



## Chapter 1

# The Agile Revolution

## Innovative Product Development

Product development teams are facing a quiet revolution in which both engineers and managers are struggling to adjust. In industry after industry—pharmaceuticals, software, automobiles, integrated circuits—customer demands for continuous innovation and the plunging cost of experimentation are signaling a massive switch from anticipatory to adaptive styles of development. This switch plays havoc with engineers, project managers, and executives who are still operating with anticipatory, prescriptive mindsets and processes geared to a rapidly disappearing era.

*Symyx creates and operates highly integrated, complete workflows that enable scientists to explore their ideas to discover and optimize new materials hundreds to thousands times faster than traditional research methods. These workflows consist of robotics that synthesize arrays of materials on a miniaturized scale, creating hundreds to thousands of tiny experiments on one silicon chip. These materials are then rapidly screened in parallel for desired physical and functional properties, including chemical, thermal, optical, electronic, or mechanical attributes. The results are captured in database systems for mining large sets of data to help scientists make well-informed decisions on the next steps of the discovery process.<sup>1</sup>*

---

1. Quote courtesy of Symyx Technologies, Inc., [www.symyx.com](http://www.symyx.com).



Symyx boasts 100 times the speed at 1% of the cost of traditional research. Drug companies used to *design* compounds for specific purposes by having scientists pore over how to make just the right one. Today they generate tens of thousands of compounds and then test them quickly using ultra-sophisticated, ultra-speedy tools such as mass spectrometers. *There are new product development economics at work here.*

In mid-2002, when Alias Systems of Toronto, Canada, started developing Alias Sketchbook Pro, a software package to be announced concurrently with Microsoft's launch of its Tablet PC operating system, the product management and software development team didn't begin with a lengthy product planning effort. The team's marketing and product strategy work evolved over several months, but its product development effort began early, and in parallel, with the strategy process. The team had a vision—an easy-to-use consumer-focused sketching product worthy of a professional graphics artist—and a deadline, the November Microsoft launch date. The product evolved in two-week iterations. For each iteration, a short planning session identified features to be developed. Then, within the “platform” architecture constraints of the operating system and Tablet PC computers, the product evolved—iteration by iteration. In the end, the product was delivered on time, met high quality standards, and has been a success in the marketplace. The product wasn't planned and built, it was envisioned and evolved. Alias didn't start with anticipated architectures, plans, and detailed specifications—it began with a vision followed shortly by the first iteration of the product. The product, the architecture, the plans, and the specifications evolved as the team adapted to the ever-unfolding reality of the market and the technology.

With Alias Sketchbook Pro, the team literally didn't know past the next iteration which features would be included in subsequent development iterations. Team members did have a clear product vision and a business plan. They did have a general idea about what features were needed in the product. They did have active involvement from product marketing. They did have an absolute time deadline and resource expenditure constraints. They did have an overall product platform architecture. Within this vision, business and technical constraints, and overall product release plan, they delivered tested features every two weeks and then adapted their plans to the reality of actual product testing. The team's process was one of evolution and adaptation, not planning and optimization.

In the automobile industry, Toyota employs set-based design in its design process—maintaining multiple design options on components until late in the development process. Similarly, BMW uses simulations to improve car crashworthiness. During one design effort, it ran 91 simulations and two real crashes. The results were a 30% improvement in design and 2.5 days per simulated crash versus 3.8 months (for simple tests)—and the 91 simulations cost less than the two real crashes (Thomke 2003).

All of these approaches to product development point to a very critical issue. When we reduce the cost of experimentation enough, the entire economics of how we do product development changes—it switches from a process based on anticipation (define, design, and build) to one based on adaptation (envision, explore, and adapt). When the cost of generating alternatives plunges and the cost of integrating them into a product is low, then great products aren't built, they evolve—just like biological evolution, only much, much faster than in nature. Biological evolution begins with experimentation (mutation and recombination), exploration (survival of the fittest), and refinement (producing more of the survivors). Increasingly, product development processes are being built using this analogy.

Time is also a driving factor in new product development (NPD). In the short and intense decade of the 1990s, the average new product time to market in the US dropped from 35.5 to 11 months (Wujec and Muscat 2002). “Corporations everywhere are engaged in a new products war,” says NPD expert and author Robert Cooper. “From soup to nuts, from can openers to automobiles, companies are at a competitive war with each other—and the advance troops are product development teams. On this new product battlefield, the ability to mount lightning attacks—well-planned but swift strikes—is increasingly key to success.... And mobility or speed enables lightning strikes designed to seize windows of opportunity or to catch an enemy off guard” (Cooper 2001).

