AGILE
Almanac

BOOK 2: PROGRAMS WITH MULTI-
AND VIRTUAL-TEAM ENVIRONMENTS
AGILE Almanac

BOOK 2: PROGRAMS WITH MULTI-AND VIRTUAL-TEAM ENVIRONMENTS

JOHN G. STENBECK, LEAD AUTHOR, AND JOE MONTALBANO, MARK PHILLIPY, DIANE BRADY, JAMES SPERRY, STEVEN M. McGEE, SR., MICHAEL WILLIAMS, GUILHERME MOTTA, DOUG MARTIN, NICHOLE TUBIOLO, DIANE McCANN, AND RICK A. MORRIS, CO-AUTHORS

First Edition

GR8 PM

Traditional. Agile. Hybrid.

Spokane, WA
Table of Contents

FOREWORD .................................................................

PART ONE – INTRODUCTION

Chapter 1: Critical Foundation: ........................................ 1
  Overview .............................................................. 1
  Technology as an Enabler of Chaotic Expectations ............ 2
  Project Management as an Antidote to Chaotic Expectations .... 5
  Scrum as an Antidote to Chaotic Expectations .................. 7
  Proven Principles as a Solution to Competitive Complexity 
    and Uncertainty ...................................................... 9
  How the Almanac Helps Practitioners and Organizations .......... 13
  Important Structural Highlights and Conventions of Book 2 .... 15

Chapter 2: Agile Myths, Misconceptions and Anti-Patterns ......... 17
  Overview .................................................................. 17
  Anti-Patterns ............................................................ 19
  Dysfunctions in the Scrum Master Role ............................. 22
  Dysfunctions in Leadership .......................................... 24
  Myths and Misconceptions .......................................... 25
  As Agile Scales…So Do the Myths .................................. 34

Chapter 3: Agile Program Management: .............................. 37
  Overview .................................................................. 37
  Benefits and Challenges .............................................. 41
  Program-level Attributes ............................................. 43
  How to Make it Work Effectively ................................... 50
  Tools and Practices Recommendations ............................ 63
PART TWO – FRAMEWORKS ................................................................. 71

Chapter 4: Scrum of Scrums (SoS) ......................................................... 73
  Overview .................................................................................. 73
  Benefits and Challenges ....................................................... 77
  Program-level Attributes ...................................................... 80
  How to Make it Work Effectively .......................................... 91
  Tools and Practices Recommendations ................................ 96

Chapter 5: Kanban ................................................................. 101
  Overview .................................................................................. 101
  Benefits and Challenges ....................................................... 104
  Program-level Attributes ...................................................... 106
  How to Make it Work Effectively .......................................... 117
  Tools and Practices Recommendations .............................. 119

Chapter 6: DevOps ................................................................. 127
  Overview .................................................................................. 127
  Benefits and Challenges ....................................................... 133
  Program-level Attributes ...................................................... 135
  How to Make it Work Effectively .......................................... 146
  Tools and Practices Recommendations ................................ 151

Chapter 7: Scaled Agile Framework (SAFe®) ................................. 155
  Overview .................................................................................. 155
  Benefits and Challenges ....................................................... 157
  Program-level Attributes ...................................................... 161
  How to Make it Work Effectively .......................................... 169
  Tools and Practices Recommendations .............................. 173

Chapter 8: Disciplined Agile ....................................................... 175
  Overview .................................................................................. 175
  Benefits and Challenges ....................................................... 185
  Program-level Attributes ...................................................... 187
  How to Make it Work Effectively .......................................... 192
  Tools and Practices Recommendations .............................. 193

Chapter 9: Large Scale Scrum (LeSS) ............................................... 195
  Overview .................................................................................. 195
  Benefits and Challenges ....................................................... 198
  Program-level Attributes ...................................................... 202
How to Make it Work Effectively ........................................... 216
Tools and Practices Recommendations ................................. 220

Chapter 10: PMBOK® Guide Sixth Edition ............................. 223
Overview .............................................................................. 223
Benefits and Challenges ...................................................... 227
Program-level Attributes ..................................................... 233
How to Make it Work Effectively ........................................ 239
Tools and Practices Recommendations ................................. 242

Chapter 11: Agile Integration Framework (AIF) ...................... 245
Overview .............................................................................. 245
Benefits and Challenges ...................................................... 251
Program-level Attributes ..................................................... 260
How to Make it Work Effectively ........................................ 285
Tools and Practices Recommendations ................................. 293

PART THREE – APPLIED SCALED AGILE .............................. 297

Chapter 12: Virtual, Distributed & Multi-Team Environments .... 299
Overview .............................................................................. 299
Benefits and Challenges ...................................................... 302
Program-level Attributes ..................................................... 304
How to Make it Work Effectively ........................................ 311
Tools and Practices Recommendations ................................. 312

Chapter 13: Estimating & Scope Management ....................... 319
Overview .............................................................................. 319
Benefits and Challenges ...................................................... 321
Program-level Attributes ..................................................... 328
How to Make it Work Effectively ........................................ 341
Tools and Practices Recommendations ................................. 343

Chapter 14: Integration Management & Scheduling ................ 347
Overview .............................................................................. 347
Benefits and Challenges ...................................................... 350
Program-level Attributes ..................................................... 351
How to Make it Work Effectively ........................................ 358
Tools and Practices Recommendations ................................. 361

Chapter 15: Risk Management .............................................. 363
Overview .............................................................................. 363
Benefits and Challenges ...................................................... 370
FOREWORD

James R. Snyder | PMI Founder

If you are looking for the elusive, and probably non-existent, silver bullet for all your project, Program, and Portfolio management issues, this is not the right book for you! However, if you want a practical approach to the management of complex project problems, and some insights into the integration of your projects across your whole enterprise, then please read on. This exciting approach to using the Agile processes offers new and well thought out ideas and best practices that will go a long way to making sense out of the ever-changing, chaotic, project management environment in which we all live and work. This high-speed, high-risk environment is the world of managing project oriented work, is a part of every profession, and impacts every part of our working life.

The author of this book, John Stenbeck, has made significant contributions to the project management profession as a major supporter of the understanding and application of Agile processes and practices across a number of industries and disciplines. His relaxed, pragmatic application of Agile techniques to complex project management problems has been well received in many major corporations, and has contributed significantly to project completion success rates. He calls it the way he sees it and is not tied to the blind application of any one set of tools and methods.

The Agile Almanac – Book 1 brought together the science behind the best Agile practices helping to define the choices for implementation of these tools to appropriate projects in appropriate ways. By gathering all the best practices of both historic project management and the newer, emerging methods for managing project oriented work; John developed a guide for application in the field, not just in the minds of developers and consultants. Now, with help from a distinguished, international group of co-authors, including Rick Morris, an evangelist for project management who is a practitioner, consultant, author, mentor and expert, they have assembled a vast collection of knowledge and practical experience to show us the next steps in using Agile methods in the bigger world of multi-project Programs. Their work builds on the use of Agile and classic technology to solve the problems of large multi-projects environments of today.

This book takes the next step up in the hierarchy of project management by moving from the management of smaller, unrelated, single projects to the
much more complex, chaotic, and inter-related planning and management of large integrated projects, Programs, and Portfolios. This book provides a well-documented look at the inter-relationships between projects and Programs, and examines the existing Frameworks for project/Program integration. It also unveils the development, design, and implementation of the emerging Agile Integration Framework for large, inter-related, multi-project Programs as a new tool to deal with the complex project mix that represents the business environment we live in today.

