GR8 PM

Traditional. Agile. Hybrid.

Why ACP is HOT!

READING ASSIGNMENT

(Pages 15-30, 45-49)

CHAPTER

2

Introducing Agile Project Management

Chapter Highlights

In this Chapter we will begin with an overview of Agile Project Management and provide a quick, familiar example to help you create a high-level mental map of the subject. Then we will explore the idea of mapping Agile to the PMBOK® and review both agile planning and estimating as well as agile execution and control practices. With that mental map in place, we will cover the origins of Agile Project Management, the history of Lean systems thinking, and the application of Lean thinking to project management.

We will conclude by summarizing the agile frameworks and tools on PMI's ACP Examination Content Outline, plus a few other frameworks, including:

- Scrum
- Extreme Programming (XP)
- Lean Software Development (LSD)
- Other Agile Frameworks
- Test Driven Development (TDD)
- Agile Modeling (AM)

Lastly, we will introduce the ***Agile Project Management Processes Grid***™, a tool for ACP Exam preparation.

Overview of Agile Project Management

A Quick Familiar Example

Years of teaching experience have shown that sharing a quick, familiar example at the 50,000 foot level helps students create a mental map as they study Agile Project Management in greater depth. Below is an example of agile processes being applied in a situation that will be familiar to you.

You and three of your friends are hosting a dinner party together. Your project objective is to put on a successful party and remain friends afterwards! You are a cross-functional team because each of you has a different skill set. One is good at making cocktails, another is good at hors d'oeuvres, the third is good at entrees, and the fourth is good at desserts. Your team is not independently wealthy so it cannot simply outsource all of the work to a catering company. Consequently, you will be doing the work yourselves. To be successful, you want to establish mutual accountability as well as task integration so that there is support for each of you when it is time for the next deliverable.

The first agile principle in this example is that the team must have the necessary skills to complete the project. Agile is not a silver bullet! If a team does not have the required skills, even agile cannot help it successfully complete projects. **The second agile principle here is that the team must be self-organized, highly-trusted, and accountable.**

Because your team is also required on other projects (such as going to work) the team has agreed to use 4 iterations (in the evening) to complete the project. They are:

- Wednesday – Plan and acquire resources
- Thursday – Produce sub-components
- Friday – Complete and deliver the party
- Saturday – Clean up and do a retrospective before the next party

An ***iteration*** is simply a ***timebox*** within which work will be completed. In the agile world, work is done in iterations and a release can be the output of a single iteration or the output of several iterations that are inter-related by design choices.

> **Iteration #1** for your party is planning. Your team sits down and talks about what kinds of cocktails, hors d'oeuvres, entrées, and desserts you will serve. Based on the outcome of that discussion, the team prepares a shopping list of ingredients needing to be purchased in order to put on a successful dinner party. Each store where the team will shop is put on a separate piece of paper and assigned to a team member to purchase those items.
>
> At the conclusion of iteration #1, the theme of the party, shopping lists, and a plan for its delivery have been established. These deliverables are referred to as a ***potentially shippable product increment.*** If a blizzard should blow into town on Thursday and prevent the party on Friday, the team has still produced a result, or output, which is useful when the project is resumed.
>
> This agile principle is incremental building and frequent delivery of ***potentially shippable product increments.*** A potentially shippable product increment is anything that has value because the customer can see or use it to understand project progress. It may also have reference value for the team after an unexpected delay in order to restart the project. Also, a foreshadowing of the practice of ***user stories*** can be seen in the shopping lists. User stories are written documents that help the team understand what work needs to be done.

Iteration #2 is preparation and logistics, specifically food preparation and set-up. Because you are working as a cross-functional team, you gather in the kitchen and help each other with washing, peeling, chopping, and storing the food ingredients you will use the following night.

At the end of iteration #2, you, once again, have an increment of value in a deliverable. All of the sub-components have been prepared or created and stored in the refrigerator.

If you should get an unexpected call that three of your guests missed their plane, forcing the party to be postponed until Saturday night, the iteration was still successful. The team produced a result, or output, which demonstrates project progress and is useful when the project is resumed. Furthermore, if the party was postponed for a week and some of the prep had to be redone due to spoilage, it would be similar to the work required to ramp a project back up after it has been stopped.

The agile principles seen here are colocated work space (the kitchen) and also interrelated yet independent deliverables that demonstrate project progress.

Iteration #3 is execution; welcoming your guests and enjoying dinner with them. Everything goes as planned, good conversation occurs, and you enjoy your guests and the time around the table together. You have delivered (or consumed) another potentially shippable product increment.

In the **agile vocabulary,** we would refer to **Friday's output** as both a **potentially shippable product increment** and as a **release.** It's considered a release because it was the cumulative effort of the first 3 iterations and it was delivered to the customer – your friends. A key idea in Agile Project Management is that a release can be planned one of two ways. This example demonstrated one approach, where the release deadline was known and fixed and therefore the exact feature set was subject to some adjustment if needed. The focus was to deliver something the customer valued when it was promised. For our example, the release was set for Friday and the customer (your friends) expected a deliverable of food and festivities, which they received.

The second approach to release planning is to define specifically what will be delivered and then analyze when it can be completed. After all, at a high level, the only way to fix both the scope and the date is to vary quality, and in **agile, quality is never varied** – it must be a working piece of functionality.

Iteration #4 is cleaning up with a retrospective meeting immediately afterwards. With agile in a normal environment – not the daily iterations we described here – **the team holds two meetings at the end of each iteration.** The first meeting is the **review meeting** where the potentially shippable product increment is presented to all interested stakeholders for their review and feedback. The review meeting is **product focused.** In our example, the customers ate the meal and were satisfied with the product, which was, in effect, a review meeting. The second meeting, which only the team attends, is the **retrospective meeting.** The retrospective meeting is **process focused.** The team uses that time to identify ways to improve how they create deliverables.

Because the project was completed by a cross-functional team, they also managed to remain friends, fulfilling a key objective. The project is complete, everyone is happy, and the experience with Agile Project Management has been both educational and successful.

This simplified example illustrates some of the key concepts and challenges of Agile Project Management. One of those key concepts is the idea of breaking up larger projects into interrelated, incremental deliverables. Those deliverables must be related and integrated in a fashion that continuously delivers value by building units of the solution. Each unit must be independent and build on prior work to move towards the final comprehensive solution. Finding the people in any organization who have the level of expertise needed to plan those types of increments is a very real challenge.

Is Agile Really Needed?

Even though there is an early precursor of agile in the concept of rolling wave planning, the last major tool recognized in the Project Management Institute's ***"A Guide to the Project Management Body of Knowledge (PMBOK® Guide Second Edition)*** was the Critical Chain in 1997[1]. That fact raises the first question we should ask, "What has changed since then?" Consider the following:

- Google launched in September, 1998
- The iPod was unveiled in October, 2001
- The BlackBerry "smartphone" was released in January, 2002
- NASA's Phoenix lander extracted Martian ice in June, 2007
- The iPad was introduced in April, 2010

Interestingly, the Apple iPad provides a "classic case study" in Agile Project Management. In Lean and agile terminology, it was a full function device that included the **minimum marketable feature** set, yet it was not a full feature tablet PC. Because it was focused on what the customer wanted, it sold 3 million devices in 80 days and almost 15 million devices in the 8 months of 2010, taking 75 percent market share of tablet PCs by the end of the year. That meant that it sold more units than all other tablet PCs combined.