But uncertainty, shrinking time schedules, and the need for iterative exploration are not restricted to new product development. New business practice implementations, such as those fostered by customer relationship management (CRM) installations, are often fraught with uncertainty of a different kind. The high rate of failures reported in CRM implementations can, in part, be attributed to anticipatory (plan-driven) project management approaches that failed to “explore” into the uncertainty caused by major business process changes. Companies tried to plan and do when they

needed to envision and explore. As authors Preston Smith and Guy Merritt (2002) write, “Innovative product development depends on exploring the uncertain to add product value and maintain competitive advantage.”

But innovation and faster development aren’t good enough. Companies have to deliver better products geared to what customers want at the time of shipment, which may or may not resemble what the team guessed they wanted when the project was initiated. Ultimate customer value is delivered at the point-of-sale, not the point-of-plan. Companies that have the ability to quickly and inexpensively evolve a product closest to the end of the development lifecycle will have a tremendous competitive advantage.

So why isn’t every company doing this? Because for most companies there is a great gap between needing and delivering new products. NPD is a multifaceted and extremely difficult challenge. The Product Development and Management Association (PDMA) estimates newly launched product failure rates of around 59%, which has changed little from the early 1990s. Also, cancelled or failed (in the market) products consumed an estimated 46% of product development resources (Cooper 2001). Yet some companies are consistently more successful than others, and a growing number of these successful companies are practicing agile development and project management methods.

The product development efforts targeted by agile methods include new products<sup>2</sup> and enhancements to products in the domains of:

- Software products (examples are PC applications, operating systems, middleware products, enterprise resource planning)
- Industrial products with embedded software (from electronics equipment to autos)
- Internally developed IT products that fit the speed, mobility, and exploration factor criteria

The key point is that opportunity, uncertainty, and risk reside in the proposed product—not in the approach to project management. Our

---

2. In Robert Cooper’s (2001) definition, new product development applies to products that have been on the market five years or less.

approach to project management needs to fit with the characteristics of the product in order to improve our chances of capitalizing on the opportunity by systematically reducing the uncertainty and mitigating the risks over the life of the project.

Companies need results from their high-pressure product development efforts, but they shouldn't come at the expense of quality. John Wooden, the legendary basketball coach of UCLA whose teams won 10 national championships, used a saying with his teams that applies to product development: "Be quick, but don't hurry." In other words, do the right things, but learn how to do them quickly. Strip away the overhead, the non-value-adding activities. Create quality products *and* do it quickly. Agile development focuses on speed, mobility, *and* quality. To accomplish this, individuals and teams must be highly disciplined—but with self-discipline rather than imposed discipline. Anyone who practices ad hoc development under the guise of agile methods is an imposter.

There is no reason to think the changes in the next ten years will be of less magnitude than those of the previous ten, although the emphasis will likely change from pure information technology to the integration of information and biotechnology. The underlying codes of information technology are zeros and ones. The underlying codes of biotechnology are A, T, C, and G (the components of DNA). When biological codes can be reduced to computer codes, as in the Human Genome Project, and then be manipulated by computer programs (as is happening), the potential impact on product development of many types is staggering. "New materials, programmed molecular factories, and self-organizing fabrication processes could change the cost and performance characteristics of everything from drugs to dragsters, paint to plastics, china to chairs" (Meyer and Davis 2003). Scientific and technological advances in the coming decade and beyond will continue to irrevocably alter product development processes, and those changes, in turn, will cause us to rethink the management of those processes.

Linear thinking, prescriptive processes, and standardized, unvarying practices are no match for today's volatile product development environment. So as product development processes swing from anticipatory to adaptive, project management must change also. It must be geared to mobility, experimentation, and speed. But first of all, it must be geared to business objectives.