This new approach scales Agile to encompass the broad, complex, and inter-dependent nature of the environment found in most major businesses, and uses some very basic project management processes to do so by using PMI’s PMBOK® Guide. The new Agile Integrated Framework developed here suggests that Critical Path Method (CPM) analysis be used to identify and clarify projects and their specific activities, which will be key in improving enterprise-wide total project performance. While the new emerging Framework for scaling Agile for large projects, Programs, and Portfolios relies heavily on both Traditional project management tools as well as the newer Agile processes and techniques, it retains the flexibility of the Agile process. It strongly supports and empowers teams to optimize project performance throughout the entire enterprise.

As Project Managers of today, we are well beyond the notion that all projects are alike and can be managed in the same way. Those of us with more than the early signs of gray hair know that just the opposite is true in the real project world. No two projects are ever the same, and no two projects can be managed using the same tools, in the same way, every time. Just as projects are unique and dynamic so must be the approach to their management. Over the last fifty years, we have seen the rise and fall of a number of “right and only” ways to manage project work. The failure of these approaches can be laid at the feet of the very nature of a project – all are unique and dynamic, and so must be their management. In the Agile Integrated Framework we see an approach, which recognizes the need for integrated, flexible solutions to project and Program management. What we have needed is a Framework that not only integrates the large and inter-connected projects in a business venture, but also the tools, processes and techniques available to manage them. This book takes us a long way towards meeting that need.

I think that you will find one of the most engaging aspects of this work is that it is not just an academic adventure into a new approach to solving a major
project management issue, but also a practical guide to making the solution work. You will find some excellent examples of how to make these new ideas for the scaling of Agile approaches to the integrated management of larger, and more complex, projects and Programs accompanied with the research-backed reasoning for making it all work.

You will learn a lot that will be immediately applicable to your business and enjoy the style and no nonsense approach of the writing! Have fun and prepare to scale your Agile processes to handle all your project work.
FOREWORD

Troy Hazard | Serial Entrepreneur, Best Selling Author

For 25 years, I have bought and sold companies, in different industries, in different countries, for different opportunities. Key to that process is to understand how to add enterprise value to the business, and manage the project from zero to hero, leaving the business in a better place than when we arrived. Sound familiar?

Along this journey the core to our commercial intelligence has been our ability recognize perception over reality, truth over fear, and commerciality over confidence. To do that, we’ve learned we need to be both practitioner and philosopher; and there is a fine line between the two.

The practitioner actually practices what they preach. The philosophizer muses over what could be.

The Agile Almanac – Book 2 is a classic representation of how to merge practice with philosophy, and turn what could be into what should be.

John Stenbeck helps you look beyond the practical message to the philosophical, personal application in your business and life, to enlighten you on a path to be the unicorn amongst peers.

If you are seeking to drive your business or career from what could be to what should be, then this book is a personal program management Roadmap. It will challenge your thought process and approach, and redefine the way you add enterprise value to every Program on the path to the outcome you seek.
FOREWORD

John Gates, Ph.D. | Partner, Avion Consulting

As a management consultant, I regularly work with leaders, teams, and organizations striving to succeed in an ever more complex and challenging environment. Specific challenges that the clients I serve face on a daily basis include increasingly lofty customer demands, pressure to continuously improve products and services, and in many cases, an expectation on the part of both internal and external stakeholders that excellent results be achieved with fewer resources.

These challenges are evident and pronounced in every industry in which we work – for-profit, not-for-profit, and everything in between. Of course, successful organizations, and those who lead them, understand that every challenge represents an opportunity. Those that lean into such challenges, and find ways to turn them to their advantage, are the ones that thrive. There are many keys to successfully navigating these turbulent waters, including having the right vision and strategy, effective leadership, and employee engagement.

But, in our experience, a key differentiator between the winners and losers in today’s turbulent business environment is effective program and project management. One of the true thought leaders in this field today is John Stenbeck, author of the best-selling “Agile Almanac – Book 1,” and now lead-author of the new “Agile Almanac – Book 2.” In this second volume in the Agile Almanac Series, Stenbeck and his co-authors, all Agile luminaries, expertly guide practitioners through Agile Program Management best practices that have been widely used and proven to be reliable. Beyond that, the authors challenge common wisdom and provide, in essence, a graduate-level education to anyone interested in the field of Program and project management.

The first volume in this Agile Almanac Series deals with single-team projects, whereas Book 2 addresses challenges associated with projects and programs being executed in environments that include the use of multiple teams with virtual, remote, and distributed elements. Written in a very practical way, such that the reader can focus on content that is most specific to one’s own environment and specific concerns, this book is an essential resource for virtually anybody in any organization with responsibility for ensuring that Agile Program Management best practices are being used to get the best possible results from projects, programs, and portfolios.
PART ONE

INTRODUCTION
Critical Foundation

Overview

Over the last 5 years I have been privileged to speak at more than 250 events and engage in focused discussions with more than 3,000 active Agile practitioners. Their collective wisdom has allowed me to combine a rarely-seen, cogent assessment of Agile factors. The elite team of co-authors, who have invested extensive amounts of time putting together this second book in the Agile Almanac, likewise endorsed this assessment then leveraged it to maximize the value of every chapter in this field guide and desk reference.

The vision of this book is to act like a Rosetta stone for Program Managers to help stakeholders, executives, and technical development professionals by providing a common lexicon to use to speak with one another in understandable terms.

Agile has been part of the technical and technology world in the past. Simply recall RAD, RUP, and Spiral and your mind will begin to fill with the many other fads and flavors-of-the-month that briefly held sway. But there is something quantifiably different about Scrum and the many variants that have come forth since the Agile Manifesto was published in 2001. Some unseen force seems to have been at work to foster the
broad adoption, acceptance, and undeniable success of Agile. This same force also seems to have moved beyond the current body of knowledge shared and published in Scrum-aligned circles.

Agile has operated with a hero-centric organizational model, which is part of the reason many practitioners and teams love it – because being a hero is intoxicating! However, hero-centric models are self-limiting and organizationally unsustainable. Scaling Agile will require a post-heroic organizational model that empowers and enables teams and individuals to be their best while also optimizing the throughput of the whole enterprise.

A required foundation for successfully implementing, then scaling, Agile is an understanding of the factor or factors driving organizations of every size, shape, and variety to use – or at least attempt to use – Agile.

This chapter will help set a foundation for success using and scaling Agile. It will also lay out the structure of this book and suggest how to use it. Mastering this content will benefit everyone in the organization in many ways.

**Technology as an Enabler of Chaotic Expectations**

Between approximately 1980 and 2015 the transistors on integrated circuits (IC) increased in density from roughly 50 thousand to nearly 10 billion. The

![Figure 1.01 | Transistors per Integrated Circuit (IC)](image-url)
trajectory, which resembled the “Moore’s Law” forecast made by Gordon Moore in 1965, has unlocked profound and nearly unimaginable improvements in the capabilities, miniaturization, and integration of devices of all kinds. (See Figure 1.01.)

The increased density on ICs expanded approximately 1,000 rough orders of magnitude (ROM), where a ROM is defined as 10 times greater than the base. (See Figure 1.02.) This expansion in processing capacity has had an impact comparable to humans harnessing fire and developing the steam engine. Both fire and steam engines unlocked profound eras of opportunity while simultaneously triggering avalanches of increased societal complexity.

Figure 1.02 | Transistors per Integrated Circuit (IC)

The improvement of ICs has unleashed an era of profound opportunity while triggering a tsunami of technical complexity.

That technical power and complexity can be seen in the growth of the Internet during the same timeframe. The number of hosts grew from 213 with ARPANET to 1 billion a few years ago, as shown in Figure 1.03. Users grew from a small cadre of technically-gifted geeks to approximately one-third of the global population while content grew from isolated, esoteric topics to every imaginable category, in every language used by humankind.