The success of the iPad speaks eloquently to the success that agile enables. It also challenges organizational leaders who may feel an expectation for them to produce achievements like Steve Jobs.[2]

Even PMI acknowledged the increased demands and complexity of the project management universe when they moved beyond the long-cherished Iron Triangle – time, cost, scope – that was a part of every edition of the PMBOK through the Third Edition. With the release of the PMBOK, Fourth Edition, PMI took the traditional view of time, cost, scope, and added quality, risk, and customer satisfaction. The triangle became a hexagon in order to express the increased complexity that project managers now face in the everyday world. Soon project managers around the world will be speaking about the ***"Hell-of-a-Hexagon"*** that replaced the ***"Iron Triangle."*** (See Figure 2.1.)

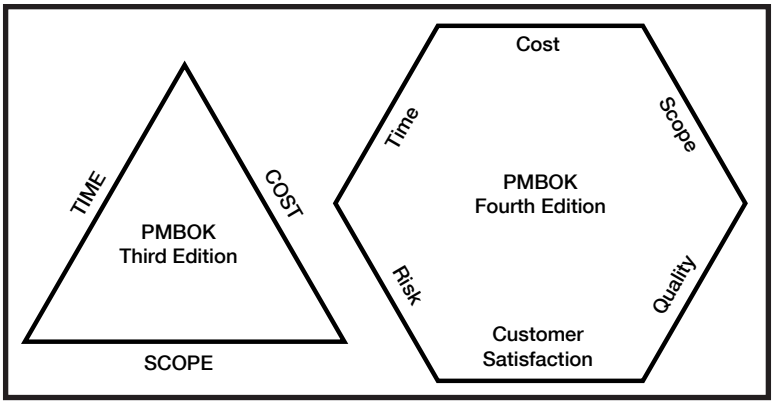

Figure 2.1 Comparing the "Iron Triangle" to the New "Hell of a Hexagon."

There is an abundance of additional evidence that points to the added complexity faced by project managers. Consider the high project failure rates documented over the last couple of decades by the Standish Group in the aptly named CHAOS Reports.[3] Or consider the report from Standish that proved only 20% of the features being delivered to users are in the "Always" or "Often Used" categories, while only 16% are "Sometimes Used," and a full 64% fall into the "Rarely" and "Never Used" categories.

Mapping Agile to the PMBOK®

Under the "Traditional" project management umbrella, PMI is the industry leader. PMI has the largest membership base, by far, of any professional user group for project managers and has developed the most recognized and best-respected credentials and certifications for practitioners in the project management field. There are, however, a host of smaller regional and local players that offer competing membership and certification choices. Regardless, PMI remains the leader and dominates the trends in identifying best practices because of its extensive research grants and educational scholarships.

Under the "Agile" project management umbrella, the Scrum Alliance (SA) is the biggest player because it has the largest membership base specifically in the agile sphere. It developed and controls the most recognized certification – the Certified Scrum Master (CSM) – for practitioners in the agile PM discipline. However, a host of smaller regional and local players offer competing memberships and certifications as well. (See Figure 2.2)

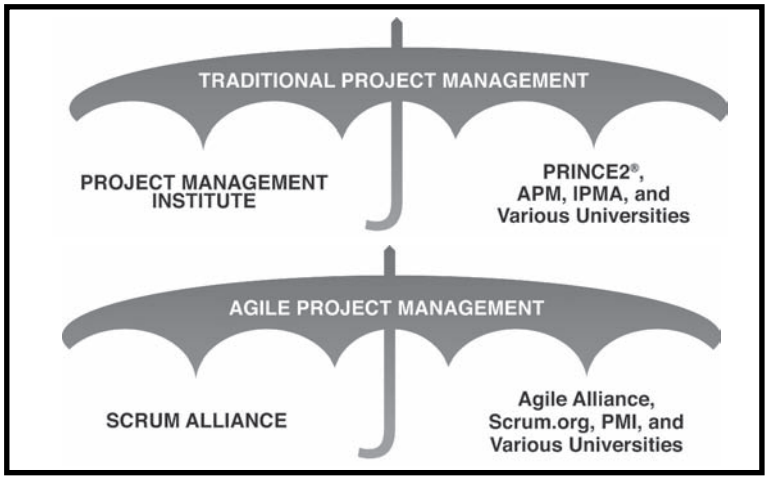

Figure 2.2 A Comparison of Traditional and Agile PM Industry Participants.

Because of the high profile that the SA has in the agile sphere, it is common to refer to Scrum and agile interchangeably – much like Kleenex® and facial tissue – but it is not always accurate to do so. ***Scrum*** is a variant, flavor, or approach within agile, however agile preceded Scrum and is broader than Scrum. Agile also includes several other notable frameworks, such as ***Extreme Programming*** (XP), ***Lean Software Development*** (LSD), and ***Feature Driven Development*** (FDD). Agile also has a group of others that make up a minute part of the market, including ***Crystal, Dynamic Systems Development Method*** (DSDM), ***Agile Unified Process*** (AUP), and ***Spiral.*** Finally, although the SA is the current leader in the agile space, PMI's new ACP certification can be expected to challenge their position and eventually dominate the landscape over the next few years.

Agile Planning and Estimating

In order to compare traditional planning and estimating to agile, we have to first understand the assumptions that underpin each method.

In the traditional world of project management represented by the PMBOK®, the first assumption is that scope can and should be defined at the very beginning of the project (Figure 2.3). Although some evidence challenges the validity of assuming that scope can be accurately defined at the beginning of a project, it continues to be the starting assumption for traditional project management. PMs who have worked on large, complex projects have experienced change order process controls, change management boards, and any number of other tactics deployed to manage and control changes in scope. Despite all the effort to manage scope changes, it often proves futile. Nonetheless, **traditional project management starts with an assumption that well-defined scope is possible.**

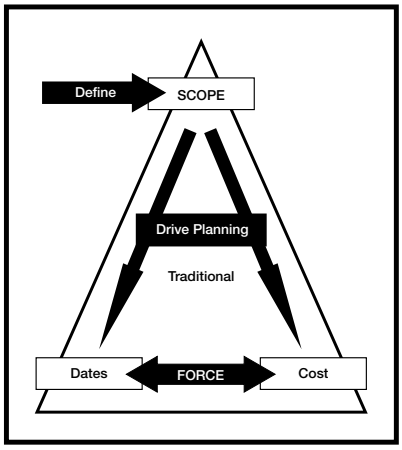

Figure 2.3 Traditional PM Assumptions.

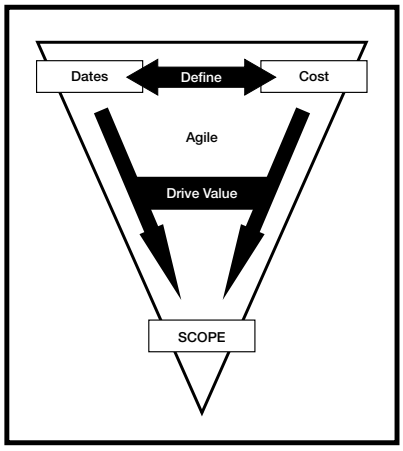

Figure 2.4 Agile PM Assumptions.