## Reliable Innovation<sup>3</sup>

There are five key business objectives for a good exploration process such as Agile Project Management (APM):

1. Continuous innovation—to deliver on current customer requirements
2. Product adaptability—to deliver on future customer requirements
3. Reduced delivery schedules—to meet market windows and improve return on investment (ROI)
4. People and process adaptability—to respond rapidly to product and business change
5. Reliable results—to support business growth and profitability

### Continuous Innovation

As I discussed in the opening section, developing new products and new services in today's complex business and technology world requires a mindset that fosters innovation. Striving to deliver customer value, to create a product that meets today's customer requirements, drives this continuous innovation process. Innovative ideas aren't generated in structured, authoritarian environments but in an adaptive culture based on the principles of self-organization and self-discipline.

### Product Adaptability

No matter how prescient a person, a team, or a company, the future will always surprise us. For some products, changes in the market, technology, or specific requirements happen weekly. For other products, the timeframe for incorporating changes varies from months to years. But for every product, the timeframes are shrinking, and the only way to survive is to strive for

---

3. Colleague and Harvard Business School professor Rob Austin introduced me to the concept and term "reliable innovation."

product adaptability—a critical design criterion for a development process. In fact, in an agile project, technical excellence is measured by both delivering customer value today *and* creating an adaptable product for tomorrow. Agile technical practices focus on lowering the cost of change (adaptation) as an integral part of the development process. In an agile project, developers strive for technical excellence, and project managers champion it.

## Reduced Delivery Schedules

As the statistics for rapidly shrinking product development times indicate, reducing delivery schedules to meet market windows continues to be a high-priority business goal for managers and executives. The iterative, feature-based nature of APM contributes to reducing delivery schedules in three key ways: focus, streamlining, and skill development.

First, the constant attention to product features and their prioritization in short, iterative timeboxes forces teams (customers and developers) to carefully consider both the number of features to include in the product and the depth of those features. Constant attention reduces the overall workload by eliminating marginally beneficial features. Second, APM—like its lean development counterparts—streamlines the development process, concentrating on value-adding activities and eliminating overhead and compliance activities. Third, APM focuses on selecting and developing individuals with the right skills for the project.

## People and Process Adaptability

Just as products need to adapt to marketplace reality over time, so do people and processes. In fact, if we want adaptable products, we must first build adaptable teams—teams whose members are comfortable with change, who view change not as an obstacle to resist but as part and parcel of thriving in a dynamic business environment. The APM guiding principles and framework encourage learning and adapting as an integral part of delivering value to customers.

## Reliable Results

Production processes are designed to be repeatable, to deliver the same result time after time after time. Good production processes deliver the anticipated result (a known result), for a standard cost, within a given time—they are predictable, if you will. Exploration processes are different. Because of the uncertainty surrounding requirements and new technology, exploration projects can't deliver a known, completely pre-specified result, but they can deliver a valuable result—one that meets customer and business requirements as they become known. Good exploration processes can deliver innovation reliably—time after time. But while performance measures for production processes can be based on actual scope, cost, and schedule versus their predicted values, exploration processes need to be measured somewhat differently.

Rather than scope, cost, and schedule, exploration projects should be measured on vision, cost, and schedule. Did the project deliver a valuable product (implemented vision) to the customer? Did the project deliver the product within the cost and time constraints placed on the project? These subtle but extremely important differences between repeatability and reliability will be revisited in Chapter 3. The bottom line: APM reliably delivers innovative results to customers within cost and schedule constraints.

## Core Agile Values

Agility is more attitude than process, more environment than methodology. In 1994 authors Jim Collins and Jerry Porras (1994) wrote *Built to Last*, a book based on their research that set out to answer the question, “What makes the truly exceptional companies different from the other companies?” One of their core findings was that exceptional companies created a foundation that didn't change and strategies and practices that did: “Visionary companies distinguish their timeless core values and enduring purpose, which should never change, from their operating practices and business strategies (which should be changing constantly in response to a changing world).”

I think one reason that agile software development has grown in recognition and use during the last few years is that the founders of the movement stated explicitly what we believed in the *Manifesto for Agile Software Development*. We stated our core values and enduring purpose. Why teams exist, what we intend to build, whom we build it for, and how we work together also form the core principles of APM. If we want to build great products, we need great people. If we want to attract and keep great people, we need a great organization. The core value of an egalitarian meritocracy runs deep in the agile movement. It is surely not the only core value that can produce products, but it is a core value that defines how the majority of agilists view themselves.

