your marketplace.

3. Evaluate Progress of Current Portfolios & Project Status

This step requires a Service Performance Inventory, Product Performance Inventory, and Master Project Schedule.

Service Performance Inventory

The Service Performance Inventory is simply the Cumulative System Performance chart, or a subset of the chart. It should include custo-
mer satisfaction scores and SLAM chart results, at least.

**Product Performance Inventory**

The Product Performance Inventory is a list of all of the primary products and services currently offered by your organization, along with information about each one such as market size, market share and annual revenue. It may be helpful to include a column showing the % change since this time last year. This list should match the public entries on the Operations Cumulative System Performance Chart, and be owned by Marketing and populated by marketing research. If revenue is kept private, a revenue rank can be substituted.

![Figure 73 - Product Performance Inventory](image)

An array of decision support tools exists to examine the market performance of products or services, such as the Boston Matrix, Bowman’s Strategy Clock, and Porter’s Three Strategies. Your product managers or marketing team should provide up-to-date reports of these products to be used for analysis.

In addition, recently completed projects or services should have a post-release ROI evaluation from the project sponsor, or an assigned business analyst. These post-delivery ROI evaluations are typically performed quarterly until trends emerge. Information collected here is critical. Sometimes new products fail to meet their intended ROI and with just a small adjustment can become high-demand products. Consequently, these evaluations sometimes trigger high-yielding follow-on projects.
Master Project Schedule

The Master Project Schedule is a list of all current projects being worked from all departments, their status, timeline, and costs to date. In some cases, costs extracted from this information gets billed back to internal departments who requested the projects from IT so that internal spending can be tracked for the organization.

The example above shows percent complete, expected revenue impact, and team size. Any metrics of interest can be tracked and reported on using a similar chart.

Projects that are failing, or candidates for cancellation can be examined and discussed with this tool, to aid in decision support.

4. Prioritize Project Opportunities (Business Value)

This step is different depending on the type of approach your organization takes to determine how it wants to organize choices.

One common set of categories to organize projects into is:

1. Compliance
2. Protect Existing Revenue (Improvement requests, robustness)
3. Add New Revenue (New products)
4. Lower the Cost of Doing Business (Efficiency)
5. Goodwill (Optional, to improve community image)

For decades, companies have debated which order is the best to use for prioritizing projects, and the debate continues today. In Stable, we recommend the Sandcone Model approach, which is the priority listed above. It was used by Kia/Hyundai to outperform Toyota over time.

Before we discuss the approach, a little portfolio theory helps. Business owners need to understand that one of the first critical investments needed, after creating their first great product or service is a marketing and sales engine. A second or third great product will not make a first great product sell if the organization has no marketing and sales engine. One of the first priorities for a company who wants to grow is to invest in a marketing and sales engine.

Secondly, efficiencies are more critical for small companies than large companies. If you have a small company, you may want to switch number 3 and 4.

Third, understand that all companies want to add new revenue, but it’s cheaper to protect existing revenue once you have a sellable asset, then it is to develop a whole new one. If you have an asset that needs a little help to improve its customer satisfaction scores, or market penetration rates, your money is better spent there than on a brand-new product.

Also, the current market share of a good product is an important consideration. If you have a product with healthy customer satisfaction (CSAT) scores and a low market share (0-16%), you are better off investing in a marketing campaign for that product then spending money developing anything new. See if you can get it into the early majority market share range (17%+) where it will begin to sell itself through word-of-mouth.
CHAPTER 23 - STABLE PORTFOLIO MANAGEMENT

Portfolio Prioritization Model

<table>
<thead>
<tr>
<th>Compliance</th>
<th>Start</th>
<th>Compliance Project?</th>
<th>Yes</th>
<th>Priority 1 Compliance Project</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protect Existing Revenue</th>
<th>Existing Product or Service w/ Low CSAT?</th>
<th>Yes</th>
<th>Priority 2 Improve Product Features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protect Existing Revenue</th>
<th>Existing Product w/ Low Market Share?</th>
<th>Yes</th>
<th>Priority 3 Improve Marketing Campaign</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Add New Revenue</th>
<th>Current Market Conditions present New Opportunity?</th>
<th>Yes</th>
<th>Priority 4 New Product or Service Project</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lower Cost of Doing Business</th>
<th>Priority 5 Efficiency Improvement Project</th>
</tr>
</thead>
</table>

End

Figure 75 - Portfolio Prioritization Model
If your existing products have good CSAT scores, and good market share, it’s time to examine the market for new opportunities. If some exist, it may be time to build a new product or service. If no good new ideas exist in your marketplace, you might consider diversifying existing products into alternate markets, or alternate price points.

If none of these options seem attractive, it may be a great time to fund a project that lowers the cost of doing business, or produces goodwill in the community, while waiting for market opportunities to emerge later. The figure on the preceding page provides a logical flow for prioritizing opportunities using the Sandcone Model.

Be aware that a corporate perspective is not the same as the customer perspective. An investment promising a great ROI from the business perspective might hold little value for the customer base. Instead, the customers may be in dire need of a different product with a lower ROI. Always remember your customers operate inside of systems and you need to understand them and their system of needs to provide complete solutions for them.

5. Evaluate Large Opportunities (Business Cases) while Managing Risk

Any proposal costing over, for example $25,000, should be submitted with a Business Case. Smaller projects can fit in between the larger projects.

A Business Case is a persuasive document prepared by a project sponsor or their business analyst to explain why the proposed project would be a good investment for the company to pursue.

Business Cases contain basic components such as:

1. Executive summary of the concept.
2. S.W.O.T. Analysis (Strengths, Weaknesses, Opportunities, Threats within the current scenario).
3. Expected ROI (Can included multiple scenarios such as minimal/medium/maximum investment).
4. Expected payback period.
5. How the project benefits will be measured, and who will measure them.
6. Risks, constraints, anticipated budget, anticipated timeline.
7. Anticipated team size.
8. A getting started plan.
The sponsoring executive presents the Business Case, or lets the invited sponsor present the Business Case to the Portfolio Management team to be scrutinized.

6. Scrutinize, Prioritize, and Select the Next Project

As business cases are presented, the committee discusses them and identifies weaknesses in each proposal. If a substantial number of weaknesses are identified, the sponsor is challenged to return with an improved proposal at a later meeting.

**Figure 76 - Opportunity Profile Radar Chart**

The group may want to establish a set of criteria critical to success, and then profile the top opportunities using a radar chart. A

**Figure 77 - Project Prioritization Matrix**

The project with the greatest relative value is weighted highest.
prioritization matrix can also be used to determine the most desirable project funding order.

The Engineering Director and Project Director should be present in the meeting to identify the teams becoming available to begin work on the highest prioritized projects.

Too much task switching within a team slows down all the work being performed by the team by creating artificial delays. It’s best to protect the team’s bandwidth so they can expedite value faster into the marketplace. Remember, we want One-piece flow.

![Portfolio Kanban](image)

**Figure 78 - Portfolio Kanban**

Some groups list ongoing projects on a Portfolio Kanban Board to protect Work In Process limits established by each project team’s natural bandwidth limitations. The Portfolio Kanban Board should govern all significant projects.

Think of each lane as a production runway, one lane per team. We’ve learned through research published by the International Software Benchmarking Standards Group (ISBSG.org) that the optimum team size is four people, with the ceiling being seven. Try to keep your core team sizes split in increments of four, and no larger than seven.

To maintain a consistent flow of project effort over time, teams that work well together are encouraged to stay working together from one project to the next. These teams are considered corporate assets and competitive advantages.
7. Make Changes to Internal Processes as needed.

A final output of a Portfolio Management meeting might include some takeaways for department managers to add to or adjust their quarterly business review goals. If a company uses a Balanced Scorecard, or some similar mechanism to report progress within their departments quarterly, they may be able to use takeaways from this meeting for that purpose.

Using Stable Framework tools and process controls will ensure all necessary considerations are made when needed.

A “Portfolio Management” Kata Card would include these checks:

- Is corporate strategy up-to-date, and visible?
- Have market trends, competitor trends, and regulatory trends been reported on within the past quarter? Are the findings visible?
- Have current status of products & services been reviewed?
- Have current status of projects in process been reviewed? Can any project benefit strategically from more resources?
- Has the business opportunity driver been identified (Add, Protect, Lower Cost, Goodwill)?
- Have new proposals been presented?
- Have new proposals been scrutinized?
- Have new projects been approved and prioritized with group input?

A “Market Update” Kata Card might include these checks for discussion:

- Market description.
- Market size (potential number of customers in terms of units sold, or total dollar value)?
- Current market trends?
- Emerging market trends?
- Largest market players and partnerships?
- Regulatory trends?
- Market vocabulary?
- Market problems or common customer frustrations?
A “Product or Service Analysis” Kata Card might include these checks for reporting:

- Market penetration rate?
- Competitive products?
- Competitive trends?
- Customer satisfaction trends?
- Customer complaint trends?
- Primary Help-desk reported customer complaints?
- Primary Operations costs?
- Primary Operations pain points?
- Net Promoter Score trends?
- ROI?
- Payback Period?

These are initial suggestions for Kata Card contents. As lessons are learned about markets, user experiences, regulatory influences, and other factors, checks can be added to appropriate Kata Cards to help the organization make better decisions going forward.

Post-project RIO evaluations will be a valuable source of updates to these Kata Cards. In highly competitive markets, experience gained in these areas sometimes is the only competitive advantage that a leading company has.

Overall, Portfolio Management can move a company towards their goals faster by maximizing their spending towards their corporate objectives, maximizing throughput by protecting work in process limits, and providing transparency between all parties on how business and IT are connected. If done properly the company will steadily grow with appropriate well-placed decisions.
Organizational transformations are challenging. In order to successfully transform your team, department, or entire organization to use a quality framework like Stable some things need to be in place.

The primary factor for success is executive support. From the top of the company down to the employees working the process steps, there needs to be a visible endorsement of the quality framework adoption program within the company.

Second, your employees need proper training. The best way to get it right from the beginning is to pursue training for your group. Training provides consistent messaging and expectations for each team member and has wonderful side effects such as improved team cohesion and an enhanced sense of employee self-worth by seeing the organization invest in each employee.

Third, your employees need to see visible progress and the positive effects of the new framework. The best way to do this is to identify metrics you can track before the transformation and then compare them to results after the transformation has been implemented.

Be sure to log the victories in an Incident Register, post it in a public place, and share them with your teams and senior management.

You may also consider hiring a Stable Coach temporarily to assist with making sure the process has been implemented correctly, and there are no bad habits forming. Think of this as insurance for your training investment.

**The Path to Stability**

So, how do we turn a chaotic environment into a well-managed world-class organization?

First, we need to have some common understandings. A system is a collection of one or more processes that are related to each other and pass along assets from various entry points to an exit point.

The proper way to engineer a system is from the end to the beginning,
meaning, you first identify what the customer wants, then you backtrack to determine how you will make that possible.

As we build a system to govern the standard work efforts, the system will, in turn, help us execute our work better. Better, in this case, means with fewer errors, and executed correctly the first time. As mistakes happen, the culture of the environment should be that the Master Chief and Process Owners ask themselves “How can we improve the system so that this can’t happen again.” In effect, we focus on improving the system instead of blaming the worker. This is a fundamental principle W. Edwards Deming taught. He told management it was their job to get the teams to improve the systems they work with so that their people couldn’t repeat mistakes.

In order to understand a system, it’s best to get outside of the system and analyze it from the outside looking in. Fish don’t know they are in water. It’s difficult for us to appreciate we are on an orb in space rotating at 1037 miles per hour.

We first must establish a system before we can improve it. We call this systematization, and it involves establishing a value proposition, a system flow, Standard Operating Procedures, and something called a Kata Card, which is a super-checklist. The Kata Card is where we add risk-averse tips learned from past experience. In the beginning Kata Cards may start out small, but they will grow over time. Kata Cards can serve as a platform for collecting data about the work. At the end of each process flow the Kata Cards are signed, dated, and given to the next downstream Process Owner, or the Master Chief if the Kata Card is the final step of the value stream.

A system can be slowed down for several reasons. Bottlenecks slow a system down. Ambiguous inputs slow a system down. Overburden can slow a system down. Inaccurate or incomplete operations due to incomplete training will slow a system down. Scrap and rework will slow a system down. Inspections and warranty work will slow a system down.

A system has only a certain amount of work it can handle at any one time, and a certain amount of work happening within it. We call the amount of work being done Work-in-Progress (WIP) and the limit we call capacity or WIP-Limit. The closer we can get to one item being completed at a time the better. We call this “One Piece Flow.” This concept enables us to put items of value into production much faster than waiting for a
batch of items to be completed. For this reason, task-switching is always discouraged.