This first assumption of well-defined scope drives the next assumption that desired dates and cost constraints can be honored. Once again, PMs who have worked on large complex projects often report the futility experienced when trying to get dates and costs to conform to a plan.

By comparison, the **first assumption for agile** methodologies is that the **customer clearly knows the date they wish to receive the solution and also the cost constraints that must be observed** (Figure 2.4). Because the customer knows these two pieces of data *firmly,* agile methodologies use dates and costs as the starting point for planning and estimating. Agile then

proceeds based on the assumption that rigorous communication with the customer will drive value by refining project scope as development makes the options, and their costs, clearer.

Working with the customer, the **agile team strives to drive value by prioritizing the most important aspects of the project scope and developing them first.** In agile, planning and estimating is focused on creating accurate estimates that are reliable because they are within an appropriate, manageable, near-term time horizon. Beyond the near term horizon, agile estimating and planning focuses on avoiding the expensive illusion of false precision. Instead it uses tools, techniques, and tactics that provide robust, reliable planning at an intelligent, appropriate cost. The agile tools, techniques, and tactics used to do so will be covered in greater detail in subsequent chapters.

Before we move on, it is worthwhile to point out, as many traditional project manager's would, that anyone who has worked on large complex projects knows that having an available, involved, and rational customer may be more elusive than trying to get dates and costs to conform to a plan that reflects changing external realities.

Agile Execution and Control

Execution and control in Agile Project Management relies on the use of **timeboxes** and **feedback cycles.** There are several types of timeboxes employed in Agile Project Management.

The highest level timebox is referred to as a ***roadmap.*** An agile roadmap is most equivalent to a program plan in the traditional project management world. Roadmaps are composed of release plans, the next lower level timebox in Agile Project Management. The size of the timebox represented by a roadmap is the sum of the release plans within the roadmap (See Figure 2.5).

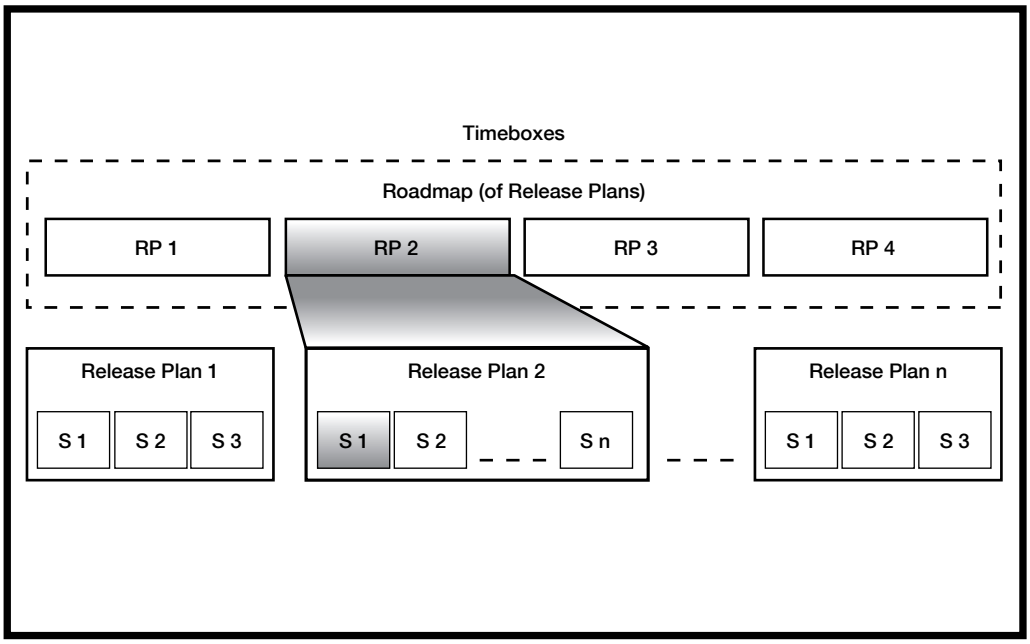

Figure 2.5 Agile Roadmap of Release Plans.

A ***release plan*** is a timebox equivalent to a project schedule in the traditional project management world. Release plans identify specific feature sets that represent a recognizable, logical component of the overall solution. Quite often, release plans represent the point at which deliverables can be used or implemented by customers. Release plans are composed of iteration plans. The size of the timebox represented by a release plan is the sum of the iterations within that release.

An ***iteration plan*** is the third timebox. Iteration plans are unique because they are a combination of a timebox and detailed work effort descriptions (See Figure 2.6). Each iteration contains the ***user stories,*** which describe the work effort for specific features or components that will be created by the agile team. Within iteration plans, user stories are decomposed into tasks, which can be estimated for the amount of work required to complete them. Iteration plans define the work that will be done in that specific timebox. Iteration plans are rolled up into release plans.

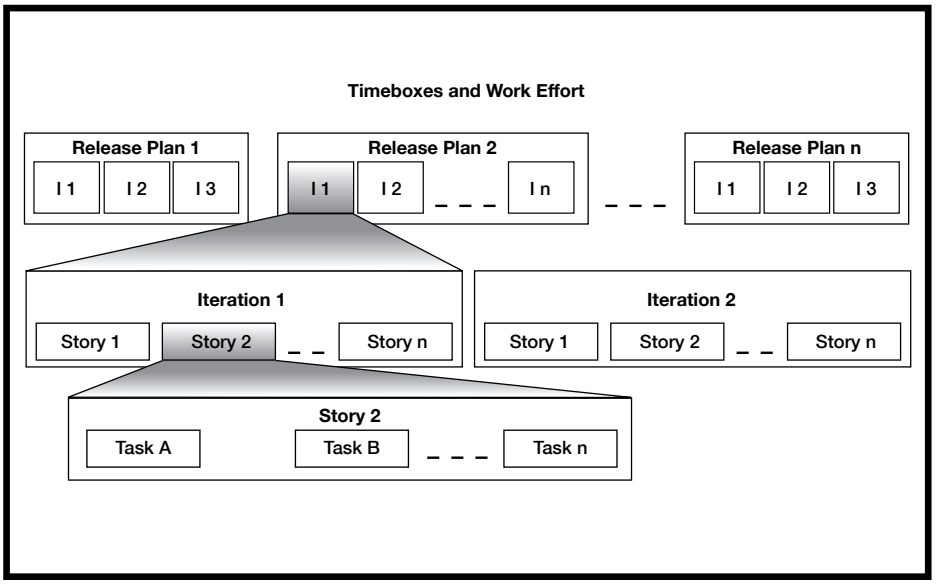

Figure 2.6 The Iteration (or Sprint) Plan.

By definition, the size of the timebox for an iteration is stable. Based on organizational norms or rules, an iteration timebox will typically be defined as either two, three, or four weeks. Once the timebox for iterations has been defined, it remains fixed because rhythm helps the team increase speed, while stability helps the team improve quality over time.

Feedback cycles occur at several distinct points in the process.