The growth of the Internet made it the most universal platform for communication in the history of humanity. It also induced a non-linear
increase in communication complexity. The amount of information available is increasing at such a rapid rate, exceeding the ability of anyone to comprehensively organize, assess, or consume it.

The communication complexity has been amplified by the number of ways the Internet can be accessed, manipulated, and leveraged to, for example, create derivatives or services based on the information. *(See Figure 1.04.)*
One critical outcome of this unbridled growth in sources of information, channels for accessing it, and tools for manipulating it has been a correlated growth in the complexity of customer expectations.

Organizations can now reach customers or constituents anywhere, anytime on the globe. Consequently, those same customers and constituents are suffering from information overload greater than any other time in history. Conversely, they can also find sources for anything imaginable, from anywhere on the globe, but the volume of choices has become a bewildering array of indecipherable confusion. Simultaneously, this group has learned, between Amazon, Google, and Facebook, that they can have anything they want almost immediately, for near-zero cost.

The result is a business environment of chaotic, unrealistic expectations enabled by seemingly unlimited technological advancements colliding with the proverbial laws of physics.

**Project Management as an Antidote to Chaotic Expectations**

Seeking a way to manage, or even benefit from, these chaotic, unrealistic expectations, businesses and organizations of every type turned to project management as the solution. One measure of this interest can be seen in the growth of the Project Management Institute (PMI) from 10,000 members in 1994 to over 750,000 certified Project Management Professionals (PMP®) in January 2017. *(Figure 1.05.)*

![Figure 1.05 | Project Management Complexity](image-url)
PMI’s approach, commonly called the Traditional or Waterfall methodology, is cataloged in various editions of *A Guide to the Project Management Body of Knowledge, (PMBOK® Guide)*. Amongst the generally accepted principles the *PMBOK® Guide* offers are guidelines that correlate levels of estimation with percentage of design completion and expected accuracy ranges for forecasts. The Rough Order of Magnitude (ROM) level of estimates are used when the Design Completion is less than ten (10%) percent and have an expected range of accuracy of +100% to – 50%. The Budgetary level of estimates are used when the Design Completion is approximately 15% to 25% complete and have an expected range of accuracy of +30% to – 15%. The Definitive level of estimates are used when the Design Completion is approximately 45% to 100% complete and have an expected range of accuracy of +15% to – 5%. (See Figure 1.06.)

![Figure 1.06 | Project Management Complexity](image)

The *PMBOK® Guide* supplies these guidelines.

<table>
<thead>
<tr>
<th>Estimation Levels</th>
<th>Design (%) Completion</th>
<th>Accuracy Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>0 – 10 %</td>
<td>+100% to -50%</td>
</tr>
<tr>
<td>Budget</td>
<td>15 – 25 %</td>
<td>+30% to -15%</td>
</tr>
<tr>
<td>Definitive</td>
<td>45 – 100 %</td>
<td>+15% to -5%</td>
</tr>
</tbody>
</table>

*The PMBOK® Guide* approach was apparently adequate during the years prior to 1990. But as the acceleration of environmental complexity became non-linear, growing evidence emerged that the estimating standards were inadequate. The biennial Chaos Report (from the Standish Group) became known as the seminal authority on the topic. When the first report was published in 1994 it created an awareness of a critical problem and fueled the perception of PMI as the source of the solution and certified PMP®s as the practitioners delivering it. In the years since, the Chaos Reports have shown the reality that project outcomes have remained mostly static. The results
show that only about one third of projects are in the category defined as “Success”, one third are “Challenged”, and one third are “Failed.”

It became clear that something more needed to be done. In a variety of places, people and organizations began using ideas from product development best practices and Lean Principles to create a better solution. And unlike the environment years before when RAD, RUP, Spiral, and a host of other iterative methods emerged but failed to gain widespread traction, a fertile environment, driven by the collision of technical capability, communication complexity, and customer demand, arose. (See Figure 1.07.)

![Figure 1.07](image-url)

**Figure 1.07 | Project Management Complexity**

**Scrum as an Antidote to Chaotic Expectations**

In 2001, seventeen software development thought leaders met at the Snowbird Ski Resort in Utah and wrote the *Agile Manifesto*, launching the Scrum Alliance into this fertile environment. The conditions were excellent and their thinking was clearly articulated, allowing the Scrum Framework to gain widespread traction.

In particular, the focus on iterative development, anchored by customer involvement, proved to be a powerful application of Lean Principles to the non-linear complexity of the environment. That non-linear complexity came to be expressed by the concept of a Cone of Uncertainty where a difficult problem was overcome by incremental steps towards the solution. Each step
forward eliminated one or more of the unknown factors or challenges. It also helped the intended beneficiary – referred to as the Customer, but included any key stakeholders – to clarify what the solution needed to accomplish in order to create value. (See Figure 1.08.)

![Figure 1.08 | Project Management and Non-Linear Complexity](image)

The combination of continuous Product improvement and continuous Process improvement produced desirable, even amazing, results for many challenging projects dealing with the non-linear distortion of time and size. It did so by leveraging Test-Driven Development (TDD) to deal with distortion caused by size. Additionally, it drastically lowered the cost of producing forecasts with an appropriate planning-level of accuracy to deal with the time distortion.

It turned out to be ‘synthesized genius’ that integrated project management and Lean Principles to solve near-term issues as the path to an overall solution. Even though the principles behind this breakthrough were not specifically identified in the Scrum Guide, the Scrum Framework worked for its many adherents. It provided a path forward to good results in a chaotic environment, but on a limited scale.

Today’s situation benefits from the 15 years of experimentation and adaptation that has occurred using Scrum and its variants. However, the situation now fits the adage often ascribed to Einstein that ‘the level of thinking that got us here is not the level of thinking needed to solve the problems that face us here.’ To solve the problems we face now, we must ask, “What is the intelligent way forward from here?”
Proven Principles as a Solution to Competitive Complexity and Uncertainty

Responding to market forces and customer demand, in 2011, PMI committed itself to developing a certification program and called it the PMI Agile Certified Practitioner (PMI-ACP®). The program was launched in 2012 and became the fastest growing certification in PMI history, even outpacing the Project Management Professionals (PMP®) standard bearer, exceeding 15,000 certification holders by the end of 2016. Because of that indisputable evidence, PMI’s leadership decided that an almost unprecedented update to the PMBOK® Guide was required.

The PMBOK® Guide Sixth Edition was released in 2017. It included Agile content in every single Knowledge Area, a change so fundamental, the PMBOK® Guide was released, in unprecedented fashion, in English and 10 other languages simultaneously.

The PMBOK® Guide Sixth Edition is an important milestone in the drive to find the most intelligent way forward, but it is not the solution. (See Figure 1.09.) It will play a significant role in developing the solution, but is not the sole source of it. Because the solution must effectively respond to changes far larger than those currently being managed using Scrum or one of the other “Big 5” Agile Frameworks – Scrum, XP, Lean Software Development, Kanban, or Hybrid – none of them is the sole source of the solution either. Agile will also play a significant role in developing the solution, but it, alone, is not the sole source of it.

Figure 1.09 | Project Management and Non-Linear Complexity
The solution will be found using integrative thinking – a “both/and” not an “either/or” model. The solution must employ principles proven to be effective across many industries and not limited to just one, such as software. It must have a comprehensive understanding of the marketplace challenges caused by non-linear complexity and uncertainty and apply proven principles supported by a set of tools that can cope with the demands of iterative, interactive planning, and development while also complying with rigorous Generally Accepted Accounting Principles (GAAP). It must accomplish all of that while integrating the important differences in planning and implementing responsive changes to customer demands in an environment where the ROM levels of change are historically unprecedented.