We live in an age in which the volume of available information stupefies us. On any relatively interesting subject we can find thousands of Web pages, tens—if not hundreds—of books, and article after article. How do we filter all this information? How do we process all this information? Core ideology and principles provide one mechanism for processing and filtering information. They steer us in the direction of what is more, or less, important. They help us make product decisions and evaluate development practices.

Principles, or “rules” in complexity theory terminology, affect how tools and practices are implemented. Practices are how principles are acted out. Grand principles that generate no action are mere vapor. Conversely, specific practices in the absence of guiding principles are often inappropriately used. While the use of practices may vary from project team to project team, the principles are constant. Principles are the simple rules, the generative rules, of complex human adaptive systems.

The *Manifesto for Agile Software Development*<sup>4</sup> established a set of four core values, which, with a single word change, form the core values of APM:

**We are uncovering better ways of developing [products] by doing it and helping others do it. Through this work we have come to value:**

---

4. ©2001 by Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas.

*Individuals and interactions over processes and tools*  
*Working [products]<sup>5</sup> over comprehensive documentation*  
*Customer collaboration over contract negotiation*  
*Responding to change over following a plan.*

**That is, while there is value in the items on the right, we value the items on the left more.<sup>6</sup>**

“This should not be construed as indicating that tools, process, documents, contracts, or plans are *unimportant*. There is a tremendous difference between one thing being more or less important than another and being *unimportant*” (Highsmith 2000). Tools are critical to speeding development and reducing costs. Contracts are vital to initiating developer-customer relationships. Documentation aids communication. However, the items on the left are the most critical. Without skilled individuals, working products, close interactions with customers, and responsiveness to change, product delivery will be nearly impossible.

While these core value statements were originally written for agile software development, they apply directly—with a bit of interpretation, and some reordering—to APM.

## Responding to Change

*Responding to change over following a plan.* This statement reflects the agile viewpoint characterized further by:

- Envision-Explore versus Plan-Do
- Exploration versus production
- Adapting versus anticipating

Every project has knowns and unknowns, certainties and uncertainties, and therefore every project has to balance planning and changing. However, balancing is required because projects also run the gamut from production-

---

5. The Manifesto’s wording is “software.” I use the word “products” here as a more general term.

6. For an in-depth explanation of the Agile Manifesto, see (Fowler and Highsmith 2001).

style ones in which uncertainty is low, to exploration-style ones in which uncertainty is high. Exploration-style projects are characterized by a process that emphasizes envisioning and then exploring into that vision rather than detailed planning and relatively strict execution of tasks. It's not that one is right and the other wrong, but that each style is more or less applicable to a particular project type.

Another factor that impacts project management style is the cost of an iteration; that is, the cost of experimenting. Even if the need for innovation is great, high iteration costs may dictate a process with greater anticipatory work. Low-cost iterations, like those mentioned earlier, enable an adaptive style of development in which plans, architectures, and designs evolve concurrently with the actual product.

Companies trying to thrive in our turbulent economy must alter both their processes and their perspectives with respect to change. We are no longer talking about 15% to 20% scope creep on projects; we are talking about everything changing—scope, features, technology, architecture (but not vision)—within the span of a few months. I'm continually surprised by the magnitude of change products and projects undergo. The common project management aim of “conforming to plan” fails dramatically in these situations.

In *Artful Making*, Harvard Business School professor and colleague Rob Austin and his coauthor Lee Devin (2003) discuss a \$125 million IT project disaster in which the company refused to improvise and change from the detailed plan set down prior to the project's start. “‘Plan the work and work the plan’ was their implicit mantra,” they write. “And it led them directly to a costly and destructive course of action.... We'd all like to believe that this kind of problem is rare in business. It's not.”

## Working Products

*Working products over comprehensive documentation.* Innovation drives companies. The core ideology at 3M has always emphasized innovation. Motorola recently launched a new cell phone innovation initiative. In early 2003, General Electric changed its motto to “Explore Imagination at Work.” Jeffrey Immelt, chairman and CEO of GE, has placed a high priority on innovation and new business. “The companies that know how to develop things are ultimately

going to create the most shareholder value. It's as simple as that," says Immelt (Budeir 2003). Many companies have innovation initiatives; fewer are willing to create processes and practices that directly support those initiatives. Switching from delivering documentation artifacts—characteristic of a serial development style—to delivering iterative versions of the real product is one of those mind and practice shifts that supports innovation.