If you examine a system from end to end, somewhere within the system you will find the slowest moving step. This is the system bottleneck. The Theory-of-Constraints, an idea put forth by Dr. Eli Goldratt in a book called *The Goal*, teaches us it is important to find the slowest system bottleneck and improve its throughput, and then find the next slowest bottleneck and repeat the process until you are producing a product or service at the rate your customer consumes it. The rate at which your customer consumes your product is called Takt Time.

*The Technology Adoption Lifecycle.*

As new technologies emerge and are adopted by the masses, a common adoption pattern can be observed. We call this the Technology Adoption Lifecycle, and it is based on an older model originally called the Diffusion Process, written about by Bohlen, Beal, and Rogers in 1957. The model states that there are five types of customers when presenting a new idea: Innovators, Early Adopters, Early Majority, Late Majority, and Laggards.

![The Technology Adoption Lifecycle](image_url)

Innovators are the geeks who like to try new things just because they are intrigued. These people generally operate in high risk-tolerant environments, which means they usually haven’t reached the point in their careers where they command budgets.
Early Adopters are people with budgets but are somewhat risk-averse. They will patiently wait for the innovators to experiment with the new technology before they try it out. They are interested in case-studies and documented proof of success.

The Early Majority is the first half of the market segment that adopts a rising star technology offered in its marketplace.

The Late Majority is the second half of the market segment that adopts a rising star offered in its marketplace. The Late Majority is usually distracted by budget needs elsewhere, older technology or policy limitations and cannot move as fast as the early adopters.

The Laggards are those who see the new technology as a benefit, but also a hassle. They are the last of the market to conform because they are complacent and happy with their current situations and will only change if forced by company policy, market demands, crisis, or technological isolation.

**The Value Realization Curves**

Another transformational dynamic to be aware of is the Time/Effort Value Realization curve, which demonstrates that it takes some
consistent effort over time at first before progressive results start to
emerge. For this reason, it’s best to measure your environment before
and after adoption, and to post victories publicly anytime your quality
program saves anyone from rework, scrap, or any other problem.

**Quality Planning**

The initial implementation of Stable is called Quality Planning and is
directed by the Master Chief. In this stage, the core structure and tools
need to be made ready for the team.

The team should not be encumbered by having to figure out where to
store process controls or where the standard Kata Card history is
located.

Of course, the Master Chief can consult with the team on all these
decisions, but in the end, if these items are not set up and in place, the
Master Chief has not prepared the team adequately.

Quality Planning should be revisited every six months or so to be kept
current.

The next section details the most common approach to setting up the
Stable Framework™.

**Initial Setup**

1. Review The Stable Framework™ – Review the Stable Framework with
   your team. Discuss the challenges in your areas and the need for
   improvement. Get their commitment to learn the framework and agree
   on a go-forward plan.

2. Create a Process Asset Library, otherwise known as a Configuration
   Management System – Identify or create a physical location where all
   the lists, templates, and other process controls will be stored and
   updated. Ideally, this should be a version-controlled document
   repository. If that is not available, use a shared network folder with
   appropriately named sub-folders that get backed-up regularly.

3. Create a Service Register - A list of Master Services, Pending Services,
4. Create an Incident Register - A side-by-side comparison of Hidden Factory problems causing rework or scrap, compared with potential problems found and remediated by the Stable Framework.

5. Create a System Schedule - A schedule or calendar showing all the future planned events, including scheduled asset maintenance and any repeating timed events like end-of-month activities, or quarterly reviews.

6. Create a System Backlog - The System Backlog is a single location organized by four work queues: The Customer Request Queue, the Scheduled Activity Queue, the Equipment Maintenance Queue, and the CCAPA Queue. These separate queues are the landing site for incoming work. These can be as simple as lists in a spreadsheet or sticky notes on a wall.

7. Create a Performance Console - Otherwise known as an Information Radiator, this is a physical area at the work-site where queues, the Kanban board, work in process, and reporting assets are posted. In Lean, this concept is called Visible Management and we want to make all work identifiable and posted here.

8. Create a Kanban Board - A Kanban board will be the centerpiece on the Performance Console. This is a multi-column tool used to track the following work items: Objectives, Selected Tasks To Do, Work In Process, and Work Completed. Optionally, columns can be added for additional categories such as Pending Approval, Approved, On Hold, To Test, To UAT (User Acceptance Testing), etc. Your environment dictates which columns would be most appropriate to construct for your team.

9. Create Historical Logs - These logs could include a Production Change log, Kata Card repository, Training logs, and a SLAM Chart repository.

10. Create Customer Satisfaction Surveys and Histograms - Customer Satisfaction should always be measured using a histogram, showing changes over time. While the Process Owner is responsible for gathering customer information to improve the process outcomes, the Master Chief is responsible for gathering Customer Satisfaction data.
11. Create SLAM Charts - SLAM (Service Level Attainment Monitoring) Charts are the primary means for reporting on the success of a service over time. All primary services should have SLAM Charts posted for them on the Performance Console and should be updated daily. SLAM Histograms should also be tracked to show trends over time. SLAM Charts are created and owned by the Process Owners.

12. Create a Cumulative System Performance Chart – Create a CSP Chart, and as it gets populated by performance information during the quarter, be prepared to present it at a Quarterly Business Review meeting, if your organization holds one.

13. Systematize Value Streams – Review the 7th step of the Stable Framework. Working with the team, identify all value streams within your area. Systematize each one by creating process maps, collecting customer information, expected Service Levels, and creating SOPs and Kata Card templates. Create Process and Asset Recovery Models, and Lessons Learned documents. *When identifying repeatable processes, or Kata’s, it’s best to start at the end of each value stream and work backwards towards the beginning.*

14. Determine Length of Master Cycle – Working with the team, identify the desired length of the Master Cycle within your areas. Anywhere from one to four weeks is good. If you work alongside an Agile team, it’s best to match the Master Cycle to their Sprint pattern so that you can report together.

15. Establish Master Cycle Meetings – Establish the Master Cycle Planning Meeting, the daily Kaizen Stand-up Meeting time, and the Master Cycle Review Meeting, and Retrospective Meetings.

16. Establish Inter-department Meetings – If they don’t already exist, you will need to establish a Change Control Committee meeting, a regular CCAPA meeting with the Service Desk, and be part of the engineering status update meetings.

17. Establish an Alert System – Off the shelf products exist for this purpose, or you can build your own. Start simple and add to it as needed. In time, it will become valuable.

18. Set up a Training Log – Establish a training registry showing who has been trained for each recurring process. Indicate the date/time so retraining can happen in six months or so.
19. Set up a Suggestion Box – Put a visible suggestion box near or on the Performance Console. Tell your group about it. Encourage suggestions and act on them.

Then, get started....

Be sure all your people get proper training. We learned from the Agile revolution that an untrained staff is far less effective than a properly trained staff.

Taking Over as Master Chief

If you assume the role of Master Chief in an existing area, these are the steps you should follow:

1. Confirm the location of the Configuration Management System.

2. Bring all Lists up-to-date.

3. Encourage all Process Owners to retrain and bring all their Standard Operating Procedures and Standard Kata Card Templates up to date, and to be sure the System Schedule is up to date.

4. Be sure the System Schedule is up to date by meeting with your executive sponsors and primary customers and recording anything relevant that your team will be involved in.

5. Be sure you have a Performance Console and it is kept up to date.

Sustaining the Stable Framework™

The key to a successful Stable Implementation is the transformation of everyone involved in the experience. Senior Management ought to understand that their support of the practices enables them to see at a glance the total and complete status of everything happening within the department clearly, leaving them free to spend their time growing their business. This is what is meant by Operational Excellence.

To the Master Chief, the Stable Framework™ means structure and
guidance in introducing and applying techniques that empower teams and enable accountability at the same time. It’s also a structured approach to transform tribal knowledge into institutional knowledge, and continually improve the most important metrics and the people within the Master Chief’s department.

For the Process Owner, it provides structure, tools, and a realization that an invisible process exists which will protect them from future mistakes if they will respect it and work with it. The Stable Framework™ also provides the assurance that mistakes will be blamed on less-effective systems needing improvement, and not the operators.

For the customer, Stable provides metrics, visibility, and as much communication as the customer wants. If managed correctly, your customers’ satisfaction levels should rise consistently over time.

For the supplier, consistent feedback from the Stable organization will provide clearer direction and insight into making the suppliers’ services and products a better fit for its markets over time.
Chapter 25 - Data Measuring & Modeling Tools

This chapter is devoted to showcasing a set of quality charts and tools useful for data collection and data analysis. They are grouped into two categories. Data Measuring tools display numeric data. Data Modeling tools display relationships.

They can be applied to Quality Planning, Incoming Quality Assessment, Quality Assurance, Quality Control, and Continual improvement.

Data Measuring Tools

Data Measuring Tools are used to interpolate real world data into an organized form for knowledge generation, reporting or further analysis.

Burn-up Chart

A Burn-up Chart is an excellent tool for tracking team velocity over time. It’s organized by units of time along the horizontal axis, and units of work along the vertical axis. Note the series along the top showing the total effort, or scope, identified and the total effort completed diagonal line near the diagonal straight line showing planned velocity.

![BurnUp Chart](image)

*Figure 81 - Burn-up Chart*
A Burn-down variant of this chart exists, but lacks the ability to show total scope of effort changes across the top.

**Capability Chart**

A Capability Chart is used to show various stages of capabilities concerning the relationship between two factors.

The example below shows the range in proportional effort forecasted to generate thousands of source lines of code (KSLOC) over time. Note the tradeoff between overstaffing, understaffing, and optimal staffing. Also note the Productivity Index (PI) driving the forecasted output.

The chart above compares results of two different Productivity Indices (PI) on work to be performed. It indicates that a development organization with a PI of 11 will complete the same amount of work in 3 months as a team with a PI of 10 will accomplish in 3.5 months.

PI indexes can be calculated for organizations using Lawrence Putnam’s SLIM Method.
Control Chart

The Control Chart was invented by Walter Shewhart in 1924 and helps determine if a process being measured is In-Control or Out-of-Control. A process In-Control has no external factors impacting it; therefore, its output variation is naturally random—as it should be. However, if a process is Out-of-Control, then some outside factors are impacting the primary process generating unwanted variation.

There are two ways to determine if a process is Out-of-Control. The first is to see if any data point is outside of the Upper Control Limit (UCL) or Lower Control Limit (LCL) of the analyzed data set.

The second method is to watch for seven consecutive data points in a row above, or below the center line (Mean), or cascading up or down. This second method is called the Rule of Seven.

The UCL and LCL are determined by measuring the average of the differences of the data, which is called the Standard Deviation, and then counting three standard deviations above and below the center of the data values (called the Mean of the data).

![Figure 83 - Statistical Process Control Chart](image-url)

Typically, a process is engineered so that its UCL and LCL are just within the customers specified variation range. We call this customer supplied
range a Specification Limit, or a Tolerance.

A company trying to improve its page load speed could measure random samples of pages loading. The company might come up with a data set like the one shown in the figure above.

This chart reading indicates that the average page load speed is 3.652 seconds but can vary from 1.137 to 6.167 and still be In-Control. If suddenly we recorded a reading of 6.2 then some anomaly is responsible for that error and could be attributed to an operating system update, or database backup, or some similar action happening in the background that is competing for system resources.

Focus Group

A Focus Group is a small group of panelists who have experienced a problem or interacted with customers who have experienced a problem. Led by a facilitator, the panel answers questions collectively to educate the others in the meeting about a recurring problem. The purpose of a focus group is to learn more about a problem, not to solve it right there in the meeting.

Hastings Chart

A Hastings Chart is used to track project work, sustainment work, and hidden factory over time.
Project effort is the work dedicated to furthering project goals. Sustainment effort is the work involved with performing daily unscheduled on-demand production-change requests from other sources. Hidden factory is the work involved with repeating steps in project or sustainment efforts because something went wrong along the way.

### HOW OUR TIME IS SPENT

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Project Efforts</th>
<th>Sustainment Efforts</th>
<th>Hidden Factory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle 1</td>
<td>72</td>
<td>2.4</td>
<td>2.4</td>
</tr>
<tr>
<td>Cycle 2</td>
<td>68</td>
<td>4.4</td>
<td>2</td>
</tr>
<tr>
<td>Cycle 3</td>
<td>74</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Cycle 4</td>
<td>76</td>
<td>2.8</td>
<td>4</td>
</tr>
</tbody>
</table>

**Figure 84 - Hastings Chart**

### Histogram

A Histogram is simply a chart showing a historical collection of data points concerning an output. Some Histograms compare multiple outputs to each other.