The *PMBOK*® *Guide Sixth Edition*, while important, lacks the required expertise to progressively elaborate from strategic long-term planning with, perhaps, 15 or more ROM levels of change in order to feed subsequent levels of planning down to operational planning with a far more manageable 1 ROM level of change. Conversely, the existing “Big 5” Agile Frameworks, while important, are well adapted only to operational planning with 1 ROM level of variation and lack tools for handling budgets or scaling planning and scheduling to the long-term with 15+ ROM levels of variation. (See Figure 1.10.)

![Figure 1.10 | Business Management: Non-Linear Complexity and Uncertainty](image-url)
Therefore, the solution to this must employ integrative “both/and” thinking where proven principles offer a comprehensive model that supports robust, reliable decision-making in spite of non-linear distortion caused by long timeframes and massive sizes of the desired outcomes.

The solution can be found at the overlapping intersection of the PMBOK® Guide, the “Big 5”, and traditional business schools, all of which have developed planning principles acknowledging various aspects of the problem while using different lexicons to describe it. (See Figure 1.11.)

![Defined Granularity is the Solution](image)

**Figure 1.11 | Business Management: Non-Linear Complexity and Uncertainty**

They all share the unarticulated understanding of the Cone of Uncertainty and the non-linear nature of time and size distortion causing it. They all also lack a full response to the technical, communication, and customer expectation complexity and uncertainty facing organizations today, as shown previously.

The vision of Book 2 of the Agile Almanac series is to provide an all-encompassing assessment of scaling Agile in the mid-term, tactical environment preceding operational planning, which was covered in Book 1.

**Part One** begins with a highly insightful discussion of the interrelationships between Project, Program, and Portfolio Management with an eye to
supporting Program Managers, Project Management Offices (PMO), directors, and business unit leaders. It also includes a highly practical discussion of Agile Myths, Misconceptions, and Anti-Patterns.

**Part Two** provides a comprehensive evaluation of the existing and emerging Frameworks – including Scrum of Scrums (SoS), Scaled Agile Framework (SAFe), Large Scale Scrum (LeSS), Disciplined Agile Delivery (DAD), Kanban, DevOps, and the *PMBOK® Guide Sixth Edition*. It then moves beyond the current Frameworks, introducing the **Agile Integration Framework**, a true “Graduate School” approach to scaling Agile across enterprises in private, public, and non-profit domains.

The **Agile Integration Framework** will address how to optimize the whole organizational process from “Ideation” to delivery of “Customer Delight”. It will demonstrate how to minimize the sub-optimization of development teams while maximizing enterprise throughput so the organization thrives in its battles to acquire the dollar-denoted votes of constituents and consumers.

**Part Three** delivers a comprehensive discussion of how to deal with environments complicated by Multiple-Teams and Virtual/Remote resources, a need for robust Estimating, Scheduling, and Scope and Integration management, and the intractable challenges of Risk, Quality, and Budget management.

**Part Four** closes out the book with two Appendices. The first Appendix provides an incomparable set of Academic Resources to support corporate trainers, university professors, and business researchers. The second Appendix is an unparalleled, top-to-bottom, roadmap and walk-through of how to organize and implement the toolset needed to cope with the demands of iterative, interactive planning and development while also complying with rigorous Generally Accepted Accounting Principles (GAAP). It details how to accomplish all of that while integrating responsive changes to customer demands in an environment where the ROM levels of change are historically unprecedented.

For any Program Manager to be a true professional, they must have proven tools and techniques that meet the needs of programs across every domain. They must also be able to effectively share and apply them in ways that empower program performance. That means they must host lunch and learns, speak convincingly to key stakeholders, and unlock important, yet intangible customer loyalty. These resources equip them to do exactly that!
Henry Ford once said, “Thinking is the hardest work there is, which is probably the reason so few engage in it!” The co-authors of the Agile Almanac – Book 2 have joined the “few” in order to provide a “short cut” to accelerate careers and save years of research and study. It is our solemn belief that this work will enable an organization, its vision, and the stakeholders it serves to reach the leading edge of market leadership.

**Understanding Almanac – Book 2**

The goal of this book is to serve the global community by offering only best practices that have been widely used and proven reliable. It presents recognized principles wherever they have been found and brings a unique perspective to the divergent, often conflicting and accusatory, viewpoints expressed by various “evangelists” who have created a “battlefield” with an absence of thought leadership.

Experience in delivering projects encompassing an immense number of industries and institutions has convinced us that Agile can, and must, be scaled as an extension of Lean Principles using integrative thinking at the overlapping intersection of the PMBOK® Guide, “Big 5” Agile Frameworks, and traditional business schools.

This book is going to increase Agile Program Management expertise and move an organization from playing checkers to playing chess when designing and executing programs!

**How the Almanac Helps Practitioners and Organizations**

The Agile Almanac book series allows practitioners and organizations to select precisely the content they need and zero in on applying it in the ways best suited to the context of their environments. We often remind students that the organization does not care whether they use Traditional methods, Agile techniques, or Aunt Suzy’s recipe. The organization only cares about results! In fact, they demand results as the validation that the Program Manager is a professional.

At the highest level, the Agile domain can be divided into three sub-domains – Single-team Projects and Exam Prep, Programs with Multi- and Virtual-Team Environments, and Portfolio Management and Enterprise Scaling. Each sub-domain is covered in its own book.
Book 1: Single-Team Projects & Exam Prep (Amazon #1 Best Seller upon release, 12/2015) Covers Agile Project Management, Lean Principles, and the “Big 5” Agile Frameworks:

- Scrum
- eXtreme Programming (XP)
- Lean Software Development (LSD)
- Kanban Basic Practices
- Hybrid Project Management

Book 1 focuses on the needs of individual practitioners, whether highly experienced or new to Agile Project Management, and provides detailed insight and analysis regarding if, when, and how to use the “Big 5” Frameworks. It also includes solid preparation for PMI’s Agile Certified Practitioner (PMI-ACP®) exam.

It has been extended with an On-Demand training course (which also became an Amazon #1 Best Seller just 5 hours after release, 09/2016).

Book 2: Programs with Multi- and Virtual-Team Environments (this book) Co-authored by Agile luminaries with a focus on the needs of Program Managers and organizations dealing with the challenges of large programs being executed in environments that include the use of multiple and/or virtual, remote, and distributed teams.

It provides insight and help for organizations whether they are a public agency, department or command competing for tax dollars in budget battles, or a commercial entity battling for consumer dollars.

Book 3: Portfolios and Enterprise Scaling (planned release in Q2, 2018)

- Agile Portfolio Management
- Agile Maturity Matrix Instrument (AMMI Assessment)
- Scaling Agile to the Enterprise

Book Three will be focused on the needs of organizational leaders and senior practitioners supporting Project Management Offices (PMOs), portfolios, and strategic decision-making. It will include an assessment instrument, analysis tools, and planning techniques that optimize total throughput within overall resource limits. It will provide insight that increases the top and bottom line as well as employee engagement.
Important Structural Highlights and Conventions of Book 2

PART ONE – INTRODUCTION
Part One covers the interrelationships between Project, Program, and Portfolio Management. These chapters are written with an eye to supporting Program Managers, Project Management Offices (PMO), directors, and business unit leaders. It also includes a highly practical discussion of Agile Myths, Misconceptions, and Anti-Patterns.