Large, front-loaded projects that spend months, and even years, gathering requirements, proposing architectures, and designing products are prone to massive failures. Why? Because teams proceed in a linear fashion with little reliable feedback—they have good ideas, but they don't test them in the cauldron of reality. Documents don't work. Products do.

Agile development and project management stress delivery of versions of the actual product, or in the case of high-cost materials, effective simulations or models of the actual product. Finishing a requirements document verifies that a team has successfully gathered a set of requirements. Completing and demonstrating a set of working product features verifies that the development team can actually deliver something tangible to the customer. Working features provide dependable feedback into the development process in ways that documentation cannot.

Again, working products don't preclude the need for documentation. Documents support communication and collaboration, enhance knowledge transfer, preserve historical information, assist ongoing product enhancement, and fulfill regulatory and legal requirements. They are not unimportant, just less important than working versions of the product.

However, there is a fundamental flaw in many people's understanding of documentation—*documentation is not a substitute for interaction*. When a customer and a developer interact to jointly develop specifications and produce some form of permanent record (documents, notes, sketches, feature cards, drawings), the documentation is a by-product of the interaction. When the customer sits down with a product manager and they write a requirements document that gets *sent* to a development group, then the document has become a substitute for interaction. In the first scenario, the documentation may be valuable to the development team. In the second, it has become a barricade to progress. Little knowledge is either gained or transferred. Furthermore, as interaction decreases, the volume of documentation often increases in a fruitless attempt to compensate.

## Customer Collaboration

*Customer collaboration over contract negotiation.* Customers and product managers drive agile development. The customer team (depending on the product type, the participants may be actual customers, product managers, product champions, or other customer proxies) and the development team form a partnership in which each has specific roles, responsibilities, and accountabilities. In highly volatile, ambiguous, and uncertain new product development efforts, the customer-developer relationship must be collaborative, not marked by adversarial contract disputes.

The goal of a project team is to deliver value to customers. While there are, in big organizations, a variety of stakeholders, the customer should reign supreme. Furthermore, the definition of customer can be reduced to a simple statement—the *customer* is the individual or group who uses the created product to generate business value or, in the case of retail products, the person who uses the product. Customers define value. Other stakeholders define constraints. When we confuse customers and stakeholders, our projects become muddled.

Customers define requirements (features) that provide value and the business objectives (schedule, cost) that assist in quantifying that value. Today, value arises from implementing features that meet this set of requirements and constraints as they evolve over the life of a project. Once a product has been delivered initially, tomorrow's benefits are a function of how quickly and cost effectively the product can be adapted to requirements and constraints that arise in the future. The formula for success is simple: deliver today, adapt tomorrow.

## Individuals and Interactions

*Individuals and interactions over processes and tools.* Ultimately, unique, talented, and skilled individuals—individually and collectively—build products and services. Processes provide guidance and support, and tools improve efficiency, but without the right people who have the right technical and behavioral skills, all the processes and tools in the world won't produce results. Processes (in moderation) and tools are useful, but when

critical decisions must be made, we rely on the knowledge and capabilities of individuals and the team.

Companies often issue flowery statements about how important their people are and then tie them down with unyielding procedures and forms. In the 1990s business went through a period of infatuation with process—much of it under the banner of business process reengineering (BPR). Process literally became more important than people. BPR proponents thought that structured processes would somehow make up for mediocre individual capabilities, but no process can overcome the lack of good engineers, product managers, customers, suppliers, and executives. Good processes should support the team rather than dictate its actions. Processes are adapted to the team rather than the other way around.

APM supports creators, not stewards. Colleague Rob Austin defines the differences between these two types of individuals:

*The problem the creators have is that they want to go for the big prize, to realize their visions in pure form; anything short of that they see as less than successful. The stewards, on the other hand, can't get excited about an innovation until ... they understand how the economic value (profit) will be created (Austin 2003).*

Stewards are critical to business success—they manage the numbers and invest prudently. Businesses without stewards often fail in their fiduciary responsibilities to employees, customers, and investors. However, without creators, the stewards have nothing to steward. Without innovation—new product development, new processes and practices, new ways of creating business value—the stewards are left maintaining an empty husk. Without the stable foundation provided by stewards, creators may spin out of control.