**Figure 85 - Histogram**
Pareto Chart

A Pareto Chart is used to show frequency of causes, prioritized by highest to lowest amounts, and is often used to track causes of scrap and defects.

![Pareto Chart](image)

**Figure 86 - Pareto Chart**

The Bar Chart portion of the Pareto is linked to the left axis, showing frequency. The hairline portion is linked to the right axis, showing percentage. Then, for example, if you want to find the source of 80% of your system outages, you would see where the hairline meets the 80% indicator on the right axis and sum the bar frequencies left of that point.

In this example, the first three bars from the left represent 80% of the reasons why the system has gone down.

Radar Chart

A Radar Chart is useful for identifying bottlenecks or problem areas that need improvement. With established criteria, each factor can be assessed for 0-100 range of value.
It also can be used to compare the characteristics of similar items.

**Scatter Diagram**

A Scatter Diagram is used to show if a relationship exists between two
variables. The following figure is comparing a programmer’s performance estimates versus actual performance. Over time a trend appears showing the actual time each work item takes which is about 1.5 times longer than the estimate. Therefore, in the future, the programmer should multiply his or her estimates by 1.5 to produce an estimate that is closer to reality.

Sensitivity Analysis

A Sensitivity Analysis is a process where you first identify all the variables that impact an outcome, then one by one you change those variables and measure the result. This provides insight into which variables have the greatest impact on an outcome.

Data in the Figure below comes from Dr. Barry Boehm’s COCOMO II sensitivity analysis conducted in the early 80’s. After analyzing hundreds of software projects his team determined that requirements skills have the greatest amount of impact on the overall effort required to produce software above any other factor listed.

In other words, a trained requirements team would reduce the overall effort required to complete the software by 29%, whereas a poorly-trained requirements team would add 42% more effort than an average team to produce the same software.

Figure 89 - Sensitivity Analysis
SLAM Charts

SLAM Charts are tools for collecting performance data for process outputs. The simplest SLAM Chart, the first chart shown below, would indicate if a process completed successfully each day. A simple dot indicates success, and a number indicates a problem and corresponds to the problem explanation at the bottom of the chart.

Figure 90 - Simple SLAM Chart

Figure 91 - Complex SLAM Chart
Note that one SLAM Chart can display performance information for multiple systems of interest.

The second example below is a more complex SLAM Chart showing statistical process control data for identifying trends and the amount of variation found in the system.

**Statistical Sampling**

Statistical sampling is useful for identifying trends in a larger population, without having to collect information on the entire population. Thanks to the central limit theorem, patterns present in a large population will be observable in a smaller sampling of the same population.

Surveys are excellent tools for this but be sure your questions are unbiased and don’t persuade the responder to select certain choices. Try to get at least 30 or more samples.

**Suggestion Box**

Be sure a suggestion box is visible and checked regularly. Provide contests for the number of suggestions that come from different teams. One simple litmus test of how well Stable is working in your environment is by measuring how many suggestions are made each month, quarter, and year.

**Kata Card (Checklist, or Checksheet)**

A Kata Card can be as simple as a checklist, but often it is more involved. Every widget moving through a production facility has information that needs to be tracked along with it. A Kata Card is used for this. They are also used to ensure the Process Owner has considered every risk factor known and ensured the problem is not present in this instance. In practice, they are sometimes called “checksheets,” “travelers,” or “routers.”
Data Modeling Tools

Data Modeling Tools are used for organizing information into contexts and relationships. These representations help to facilitate discussions, generate knowledge, and support decisions.

Activity Network Diagram

An Activity Network Diagram is used to sequence activities, and then determine critical path, early start, late start, early finish, late finish, and total float time for each activity.

Activity Network Diagrams can be excellent tools for sequencing complex logic or logistical implementation procedures. A similar diagram that is used for the same purpose is called an Activity-On-Arrow diagram.
Affinity Diagram

An Affinity Diagram is a group brainstorming tool that collects ideas, opinions, and issues from a group and organizes them into topical clusters.

To perform the activity, each group member is given a stack of blank sticky-notes or index cards and encouraged to write a separate idea down on each card. After enough ideas are written, everyone’s ideas are combined on a table-top or a wall, and a facilitator separates the ideas into topical clusters.

This tool is especially helpful for getting a lot of ideas or information from a group in a short amount of time, or if a group cannot agree on categories.

Figure 94 - Affinity Diagram

Cause and Effect Diagram, or Ishikawa Diagram

Also called a “Root Cause Diagram,” “Ishikawa Diagram,” or “Fishbone Diagram,” this diagram was invented by a Japanese shipyard quality manager named Kaoru Ishikawa.

These diagrams are useful when a problem originates within a process and a group is collectively brainstorming the root cause.

There are two types of Ishikawa Diagrams. Theme-based diagrams, and process-based diagrams. Theme-based diagrams contain a collection of common areas for which a problem might occur. Process-based Ishikawa Diagrams contain all the steps of a process to be considered when searching for a root cause.
A typical approach for using an Ishikawa diagram is to have your team brainstorm challenging problems experienced with a particular process. Each problem is listed on the diagram under the appropriate area.

After the team has exhausted their list of known problems, they take a second pass and brainstorm solutions to each problem.

The Ishikawa Diagram is often combined with the 5 Why's questioning technique.

![Figure 95 - Cause and Effect, or Ishikawa Diagram (Theme Based)](image)

![Figure 96 - Cause and Effect, or Ishikawa Diagram (Process Based)](image)
Context Diagram

The Context Diagram is useful for expressing the boundaries of a system and the external entities with which it interacts.

The circle in the center represents the solution your team needs to build and the boxes around it represent external systems your solution needs to interact with.

The external systems may represent existing systems or systems a different team is responsible for building.

The arrows represent interaction points, and a diagram may display more than one set of arrows connecting the circle to a particular box.

Figure 97 - Context Diagram
A Facilitative Workshop is a meeting called for many purposes. In it, a team can brainstorm for a solution, conduct an As-Is analysis, provide a To-Be walkthrough, or do any other group-interactive activity. Some Facilitative Workshops span multiple days.

Flowchart

Flowcharts are used to map out a process flow so that it can be analyzed, understood, and improved. Each icon represents a distinct type of activity such as a decision, a process step, a document to be approved, etc.

Modern flowcharts are typically organized by lines called swim lanes which indicate who is responsible for what activities. This can be helpful to clarify the division of labor within a team, department, or company.

Flowcharts can also be used to describe the steps in a process.

Similar to swimlanes, some flowcharts have phase separators perpendicular to swimlanes which segment steps from one phase to another.

The flowchart shown on the next page is a software testing process flowchart. Note the size difference with this diagram compared to the Sequence Diagram that follows it.
This next figure is a UML Sequence Diagram, but it makes an excellent flowcharting tool showing process ownership and checkpoints. This
A Gantt Chart is a common approach to laying out anticipated work in a dependent order. Gantt Charts are useful for showing project task breakdowns, dependencies, and to show the critical path through the workload, which helps keep the project on schedule.
Interrelationship Diagram

This tool is a way to show causes and effects within a scenario.

A problem is written in a circle. Then one or more results of that problem are written in another circle, and an arrow is drawn to connect each cause-and-effect relationship. The process is then repeated on each one of the newly created circles until all cause-and-effect relationships are represented and connected.

A count is made of the incoming arrows versus the outgoing arrows for each circle, or node. Nodes with high incoming arrows represent the biggest areas of concern and are good targets for measurement. Nodes with the highest outgoing arrows represent the biggest sources of problems.

![Interrelationship Diagram](image)

**Figure 101 - Interrelationship Diagram**
This is an excellent group collaboration tool. Many causes and effects can be exposed and discussed in a short amount of time with input from an entire team. Some people call this the “vicious cycle diagram.”

Consider creating this diagram along with an Ishikawa diagram while flow-diagraming a system for improvement ideas. All three charts complement each other.

Matrix Diagram

A Matrix Diagram is used to organize and show relationships between factors. There are seven types of matrix diagrams: L, T, Y, X, C, R, and Roof shaped, depending on how many groups must be compared.
Prioritization Matrix

A Prioritization Matrix is a tool used to rank options against each other. When a Prioritization Matrix like the one shown below is populated (all blank cells contain 0 or 1) the relative value column on the right will show a list of projects prioritized from 10 down to 1.

![Prioritization Matrix](image_url)

Process Decision Program Chart (PDPC)

A Process Decision Program Chart is used to pre-plan risk contingencies. Any anticipated risk can have a plan made to mitigate the problems posed by the risk.

PDPC Charts are excellent compliments to addressing identified risks uncovered during a risk analysis and are often used with a Failure Mode Effects Analysis (FMEA).

The mark of seasoned professionals in any part of IT is a healthy respect for risk management and their generation of plans to anticipate and mitigate risks.

If you think about it, the reward for completing a large and risky project successfully is a larger and even riskier project. This continues throughout an IT professional’s career until risk management becomes a time-consuming and serious part of planning and reporting. It’s best to learn how to do it properly now.
Implementation

Go On site
(Objective 1)

Setup On-site Internet Connection
(Objective 2)

Miss Flight
(Problem 1.A)

Staff Not There
(Problem 1.B)

Internet Service Provider Not Completed Install
(Problem 2.A)

Arrive and Check-in with Mgr 1.5 hours before flight
(Countermeasure 1.A.1)

Cut in front of Zone Passengers
(Countermeasure 1.A.2)

Bring Office Manager’s Phone Number
(Countermeasure 1.B.1)

Don’t Book Flight Until ISP Setup is Completed
(Countermeasure 2.A.1)

Figure 104 - Process Decision Program (PDPC) Chart

Tree Diagram

A Tree diagram is used to examine two or more variables. The first variable is represented at the first split. The second variable is represented at the second split, and so on. The result is that all possible resulting combinations are represented at the end (rightmost) nodes.

Figure 105 - Tree Diagram
**5S Housekeeping.** A set of workplace organization methods that use a list of five phrases, all beginning with “S”: Sort, Set in order, Shine, Standardize, Sustain. Adapted from Japanese: Seiri, Seiton, Seiso, Seiketsu, Shitsuke.

**Acceptance Screen.** A mechanism used to mistake-proof an incoming or outgoing asset so that it will interact correctly with its intended environment. In Japanese this is called “Poka-Yoke.”

**After Action Review (AAR).** A meeting called right after an event where the facilitator presents what happened compared to what was supposed to happen, and the group discusses why it happened the way it did, if it could have happened better, and how it could have happened better.

**Agile Mindset.** A set of practices that enable groups to produce and deliver what is most valuable to the customer, again and again. Typically, this includes close engagement with customers, short feedback cycles, multiple levels of planning, and pivoting quickly when changes are needed.

**Alert System.** A system put in place to show the real-time status of assets in an environment. This solution could be a scoreboard monitor hanging in a work area, or a message emailed or texted to a group list, or something similar.

**Apollo Team.** Also called the Apollo Syndrome, a phenomenon identified by Dr. Meredith Belbin where teams of highly capable individuals can collectively perform badly. This is attributed to excessive time in destructive debates, omitting needed work, working in silos, and avoiding necessary confrontation.

**Asset Controls.** All the governing information and instructions pertaining to an asset. This includes Asset Recovery Models, Process Asset Matrix, Asset Maintenance Logs, and Asset Purchase and Warrantee Information.

**Asset Recovery Model (ARM Sheet).** A guide associated with an asset, for quickly diagnosing and remedying one or more problems that may occur while using the asset. The model is based on past problems and it
gets added to as new problems and solutions are discovered. Its purpose is to shorten asset downtime.

**Blacklisting.** A security access technique where all entities are granted access except those on a list. See also Whitelisting.

**CCAPA Committee.** A committee staffed to address nonconforming problems discovered by customers while using the product or service. CCAPA means Correction, Corrective Action, and Preventative Action, which is a pattern for dealing with problems. Correction is fixing the specific customers situation. Corrective Action means changing production so that no one else experiences that problem in the future. Preventative Action means changing any other similar processes that may benefit from a similar Corrective Action. A CCAPA Committee should meet daily, weekly, or once each Master Cycle.

**Checkpoint.** A coordination point between processes or process steps where a completed Kata Card is handed off to the next Process Owner at the receiving process step. At the final checkpoint, the accumulated stack of Kata Cards is given to the Master Chief.

**Coaching Kata.** A pattern used by a Master Chief to motivate a Process Owner, in a structured manner, towards a goal.

**Configuration Management System (CMS).** A storage repository for Master Lists, Process Controls, and Asset Controls.

**Control Points.** A strategic checkpoint measured to provide summary information for the organization.

**Critical Success Factor (CSF’s).** An action necessary for a project team or organization to achieve its mission. A CSF can also be thought of as a Key Result (See “Objectives and Key Results (OKR’s)”).