PART TWO - FRAMEWORKS
Part Two covers Scrum of Scrums (SoS), Kanban, DevOps, Scaled Agile Framework (SAFe), Disciplined Agile Delivery (DAD), Large Scale Scrum (LeSS), the PMBOK® Guide Sixth Edition, and the Agile Integration Framework (AIF).

The structure of each chapter in Part Two will follow a shared format in order to aid the reader with making cross-comparisons. The structure is:

Overview
Benefits / Challenges
Description of Program-level Attributes for:
  - Roles
  - Workflow
  - Ceremonies
  - Artifacts

How to Make it Work Effectively
Tools and Practices Recommendations

PART THREE – APPLIED SCALED AGILE
Part Three, with a perspective focused on program-level execution, covers Agile with Multi-Team Environments, Estimating and Scope Management, Integration and Scheduling, and Risk, Quality, and Budget Management.

The structure of each chapter in Part Three will share the same format as Part Two.

PART FOUR – APPENDICES
Each Appendix in Part Four will be an unparalleled, top-to-bottom, resource that is historically unprecedented. Appendix A – Academic Resources provides
an immense cache of specific resources to support professors and trainers. Appendix B – Advanced Toolset provides specific, detailed information for configuring the tools teams, Program Managers, and PMOs need.

PART FIVE – ABOUT THE AUTHORS & RESOURCES

Part Five includes sections for About the Authors, Editor, and Graphic Designer, and Glossary and Index.
Agile Myths, Misconceptions, and Anti-Patterns

Overview

This chapter shares, explores, analyzes, and debunks the most common myths, misconceptions, and anti-patterns plaguing serious Agile practitioners and the organizations they serve. The information and fact-based arguments presented will assist readers in becoming effective change agents by empowering them to overcome the typical resistance encountered when stakeholders have heard or experienced the negative repercussions caused by these myths, misconceptions, and anti-patterns.

It is critical to the success of any Agile leader shouldering the responsibility of leading a journey toward Agile transformation, or even a limited Agile adoption, that they be forewarned because as the old saying goes, “Forewarned is fore-armed!” Anti-patterns will be explored in a role-based examination of common dysfunctions. It will be followed by an exploration of common Agile myths and misconceptions. The final section will address myths and misconceptions specific to scaled Agile environments.

Before beginning, here’s a note about dogma. Agile is a values and principles based philosophy complete with its own “Manifesto.” Most popular Agile Frameworks even have their own corresponding set of values and principles. Because these Frameworks and Agile thinking are based on values, many “dogmatic” wars
develop among colleagues, especially in the early stages of an Agile adoption. “That’s not Scrum!” is often heard with its sibling, “That’s not even Agile!” usually following not far behind. Many such battles are waged between, and among, people who have just returned from their first Agile training and notice that what is being implemented, especially if it is at scale, is not exactly the Agile they were taught. Further, only a small portion of projects are done with a pure Agile approach, with many using a hybrid approach. Readers are urged to view Agility as a continuum and the journey to Agility as one that is never complete.

When implementing Agile in an organization, the question that should be asked isn’t necessarily, “Is this pure Agile?” The answer will usually be “No” because it is the wrong question. Asking better questions leads to better answers. Some questions to consider include the following. If their answers are yes, they are probably a good starting point.

- “Is this method more in line with Agile values and principles than what was being used before?”
- “Does this approach provide transparency along with opportunities for inspection and adaptation?”
- “Does this method move the organization closer to building an environment that embraces the four Agile values and the 12 Agile principles?”
- “Is this method the best method for this project and this team?”

![Figure 2.01 | Total Projects by Framework](image-url)

When evaluating Framework choices many organizations and practitioners find the real consideration is not whether it is ideologically pure, but rather,
whether it will enhance customer value and business results better than the current method.

An experienced Agile Coach can look at an organization and predict the success or failure of an Agile transformation effort with much greater accuracy than your local meteorologist can predict rain. The red flags that Coaches use to perform that assessment are discussed in the following section. They are the same red flags a Program Manager must use to assess and re-assess team needs. With that understanding, here are proven content and suggestions to help guide this process.

**Anti-Patterns**

**Dysfunctions in the Product Owner Role**

The Product Owner role has the highest correlation to the success or failure of most Agile implementations because it is the role responsible for maximizing value and being the “voice of the Customer” and primary source for setting the team’s direction. This makes it vitally important that the Program Manager and the Product Manager be closely involved with selecting, grooming, and coaching the Product Owners. *The Agile Almanac – Book 1* covers the Product Owner role in detail.

---

**Story from the Trenches**

“I was running an internal Introduction to Scrum course, where departments that were interested in adopting or were about to start a project using this approach, would come and learn about the topic. We would talk about tools, techniques, new language, what it meant, and what would change. During the introductions, one of the younger men in the room sat at the back “suited and booted,” looking very formal. He said it was likely he would be a Product Owner as he was currently a Team Leader.

As the day progressed, we started on the topic of culture change and how the Scrum culture differs from the waterfall (Traditional) project world. This gentleman said he didn’t think the culture was an issue as this was just a project, everything else remained the same. So, I asked him, ‘do you wear the suit, shirt, tie, polished shoes on a Saturday?’ He said ‘no’. I asked, ‘why not’. ‘Because it is not as comfortable as jeans, trainers, and a T-shirt,’ came the response. ‘Oh,’ I said, ‘so this
companies. Rick Allan, Southampton, UK

Common mistakes the Program Manager needs to be aware of when assigning the Product Owner role include:

**Picking the Wrong Person to be Product Owner**

Finding the right person to be the Product Owner is as difficult as it is important. Poor choices are, perhaps, the most common mistake in Agile, including:

- **Picking an Executive to be the Product Owner.** At first it might seem logical for a department manager or executive to be the Product Owner. Experience has shown that this is seldom the best decision and is, in many cases, the worst. Product Owners must dedicate intense, almost exclusive, focus because they are the person closest to the intersection of customer aspirations and product functionality. The Product Owner is the voice of the customer, users, and key stakeholders so they must be available to work with the development team daily.

Availability is a ‘luxury’ an executive rarely has. Most executives and department managers are too busy with other day-to-day responsibilities to honor the commitment of a Product Owner to be available to interact with the team every day. The importance of this daily interaction, and of the Product Owner’s availability to engage with the team, cannot be overstated.

- **Picking a Project Manager to be the Product Owner.** Sometimes people who were Project Managers before an Agile adoption are in-tune enough with the product and its
user base to become a good Product Owner, but this is an exception rather than a rule. This role demands someone who knows enough about the needs of the product and users that they can make good decisions, while also being internally motivated to want to fill this role. Many Project Managers see the Product Owner role as a step down, so they resent it instead of embrace it.

- **Picking a Business Analyst to be the Product Owner.** Senior Business Analysts have the required technical skills to fulfill the role, which they perceive as a promotion and are motivated to take it. The challenge is that this role demands someone who not only knows enough about the needs of the organization and the users, but someone who can, and will, make the tough, sometimes on-the-spot, decisions needed by the Development Team without having to consult others. Many Business Analysts act as a “Product Owner by Proxy” because it is not in the nature of the Business Analyst role, and perhaps their make-up, to make decisions and accept responsibility for them, which becomes a fatal flaw for a Product Owner.

**Product Owner Not Present or Not Engaged**

The Product Owner must choose to prioritize fulfillment of their obligations to the team. They must make time as needed to sit with the team and answer questions. An example of how they might spend that time includes studying the results of Spikes then discussing their impact on Backlog items. Regardless of how much the Product Owner knows about the product and the needs of its users and the organization, if they are unavailable or disengaged, it is a recipe for disaster. The Product Owner’s number one responsibility is to the team.