The agile movement supports individuals and their interactions through dedication to the concepts of self-organization, self-discipline, egalitarianism, respect for individuals, and competency. “Agile” is a social movement driven by both the desire to create a particular work environment and the belief that this “adaptive” environment is critical to the goal of delivering innovative products to customers.

## Agile Project Management

The central question for project managers, project team members, and executives is, “How does project management add value to a project?” Unfortunately, many development engineers consider project management to be a roadblock—a hindrance, not a help. Project managers are viewed as administrators who put together detailed task schedules, create colorful resource profiles, bug team members about micro-task completions, and write reams of status reports for upper management, not as direct contributors to delivering value to customers. Teams often view project management as overhead. As Michael Kennedy (2003), author of *Product Development for the Lean Enterprise* (about Toyota) says, “Our product development philosophy is based more upon administration excellence than technical excellence, and it’s getting worse.”

“Buy pizza and stay out of the way” expresses too many product engineering teams’ view of “good” project management. Rather than overhead, project management should be seen as offering inspirational leadership focused on delivering customer value. We have missed this point because many project management practices and project managers are focused on compliance activities, not delivering value. They are project administrators, not project managers. Customers pay for value; everything else is overhead—necessary overhead in some respects, but overhead nonetheless. Team activities or deliverables that help comply with government regulations are necessary, but they rarely add customer value. Documentation that conforms to legal requirements may be necessary, but it may not add value—at least not directly. Status reports assist managers in meeting their fiduciary responsibilities, but they don’t add value. Endless milestone approvals may delude management into thinking they are in control, but they don’t add value.

But how do we measure customer value and project management’s contribution to that value? Motorola’s ill-fated, multibillion-dollar Iridium project was a spectacular failure in the market. Meanwhile, the movie *Titanic*, which was severely over budget and time—and viewed by early pundits as a \$200 million flop awaiting release—was the first movie to generate over \$1 billion in worldwide revenue. By common compliance-based project management practices of budget, scope, and schedule performance,

*Titanic* was a failure. Within some circles, Iridium was considered a success because it fulfilled the original specifications.

All products face similar demands—customer needs, profit, development speed, constant change, high quality—that require high levels of competency in multiple domains of expertise. These often contradictory constraints and demands on product development—speed *and* quality, great functionality *and* low cost, uncertainty *and* predictability, mobility *and* stability—have created the need for project managers and project management practices that focus on delivering value.

APM is a set of values, principles, and practices that assist project teams in coming to grips with this challenging environment. The core values of APM address both the need to build agile, adaptable products and the need to create agile, adaptable development teams.

## Agility Defined

There is no *Agility for Dummies*. Agility isn't a silver bullet. You don't achieve it in five easy steps. So what is it? For myself, I've characterized agility in two statements:

*Agility is the ability to both create and respond to change in order to profit in a turbulent business environment.*

*Agility is the ability to balance flexibility and stability (Highsmith 2002).*

In an uncertain and turbulent world, success belongs to companies that have the capacity to create change, and maybe even chaos, for their competitors. Creating change disrupts competitors (and the entire market ecosystem); responding to change guards against competitive thrusts. Creating change requires innovation: developing new products, creating new sales channels, reducing product development time, customizing products for increasingly smaller market segments. In addition, your company must be able to respond quickly to both anticipated and unanticipated changes created by your competitors and customers.

An example of a product development effort in which all the aspects of agility come into play is that of small, portable DNA analyzers. These instruments, which are still five or more years away from commercial deployment,

could be used for analyzing suspected bioterror agents (e.g., anthrax), making a quick medical diagnosis, or performing environmental bacterial analysis. These instruments must be accurate, easy to use, and reliable under wide-ranging conditions, and their development depends on breakthroughs in nanotechnology, genome research, and microfluidics. A half-dozen companies are racing to build versions of a DNA analyzer, often with grants from agencies such as the US National Institutes of Health (Gardner 2003). Developing these leading-edge products requires blending flexibility and structure, exploring into various new technologies, and creating change for competitors by reducing delivery time. These are not projects that can be managed by traditional, prescriptive project management methodologies.

Some people mistakenly assume that agility connotes a lack of structure, but the absence of structure, or stability, generates chaos. Conversely, too much structure generates rigidity. Complexity theory tells us that innovation—creating something new in ways that we can't fully anticipate, an emergent result—occurs most readily at the balance point between chaos and order, between flexibility and stability. Scientists believe that emergence, the creation of novelty from agent interaction, happens most readily at this “edge of chaos.” The idea of enough structure, but not too much, drives agile managers to continually ask the question, “How little structure can I get away with?” Too much structure stifles creativity. Too little structure breeds inefficiency.