**Crux Design.** An analysis, design, and estimation technique used to quickly model a reasonable representation of a system and provide a corresponding estimate for time and resources needed to construct it.

**Culture.** The anticipated interactions between two or more people. In organizations, culture is typically polarized on two axes: Independent vs. Team-focused, and Flexible vs. Rigid. Culture is ever-changing and can be altered over time through shared memories and repeatedly emphasizing important topics in conversations.
**Cumulative System Performance Chart.** A chart showing Master Cycle performance trends over time, to be updated at the end of every Master Cycle. Typically, this chart or a variation of it is brought to a Quarterly Business Review (QBR) meeting.

**Customer Value.** Utility a paying customer receives in exchange for currency.

**D.I.R.F.T.: Do It Right The First Time.** A philosophy developed by Phillip Crosby to provide an environment where operators performing repetitive processes could perform them consistently, without setbacks or failures.

**Disciplined Agile.** Also called Disciplined Agile Delivery, an agile development model promoted by the Project Management Institute® which provides an array of approaches to use depending on the target product lifecycle.

**Facilitative Workshop.** A group-based meeting held for many purposes, namely brainstorming, knowledge-transfer, walk-throughs, or other planning discussions. Some Facilitative Workshops can span several days.

**Failure Modes and Effects Analysis (FMEA).** A systematic method for evaluating a process to identify where and how it might fail, and to assess the relative impact of different failures, in order to identify parts of the process most in need of change or contingency planning.

**FDA 21 CFR Part 11.** A set of digital data security standards established by the Federal Food and Drug Administration.

**Focus Group Meeting.** A meeting held with a small group of people familiar with an issue to understand the issue better.

**Hidden Factory.** Excess work, rework, scrap, wait-time, gold-plating and other cost-draining practices that represent effort expended that the customer does not need to pay for.

**HIPAA.** Health Insurance Portability and Accountability Act of 1996 is legislation passed to protect medical information.

**Hot-fix Haze.** Any environment without a functioning quality program, characterized by frequent rework, fire-fighting, and heroic efforts to deliver value.
**Incoming Quality Assessment (IQA).** Inspecting incoming materials and information to be sure they meet requirements before incorporating them into an internal product build.

**Institutional Knowledge.** Information learned on the job that has been transferred to standard operating procedures, lessons learned, risk lists, asset recovery models, and other permanent records so that the entire company will benefit from what was learned. Sustained Continual Improvement is based on adding to and revising Institutional Knowledge.

**Incident Register.** An Incident/Incident Avoidance Register is a two-part document showing the efforts and costs lost during a measured time period due to repeated steps needed to correct a problem, and a list of estimated effort saved due to potential incidents detected and avoided thanks to quality program efforts.

**Just-In-Time Manufacturing.** An industrial technique where external material suppliers are challenged to deliver needed goods right as they are needed for production, and not sooner. This eliminates expensive warehousing costs.

**Kaizen Blitz.** A team improvement effort organized by a request for help from a Master Chief or a Process Owner. A Kaizen Blitz is different from a Kaizen Project in that a Blitz is triggered as needed and should only last a few days, whereas a Kaizen Project is pre-planned and may consume an entire Master Cycle.

**Kaizen Project.** A problem within the department, or with a product or service, that the team or everyone on the team takes on to improve during the Master Cycle. This challenge is identified during the Cycle Retrospective Meeting and reported on during the next Cycle Review Meeting.

**Kaizen Team.** A team created as needed to solve a problem. These teams seem to produce the best results if they are staffed by volunteer participants. Sometimes these are called Ad-hoc Kaizen Teams, or Ad-hoc Teams.

**Key Performance Indicators (KPI’s).** Measurable data collected to determine how organizational or team goals are progressing.

**Lessons Learned Log.** A historical record of knowledge gained over time. Typically, this is part of a Process or Asset Recovery Model, and critical
findings to watch out for in the future are added to the Kata Card for that step.

**Market Summary Report.** A summary of happenings or news from the customer base delivered every morning by the Master Chief to the Process Owners in the morning Kaizen Stand-up Meeting. This information is sourced from senior management, operations, and the Service Desk.

**Master Cycle.** A cadence rhythm established for reporting and measuring purposes. It should last between one to four weeks. If your group works alongside an Agile group, you should synchronize your cycles with theirs so that you can report together.

**Master Kaizen.** In a large Stable Framework™ implementation, it may be necessary to stagger Kaizen Meetings. The Master Kaizen Meeting is a meeting of representatives from each Kaizen Meeting who report on the highlights of what their teams are doing.

**Master Lists.** Lists of important information needed to be maintained within a system.

**Net Promoter Score.** A score measuring how probable a customer base is to recommend a business.

**Nonconforming Issues.** Problems found by customers and brought to the attention of the Process Owner, usually via the Service Desk. These problems are prioritized and addressed by the Process Owner, or the CCAPA Committee.

**Objectives and Key Results (OKR’s).** A goal-setting pattern consisting of an Objective, and a short list of self-evident Key Results needed to meet the objective.) The objective is considered either “met” or “not met.” In a well-run company, OKR’s are established and made public at all levels of the company. This allows line employees and middle management to ensure their OKR’s match senior level OKR’s. This synergy enables well run companies to meet their goals faster. Key Results can also be thought of as CSF’s. (See “Critical Success Factor (CSF)”)

**One-piece Flow.** Also called One-piece flow. The idea that producing one item at a time provides value to the marketplace faster than batching items through production. Early items within the batch must wait for all
the other items to be completed in the batch before they can all move to production. For this reason, we minimize task switching and minimize work-in-progress as much as possible.

**Operational Excellence.** Producing exactly what the customers want (effectiveness), and doing so with minimal byproduct (efficiency). An Agile Mindset enables the first goal, while a Stable Mindset enables the second. A quality program is needed to obtain the transparency and conscientiousness required to ensure operational excellence.

**Performance Console.** Sometimes called an Information Radiator, this is a dedicated space on a wall or rolling board for the team to report on the performance of their processes, customers, and supplier services.

**Personal Coefficient.** A factor introduced by Fredrick Taylor in his book *Scientific Management*, which attributes differences between separate individuals performing the same task.

**Personas.** Fabricated personalities created by the team and used to calibrate user experience expectations and gain insights into additional user needs.

**Plan-Do-Check-Act.** A hypothesis testing pattern created by Walter Shewhart and made famous by W. Edwards Deming. When you have an idea you “Plan” how to try it, “Do” the experiment, “Check” the results, and then “Act” by changing production if the result was desirable.

**Planning Package.** A vague unit of work within a Work Breakdown Structure (WBS) that has not yet been decomposed into one or more work packages, due to excessive remaining ambiguity.

**Poka-Yoke.** A Japanese word that means mistake-proofing, or error prevention. This is a technique where the use of a design change or added tool will prevent inadvertent human error. For example, the plastic shim in a USB port and plug, that prevents the user from inserting the plug the wrong way is a Poka-Yoke. In Stable we call these Poka-Yokes “Acceptance Screens.”

**Process Controls.** All the information and instructions governing the use, maintenance, and recovery of a process. This includes Standard Operating Procedures, Process checklists we call Kata Cards, Service Levels, Recovery Models, etc.
**Process Kata.** When a Process Owner executes a repeatable process, this is called a Process Kata. A certificate indicating it was completed successfully is the completed, signed Process Kata Card.

**Process Quality.** How well the processes were performed, according to the written procedure and with the least amount of excess, waste, rework, delays, or any other unnecessary byproduct. This is measured during Quality Assurance efforts.

**Process Register.** A list of all repeatable process steps that comprise a service. See Service Register.

**PRE, or Process Reliability Engineering.** Quality frameworks like The Stable Framework™ provide a toolset for shoring up a repeatable process so that the operators performing it always Do It Right the Frist Time (D.I.R.F.T)

**Product Quality.** How well the product matches the expectations of its customers. This is measured during Quality Control efforts.

**Product Backlog.** A list containing intentional functionality to add to a project workload.

**Pull System.** A manufacturing system triggered to produce the next product when the next sale occurs. This configuration is the most efficient due to minimal warehousing needs and assumes all upstream materials and components are in place queued for the next sale.

**Push System.** A manufacturing system based on queuing batches of products and storing them until the next sales are made. This system is less efficient than a pull system.

**Quality Debt.** The fact that costs involved with repairing a defect rise almost exponentially as they cascade down from where they were introduced to later value stream steps.

**Quality Planning.** The initial setup and identification of quality goals within an environment, or a project. In the Stable Framework™ this involves establishing your Configuration Management System, Master Services, Supplier Services, Service Levels, Standard Operating Procedures, Kata Cards, Asset Recovery Models, Work Queues, Kanban Board, Suggestion box, Master Cycle, Cycle Planning Meeting, Daily Kaizen Meeting, Cycle Review, and Cycle Retrospective Meetings.
Radical Candor. A feedback model developed by Kim Scott, made popular in her book by the same name, *Radical Candor*.

Requirements Rot. A natural phenomenon where changes in the environment, marketplace, government regulations, and human interests lead to requirements becoming less relevant over time if neglected.

Service Register. A list of various services your group has stewardship over. Typically, this is organized by Master Services, Pending Services, Retired Services, and Supplier Services. Each service may have its own Process Register, or list of all repeatable process steps used to provide that service.

Service Level Agreement (SLA). A formal or informal contract between a customer and a supplier for a certain level of service to be expected.

Service Level Attainment Monitoring (SLAM) Chart. A chart showing the Master Cycle performance of a process of interest, against its Service Level Agreement, or Service Level Goal.

Service Level Goal (SLG). An informal standard internally set by the production team to provide services at a particular level.

Service Level Histogram. A histogram displaying the past trend of service level results.

Service Oriented Architecture. A style of software design where services are provided to other components across the enterprise, via a common documented protocol.

Shift-Left Thinking. The maturation of awareness that quality practices should be happening farther upstream to prevent downstream issues. This is related to "Prevention over Inspection."

Single-Piece Flow. See One-Piece Flow.

SOC 2. An auditing procedure developed by the American Institute of CPA’s (AICPA) which defines criteria for managing customer data based on five principles: security, availability, processing integrity, confidentiality, and privacy.

SRE, or Site Reliability Engineering. A concept introduced by Google expressing all of the effort and practices involved with developing and supporting robust IT service platforms.
SPRE, or Site & Process Reliability Engineering. The combination of Site and Process Reliability Engineering. PRE + SRE = SRPE.

Stable Mindset. A collection of practices that provide process governance and produce sustained continual improvement over time. These include identifying critical processes, systematizing them, improving them over time, and measuring performance improvements.

Stable-Agile. Also called “Stable-Scrum,” this is a name given to the practice of combining Agile and Stable methodologies.

Standard of Care. A medical term indicating a pre-set list of activities which should be performed specific to a patient’s condition.

Standard Work. Repeatable steps within a process.

Standardize-Do-Check-Act. A variation on the Plan-Do-Check-Act model used for Quality Planning while creating Standard Operating Procedures (SOP’s). The Process Owner “Standardizes,” or creates the SOP, “Does” the process step, “Checks” to be sure the SOP matches the step, and if not “Acts” by updating the written SOP, until it is accurate enough that somebody skilled in the art, but not necessarily from that environment could repeat the steps with that level of written instruction.

Story Point. A reasonable amount of work that can be accomplished by one skilled in the art, with minimal distraction, in about a day.

Supply Chain. The collection of upstream suppliers providing needed technologies, information, supplies, or services as inputs to your processes. In the Stable Framework™, we call these Supplier Services.

T.I.M.W.O.O.D.S. An acronym to help remember the eight wastes in Lean Manufacturing Theory. They are: Transport, Inventory, Motion, Waiting, Over Production, Over Processing, Defects, and Skills Underutilized.

Takt Time. This is the speed at which your customers consume your products. Used as a calibration mechanism for performance tuning the rate at which you should be supplying services.

Team Development Model. A pattern identified by Bruce Tuckman describing how teams Form, Storm, Norm, Perform, and then Adjourn through natural transitions.
**Tribal Knowledge.** Knowledge developed while on the job and stored in the minds of workers. The goal of Process Owners is to systematize their knowledge so that it is transferred into Institutional Knowledge. Once this transition occurs, real Process Improvement can begin.

**Trigger.** The starting mechanism to begin a process.

**Value Proposition.** The incentive that creates a customer. Each process should have a value proposition.

**Value Stream.** The collection of upstream and downstream processes, process steps, materials, information, and suppliers producing a product or service of interest to a customer.