**More Than One Product Owner Per Team**

One of the core values that a Product Owner provides is clarity and consistency. Because of the high level of complexity and uncertainty in the development environment, the team must have a single voice to articulate the vision and make decisions about the incremental steps that will be used to fulfill it. When a second Product Owner gets involved, under the
mistaken assumption that it will help the primary Product Owner, a chaotic phenomenon ensues called “diffusion of responsibility,” characterized by a diminished degree of responsibility, ownership, and engagement of the Product Owner. The Program Manager must recognize this situation and cure it by redefining roles such that there is only one Product Owner for any given unit of functionality.

There is a Portuguese saying that, “A dog with two owners dies hungry.” This is because each owner assumes the dog will be taken care of by the other. This is also one of the key reasons for the Product Owner being a single individual and not a committee. The Product Owner increases the speed of the team by providing timely, reliable decision-making. If the Program is producing a deliverable simply too large or complex for a single Product Owner to handle, it is necessary, and far wiser, to decompose the deliverable into independent, de-coupled units of functionality, each with their own Product Owner and Program Roadmap defining the integration points where the final deliverable will be reviewed, tested, and assembled. Chapter 4 will cover this in more depth.

**Dysfunctions in the Scrum Master Role**

Second only to the Product Owner role, the impacts of the Scrum Master, both positive and negative, have a major effect on Agile organizations. This is one of the reasons consulting with a good Agile Coach is discussed at length in Chapter 3.

Dysfunctions that can be traced to the Scrum Master role include:

**Sprints Extending Beyond Timebox**

This is commonly seen in organizations as they first start with Agile. This occurs when the Development Team includes too much work into their Sprint Backlog and cannot finish within the agreed upon timebox. If the Sprint Goal is not achieved at the end of the Sprint, the Product Owner may extend the timebox until the work is finished. This breaks the development cadence and impedes the learning curve leading to more reliable sizing
of future Sprint Backlogs. The Program Manager should not allow this to happen because it undermines the process for building a planning process that produces robust, reliable forecasts used to influence stakeholder expectations.

**Daily Stand-ups Not Effective or Happening Consistently**
One of the clearest warning signs that the Program is at risk is when Daily Stand-up meetings are not consistently held or are ineffective. The Daily Stand-up meeting serves as the team’s cornerstone for synchronization and is the primary opportunity for inspection and adaptation. It also serves as an “alert mechanism” to bring obstacles to the surface. When the Daily Stand-up does not happen, impediments are allowed to grow unrecognized until they become blockers like the iceberg that sunk the Titanic.

**Dysfunctional Behaviors Not Addressed**
Because Agile programs operate in an environment of high complexity and uncertainty, the team dynamic plays a vital role in producing progress amid the Cone of Uncertainty. That means the Program Manager must cultivate an environment of personal safety so crucial conversations happen as needed to enable the Development Team to be fully functional. That sometimes means doing a constructive intervention focused on the behaviors needed to fulfill the team’s obligations to the Program. When this doesn’t happen, multiple dysfunctions can arise.

The Program Manager must provide a set of transparent guidelines for Scrum Masters seeking direction from Project Managers or the Program Manager. Those guidelines must explain when it is the right time to escalate to the Program Manager, such as defining a reasonable timeframe for the team to resolve the dysfunction themselves or a threshold for how many teams are affected by the dysfunction. The guidelines should be developed in collaboration with the Project Managers and Scrum Masters across the Program so they reflect the good judgment and perspectives of each viewpoint.

When dysfunctions cannot be resolved by the Program Manager,
Program risk increases substantially and should be escalated to the Sponsor level.

**Dysfunctions in Leadership**

Leading the transformation to a Lean-Agile organization is no small feat and is difficult even when the Program Manager is knowledgeable, experienced, and engaged. The Program Manager is responsible for cultivating leadership at all levels. They not only understand Lean-Agile principles, but actively apply them to support the organizational change the Program needs to succeed. Here are some common leadership dysfunctions.

*Not Actively Supporting and Embracing the Change*

This is a big one! An Agile Program will drive a transformation that cannot happen without leadership. Simply signing off and moving on is the lip service that will induce failure. The Program Manager must actively work to incrementally, and where required, dramatically change the mindset across every level of the Program in word and deed, every day.

*Story Points Misunderstood*

This is one of the biggest challenges for business leadership. The Program Manager, aided by the Project Managers, must consistently and accurately portray the use of Story Points in order for the Agile adoption supporting the Program to succeed. There are two commonly misunderstood facets of Story Points. One is that they are non-linear so a Fibonacci sequence or similar scale is used to assign them. The other is that they are unique to each team. For example, in a four-week Iteration, if one team completes 35 Story Points writing simple reports while another team completes 35 Story Points writing difficult algorithms, it doesn’t make sense to assume they had the same perspective on the difficulty of a Story Point.

Convincing executive management that Story Points are valid for planning purposes and useful as a performance or progress metric, and they cannot, and should not, be normalized across teams is a challenge for the Program Manager to own.

*Inadequate Training*

This surprisingly common dysfunction occurs despite mountains of evidence demonstrating a powerful, positive ROI and the very real, very steep cost for failure. The conundrum is that leadership often requires training for their
staff, but fails to invest in themselves. The logic that “The Scrum Guide is only 16 pages long, so perhaps only the Development Team needs two days of training on it” is deeply flawed. For Programs to succeed, Program Managers must invest in themselves and be persuasive in arranging the correct type of training for senior stakeholders.

This dysfunction presents itself when leadership lacks the ability to effectively embrace change by cultivating an environment of personal safety where teams can exercise the Agile mindset of “Fail fast to succeed sooner” without fear of career suicide. Leadership must reward the team for taking risks and enduring setbacks because these make the Agile Program succeed.

Myths and Misconceptions

Agile attracts myths and misconceptions like a picnic attracts ants. Some of the most common ones make Agile about as welcome as the ants too. Program Managers are encouraged to use this section to prepare for effective dialogues with stakeholders and teams. This chapter provides insights that will resonate with them and increase the Program Manager’s ability to influence, persuade, and lead them.

Agile is Only for Software

The fact that the visionaries who gathered in Snowbird, Utah in 2001 to write the Agile Manifesto were software development professionals frustrated with the failings of Traditional Project Management does not mean Agile is for software only. Many of the practices that have grown out of the Agile Manifesto can be directly traced to Lean Manufacturing, the Toyota Production System, and W. Edwards Deming in the 1950s.

Here are two good examples of Agile in non-software environments.

- The SAAB JAS 39 Gripen fighter jet (shown in Figure 2.06), was built using Agile methods and has the lowest cost of any Western fighter jet according to a study published by IHS Jane’s in 2012.¹ The Gripen operating costs are just 65% of the F-16E!² An analysis of the Agile application to this project concluded,

Fig 2.06 | SAAB JAS 39 Gripen Jet Fighter
“Through Agile practices Saab…[achieved] an aircraft delivered for lower cost, with higher speed, and greater quality.”

- In 2012, an all-volunteer Seattle team entered the Progressive Insurance X $10 Million Prize competition with the goal of building a 100 MPG street legal car (shown in Figure 2 – 07). The self-funded team used Agile methods and, in three months, tied for 10th place against many other, well-funded teams using Traditional methods with more time. The final product was built entirely with off-the-shelf parts, got 100 MPG, did 0 – 60 in less than 5 seconds, and had a top speed of 149 MPH.