One of those points is the ***daily meeting*** for the agile team. The daily meeting is sometimes referred to as a stand-up meeting or a Scrum meeting and is held to synchronize the activities of all of the team members. It also allows measurement of work progress against the iteration plan.

Another of those feedback points is the iteration ***review meeting,*** which occurs at the end of each iteration timebox and is **product** centric. At the end of each iteration, the agile team presents the completed deliverables to all interested stakeholders. This allows the stakeholders to see the most recent work product of the team and give feedback on how well it meets their needs and expectations. It provides transparency between the

stakeholders' needs and the agile team's work, allowing adaptation to occur while it is easier and less expensive to make changes.

The third feedback point is the iteration retrospective meeting. The retrospective also occurs at the end of each iteration and is process centric. During the retrospective meeting, the agile team, Scrum Master, and Product Owner discuss process improvement ideas. It provides an opportunity for all members of the team to identify what changes would produce better work products, reduce errors, or improve communications.

The Origins of Agile Project Management

History of Lean Systems Thinking

Although arguments can be made that rigorous process thinking went into building the pyramids and many other significant accomplishments of antiquity, Henry Ford and his team of engineers – in particular Henry Gantt – are considered the first to truly integrate an entire production process. Around 1913 they integrated the ideas of interchangeable parts, standardized work units, and automated conveyors in order to produce what they described as production flow. The dramatic productivity gains made with a moving assembly line were revolutionary.

The revolution went beyond mere productivity gains and created insight into the theoretical foundations of mass-production manufacturing. Ford used process sequencing, single-purpose machines, and control-gate decision points to deliver required parts and sub-assemblies directly to the assembly line. Each of these ideas was a revolutionary break from prior standard practices of process grouping, general purpose machines, and batch-production decision points, which delivered parts still needing subassembly to the assembly line.

However, there were two significant problems with Ford's approach. First, the production system could not accommodate variability and second, the manufacturing machines could not handle complexity. This was epitomized in Henry Ford's, now infamous, statement that customers could have a Model T in any color they wanted…as long as they wanted black! But color wasn't the only limitation. All Model T chassis had to be essentially identical even though customers could choose one of four body styles. Because the single purpose machines only worked on a single part, changes were not possible and the model cycle for the Model T ended up being longer than 15 years. In the end Ford's system, while much more efficient than his competitors, lost out to other automakers who responded with many models and many options for each model.

The production systems of Ford's competitors handled variability and complexity, but at the cost of much longer throughput times. Their larger, faster machines lowered costs per unit but continually increased throughput times and inventories. Compounding the problem, the lag time between process steps, because of the complex routing of parts, required immense management effort that eventually spawned the computerized Materials Requirements Planning (MRP) systems that have become common.

In the midst of this manufacturing milieu, Sakichi Toyoda, founder of Toyota, and his son Kiichiro Toyoda, worked to build upon what the Ford had done using ideas from W. Edwards Deming.[4] Although they were unimpressed when they observed an American mass production assembly line, they were struck by an idea while shopping in a supermarket. They observed the simple idea where a customer took whatever soda they wanted and it was automatically replenished to await the next time a customer decided to take one.

As the Toyodas and Taiichi Ohno studied this situation, it occurred to them that a series of simple innovations might make it possible to provide both continuity in process flow and a wide variety in product offerings.[5] The result was the Toyota Production System (TPS).

The revolutionary idea within the Toyota Production System was that the focus should be shifted from individual machines and their utilization, to optimizing the product flow through the whole manufacturing process. Using concepts that aligned parts and sub-assembly production to the actual volume needed, applied self-monitored quality, and integrated process sequencing with quick changeovers, TPS created a system where each step "requested" materials from the previous upstream step to meet current needs. The outcome was low cost, high variety, high quality, and very short throughput times, which allowed them to respond to changing customer desires. As an added benefit, managing the immense MRP systems became much simpler and more accurate.

Today the ideas developed by W. Edwards Deming and the Toyota Production System are generally referred to as ***"Lean."*** The concepts of Lean were first described by James Womack in the book *The Machine That Changed the World.*[6] A few years later, Womack and Daniel T. Jones defined the five core Lean principles in their classic Lean Thinking.[7]

The Five Core Lean Principles are:

1. Define the value the customer desires.
2. For each product, identify the value stream that provides customer value and challenge all of the wasted steps not directly providing it.
3. After removing the wasted steps, make the remaining value-added steps flow continuously through to the product.
4. Wherever possible, use "pulling" between steps to create continuous flow.
5. Continuously move toward perfection by reducing the number of steps, and the amount of time and information needed, to provide the customer value.

Because these five principles provide the theoretical foundation that influenced Agile Project Management, they are important to know and remember.

The terms pushing and pulling are Lean manufacturing concepts. ***Pushing*** signifies a "make to stock" (MTS) supply chain philosophy where production is not based on actual demand. Pulling is a "make to order" (MTO) approach where production is based on actual demand.

Lean thinking has spread through every industry, and nearly every country, causing leaders to adapt the tools and principles from manufacturing into services, healthcare, construction, and even charitable, institutional, and government settings. But Lean has only begun to influence senior managers and leaders compared to what the future will likely hold as time-to-market becomes a critical competitive differentiator.

The Lean principles summarized above have evolved into a set of core beliefs that should also be well understood in preparation for the ACP exam. Those core beliefs can be articulated as:

- The measure of success for any system or process is the amount of time between when ideas come in and when value is received by the customer.
- Any ad hoc system or process will produce unacceptable delay in customer value because it cannot be studied or improved upon. Therefore, processes must be defined in order to improve customer value.

- Most process errors are caused by the system, not the people who work in the system. Therefore, the people doing the work are the best qualified to define how to improve the system.
- The goal is to optimize the whole system, not merely improve individual steps. Therefore, optimizing the whole system or process by looking at when steps occur is a better path to improvement than trying to optimize the efficiency of each step.
- Because the goal is to optimize the whole system and because the people doing the work are the best qualified to improve the system, management must work with the team in order for the system to improve.
- Teams, as well as systems, have inherent capacity limits that cannot be violated without subverting quality and sustainability. Therefore, teams are most efficient when the amount of work expected is within their capacity and efficiency is best improved by minimizing the amount of non-value or low-value work in process at any time.

These core beliefs create an agile paradigm where managers and teams work together toward the goal of maximum customer value. That fact is true whether the Lean principles are applied to software development, healthcare delivery, professional service delivery or any other field.

Application of Lean Thinking to Project Management

Proponents of traditional project management cite its success in the fields of engineering and construction as an indicator of its applicability to fields like software development. Since teams take requirements and build products that customers can use (not unlike engineering, construction, or other fields of product/service development and delivery), the theory implies traditional project management should work well. There are a number of problems with this thought process.

First, unlike construction where detailed blueprints are available before construction begins, or engineering where models or algorithms are available to specify specific processes before manufacturing begins, software development usually starts without clearly defined requirements or models that hold the rules for the complex variables that are involved.

Second, the immense variability and complexity of developing software makes the challenges faced by Ford pale in comparison. The variability is driven by the constraint of being human, which is to say we cannot perfectly perceive the best solution to a complex problem without going through incremental stages of development. The complexity is fueled by the wide range of variables, such as situational context and multiple platforms (i.e., PC, web, and mobile), which the solution must handle.