**Victim Triangle.** Also called the Drama Triangle, a model of human dysfunctional interaction where an underperforming “victim” complains about a “persecutor” disabling the “victim’s” ability to perform an agreed-upon task and asks the manager to “rescue” the “victim” from the consequences of a neglected task by excusing it this time. Instead, the manager should challenge the “victim” to express what he or she could have done differently under the same oppressive circumstances to still meet the commitment, and then challenge the “victim” to do that next time.

**Whitelisting.** A security access technique where no entity except those on a list are granted access. See also Blacklisting.

**Work Breakdown Structure (WBS).** The total scope of a project to be delivered, organized into logical deliverable structures called Work Packages, or Planning Packages.

**Work In Progress (WIP).** Also called Work in Process, this term describes all the work presently being performed by the team, or an individual on the team. WIP-Limits are established to ensure the system is streamlined, but not overburdened. Overburdening a system causes unnecessary delays due to task switching by humans and equipment.

**Work Package.** The subcomponent within a Work Breakdown Structure to be completed or delivered, usually composed of one or more activities.

**WIP-Limit.** The amount of bandwidth available or dedicated to a type of work. If a WIP-Limit is breached, a bottle-neck emerges inhibiting the flow of value through the system.
Bibliography

Introduction

- Jeff Sutherland, *The Roots of Scrum: Takeuchi and Self-Organizing Teams*, Mar 7, 2010 ScrumInc.com blog
- *The 12 Principles Behind the Agile Manifesto*,
- “Stable-Agile” and “Stable-Scrum” terms coined by Lynn Wright

Chapter 1 – The Problem

- Dr. Atul Gawande; *The Checklist Manifesto: How to Get Things Right*; Jan 2011; Picador
- The Standish Group International,

Chapter 2 – The Hidden Factory in IT

Chapter 3 – System Thinking and Value Streams

- Mark Schwartz; *The ART of BUSINESS VALUE*; 2016; IT Revolution

Chapter 4 – Six Forms of Process Improvement

- John L. Lee; *Rising Above It All*; Oct 2012; iUniverse
- Dean R. Spitzer; *Transforming Performance Measurement*; 2007 Amacom
- Douglas W. Hubbard; *How to Measure Anything*; 2010; Wiley & Associates
- John Doerr; *Measure What Matters*; 2008; Bennett Group, LLC
Chapter 5 – The Components of a Quality Program

- W. Edwards Deming; *Out of The Crisis*; Aug 2000; MIT Press

Chapter 6 – Common Quality Programs

Chapter 7 – The Stable Framework™

- Atul Gawande; *The Checklist Manifesto*; Jan 2011; Metropolitan Books
- Kevin J. Duggan; *Design for Operational Excellence: A Breakthrough Strategy for Business Growth*; Sep 2011; McGraw-Hill
- Masaaki Imai; *Kaizen: The Key to Japan’s Competitive Success*; 1986; Random House Business Division

Chapter 8 – The Future Domain

- Dr. Atul Gawande; *The Checklist Manifesto: How to Get Things Right*; Jan 2011; Picador

Chapter 9 – The Present Domain

Chapter 10 – The Past Domain

Chapter 11 – The Four Core Meetings in Stable

Chapter 12 – The Process Owner

- Mike Rother’s; *Toyota Kata: Managing People for Improvement, Adaptiveness, and Superior Results*; 2010; Rother & Company
- Eliyahu M. Goldratt and Jeff Cox; *The Goal: A Process of Ongoing Improvement*; Jun 2014; North River Press
- Alistair Cockburn, *Crystal Clear: A Human-Powered Methodology for Small Teams*; Oct 2004; Addison-Wesley Professional

Chapter 13 – The Master Chief

- Doerr, John; *Measure What Matters*; 2018; Bennett Group
- VitalSmarts™; Influencer: The Power To Change Anything; 2008; McGraw-Hill
- Gostick, Adrian; *The Carrot Principle: How the Best Managers Use
Recognition to Engage Their People, Retain Talent, and Accelerate Performance; 2009; Free Press

- Blake, R., Mouton, J.; The Managerial Grid: The Key to Leadership Excellence; 1964; Gulf Publishing Co
- Dr. Eddie O’Connor, Ph.D.; The Psychology of Performance: How to Be Your Best in Life; 2017; The Teaching Company
- James Pizzola, Managerial Leadership in the Workplace; 2023; Children’s Books for Adults
- Kim Scott; Radical Candor; 2019; St. Martin’s Press
- Jim Collins; Good to Great: Why Some Companies Make the Leap and Others Don’t; Oct 2001; Harper Business
- Jack and Suzy Welch; Winning; 2005; Harper Business
- Jean Tabaka; Collaboration Explained: Facilitation Skills for Software Project Leadership; 2006; Pearson Education Inc.
- Vince Lombardi, Jr.; What It Takes to Be Number One: Vince Lombardi On Leadership; Sep 2003; McGraw-Hill
- Apollo Syndrome – Information Sheet; Warwick Clinical System Improvement; https://warwick.ac.uk/fac/med/research/hsri/emergencycare/research/sdo/csi2/education/education-delivery/handouts/apollo_syndrome.pdf
- Marvin Karlins; Human Use of Human Resources; Mar 1981; McGraw-Hill
- Kenneth Blanchard Ph.D. and Thad Lacinak; Whale Done: The Power of Positive Relationships; Feb 2002; Free Press
- Robert K. Greenleaf; Servant Leadership: A journey into the nature of legitimate power and greatness; 1977; Paulist Press
- W. Edwards Deming; Out of The Crisis; Aug 2000; MIT Press
- Jack Welch and Suzy Welch; Winning: The Ultimate Business How-To Book; Oct 2009; HarperCollins Publishers
- Leadership Model contributors: Evelyn Cristina Acosta Maldonado, Vicente Pena, Jacob Anderson

Chapter 14 – Committees, Groups, and Kaizen Events

- Masaaki Imai; Gemba Kaizen: A Commonsense Approach to a Continuous Improvement Strategy; Jun 2012; McGraw-Hill

Chapter 15 – The Stable, or “Hybrid” Project

- PMI-ACP®, PMI®, PMP® are registered trademarks of the Project Management Institute

Chapter 16 – Using Stable in Operations
Chapter 17 – Using Stable in Implementation

Chapter 18 – Using Stable in DevOps


Chapter 19 – Using Stable in DevSecOps

- Samir Kumar Rakshit; Ethical Hacker’s Penetration Testing Guide: Vulnerability Assessment and Attack Simulation on Web, Mobile, Network Services and Wireless Entworks; June 2022; BPB Online Press

Chapter 20 – Using Stable in the Support Desk

Chapter 21 – Using Stable in Product Development

Chapter 22 – Using Stable with Agile Development

- PMI-ACP®, PMI®, PMP® are registered trademarks of the Project Management Institute

Chapter 23 – Stable Portfolio Management

- International Software Benchmarking Standards Group, [http://www.ISBSG.org](http://www.ISBSG.org)

Chapter 24 – Getting Started with Stable

Chapter 25 – 20 Data Collection & Analysis Tools

- Timothy J. Clark; *Success Through Quality: Support Guide for the Journey to Continuous Improvement*; Feb 1999; ASQ Press
- Lawrence H. Putnam and Ware Myers; *Measures for Excellence: Reliable Software on Time, Within Budget*; Oct
Index

.NET Reflection ........................................ 224

5

5 Why’s .................................................. 131
5S Housekeeping ........................... 102, 285

8

80/20 rule ................................................. 3

9

99.999% ................................................... 51

A

A3 Problem Solving ................................ 133
Acceptance Screen ............................ 50, 285
Acceptance Screens .......................... 50
access stewardship .......................... 223
Accessibility ........................................ 23, 24
accountability .................................... 77
Activity Network Diagram ................... 273
Ad Hoc Kaizen Teams ..................... 101
Adjourning ........................................ 172
Adolescent leaders ............................ 162
Affinity Diagram ............................... 274
Affinity Diagraming ......................... 199
After Action Review (AAR) ............ 135, 285
Agile ............................................... 10, 12
Agile Manifesto ................................ 10, 11
Agile mindset .................................... 10, 15
Agile Mindset .................................. 285, 290
Agile Product Development Using the
  Stable Framework™ ............................ 237
Agile project management ................ 195
Alert System .................................. 104, 259, 285

Alistair Cockburn ................................. 115
Ambiguous inputs ............................ 254
Anchoring Bias ................................. 129
anomaly ........................................ 36
Antithesis ........................................ 129
anxiety ........................................... 42
Anything Repeatable Can be Improved ... 77
Apollo Team .................................. 172, 285
Apollo Teams ................................. 172
architecture review ....................... 232
Assembly Line .................................... 1
Asset Control .................................. 83
Asset Controls ..................... 70, 85, 90, 285
Asset Maintenance Queue .................. 92, 93, 96
Asset Recovery Model .................. 37, 48, 91, 285
Asset Recovery Models .................. 48
Assurance ....................................... 32
Attunement ....................................... 32
Automation ...................................... 50
availability ...................................... 223
Availability% (AV%) ........................ 29
average page load speed ............ 266

B

B-17 Flying Fortress ......................... 13
Behr, Kevin ....................................... 298
Belbin, Dr. Meredith .......................... 285
Belligerent Employee ....................... 179
Berry, Michael ..................................... v
bias ...................................................... 41
blacklisting ...................................... 224
Blacklisting ........................................ 286
Blanc, Honoré ..................................... 1
Blanchard, Ken .................................... 177
Blanchard, Kenneth Ph.D .................. 297
Boeing .............................................. 13
Bohlen, Beal, and Rogers ................. 255
Boston Matrix ................................... 244
bottleneck ....................................... 45
Bottlenecks ................................ 254
Bowman’s Strategy Clock .................. 244
C