The important take-away is that Program Managers now have a vast array of new Agile tools in their toolbox and should use them to guide, help and influence executives, Portfolio Managers, Product Managers, and other stakeholders wherever those tools add value, regardless of the industry, product, or service.

**Too Many Meetings**

A common complaint Program Managers will hear is, “I am too busy to participate in so many meetings. I won’t have time to work.” Program Managers must be persuasive in explaining that meetings reduce risk and rework, which is waste, by ensuring that vital communication happens in the large, complex, and uncertain environment of the Program. Agile is a big culture shift and Program Managers have to own the responsibility of leading the change. Regular meetings are an essential feature of Agile. It is critical for team members to communicate consistently in order to succeed. Properly managed Agile meetings are productive collaborations in which critical work synchronization, alignment, and integration is actually accomplished.

**Two Weeks is Too Short**

The most common timebox for Iteration length is two weeks. Many companies mistakenly think they must use two-week Iterations, which are too short for their industry. However, Agile does not specify one Iteration length as good and another as bad. The key is to understand that the shorter
the timebox, the faster feedback is received, teams learn, and processes improve. However, shorter Iterations increase the number of Iterations and, thus, the number of meetings. Meetings increase overhead or, in Lean terms, increase waste. Therefore, the only reason to shorten Iterations is when the increased learning and process improvement provide a positive ROI.

In recent years, some Agile consulting firms have been training teams by initially using daily Iterations to speed up acquisition of all the needed Agile skills. Basically, by going through all the Agile ceremonies - working with Product and Iteration Backlogs, and holding review and Retrospective meetings - with each daily Iteration, the team hyper-accelerates through the learning curve. While it is not sustainable in the long term, it is great for teaching, getting to know team members, and sharing the importance of Agile values and principles.

With a little discipline, proper training, and good engineering practices, it is possible to have a sustainable and healthy work environment with timeboxes that are the appropriate length, even in two-week Iterations!

**Every Iteration Must Deploy into Production**

The Scrum Guide, on page 8, states, “The heart of Scrum is a Sprint, a timebox of one month or less during which a “Done”, a useable and potentially releasable Product Increment, is created.”

Every Sprint needs a high-level Sprint Goal or Sprint Objective. The set of User Stories promoted from the Product Backlog to the Sprint Backlog needs to be the set providing the highest value for the end-user and organization. However, the idea that each Sprint will deliver value does not mean that every Sprint will deploy in production. Many times, Agile teams deliver fully shippable code at the end of their Sprint, but the Product Owner decides to release it at a later date.

Depending on the system, adding or changing a feature may require training users and support activities, which might be too costly to do with every Sprint (regardless of the length). On the other hand, many teams who work with XP and Agile perform multiple deployments into production each day. The difference here is the kind of system and confidence provided by automation and good practices that lead to the right amount of quality, security, and performance.
Amazon provides an insightful case study of the interface between Iterations, releasing into production, and having Program level standards. Amazon releases into production from as many as 1,000 development teams per day. That’s right, *per day!* But the vast majority of those Releases impact simple user preferences or fixes to the appearance or operation of non-core functions. Conversely, any Iteration where the deliverable will impact core functions, such as security, credit card processing, or the personally-identifiable-data of customers, sellers or advertisers, absolutely may not be released until it has been approved by a predefined set of stakeholders along a prescribed escalation path.

Program practices such as source control management and continuous integration will play a big role in defining the mandatory quality and security of the Program’s Potentially Shippable Increments (PSI) and will, therefore, directly influence the frequency of those deliverables being released into production. Be careful not to mix up continuous deployment with continuous delivery.

**Agile Projects are Difficult to Budget**

Agile provides the most value for Programs where there is unavoidable high complexity and uncertainty. Agile embraces the perspective that scope will change as learning and discovery happen in the midst of high complexity and uncertainty. By aligning budget and schedule expectations with appropriate planning-levels of granularity, Agile Programs can develop robust, useful project and financial plans where the greatest risk of unpredictable and unmanageable variances has occurred for most organizations. In other words, Agile Program Management improves budgeting accuracy on the projects that have historically been the toughest to get right.

A Program with a fixed budget will deliver as many high priority Features as possible from the Minimum Viable Product (MVP) and when the budget is spent, the Sponsor or key stakeholders will decide if further work should occur.

Likewise, a Program with a fixed timeframe will work to deliver as many high priority Features as possible from the MVP until the predetermined deadline is reached. The Sponsor or key stakeholders will then decide if further work should occur.
For Program Managers, the brilliance behind this model is that the remaining Backlog items will be judged on their own merits. If the value of the remaining Backlog items does not exceed the cost of producing them, the Program ends.

**Budgeting is Difficult Because There is no End**

Some Programs support Products, tangible or intangible, that have their own Product Lifecycle based on the ongoing financial viability of the Product or Product Suite. The challenge for Program Managers is that, by definition, Programs and Projects have a beginning and an end. But when the Program supports a Product that does not have a predefined end, then neither does the Program. In such cases the Program and Program Manager must accept that the Product Lifecycle will cause a continual need to develop and improve the Product. In such circumstances, the Program adds great value because, rather than disband and reform teams over and over, it keeps the Development Team together. This avoids the waste of repeatedly forcing teams to go through the forming, norming, and storming phases of team development. It allows the Product to benefit from unbroken periods of robust deliverables from high performing teams.

It also means the Program can provide known, dependable development with a consistent cost that can be reliably forecasted and budgeted. Release Plans can be developed to aid financial planning for platform and infrastructure improvements supporting the planned, additional Features.

**Accounting Cannot Capitalize Development Done with Agile**

Generally speaking, using a scaled Agile Framework implies the organization has made a commitment to create competitive advantage. However, many organizations feel that once the decision to use Agile methodologies has been made, the ability to capitalize the development work is lost. This is due to the flawed thinking that it too difficult to quantify or track Development Team work and split it between work that qualifies under capitalization rules and work that must be expensed. The concept of Story Points also tends to create a “translation issue” for accounting. Luckily, going Agile does not mean sacrificing the ability to capitalize work as long as the correct focus is established between accounting and the Development Teams. With some simple configurations of the right development tool (as discussed in
Appendix B) the information can flow freely to accomplish both accounting and development reporting tasks.

There are three main ways that organizations have been capitalizing work in Agile environments.

- Normalize Story Points using a formula, such as capitalize 80% of completed Story Points per Release and expense the other 20%, after the formula is approved by accounting and the auditing firm.

- Use an integrated timesheet in a hybrid project management environment. Since many organizations are still tracking time to capitalize development efforts, an integrated timesheet, using line items for each Agile Feature, Function or Capability being developed, can report a value for capitalization and another for expense.

- Breakdown the Tasks under a User Story to create the percentage of effort that is capital and expense per release. Then the tool records who is on the team, the team cost, and the capital and expense transactions in a fully automated way.

Creating these solutions requires collaboration between the accounting, audit, management, and development teams. The right tool, properly configured, can easily automate these functions so the organization maintains the key advantage of capitalizing development efforts.

**Agile is Faster and Cheaper than Other Methods**

Agile Frameworks are often adopted because of a desire to bring products to market faster with the expectation that they will also be cheaper than if other methods were used. No one can fault that logic. It is important, however, for Program Managers to understand exactly what Agile can and cannot do before making the decision to use an Agile Framework.

While it is true that Agile methods will, in the right circumstances, help organizations prioritize product Features by value and teams develop small, fully functioning units of value in short timeframes, the idea should be clarified. The catch is that the value will be delivered incrementally and iteratively. Smart organizations will find ways to monetize their products at the lowest possible level and use those early revenues to improve ROI or fund development of further Features. That does not mean that Agile methods will “auto-magically” help organizations iterate towards a completed product faster.
than Traditional methods. What it does mean is that when the Agile Product Owner or Program Manager sees that the point of diminishing marginal returns has been reached, they will choose to assign the team to other work with a higher ROI. Because of this, comparing products developed using Agile versus Traditional methods is not an apples-to-apples comparison.