However, success in the software world – with all of its immense variability and complexity – has proven that Lean systems thinking, as embodied in Agile Project Management, can clarify what solution is needed and the process of discovery that can produce it. **Because the end goal is delivering value to a customer, Lean and agile processes are applicable to a great many fields outside software development.**

Applying the basic principles from Lean Manufacturing to project management requires the practitioner to **accept the idea that fast, flexible flow in the development process** – sometimes called the **development pipeline** – is possible. Many new practitioners have great skepticism about the whole idea that their specific industry could be modeled or managed as a fast, flexible pipeline. They discount that a value stream could be mapped or that mapping and refining it using Lean concepts as a guide would bear any useful benefits.

In order to help you prepare for the ACP exam, as well as prepare to apply Lean concepts in the workplace, we will now describe the Agile Project Management practices that are implied in Lean thinking. As we do this, you will see how agile practices have grown out of Lean principles. The reason for developing this understanding is twofold; first, when a question on the ACP exam describes a situation where an agile practitioner has found himself in a situation where standard agile practices won't work, you can use the Lean principles to guide you to the right answer and second, the development of this understanding will lead to better responses to workplace challenges.

Agile Project Management begins with the Lean concept that creating a sustainable stream of products requires directing business resources and focusing development teams so that results are based on **prioritized business needs which are defined to create customer value.** That, in turn, requires focusing on speeding up time-to-market by removing delays in the development process.

Another important concept implied by Lean, but made an explicit goal in Agile Project Management, is to **improve communication.** Quantitatively, improved communication reduces risk while also improving quality and dramatically increasing the likelihood of achieving real customer value.

Unfortunately, Agile Project Management practices tend to focus on communication at the local level – within the team, with the customer, and to a lesser degree, between multiple teams. Current Agile Project Management practices offer only limited support for improving communication across the enterprise or across the entire value stream. This weakness is being addressed by new practices in agile program and portfolio management.

Agile Project Management also embraces the Lean concepts of **deferring commitment** and **eliminating waste** as good ideas. Many practitioners are distracted by those word choices and fail to consider the evidence before making a judgment. But, as we mentioned earlier, when one considers the high project failure rates documented by the Standish Group in the CHAOS Reports and their research that showed 64 percent of the features being delivered to users fall into the "Rarely" or "Never Used" categories, the only viable conclusion is that when the Lean concepts of eliminating waste and deferring commitment are properly understood, they create value.

In Lean, and therefore also in Agile Project Management, ***deferring commitment*** means that decisions are made at the **right time,** sometimes referred to as the "last responsible moment." Although this idea is counter-intuitive to many project managers because traditional project management has spent decades developing massive specifications at the beginning of a project, it is entirely sensible. Ask any experienced project manager whether it is better to (a) plan and estimate a project when very little is known about the problem or solution, or (b) plan and estimate a project when good information is known, and they will laugh at you because the answer is so obvious! Millions of change orders – change orders that could have been avoided – also validate the sensibility of this concept.

The concept is to **resist making decisions too early,** when needed information is not available, simply to create a sense of security or precision that will often turn out to be false. Conversely, the concept also warns against making decisions too late and incurring avoidable, higher costs, which usually occurs because the decision maker was too risk averse and wanted more accurate information used in the estimate.

The financial rationale for deferring commitments is quite straightforward. It is well known that with the application of additional effort – that is, time spent by resources that are usually expensive

and in short supply – the accuracy of any estimate can be increased. As shown in Figure 2.7, 10 hours spent on an estimate might create 80% accuracy, while spending 20 hours might improve the accuracy to 90%. The problem is that a 12% improvement in accuracy (i.e. (90% - 80%)/80%), has required a 100% increase in cost, from 10 to 20 hours.

If that wasn't onerous enough, consider the fact that the **value of any estimate decreases with the passage of time.** Estimates are most valuable early in the project when little is known, and least valuable late in the project when much is known. So the focus in **Agile Project Management is to produce detailed estimates only when enough is known to get useful accuracy at a reasonable cost.** In Agile Project Management deferring commitment can be readily applied in defining requirements and doing analysis and estimating.

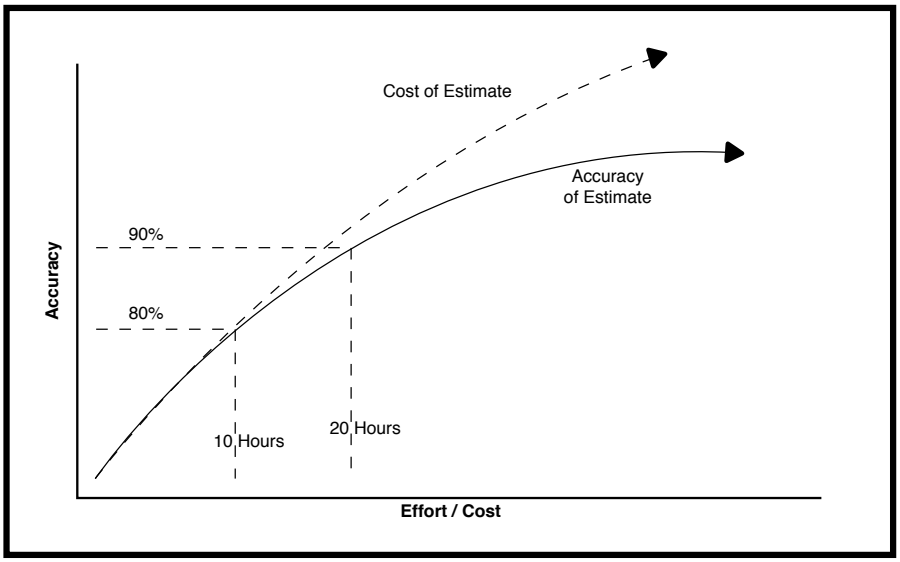

Figure 2.7 Accuracy Versus Cost of Estimate.

The goal of defining requirements and creating estimates should be to prioritize where resources are invested. If we simply stop to consider whether it is really necessary to define every single customer requirement, the answer should be a clear, "No." What we need is great clarity about the requirements that will impact development resources on a reasonable time horizon – 60, 90, or 120 days.

Some requirements are more important, more urgent, or more technically complex than others. As a guide, **Agile Project Management should start with those requirements that are the most important to the customer, involve safety, and create technical risk for things like scalability.** Then, once those priorities are defined, move to requirements that improve marketability, performance, and flexibility. Finally, focus should shift to requirements that leverage opportunity or create comfort and luxury.

Recognize that commitments cost money because they spend time doing some kind of work, which can never be restored or the cost undone. Therefore, commitments should be spent doing work on the requirements that will bring the greatest value to the customer. Agile Project Management directs resources to the requirements that customers define as most important.

As we have shown, **Agile Project Management embraces deferring commitment.** Now, let's see why the concept of eliminating waste is a good idea.

Most of the projects being managed today involve a considerable degree of architectural or technical risk, with risk being a good indicator of potential waste. After all, few things are as wasteful as building the wrong thing or building something no one will ever use. Therefore, **eliminating waste has primary importance** as a guideline for the Agile Project Management practitioner.