<table>
<thead>
<tr>
<th>Term</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buckingham, Marcus</td>
<td>182</td>
</tr>
<tr>
<td>Burnup Chart</td>
<td>203</td>
</tr>
<tr>
<td>Business Case</td>
<td>248</td>
</tr>
<tr>
<td>Business Value</td>
<td>240</td>
</tr>
<tr>
<td>Capability Chart</td>
<td>264</td>
</tr>
<tr>
<td>Capability Maturity Model integrated</td>
<td>9, 65</td>
</tr>
<tr>
<td>Capable</td>
<td>123</td>
</tr>
<tr>
<td>Carnegie-Mellon University</td>
<td>65</td>
</tr>
<tr>
<td>Cause and Effect Diagram</td>
<td>274</td>
</tr>
<tr>
<td>CCAPA</td>
<td>144</td>
</tr>
<tr>
<td>CCAPA Committee</td>
<td>57, 101, 190, 286</td>
</tr>
<tr>
<td>CCAPA event</td>
<td>100</td>
</tr>
<tr>
<td>CCAPA Queue</td>
<td>92, 93, 96, 184</td>
</tr>
<tr>
<td>Center Line (Mean)</td>
<td>265</td>
</tr>
<tr>
<td>certificate-based workflow</td>
<td>224</td>
</tr>
<tr>
<td>Challenger</td>
<td>9</td>
</tr>
<tr>
<td>Challenging Employee</td>
<td>178</td>
</tr>
<tr>
<td>Change Control</td>
<td>144</td>
</tr>
<tr>
<td>Change Control Committee ...</td>
<td>101, 190, 191</td>
</tr>
<tr>
<td>Chaos</td>
<td>22</td>
</tr>
<tr>
<td>Chaotic environments</td>
<td>21</td>
</tr>
<tr>
<td>Character</td>
<td>158</td>
</tr>
<tr>
<td>Chartering</td>
<td>198, 203</td>
</tr>
<tr>
<td>Checklist</td>
<td>272</td>
</tr>
<tr>
<td>checklists</td>
<td>29, 53</td>
</tr>
<tr>
<td>Checklists</td>
<td>76</td>
</tr>
<tr>
<td>Checkpoint</td>
<td>286</td>
</tr>
<tr>
<td>Cheksheet</td>
<td>272</td>
</tr>
<tr>
<td>Chernobyl</td>
<td>9</td>
</tr>
<tr>
<td>Chrysler</td>
<td>8</td>
</tr>
<tr>
<td>CI/CD Pipeline</td>
<td>217</td>
</tr>
<tr>
<td>CI/CD Pipelines</td>
<td>219</td>
</tr>
<tr>
<td>circus of reactions</td>
<td>58, 62</td>
</tr>
<tr>
<td>Clark, Timothy J</td>
<td>298</td>
</tr>
<tr>
<td>CMMI</td>
<td>9</td>
</tr>
<tr>
<td>coach, accountability</td>
<td>174</td>
</tr>
<tr>
<td>coach, subject-matter</td>
<td>174</td>
</tr>
<tr>
<td>Coaching Kata</td>
<td>126, 169, 286</td>
</tr>
<tr>
<td>coaching, accountability</td>
<td>159</td>
</tr>
<tr>
<td>coaching, subject-matter</td>
<td>159</td>
</tr>
<tr>
<td>COBIT</td>
<td>9</td>
</tr>
<tr>
<td>Cockburn, Alistair</td>
<td>296</td>
</tr>
<tr>
<td>code review</td>
<td>233</td>
</tr>
<tr>
<td>coffee</td>
<td>40</td>
</tr>
<tr>
<td>cognitive biases</td>
<td>129</td>
</tr>
<tr>
<td>Collins, Jim</td>
<td>162, 166, 182, 297</td>
</tr>
<tr>
<td>Colt, Samuel</td>
<td>1</td>
</tr>
<tr>
<td>committed</td>
<td>157</td>
</tr>
<tr>
<td>Common Causes</td>
<td>41</td>
</tr>
<tr>
<td>communication</td>
<td>159</td>
</tr>
<tr>
<td>Competence</td>
<td>158</td>
</tr>
<tr>
<td>Concept Review</td>
<td>232</td>
</tr>
<tr>
<td>confidentiality</td>
<td>223</td>
</tr>
<tr>
<td>Configuration Management Database</td>
<td>37</td>
</tr>
<tr>
<td>Configuration Management System</td>
<td>37, 48, 56, 85, 90, 126, 257, 286, 291</td>
</tr>
<tr>
<td>Configuration Management System (CMS)</td>
<td>69</td>
</tr>
<tr>
<td>Confirmation Bias</td>
<td>129</td>
</tr>
<tr>
<td>Constancy of Purpose</td>
<td>157</td>
</tr>
<tr>
<td>Context Diagram</td>
<td>276</td>
</tr>
<tr>
<td>continual improvement</td>
<td>56, 79</td>
</tr>
<tr>
<td>Contiual improvement</td>
<td>79</td>
</tr>
<tr>
<td>Continual Improvement</td>
<td>204, 205</td>
</tr>
<tr>
<td>continual improvement, sustained</td>
<td>56</td>
</tr>
<tr>
<td>Continual Improvement, Sustained</td>
<td>56</td>
</tr>
<tr>
<td>Continual Self-Improvement</td>
<td>162</td>
</tr>
<tr>
<td>Continual-flow</td>
<td>44</td>
</tr>
<tr>
<td>Continuous Integration, Continuous</td>
<td>217</td>
</tr>
<tr>
<td>Delivery</td>
<td>217</td>
</tr>
<tr>
<td>Control Chart</td>
<td>265</td>
</tr>
<tr>
<td>Control Points</td>
<td>81, 86, 286</td>
</tr>
<tr>
<td>Correction</td>
<td>93, 191</td>
</tr>
<tr>
<td>Corrective Action</td>
<td>93, 191</td>
</tr>
<tr>
<td>Cost Savings of Quality</td>
<td>58</td>
</tr>
<tr>
<td>costs</td>
<td>42</td>
</tr>
<tr>
<td>Cotton Gin</td>
<td>1</td>
</tr>
<tr>
<td>Critical Success Factor (CSF’s)</td>
<td>286</td>
</tr>
<tr>
<td>Critical Decision Log</td>
<td>114</td>
</tr>
<tr>
<td>Critical Fix</td>
<td>110</td>
</tr>
<tr>
<td>critical thinning model</td>
<td>129</td>
</tr>
<tr>
<td>critical thinking</td>
<td>170</td>
</tr>
<tr>
<td>Critical thinking</td>
<td>128</td>
</tr>
<tr>
<td>Crosby, Philip</td>
<td>6</td>
</tr>
<tr>
<td>cross-site scripting</td>
<td>224</td>
</tr>
</tbody>
</table>
INDEX

cross-trained ............................................ 98
Crux Design ............................................ 286
Culture ................................................. 164, 165, 286
Cumulative System Performance Chart 112, 244, 259, 287
Cumulative System Performance Chart Repository ......................................... 114
customer ................................................................ 124
Customer expectations ........................................ 42
customer feedback ............................................. 120
Customer List ........................................... 89
Customer Request Queue .................................. 92, 96
customer satisfaction ......................................... 32
Customer satisfaction ........................................... 42
customer satisfaction questionnaire ................. 32
customer satisfaction rating ................................ 110
customer satisfaction ratings .............................. 108
Customer Satisfaction scores ........................................... 138
Customer Satisfaction Surveys ............................. 258
Customer Value ............................................. 240, 287
customers ................................................................ 31
Cycle Planning Meeting ................................... 117
Cycle Retrospective ..................................... 117, 144
Cycle Review Meeting ................................... 93, 117
cycle times .................................................. 42
Cycle totals .................................................. 110
cycle velocity .................................................. 203
Cycle-time Metrics ........................................... 37
delays .................................................................. 14, 24
delegate .......................................................... 175
Deming .......................................................... 1, 3, 4, 5, 6, 7, 8, 9, 23, 24, 25, 32, 39, 40, 51, 53, 59, 67, 79, 81, 137, 157, 162, 176
Deming, W. Edwards ........................................ 3, 24, 37, 49, 58, 79, 137, 254, 290, 296, 297
Dependability .................................................. 183
Desert Storm ................................................... 60
detractors ......................................................... 138
Development ................................................. 223
DevOps .......................................................... 217
DevOps Using the Stable Framework .................... 219
DevSecOps .................................................... 223, 224, 226
Diffusion Process ........................................... 255
digital password best practices ......................... 224
DIRFT concept ................................................... 6
Disaster Recovery ............................................ 207
disaster recovery planning .................................. 91
Disciplined Agile ............................................ 235, 287
Disorder, Chaotic ............................................ 195
Disorder, Disorganized ...................................... 195
documentation .................................................. 36
Doerr, John ..................................................... 150
dorking, Google .............................................. 224
downstream customer ......................................... 29, 32
Downstream Customer ........................................... 32
Drucker, Peter .................................................... 149, 163
Duggan, Kevin J. ............................................. 296

d.I.R.F.T
Do It Right the First Time .................................... 287
Daily Kaizen ..................................................... 186
Daily Kaizen Standup ...................................... 97
Daily Kaizen Standup Meeting ........................................... 117
data ..................................................................... 223
Data Analysis Tools ........................................ 273
Data Collection Tools ........................................ 263
de Gribeauval, Jean-Baptiste Vaquette .......... 1
Debois, Patrick ................................................. 298
Defects ............................................................ 48
defensive institutional knowledge ......................... 203
Define Scope ................................................... 199
edgment .......................................................... 253

e.E
Early Adopters .................................................. 255
Early Majority .................................................. 255
effective leaders ................................................. 162
effectiveness ..................................................... 26
efficiency ........................................................ 26
Elimination ....................................................... 49
Eliyahu Goldratt ................................................... 44
Elton, Chester ..................................................... 151
emotionally bond .............................................. 138
Empathy .......................................................... 32
encryption types/purposes .................................. 224
end to the beginning ........................................... 253

303
THE STABLE FRAMEWORK™

Epson ......................................................... 9
executive support ............................................. 253
Expected payback period .............................. 248
Expected ROI ................................................. 248
Extreme Aversion ............................................. 129
facilitated workshop ........................................... 135
Facilitation ................................................ 161
facilitation skills ................................................. 160
Facilitative Workshop ...................................... 287
Failure Modes and Effects Analysis (FMEA) ................. 135, 287
failures ......................................................... 42
FDA 21 CFR Part 11 ........................................ 287
FDA 21 CFR Part 11 Compliance .............................. 223
final project report ................................................. 203
Final Project Report ............................................. 204
First Break All the Rules ................................... 182
First Impression Bias ......................................... 129
Fishbone Diagram .............................................. 132
Five 9 ................................................................. 51
Five dimensions of customer satisfaction ......................... 32
Five Stable Principles ........................................... 77
Flow ................................................................... 35, 97
Flowchart ...................................................... 277
flowcharts ......................................................... 37
focus group ......................................................... 130
Focus Group ..................................................... 266
Focus Group Meeting ....................................... 287
Focus Groups .................................................... 133
Ford ................................................................. 8
Ford Motor Company .......................................... 7
Ford, Henry ........................................................... 2, 4
Formidable When Necessary .................................. 160
Forming .............................................................. 171
fourteen points of leadership, Deming ..................... 162
Framing Bias ...................................................... 130
Frequency of failure .............................................. 42
Fun ..................................................................... 181
Future Domain ................................................... 85, 183
fuzzing, website & webservice ................................. 224
Gantt Chart ...................................................... 203
Gantt, Henry Laurence ....................................... 196
Gawande, Atul ................................................. 89, 296
Gawande, Dr. Atul .............................................. 13, 296
Gemba ................................................................. 38
Gemba Walk ...................................................... 193
General Electric ................................................. 7, 161
General Motors ................................................... 8
Go and See ......................................................... 38
Goal Statement .................................................. 168
Goal, The ............................................................... 97
goals, long-term .................................................. 242
goals, Short term .................................................. 242
Goldratt, Dr. Eli ..................................................... 255
Goldratt, Eli ........................................................... 6
Good to Great .................................................... 162, 182
Google ............................................................ 11, 183
Gostick, Adrian .................................................. 151
Green Bay Packers ............................................. 169
Greenleaf, Robert K. ............................................. 297
Group Conflict Resolution .................................... 170
Group Problem Solving ....................................... 170
Groupthink Bias .................................................. 129
Grove, Andy .......................................................... 149
Harry, Dr. Mikel J. ................................................. 7
hashed ................................................................. 225
Hastings Chart ................................................. 112
Heavy Process ..................................................... 58
hidden factory ................................................. 14, 15, 35, 46
Hidden Factory .. 18, 20, 53, 55, 57, 82, 112, 143, 287
Hill, Major Ployer P ............................................. 13
HIPAA Compliance .............................................. 223
Histogram ............................................................. 267
Histograms ......................................................... 258
historical logs .................................................... 53
Historical logs ..................................................... 114
Historical Logs .................................................... 258
Honda ................................................................. 9
Historical Logs .................................................... 258
Honda ................................................................. 9

304
INDEX

Hot-fix Haze ........................................... 287
How to be Number One ......................... 169
Hubble Telescope ................................. 131
Humble, Jez ............................................ 298
Hybrid Project ........................................ 195
hypothesis ................................................ 40
Hyundai.................................................... 63

IDOR/BOLA/RCE authorization spoofing 224
Imai, Masaaki .................. 79, 296, 297
Imai, Masaaki ........................................... 38
Impact....................................................... 183
imperical learning ...................... 224
Implementation ..................................... 213
Implementation Using the Stable Framework ........................................ 213
improvement . 78, 79
Incident Register .. 57, 71, 73, 109, 114, 143, 188, 258, 288
Incident/Incident Avoidance Register ...... 57
Incoming Quality Assessment ..... 31, 53, 72
Incoming Quality Assessment (IQA) 52, 288
Influencer: The Power to Change Anything .......................................................... 151
Information Radiator ............................. 115
Information Technology Infrastructure Library (ITIL) ........................................ 64
Initial setup ............................................ 257
Innovators........................................... 255
insight, ..................................................... 77
inspection ................................................ 54
Inspection ................................................. 2
institutional knowledge .............. 53, 69
Institutional Knowledge ........... 55, 288
Intel .......................................................... 149
Intelligence ............................................... 169
International Software Benchmarking Standards Group ........................................ 250
Interrelationship Diagram ........... 42, 280
Inventory ................................................. 47
investment ......................................... 35
Irresponsible Employee ............. 178

ISBSG.org..................................................... 250
Ishikawa Diagram............................ 121, 274
Ishikawa, Kaoru ........................................... 5
ISO 9000 ..................................................... 63
Issue ......................................................... 93
ITIL............................................................. 9

J

Jack Welch .............................................. 297
Jefferson, Thomas .............................. 1
JetBlue ..................................................... 151
Jobs, Steve .............................................. 180
Juran Trilogy ............................................. 5
Juran, Joseph ............................................. 3
JUSE......................................................... 5
Just-In-Time ............................................. 5
Just-In-Time Manufacturing ................... 288