**It’s Okay to Tell the Executives (or Stakeholders) to Wait and See**

Some Agile aficionados seem to think Agile empowers them to tell executives and stakeholders, “You’ll just have to wait and see what turns out because Agile means no long-term planning.” Perhaps they work in places where the boss says, “Take all the time and money you need to complete the project!” For the rest of us, that is just a dream.

Agile thrives in environments where frequent changes and technological uncertainty are common. Remember, Principle #2 of the Agile Manifesto says, “Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.” This does not mean, however, that there is no long-term planning in Agile or that Agile Program Managers are unable to give their stakeholders a Product Roadmap or Release Plan.

In Scrum, for example, planning is done at five, increasingly granular levels – Product Vision, Product Roadmap, Release Plan, Sprint Plan, and Daily Stand-up. Unlike a Release Plan in a Traditional project, a Scrum Release Plan has scope, not date or quality, as its variable. For example, a Product Owner can tell executives and stakeholders that the Release Plan for the next two months calls for Features E, F, G, and H to be released. The only variable is how much of each Feature will be released. The Development Team may be able to complete all of Features E, F, and G, but possibly only half of the planned functionality for Feature H. Since Feature H is the lowest priority Feature, the team will build the MVP first, adding other, lower priority functionality later. In this way Agile Program Managers can communicate to their stakeholders a high-level Release Plan that they can work with so long as they accept that scope is variable.
**Agile Uses “Generalizing Specialists”**

While the origin of the term “generalizing specialists” cannot be definitively assessed, it is commonly interpreted to mean a person with deep expertise in a particular area who also has experience within an array of other, usually related, areas. The hypothesis would appear to be that the type of person able to master multiple skills has a strong ability to learn new skills quickly. The idea seems to have emanated from the software industry referring to programmers who also do other development-related tasks, such as testing.

Generalizing specialists may be a good idea, but is a misconception in the sense of being a less-than-robust response to the central challenge. In this case, the central challenge is how to avoid the risk of dependency on sole-source providers of particular skills because they can leave the organization, creating a major operational deficit. Let’s look at this in analogies from different viewpoints.

From an accounting perspective, each member of the team is a resource paid in proportion to the tacit knowledge they can supply. An increase in tacit knowledge, also known as experience or battle scars, is why a typical engineer with 10 years of experience gets paid more than a recent college graduate in the same field. If the more experienced engineer leaves an organization, it is likely harder, and more expensive, to fill the void.

An alternate, and, perhaps, stronger solution for this problem is tribal knowledge. It is not uncommon during Olympic track and field relay races for the team with the fastest runners to fail to win the race. But 100 percent of the time, the winning team is a team that did not drop the baton. The baton is the symbol of tribal knowledge and the runners are the sole-source providers of skill.

One of the powerful factors common to Agile approaches is cross-functional teams. Each member of the cross-functional team contributes tacit knowledge while the system of Iteration Planning and Stand-up Meetings, plus other tools and activities, such as User Stories and Planning Poker, creates tribal knowledge. This is so because over the course of time, for example in software development, the programmer explains to the quality assurance team member that if they changed the interface of their activities in a certain way it would aid the programmer, and vice versa. This continuous process improvement creates tribal knowledge reducing the impact of a highly paid, sole-source skill provider leaving the organization.
because it reduces the transition cost and learning curve for the new hire. That is so because when the new programmer arrives, the quality assurance team member explains how the interface works and high-speed productivity resumes without the risk of the baton being dropped.

This approach also compensates for a nearly-fatal flaw in the assumptions underpinning the “generalizing specialists” solution. In a great many environments, sole-source skill providers are as different as zebras and giraffes. Asking them to be generalizing specialists is like hoping for a striped giraffe. Agile offers a much more reliable solution with a systemic approach, creating easy-to-hold batons of tribal knowledge.

**Agile Opposes “Big Design Up Front (BDUF)”**

Perhaps the root of this misconception can be traced back to the Scrum Guide published in November 2009 by Scrum.org, owned and shared by Jeff Sutherland and Ken Schwaber. On page 8 it said, “Release planning is entirely optional. If Scrum team(s) start work without the meeting, the absence of its artifacts will become apparent as impediments that need to be resolved. Work to resolve the impediments will become items in the Product Backlog.”

Apparently, somehow, somewhere, someone saw and started parroting only, “Release Planning is entirely optional,” missing the critical warning that followed stating that if teams started work without the Release Plan artifacts it would become an impediment to resolve. Implementing single-team Agile, much less scaling it, with an Iteration-plus-Backlog-only mindset is a simple recipe for creating disasters. The lack of good Release Planning is rampant in parts of the Agile community, rejecting it as unnecessary, which contributes to discrediting Agile.

The idea that anything but the smallest, most inconsequential development could be done with no Big Design Up Front (BDUF) is even more dangerous and ludicrous than heavy-weight planning done under the banner of Release Planning. Two examples add insight. First, XP uses the principle of Incremental Architectural Planning as its version of a BDUF. Second, Lean Software Development applies the principle of Optimize the Whole, which implies the assumption of a BDUF of the whole.

To succeed in scaling Agile, professional practitioners must be prepared to respond to this canard whenever they see it raise its ugly head! The solution is to cultivate an understanding of how to align planning granularity with decision-making granularity, which will be covered in depth in Chapter 11.
As Agile Scales… So Do the Myths

Agile Doesn’t Work with Distributed Teams

The annual *State of Agile Development* report, funded by VersionOne, and the *State of Scrum* report issued by the Scrum Alliance are both generally accepted as reliable sources of information regarding Agile.

According to the 2017 *State of Agile Development* report, “86% of respondents had at least some distributed teams practicing Agile” while “51% reported using Agile to manage outsourced IT projects.” The 2017 State of Scrum report stated that 54% of respondents said Scrum teams had members who were “…distributed across different sites and/or geographic areas”. While it is generally accepted that collocated teams provide many benefits in Agile development, in practice it is not mandatory. To be clear, collocation is still preferred, but distributed teams have become far more common.

Driven by industry trends toward distributed workforces, and remote and offshore workers, Agile practices have adapted. Most companies decided that the benefits of Agile outweigh any challenges remote teams or workers introduce. A variety of technologies and software has been introduced to help these distributed Agile teams communicate. Some of the scaled Agile Frameworks, such as SAFe®, make provisions for including remote and vendor teams in planning.

Impossible to Integrate Agile and Non-Agile Teams

Integrating Traditional and Agile teams in a Program is unavoidable. Therefore, Program Managers are required to lead the planning and implement carefully crafted communication-by-design instead of communication-by-accident in order to ensure success.

In practice, the most important principle is to map out the integration points using the old school approach of building a logic network then enhancing it with the Agile techniques for low-tech, high-touch methods and a focus on being effective before being efficient. Being effective in delivering Customer Value must have priority over being efficient in the development how-to approach.

A complete presentation of how to do this is included in Chapter 11.
Conclusion

This chapter has covered the most common and highest-value risks and anti-patterns in Agile. The tools provided to avoid them, or at least begin a dialogue about them with stakeholders, will be of great value to many Program Managers. For more Agile Myths, Misconceptions, Anti-Patterns, some interesting Horror Stories, or to contribute your own, please join the GR8PM Tribe at GR8PM.com.