Waste comes in many forms. In software, it is code that is more complex than needed, causing undue defects and creating extra quality control work. In manufacturing, it is non-value-added work spent to create a product. In other contexts, waste is unneeded paperwork or documentation, or missing paperwork or documentation that creates errors or rework, or a failure to create clarity that would have increased the speed at which the deliverable could have been created. Wherever waste is, Agile Project Management seeks ways to improve the system and eliminate it, because it is likely that errors will be repeated until the system that caused it is fixed.

Lean asserts that the most common and perhaps largest waste in traditional project management is the effort spent on detailed planning done too early in the project. To consider this idea from a Lean point of view, ask yourself, "When is it best to estimate, when little is known about the problem and solution domains or when much is known?" The answer is so obvious that when we ask that question in class, many students are hesitant to answer because they suspect it is a trick question.

Most project managers would acknowledge, especially on large, long projects, that accurate information is the least available during the early stages. Customers often have only a vague notion of how to describe the best solution. They often use the phrase, "I'll know it when I see it," which is referred to as the "IKIWISI syndrome" (pronounced icky-whizzy) to express their lack of clarity. Yet, this is when traditional project management often produces detailed requirements documents, very specific contract language, and detailed project plans.

Agile Project Management avoids this type of waste using a technique called emergent design. **Emergent design limits resource commitments and costs to those features that are currently necessary.** A comparison of the two concepts is shown in Figure 2.8, where a traditional waterfall approach that begins with a large effort to define everything and ever smaller efforts spent on elaboration over time, is contrasted with an agile approach, which only elaborates those things that are needed for reasonable clarity on a practical time horizon, such as 60, 90, or 120 days.

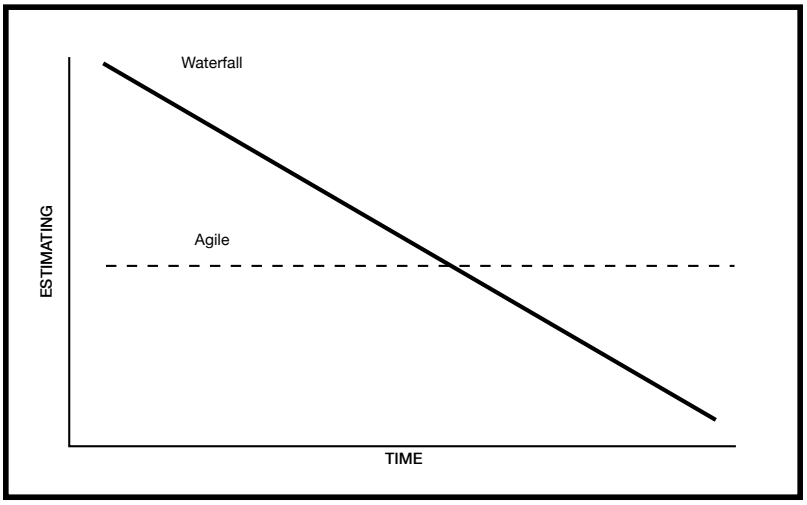

Figure 2.8 Estimating Effort/Resources over Time.

Emergent design has a dual meaning. It means that results emerge from an internal creative work process rather than being the result of an external blueprint. It also means that the design artifactsare more than the sum of the parts or a permutation of existing factors cobbled together by a team. It implies that the design process is creative and cannot be done by rote or by accident so it must be intentional.

When emergent design is applied in software development, it also integrates the discipline of ***design patterns*** to create application architecture that is durable and flexible, as well as automated acceptance and unit testing, to improve code clarity and reduce defects. **The use of design patterns enables code to more easily adapt to change by reducing complexity.** Reduced complexity is achieved by limiting coding to defined, current needs. Automated testing validates the design pattern, making the code safer to change should the need arise. Therefore, emergent design and automated testing combine to enable the deferral of commitments until implementation variables are reasonably understood.

A central tenet of Agile Project Management is that knowledge about the context of the problem and the variables of its solution are discovered and created as an integral part of the development process. **Agile Project Management solutions are created, built, or developed in stages so that as the customer uncovers specific real needs (remember IKIWISI?) the team can design and write code accordingly.** The analogy below may help clarify your understanding.

According to a recent article, at any given moment there are 500,000 passengers riding in airplanes over the United States. Due to factors like headwinds and crosswinds and collision avoidance routing, each of those flights is off course more than 90% of the time. The reason the vast number of those flights – or projects – are completed as intended is because the pilot uses accurate, transparent, real-time data to make minor course corrections as needed.

In Agile Project Management, the person who is the ***voice of the customer,*** called the ***Product Owner*** in the Scrum lexicon and ***customer/proxy*** in this book, is continuously taking in stakeholder feedback data and using it to provide the team with minor course corrections.

By doing Agile Project Management this way, the team delivers value quickly and avoids building things of little (or no) value. Remember that creating customer value is more of a discovery process than a building process. Software, buildings or medical devices have little inherent value. Value occurs when any of these enable the delivery of a product or service that solves a customer problem. Therefore, it may be more useful to think of product or service development – whether the deliverables are tangible or intangible – as a set of activities used to uncover the real needs, and the real problems, of customers and furthering the strategic goals of the organization by addressing them.

Applying Lean to Agile Project Management implies accepting the mindset that it is necessary to deliver increments of the solution, early and often, so that the customer can experience specific aspects of the solution and reduce IKIWISI at each stage.

Delivering increments early and often requires development to be done in iterations, which is referred to as ***iterative development.***

The financial reason for doing iterative development is that customer value can be realized more quickly. Doing so improves market penetration, generates greater credibility for the business, creates

strong customer loyalty, and increases profit margins. It also typically allows revenue streams to begin sooner, which, in turn, offsets the cost of subsequent development and reduces the total capital commitment required, directly increasing return on investment (ROI).

The financial benefits of Agile Project Management demonstrate the value of Lean's focus on time and timing. Time is one of the core focuses in a Lean approach. Instead of the traditional project management focus on resource utilization – which has driven matrix-type organizational structures and multitasking of workers – **Lean zeroes in on reducing the elapsed time from idea generation to delivery of value to the customer.** Of course, since time is money, when the team goes faster by using an improved system or process, costs go down.

Agile Project Management also uses the Lean focus on time and timing to reduce risk, in part, by eliminating delays that create waste. For example, some common delays that create waste in software are: requirements waiting to be verified as correct, work stoppages because a clarification is needed from a customer or analyst, and waiting for code that has been written, but needs to be tested. In healthcare, some common examples of waste are: delays experienced when a patient is waiting for insurance coverage to be verified as correct, when a pharmacy can't dispense a prescription because a clarification is needed from the doctor, and when a pharmaceutical has passed testing, but is waiting to be approved by a regulator. These delays represent both risk and waste because delays increase the likelihood that something will be misunderstood and, in turn, multiply the potential of something going wrong.

By using iterative development steps, Agile Project Management creates the ability to make minor changes that move in the direction of the real solution without wasting effort. Borrowing from the Lean Manufacturing vocabulary, agile seeks to **minimize work-in-process (WIP).**

For Agile Project Management, WIP means those things that are described as 60% done or 80% done, or some other percent done in a traditional project status meeting. Because the customer cannot reduce IKIWISI and progress toward the real solution cannot be accurately measured, WIP has no value even though it has cost.