K

Kaizen ........................................... 36, 38, 76, 79
Kaizen Blitz .......................................... 74, 80
Kaizen Circles ........................................ 5
Kaizen Initiatives ..................................... 144
Kaizen Kata ........................................... 121, 288
Kaizen Meeting ...................................... 76
Kaizen of Kaizens ................................ 192
Kaizen project ...................................... 204
Kaizen Project ........................................ 288
Kaizen Projects ..................................... 74, 135
Kaizen Stand-up .................................. 103
Kaizen Standup meeting ..................... 202
Kaizen Stand-up Meeting ................. 76
Kaizen Team ....................................... 101, 288
Kaizen Teams ........................................ 73
Kanban board .................................... 185
Kanban Board ....................................... 211, 258
Kata ......................................................... 99
Kata Card .......... 41, 83, 89, 127, 224, 227, 252, 272
Kata Card Repository .......................... 114
Kata Card Template ....................... 83, 100
Kata Card, Market Update ............. 251
Kata Card, Portfolio Management ........ 251
Kata Card, Product or Service ........... 252
Kata Cards ........................................ 27, 53, 58, 73, 76, 99
Kata Historical Repository................ 100
Key Performance Indicators .......... 86, 240
Key Performance Indicators (KPI’s) .... 288
Kim, Gene ........................................ 217, 298
Kohn, Alfie ........................................... 6
KPI ....................................................... 86
KPI’s .................................................... 240
Krafcik, John ..................................... 7
Krafcik, John F. ................................... 63

L

Laggards ........................................... 255
Larry Smith ......................................... 11
Late Majority ..................................... 255
laughter ............................................. 181
leadership, Effective ....................... 148
leadership, Ineffective .................... 148
lean ..................................................... 35
Lean ..................................................... 7, 36, 46, 59, 63
learning organization ..................... 54, 69
Lee, John ............................................. 51
Lee, John L. .......................................... vi
Lessons Learned Log ....................... 288
Lewin, Kurt ......................................... 153
life skills ............................................. 123
Lombardi Jr., Vince ......................... 169, 297
Lombardi, Vince ................................. 169
Lower Control Limit (LCL) ............... 265

M

Magnavox ............................................. 7
Management by Objective ............... 149
market summary report .................. 118
Market Summary Report ................... 289
Marlins, Marvin ................................. 297
marriage ............................................. 170
Master Chief ................................. 145, 160, 173, 224
Master Cycle ......................... 98, 150, 201, 209, 259, 289
Master Cycle Meetings .................... 259
Master Cycle Planning ..................... 203
Master Cycle Planning Meeting .......... 117, 186, 209
Master Cycle Planning Meetings ........ 91
Master Cycle Retrospective ............... 204
Master Cycle Retrospective Meeting .... 120, 144
Master Cycle Review ....................... 202, 204
Master Cycle Review Meeting .......... 120, 143
Master Cycles .................................... 112
Master Kaizen ................................... 289
Master Kaizen Meeting .................... 192
Master Lists ....................................... 289
Master Project Schedule .................. 202, 245
Master Schedule ............................... 85, 91, 258
Master Services List ......................... 86, 90, 126
Matrix Diagram .................................. 281
Maxwell, John C. ............................... 164
Maxwell, John C. ............................... 148
Meaning ............................................. 183
Measure What Matters .................... 150
Measurement ..................................... 41
Measurements ................................... 181
measurements, two types .................. 141
Measures ........................................... 41
meetings .......................................... 117
memory encryption ......................... 224
metrics .............................................. 82
Mission Statement ......................... 167, 242
Mistake Proofing ............................. 36, 50
mistakes ............................................ 37
MIT ..................................................... 63
Models of Excellent Teams .............. 182
Moore, Christian .............................. 180, 297
motion .............................................. 47
Motorola ............................................ 7, 64
Muda ............................................... 46
Myers, Ware ...................................... 298

N

Naivity Bias ....................................... 129
Napoleon ............................................ 1
NBC .................................................... 7
INDEX

NEC ........................................................ 7, 9
Needy Employee .................................... 177
Net Promoter Score .............................. 137
Net Promoter Score (NPS) .... 108, 138, 289
No Contest .............................................. 6
Noel Burch ............................................ 20
Nominal Group Technique (NGT)........... 130
Nonaka, Ikujiro ...................................... 9
Nonconforming issue ......................... 102
Nonconforming Issues ......................... 289
Non-conforming Output ....................... 41
Non-Conforming Product Management .... 56
non-rolled throughput yield ................... 29
Norming ............................................... 171
North Star ............................................ 150, 167

O

O’Connor, Dr. Eddie ............................... 153
Objectives ............................................ 98
Objectives and Key Results ................. 149
Objectives and Key Results (OKR’s)... 86, 240
Objectives and Key Results (OKR’s) 107, 289
observation and curiosity .................. 175
Ohno, Taiichi ....................................... 4, 63
OKR framework .................................... 150
OKR’s .................................................. 149
One Piece Flow ................................. 254
one-on-one’s ..................................... 180
One-piece flow ............................... 44, 211
One-piece Flow .................................. 289
Operational Excellence 9, 12, 15, 18, 26, 70,
84, 116, 290
Operations ........................................... 207, 223
Operations Using the Stable Framework ................................. 208
Opportunity ....................................... 240
Opportunity Profile Radar Chart .......... 249
Organizational transformations .......... 253
Out of the Crisis .............................. 8, 67
out-of-control ................................. 265
over processing ................................. 47
Overburden ...................................... 254
Overconfidence Fallacy ...................... 129
Overproduction ................................. 47
OWASP ............................................ 224
OWASP’s Top 10 Mobile Risks ............ 224
OWASP’s Top 10 Security Risks .......... 224
Oxford English Dictionary .................... 76
PAL, Process Asset Library .................. 69, 85
Panasonic ........................................... 7
Pareto Chart ....................................... 268
Pareto Principle .................................. 3
passives .......................................... 138
password hashing & salting storage ...... 224
password weak/strong principles ...... 224
Past Domain ..................................... 107, 188
Path to Stability .................................. 253
PBP .................................................... 51
PCI Compliance ................................ 223
PDCA ........................................... 37, 38, 40, 41
Pending Services ................................ 87
penetration testing ............................ 224
Performance Console .... 115, 184, 187, 210,
258, 290
performance management ........... 69, 149
Performing ........................................ 171
Personal Coefficient ......................... 2, 290
Personal Leadership ......................... 164
Personas ......................................... 290
Personify, Blame, then Punish ............ 51
persuasive ....................................... 159
Persuasive ........................................ 159
Phase-gate ......................................... 27
Phoenix Project, The .......................... 97
Plan-based project management ....... 195
Plan-Do-Check-Act ... 3, 37, 39, 78, 290, 293
Plan-Do-Check-Act (PDCA) .............. 37
Planning Fallacy ............................... 129
Planning Package ......................... 290
Platform as a Service (PaaS) .......... 217
Poka-Yoke ...................................... 5, 290
Porter’s Three Strategies ................... 244
Portfolio Management ................. 66, 242, 252
positive psychology .................... 176
Product Performance Inventory ................................................. 244
Product Quality .................................................................. 54, 291
Production Change Log ......................................................... 114
Profound Knowledge ......................................................... 114
Project Management Institute® ............................................. 196
Project Manager .................................................................. 198
Project Managers ................................................................. 203
Project Prioritization Matrix .................................................... 249
Project Schedule & Gantt Chart ................................................. 197
Promoters ............................................................................ 138
Propinquity ................................................................. 32
Protector When Necessary ....................................................... 161
Psychological Safety ............................................................... 183
Psychology ......................................................................... 40, 159
Pull system ........................................................................... 96
Pull System ............................................................................. 291
Push system .......................................................................... 95
Push System .......................................................................... 291
Putnam, Lawrence H. ............................................................ 298
Q

quality .................................................................................. 51
Quality .................................................................................. 8
Quality Assurance ................................................................. 53, 237
Quality Assurance (QA) ......................................................... 52
quality charts ......................................................................... 263
Quality Circles ....................................................................... 5
Quality Competency ............................................................... 20
Quality Control ...................................................................... 54, 230
Quality Control (QC) ............................................................. 52
Quality Control Handbook ........................................................ 3
Quality Debt .......................................................................... 19, 20, 291
Quality framework ................................................................. 20
quality kills ............................................................................ 4
Quality Management ............................................................. 58
Quality Planning ..................................................................... 37, 52, 86, 89, 102, 257, 291, 293
Quality system ....................................................................... 81
Quarterly Business Review ....................................................... 242

R
radar chart ............................................................................. 249
INDEX

Radar Chart .................................................... 268
Radical Candor .............................................. 176, 292
Rakshit, Samir Kumar ........................................ 298
RCA .............................................................. 7
Reactance Bias .................................................. 129
reality .......................................................... 41, 130
receiver processes ........................................ 35
Red-Amber-Green ............................................. 202
redirection ..................................................... 177
Rejector of Distractions .................................... 158
Rejector of Patterns of Mistrust ......................... 159
Release stability .............................................. 82
Reliability ...................................................... 32
Remediation .................................................... 48
Repetition ..................................................... 78
Requirements Review ....................................... 232
Requirements Rot ............................................ 292
Resilience ....................................................... 48
Respect People ................................................. 79
Responsiveness .............................................. 32
Retired Services .............................................. 88
retraining ..................................................... 41
Retraining ..................................................... 102
rework .......................................................... 2, 14, 20, 24, 31, 58, 254
Rework Metrics .............................................. 37
Rising Above it All ........................................... 51
risk management ............................................. 20
Risk Management ............................................ 203
Risk Register .................................................... 203
risks ............................................................ 224
Risks ............................................................ 203
Robert K. Greenleaf .......................................... 160
robustness ...................................................... 24
Robustness ..................................................... 23
ROI evaluation ............................................... 244
Roll Through% (RT%) ........................................ 29
root cause ...................................................... 36
root cause analysis ......................................... 41
Root Cause Analysis ........................................ 130
Root Cause Diagram ........................................ 274
root causes .................................................... 35
Rother, Mike ................................................. 296
Rugby ............................................................ 10
Rule of Seven .................................................. 265