This need to balance at the edge of chaos to foster innovation is one reason process-centric methodologies often fail. They push organizations into over-optimization at the expense of innovation. Agile organizations don't get lost in some gray middle ground; they understand which factors require stabilization and which ones encourage exploration. For example, in a high-change product development environment, rigorous configuration management stabilizes and facilitates flexibility just as a focus on technical excellence stabilizes the development effort. The concepts and practices described in this book are designed to help project teams understand this balancing between flexibility and stability. They help answer the question of what to keep stable and what to let vary.

Agility means being responsive or flexible within a framework or context. As I've said elsewhere, “The problem with many traditional software development and project management approaches is that they have too narrowly defined the context; that is, they've planned projects to a great level of

task detail, leaving very little room for agility to actually happen” (Highsmith 2002).

Balancing at the edge of chaos between flexibility and stability requires people who are good improvisers—who have the ability to deal effectively with the ambiguity, and the paradox, of pursuing two seemingly dissimilar goals at once. Organizations that support these improvisers have three key traits:

- An adaptive culture that embraces change
- Minimal rules that encourage self-organization, combined with the self-discipline to closely adhere to those rules
- Intense collaboration and interaction among the project community

Each of these traits will be further described in the chapters on the guiding principles of APM.

## The APM Framework

Project management processes and performance measures are different for exploration- and experimentation-based approaches than they are for production- and specification-based ones. Production-oriented project management processes and practices emphasize complete early planning and requirements specification with minimal ongoing change. Exploration-based processes emphasize nominal early planning, good enough requirements, and experimental design with significant ongoing learning and change. Each approach has its place, but the lifecycle framework for the latter has a very different flavor from the former. The APM framework consists of five phases, each with supporting practices. They are Envision, Speculate, Explore, Adapt, and Close.

These phases resemble a scientific investigative process more than a production management process. The Envision phase results in a well-articulated business or product vision—enough to keep the next phases bounded. In the Speculate phase, the team hypothesizes about the specifications of the product, knowing that as the project continues both technology and customer specifications will evolve as new knowledge is gained. The Explore phase then becomes a parallel and iterative operation in which the

preliminary specifications and design are implemented. Components labeled “uncertain” are subject to more experimentation than others whose specifications or design are more certain. In the Adapt phase, the results of these experiments are subjected to technical, customer, and business case review, and adaptive actions are incorporated into the next iteration.

One of the most common questions about APM is, “What about the planning, architecture, and requirements phases?” The simple answer is that these things are activities and not phases. An agile approach can easily include as much time for these activities as in a conventional serial phase approach, *but* the activities are spread across multiple iterations.

A second area of concern is the risk of rework in agile development if the initial architecture work (the discussion in this section could refer to architecture, plans, or requirements) misses a critical item. But there is an equal if not greater risk in serial development that often goes unnamed—that of getting the up-front architecture wrong. In a serial process, the validation of the early architecture decisions comes late in the project lifecycle, when the actual building occurs. By that time, a tremendous amount of time and money has been spent. Changing the architecture then becomes a major, and costly, decision—if it is possible at all.

Within this general APM framework, the successful completion of each phase depends upon a series of key practices that actually guide the work effort. Some of these practices are simple and straightforward, like the product vision box and project data sheet, which I’ll discuss in Chapter 5. Others such as customer focus groups and iterative planning are based directly on the core values of agile organizations, while such practices as workload self-management and participatory decision making focus on building an agile, adaptive culture. Values and guiding principles describe the *why* of APM, and practices describe the *how*.

## Thriving in a Chaordic World

Former Visa International CEO Dee Hock (1999) coined the word “chaordic” to describe both the world around us and his approach to managing a far-flung enterprise—balanced on the precipice between chaos and order. Our sense of the world dictates management style. If the world is perceived as static, then

production-style management practices will dominate. If the world is perceived as dynamic, however, then exploration-style management practices will come to the fore. Of course, it's not that simple—there is always a blend of static and dynamic, which means that managers must always perform a delicate balancing act.