Whereas a traditional project may spends months or years going from 20% done to 30% done to 90% done, it isn't until it is 100% done that the customer can truly ascertain if real value has been created. There are myriad examples of projects that accumulated astronomical costs while in a state of WIP only to be judged by the customer as having no value when the deliverables were finished. In those cases, the work done by the team has gone directly from WIP to waste, and despite elaborate risk management protocols, potential risk germinated into very real problems.

One well known example of this was Motorola's multibillion-dollar venture into satellite-based phones – the Iridium project. While some might argue that Iridium was a success from a technical perspective, despite remarkable financial forecasts and intense project management, the entire venture turned out to be a financial debacle because it failed to deliver customer value.

This could have been avoided by using Agile Project Management's iterative development process, where the customer is given something at the end of each iteration that can be used, seen, applied or sampled, in order to produce clarity about needed course adjustments. The impact is organic risk mitigation and systematic value generation.

The Agile Process Map™ & Agile PM Processes Grid™

Agile Process Map™

Now that you have been exposed to a significant part of the agile lexicon, it is time to put those ideas together in an Agile Process Map™ that can be used to guide and integrate the detailed learning in the rest of this course.

As you may have gathered from what you have learned so far, the agile worldview centers around the team's work as the point of value creation. So from the team-centric view, the macro perspective is a process that moves from a "steady state" to a "transition state" to a "steady state."

The first steady state encompasses the activities of the customer/proxy, or in Scrum vernacular the Product Owner (PO), who is receiving, analyzing, and prioritizing the features required for a successful solution. That work is kept in an artifact referred to in Scrum as the ***product backlog.*** The Product Backlog is equivalent to the product specification or requirements list in traditional project management. It is, however, significantly different because the PO is continuously ***grooming*** it based on information being received from internal and external sources. As priorities change, system features can be promoted or demoted. As the project moves forward, features that were on the future horizon enter the current horizon and are analyzed and estimated. So the Product Backlog is in a state of flux, but from the team's perspective it is in a steady state. That is because the team interacts with the Product Backlog only at the beginning of each iteration in order to negotiate with the PO and decide which features will be included in the next iteration. Once those features are agreed upon and fully committed to, they cannot be changed. This is shown in Figure 2.14.

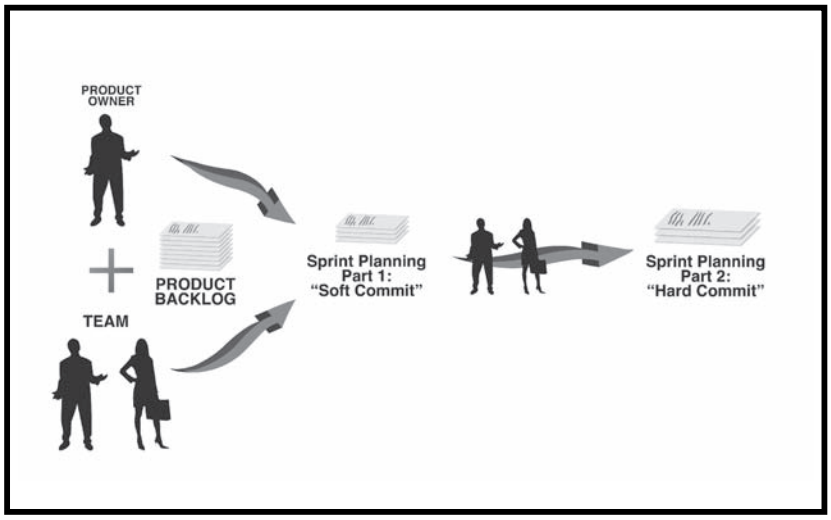

Figure 2.14 Steady State #1.

So for the PO, the backlog is in a state of transition, but for the team that flux is outside their concern. Once the team has fully committed to the ***Iteration Backlog,*** which is the portion of the Product Backlog being developed in the current iteration, enters a state of transition. The team has committed to doing whatever is necessary to change the current state of those features into the future state as the goal of that iteration. At this point, conversely, for the PO the iteration (or in Scrum vernacular sprint) backlog is in a steady state and may not be changed.

The desired future state of those features is, by definition, the potentially shippable product increment, which is the goal of that iteration. The team will focus all of its energy on building that part of the solution or system and at the end of the iteration, will demonstrate for all interested stakeholders what has been created at a review meeting. The **review meeting** is a product-centric meeting where acceptance of the deliverables and considerations for future enhancements is discussed. As far as the team is concerned, once they have demonstrated that the features work as agreed upon at the beginning of the iteration, the second steady state has been achieved. This is illustrated in Figure 2.15.

Between those two steady states is the state of transition, which is where the agile team lives its life. In the state of transition, there are only two constants – the duration and the goal of the iteration. Everything else is in a constant state of change, **driving the need for the team's daily meeting to synchronize and plan.** Each day when the team meets, each member briefly explains what they have done since the last meeting, what they will do before the next meeting, and any impediments interfering with their ability to be effective and productive. The team will use that information to self-manage. They will re-plan as needed, synchronize hand-offs, rally to support one another, and also hold one another accountable. Each day, each member is expected to make reasonable progress towards the fully committed, agreed-upon iteration goal. As needed, the team will also meet with the PO to clarify questions or concerns about elements, components or behaviors of the system in order to make sure it will meet the ***definition of done*** that the PO supplied at the beginning. Sometimes discoveries or insights will come out of the daily meeting. Those discoveries and insights are forwarded to the PO for use in grooming the Product Backlog.

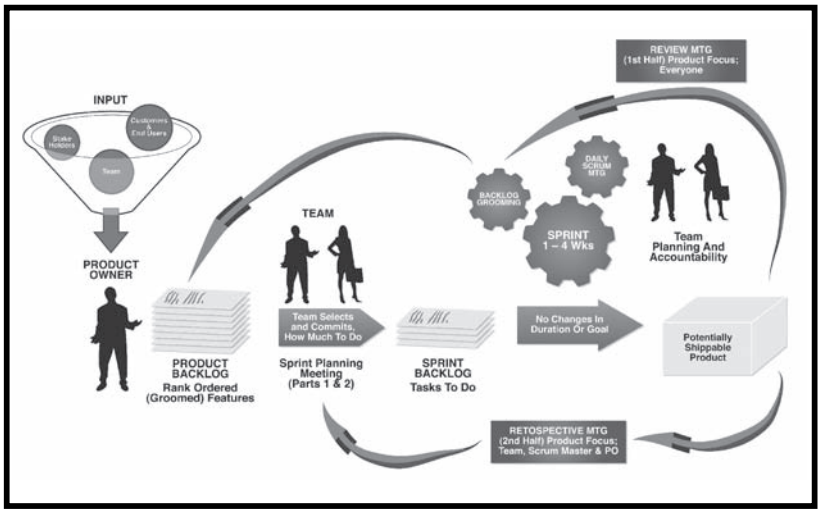

Figure 2.15 The Agile Process Map™ Shows the Second Steady State.

By having daily results that are measurable, external control is unnecessary because internal self-discipline has been created.