S

S.W.O.T. Analysis ........................................... 248
safety, team ................................................... 179
salted ......................................................... 225
Sandcone ..................................................... 246
Sandcone Model ............................................. 66
Scatter Diagram .............................................. 269
Scheduled Activity Queue .................................. 92, 96
Schwaber, Ken ................................................ 10
Scientific Management .................................... 2
Scoreboards ................................................... 104
Scott, Kim ..................................................... 176, 297
scrap ........................................................... 14, 24, 58
Scrap ........................................................... 254
Scrap Metrics .................................................. 37
Scrum ........................................................... 10
SDCA ........................................................... 37, 38
security ........................................................ 223
Sensitivity Analysis .......................................... 270
Servant .......................................................... 160
Servant Champion ........................................... 155
Servant Champions ........................................... 70
Servant Leadership ......................................... 160, 173
Service Catalog .............................................. 64
Service Desk ................................................... 56, 64, 93, 94, 118
Service Level .................................................. 43
Service Level Agreement .................................. 37, 43, 82, 292
Service Level Agreement (SLA) ......................... 210
Service Level Attainment Monitor (SLAM) ............. 210
Service Level Attainment Monitoring .................... 292
Service Level Goal .......................................... 37, 43, 292
Service Level Goal (SLG) .................................. 210
Service Level Histogram ................................... 292
service levels .................................................. 82
Service Levels ............................................... 43
Service Oriented Architecture ......................... 292
Service Performance Inenotry ............................ 243
Service Register ............................................. 69, 85, 86, 90, 257, 292
Serviced Level Goal ......................................... 82
SERVQUAL .................................................... 32
Set in Order .................................................... 104
Shewhart ....................................................... 2, 3, 7, 39, 78
<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shewhart, Walter</td>
<td>37, 265, 290</td>
</tr>
<tr>
<td>Shewhart, Walter</td>
<td>2, 3</td>
</tr>
<tr>
<td>Shift-Left</td>
<td>223</td>
</tr>
<tr>
<td>Shift-left Thinking</td>
<td>11</td>
</tr>
<tr>
<td><strong>Shift-Left Thinking</strong></td>
<td>292</td>
</tr>
<tr>
<td>Shine</td>
<td>104</td>
</tr>
<tr>
<td>Shingo Research and Professional Publication Award</td>
<td>51</td>
</tr>
<tr>
<td>Shingo, Shigeo</td>
<td>5</td>
</tr>
<tr>
<td>Simplification</td>
<td>49</td>
</tr>
<tr>
<td>Single-piece flow</td>
<td>44</td>
</tr>
<tr>
<td><strong>Single-Piece Flow</strong></td>
<td>292</td>
</tr>
<tr>
<td>SIPOC Model</td>
<td>26</td>
</tr>
<tr>
<td>Site &amp; Process Reliability Engineering (SPRE)</td>
<td>11</td>
</tr>
<tr>
<td>Site Reliability Engineering (SRE)</td>
<td>11</td>
</tr>
<tr>
<td>Six Sigma</td>
<td>7</td>
</tr>
<tr>
<td>Six-Sigma</td>
<td>64, 65</td>
</tr>
<tr>
<td>Skills</td>
<td>48</td>
</tr>
<tr>
<td>SLAM Chart</td>
<td>64, 88</td>
</tr>
<tr>
<td>SLAM Chart Repository</td>
<td>114</td>
</tr>
<tr>
<td>SLAM Charts</td>
<td>76, 108, 119, 120, 143, 259, 271</td>
</tr>
<tr>
<td>Smith Jr., William B.</td>
<td>7</td>
</tr>
<tr>
<td>Smoke Test</td>
<td>229</td>
</tr>
<tr>
<td><strong>SOC 2</strong></td>
<td>292</td>
</tr>
<tr>
<td>SOC 2 Compliance</td>
<td>223</td>
</tr>
<tr>
<td>Software as a Manufacturing Process</td>
<td>11</td>
</tr>
<tr>
<td>Software as a Service (SaaS)</td>
<td>217</td>
</tr>
<tr>
<td>Software Engineering Institute (SEI)</td>
<td>65</td>
</tr>
<tr>
<td>Software Implementation Process Map</td>
<td>27</td>
</tr>
<tr>
<td>Sony</td>
<td>7, 59</td>
</tr>
<tr>
<td>SOP</td>
<td>136</td>
</tr>
<tr>
<td>SOPK</td>
<td>40</td>
</tr>
<tr>
<td>Sort</td>
<td>103</td>
</tr>
<tr>
<td>Space Shuttle Challenger</td>
<td>8</td>
</tr>
<tr>
<td>Spafford, George</td>
<td>298</td>
</tr>
<tr>
<td>SPC Chart</td>
<td>42</td>
</tr>
<tr>
<td>Speak with Data</td>
<td>80</td>
</tr>
<tr>
<td>Special Causes</td>
<td>41</td>
</tr>
<tr>
<td>spoofing, local PC clock</td>
<td>225</td>
</tr>
<tr>
<td><strong>SPRE</strong></td>
<td>11</td>
</tr>
<tr>
<td><strong>SPRE, Site &amp; Process Reliability Engineering</strong></td>
<td>293</td>
</tr>
<tr>
<td>Sprint</td>
<td>150</td>
</tr>
<tr>
<td>Sprint board</td>
<td>185</td>
</tr>
<tr>
<td>SQL injection</td>
<td>224, 225</td>
</tr>
<tr>
<td><strong>SRE</strong></td>
<td>11</td>
</tr>
<tr>
<td><strong>SRE, SiteReliability Engineering</strong></td>
<td>292</td>
</tr>
<tr>
<td>SSL/TLS/HTTPS</td>
<td>224</td>
</tr>
<tr>
<td>Stable</td>
<td>81</td>
</tr>
<tr>
<td>Stable Agile</td>
<td>12</td>
</tr>
<tr>
<td>Stable Framework</td>
<td>82</td>
</tr>
<tr>
<td>Stable Framework™</td>
<td>14, 20, 22, 67</td>
</tr>
<tr>
<td>Stable Framework™ Leadership Model</td>
<td>155</td>
</tr>
<tr>
<td>Stable mindset</td>
<td>15</td>
</tr>
<tr>
<td>Stable Mindset</td>
<td>290, 293</td>
</tr>
<tr>
<td>Stable Proposition</td>
<td>20</td>
</tr>
<tr>
<td>Stages of Competency</td>
<td>21</td>
</tr>
<tr>
<td><strong>Stages of Competency model</strong></td>
<td>20</td>
</tr>
<tr>
<td>Standard of Care</td>
<td>100, 293</td>
</tr>
<tr>
<td>Standard Operating Procedure</td>
<td>100, 198</td>
</tr>
<tr>
<td>Standard Operating Procedure (SOP)</td>
<td>37, 41, 83, 89</td>
</tr>
<tr>
<td>Standard Operating Procedures</td>
<td>27, 58, 70, 76, 102, 224</td>
</tr>
<tr>
<td>standard unit of measure</td>
<td>29</td>
</tr>
<tr>
<td><strong>Standard Work</strong></td>
<td>293</td>
</tr>
<tr>
<td>Standard Work Checklist</td>
<td>37</td>
</tr>
<tr>
<td>Standard Work Checklists</td>
<td>27</td>
</tr>
<tr>
<td>Standard Workflow Step</td>
<td>83</td>
</tr>
<tr>
<td>Standardize</td>
<td>104</td>
</tr>
<tr>
<td><strong>Standardize-Do-Check-Act</strong></td>
<td>293</td>
</tr>
<tr>
<td>Standardize-Do-Check-Act (SDCA)</td>
<td>37</td>
</tr>
<tr>
<td>Standish Group International</td>
<td>17</td>
</tr>
<tr>
<td>State of the Market report</td>
<td>209</td>
</tr>
<tr>
<td>Statistical Process Control</td>
<td>7, 64</td>
</tr>
<tr>
<td>Statistical Process Control Chart</td>
<td>2</td>
</tr>
<tr>
<td>Statistical Sampling</td>
<td>272</td>
</tr>
<tr>
<td><strong>Status Quo Bias</strong></td>
<td>129</td>
</tr>
<tr>
<td>Step Time (ST)</td>
<td>29</td>
</tr>
<tr>
<td>Storming</td>
<td>171</td>
</tr>
<tr>
<td>story points</td>
<td>99</td>
</tr>
<tr>
<td><strong>Story Point</strong></td>
<td>293</td>
</tr>
<tr>
<td>Storyboarding</td>
<td>200</td>
</tr>
<tr>
<td>Strategy Review</td>
<td>242</td>
</tr>
<tr>
<td>Suggestion Box</td>
<td>105, 260, 272</td>
</tr>
<tr>
<td>Suggestions</td>
<td>105</td>
</tr>
</tbody>
</table>
Sunk-cost Bias ........................................ 129
supplier processes .................................. 35
Supplier Services List.............................. 88
Supply Chain ......................................... 31, 293
support desk ......................................... 227
support, External ..................................... 227
support, Internal ..................................... 227
surveys .................................................. 110, 123
Sustain .................................................. 104
sustained continual improvement ............ 56, 79
Sustaining the Stable Framework ............. 260
Sutherland, Jeff ...................................... 10
Swimlanes ............................................. 27
Synthesis ............................................... 129
system ................................................... 24, 253
System Backlog ................................. 85, 92, 96, 258
System Interactions .................................. 25
System Learning ...................................... 41
System Learning & Improvement ............. 39
System Lists .......................................... 86
System of Profound Knowledge ............. 39, 40
Systematization ..................................... 27, 36
systematize ........................................... 35
Systematize Value Streams .................. 259
systematized ......................................... 36, 56
systematizing ....................................... 82
System-Flow Diagrams ......................... 27
Systemitization ...................................... 81
systematization ..................................... 54
systems .................................................. 24
Systems Thinking .................................. 3, 4, 23, 24
Systems-Thinking ................................. 223
team branding ....................................... 167
Team Development Model ...................... 170, 293
Team Empowerment Model .................. 173
Team Rivalries, Repairing ...................... 172
Technical Debt ...................................... 19
Technology Adoption Lifecycle ............. 255
teeth and backbone traits ..................... 156
The “Agile” Mindset ............................... 9
The “Stable Proposition” ....................... 20
The “Stable” Mindset ....................... 10
The Carrot Principle ............................. 151
The Checklist Manifesto ....................... 13, 89
The DevOps Handbook ......................... 217
The Goal ............................................ 6, 44, 255
The Hidden Factory ............................. 14
The Human Use of Human Resources .......... 172
The Phoenix Project ............................. 217
the Stable Framework™ ...................... 224
The Stable Framework™ ...................... 11
Theory of Constraints ......................... 6
Theory-of-Constraints .......................... 44, 255
Thesis ............................................... 129
This is continual improvement .............. 127
Three Ways ......................................... 217
Time/Effort Value Realization Curves .... 256
Toshiba ............................................... 59
Total Quality Management .................... 3
Tower, Les ............................................ 13
Toyota .................................................. 9, 59, 65
Toyota Kata .......................................... 38
Toyota Production System ............... 4, 5, 7, 10
Toyota Production System (TPS) ............ 63
Toyota, Eiji .......................................... 4
TPS ..................................................... 5, 10
TQM .................................................... 3
Training ............................................. 56, 102
Training Log ......................................... 114, 224, 259
Transport ............................................. 46
traveler ............................................. 89
Tree Diagram ......................................... 283
tribal knowledge .................................. 56, 69
Tribal Knowledge ............................... 56, 294
trigger ............................................... 24, 27
Trigger ............................................... 294
<table>
<thead>
<tr>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>unvalidated URL redirects</td>
</tr>
<tr>
<td>upper control limit (UCL)</td>
</tr>
<tr>
<td>upstream supplier</td>
</tr>
<tr>
<td>upstream supplier</td>
</tr>
<tr>
<td>use case</td>
</tr>
<tr>
<td>user stories</td>
</tr>
<tr>
<td>utility value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>validation</td>
</tr>
<tr>
<td>value</td>
</tr>
<tr>
<td>value add (VA)</td>
</tr>
<tr>
<td>value proposition</td>
</tr>
<tr>
<td>value proposition</td>
</tr>
<tr>
<td>value propositions</td>
</tr>
<tr>
<td>value propositions</td>
</tr>
<tr>
<td>value realization</td>
</tr>
<tr>
<td>value statement</td>
</tr>
<tr>
<td>value stream</td>
</tr>
<tr>
<td>value streams</td>
</tr>
<tr>
<td>values</td>
</tr>
<tr>
<td>value-stream maps</td>
</tr>
<tr>
<td>variation</td>
</tr>
<tr>
<td>vectors of system excellence, two</td>
</tr>
<tr>
<td>velocity</td>
</tr>
<tr>
<td>verification</td>
</tr>
<tr>
<td>vertical communication</td>
</tr>
<tr>
<td>victim triangle</td>
</tr>
<tr>
<td>victories</td>
</tr>
<tr>
<td>virtual teams</td>
</tr>
<tr>
<td>vision statement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>wait time (WT)</td>
</tr>
<tr>
<td>waiting</td>
</tr>
<tr>
<td>warranty value</td>
</tr>
<tr>
<td>Washington, George</td>
</tr>
<tr>
<td>Waste</td>
</tr>
<tr>
<td>waste</td>
</tr>
<tr>
<td>WBI-Zogby</td>
</tr>
<tr>
<td>web encryption</td>
</tr>
<tr>
<td>Welch, Jack</td>
</tr>
<tr>
<td>Welch, Jack and Suzy</td>
</tr>
<tr>
<td>Wells, John</td>
</tr>
<tr>
<td>Whale Done</td>
</tr>
<tr>
<td>Whitelisting</td>
</tr>
<tr>
<td>Whitney, Eli</td>
</tr>
<tr>
<td>WIKI</td>
</tr>
<tr>
<td>Winning</td>
</tr>
<tr>
<td>WIP</td>
</tr>
<tr>
<td>WIP-limit</td>
</tr>
<tr>
<td>work breakdown structure</td>
</tr>
<tr>
<td>Work in Progress (WIP)</td>
</tr>
<tr>
<td>Work Package</td>
</tr>
<tr>
<td>work package</td>
</tr>
<tr>
<td>workload analysis</td>
</tr>
<tr>
<td>Workload Analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yamada, Toshiro</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zen</td>
</tr>
<tr>
<td>Zenith</td>
</tr>
</tbody>
</table>
CERTIFICATIONS AND A BELT SYSTEM ARE AVAILABLE TO ENSURE THE INTEGRITY OF THE FRAMEWORK AND THE EFFECTIVENESS OF RESULTS WHEN PRACTICED.

FOR MORE INFORMATION:

WWW.STABLEFRAMEWORK.ORG

WWW.STABLEINSTITUTE.ORG
Michael J. Berry has worked in the software industry since 1987. During this time, he developed enterprise software for banks used in over 2500 branch sites in six states, and led teams that developed and supported manufacturing software used on five continents, and medical software used in over 600 clinics. He has written Lean manufacturing software for companies that have product components on Mars. He has consulted with and trained thousands of technology professionals during the past two decades.

Michael is a past adjunct professor at the University of Utah, Utah Valley University, and Neumont College of Computer Science. He is an active member of the Project Management Institute, an IEEE Senior Member, and a John C. Maxwell leadership coach. In 2007 he founded Red Rock Research, a professional software development training organization. Michael holds a B.S. in Information Systems and Technologies from Weber State University.

Michael can be reached at Michael@StableFramework.org