The final part of the process is the **retrospective meeting.** The retrospective meeting is attended by the team – and sometimes the PO – only. It is a process-centric meeting where the team identifies how it can improve its process of creating potentially shippable product increments. Typically, the review meeting and the retrospective meeting are the first and second halves of a single meeting for the team.

Agile Project Management Processes Grid™ for the ACP Exam

In 1986 PMI produced the first draft of the world changing *A Guide to the Project Management Body of Knowledge.* Throughout the ensuing 25 years, project management professionals have had to master the dreaded PMBOK® Framework grid in order to survive the rite of passage called the PMP Exam. Project management may not be rocket science, but that's not because the PMP exam is easier!

In 2010, PMI took another historic step by announcing that it would embrace the agile methodology as a source of credible, valuable, and necessary frameworks and tools for project management professionals to have in their repertoire of skills. In 2011, PMI announced a formidable process for demonstrating the acquisition of those skills with a new path to certification that includes the (some would say, soon to be dreaded) ACP Exam.

As mentioned earlier, our purpose in writing this book was twofold. Our first focus was to help students adequately and efficiently prepare for the ACP exam. Secondly, we knew students would benefit from having a Desk Reference to support their work handling daily challenges of being an agile PM. In order to fulfill the first requirement of passing the exam, we created the Agile Project Management Processes Grid™, which is also a useful reference tool while you are working.

It is with great pride that we introduce the ***Agile Project Management Processes Grid™!***

Using the Agile Project Management Processes Grid™

As we mentioned earlier, one of the best test preparation exercises a student can do is to practice reproducing the Agile PM Processes Grid™ from memory. It gives students a chance to see how much content they have retained, and most importantly can process from memory onto a blank sheet of paper. It is an excellent self-assessment of what you have retained to that point from studying.

Because the testing center will provide several blank sheets of paper or a small whiteboard with dry marker and eraser, anything the student can quickly reproduce or brain dump is an aid in passing the test as stress builds up. At the end of each chapter, the student will be challenged to take 3 minutes and see how much of the grid they can reproduce from memory. When the 3 minute time limit expires, use the grid above to correct what you did and study specific additional cells to add to it the next time.

It is an exercise well suited to repeating and practicing over lunch, during breaks, and periodically throughout the day. Doing so extends your study time and reinforces your learning.

Memorizing the Agile PM Processes Grid™ can be immensely simplified by using an age old practice called mnemonics. ***Mnemonics*** are a technique for assisting human recall by using or similar device. The grid has processes and areas of knowledge and skill that identify the columns and rows. To apply a mnemonic device a student takes the first letter of each process name and creates a sentence where each word starts with the same letter. The trick is to make it as visual and memorable as possible so weird and wild is helpful.

Knowledge & Skills Areas	Agile Project Management Processes Grid™				
	Initiate	**Plan**	**Iterate**	**Control**	**Close**
External Stakeholders Engagement	• Stakeholders Identification • Vision Statement • Project Data Sheet • Active Listening	• Product Roadmap • Minimally Marketable Feature (MMF) • Prioritization	• Product Backlog Grooming	• Product Demonstrations	• Deliverables Acceptance
Value-Driven Delivery	• Value Analysis • Business Case • Contracts	• Release / Iteration Plans • Planning Activities • Decomposition • Progressive Elaboration	• Cycle Time Measurement • Work-in-Process (WIP) Limits • Cumulative Flow Diagrams	• Product Feedback • Accounting and Contracting Control • Earned Value Mgt. (EVM)	• Product Release
Adaptive Planning	• Team Acquisition • Project Kick-off Meeting • Incremental Delivery • Time Boxing	• User Stories • Iteration Backlog • Definition of Done • Estimation • Sizing • Wideband Delphi • Planning Poker • Story Points • Ideal Days • Affinity Estimates	• Burn Down Charts • Task / Kanban Boards • Test-driven Practices • Agile Modeling • Wireframes	• Information Radiators Monitoring	
Team Performance	• Coach Recruiting • Servant / Adaptive Leadership • Emotional Intelligence	• Coaching / Facilitation • Collaboration / Negotiation • Motivation / Empowerment	• Coaching / Mentoring • Conflict Resolution	• Task Board / Burn Down Charts Updates • Velocity	• Team Evaluations • Performance Incentives • Self assessment
Risk Management	• Organizational Practices • Regulatory Discovery • Quality Standards	• Risk-adjusted Backlog • Regulatory Compliance	• Problem Solving • Continuous Integration • Risk-based Spike • Risk Burn Down Charts • Verification and Validation	• Obstacle Removal • Variance and Trend Analysis • Escaped Defects	
Communi-cation	• Colocated / Distributed • Participatory Decision Making	• Communication Protocols • Information Radiator • Team Space • Agile Tooling	• Daily Stand-up, Iteration Review, and Team Retrospectives • Osmotic Communication	• Knowledge Sharing	• Retrospec-tives (Project, Release, and Iteration Lev-els)
Continuous Improvement	• Identify Agile Ceremonies	• Value Stream Mapping • Cross-functional Team Formation • Metric Definition	• Metric Tracking	• Process Analysis	• Process Tailoring

Figure 2.16 Agile Project Management Processes Grid™

For example, the processes start with the letters I, P, I, C, and C so five possible mnemonics are:

- **I Prefer Ice Cold Cheese!**
- **I Plan Intently to Control Costs**
- **I Pondered Inscrutable Clairvoyant Clues**
- **I Prefer Ingesting Cold Chivas!**
- **Intoxicated Pink Iguanas Came Calling**

The knowledge areas can be represented with the letters S (for stakeholders), V, A, T, R, C, and I (for improvement) so five possible mnemonics are:

- Students Variously Attempted To Recall Critical Information
- Stately Venetian Attorneys Tried Recalling Circus Invocations
- Super Venomous Angels Tortured Recalcitrant Crowned Idiots
- Saturated Vermin Attempted To Reverse Course Instinctively
- Simply Venturing Abroad Teaches Reasonable Civilians Insight

Two common mistakes need to be pointed out. First is not choosing a single, specific mnemonic and committing it to memory. It is counterproductive to try to use multiple mnemonics because it creates mental clutter and confusion. Make a decision, pick one, and then keep playing with it in your mind until it sticks like the Disney song "It's a Small World After All." Second is making it mundane. The mnemonic needs to be visually wild or better yet a crazy mental cartoon with motion and sound.

With your mnemonic ready, you draw 5 vertical lines, and 7 horizontal lines to divide up the paper, then write the first letter at the top of each column and the beginning of each row, and begin filling in everything you can remember. With each cycle your recollection will grow stronger!

Chapter Close-Out

Agile PM Processes Grid™ Exercise

Please take out a blank piece of paper, set a timer for no more than 3 minutes, and see how much of the grid you can reproduce from memory. To make the most of this Agile PM Processes Grid™ exercise, please simulate being in the testing environment. Close your book and all your notes. Visualize the Proctor handing you the blank sheets of paper and taking your seat in the testing site. Begin by drawing the grid, 6 columns and 8 rows, and then fill in everything you can. After the 3 minutes ends, use your book and notes to complete the grid. Study it as you do so.