In the last decade, a vanguard of scientists and managers have articulated a profound shift in their view about how organisms and organizations evolve, respond to change, and manage their growth. Scientists' findings about the tipping points of chemical reactions and the "swarm" behavior of ants have given organizational researchers insights into what makes successful companies and successful managers. Practitioners have studied how innovative groups work most effectively.

As quantum physics changed our notions of predictability and Darwin changed our perspective on evolution, complex adaptive systems (CAS) theory has reshaped scientific and management thinking. In an era of rapid change, we need better ways of making sense of the world around us. Just as biologists now study ecosystems as well as species, executives and managers need to understand the global economic and political ecosystems in which their companies compete.

A complex adaptive system, be it biologic or economic, is an ensemble of independent agents:

- Who interact to create an ecosystem
- Whose interaction is defined by the exchange of information
- Whose individual actions are based on some system of internal rules
- Who self-organize in nonlinear ways to produce emergent results
- Who exhibit characteristics of both order and chaos
- Who evolve over time (Highsmith 2000)

For an agile project, the *ensemble* includes core team members, customers, suppliers, executives, and other participants who interact with each other in various ways. It is these interactions, and the tacit and explicit information exchanges that occur within them, that project management practices need to facilitate.

The individual agent's actions are driven by a set of internal rules—the core ideology and principles of APM, for example. Both scientific and management researchers have shown that a simple set of rules can generate

complex behaviors and outcomes, whether in ant colonies or project teams. Complex rules, on the other hand, often become bureaucratic. How these rules are formulated has a significant impact on how the complex system operates.

Newtonian approaches predict results. CAS approaches create emergent results. “Emergence is a property of complex adaptive systems that creates some greater property of the whole (system behavior) from the interaction of the parts (self-organizing agent behavior). This emergent system behavior cannot be fully explained from measured behaviors of the agents. Emergent results cannot be predicted in the normal sense of cause and effect relationships, but they can be anticipated by creating patterns that have previously produced similar results” (Highsmith 2000). Creativity and innovation are the emergent results of well-functioning agile teams.

Another pair of words that indicate this perspective difference are optimization and adaptation. Optimization processes focus on efficiency and cost reduction. They are documented, measured, refined, and repeated. Adaptation processes focus on innovation, exploration, speed, and constantly reacting to meet external changes. Optimizing processes thrive in low-change, predictable environments, whereas adaptive processes thrive in high-change, uncertain ones.

An adaptive development process has a different character from an optimizing one. Optimizing reflects a basic prescriptive Plan-Design-Build lifecycle. Adapting reflects an organic, evolutionary Envision-Explore-Adapt lifecycle. An adaptive approach begins not with a single solution, but with multiple potential solutions (experiments). It explores and selects the best by applying a series of fitness tests (actual product features or simulations subjected to acceptance tests) and then adapting to feedback. When uncertainty is low, adaptive approaches run the risk of higher costs. When uncertainty is high, optimizing approaches run the risk of settling too early on a particular solution and stifling innovation. The salient point is that these two fundamental approaches to development are very different, and they require different processes, different management approaches, and different measurements of success.

Newtonian versus quantum, predictability versus flexibility, optimization versus adaptation, efficiency versus innovation—all these dichotomies reflect a fundamentally different way of making sense of the world and how to manage effectively within it. Because of high iteration costs, the traditional

perspective was predictive and change averse, and deterministic processes arose to support this traditional viewpoint. But our viewpoint needs to change. Executives, project managers, and development teams must embrace a different view of the new product development world, one that not only recognizes change in the business world, but also understands the power of driving down iteration costs to enable experimentation and emergent processes. Understanding these differences and how they affect product development is key to understanding APM.

## Our Journey

The objective of this book is to outline the values, principles, and practices of Agile Project Management—to describe what I believe constitutes a better approach to project management *within a specific context*, which I will also endeavor to describe. If this context does not fit your organization, then APM may not be for you. However, if the context does fit, you may find that the concepts and practices of APM will help you achieve your goals of delivering innovative products to customers and improving your work environment.

Many, if not most, of the practices of APM are not new. Iterative lifecycle development, for example, has been around since the 1950s (Larman 2004). Good practices for building a project community have developed steadily over the years. Fundamental project management practices have evolved, and many are useful in fast-moving, rapidly changing projects.

There are fewer than 20 practices described in this book. Why, then, are there 1,500-page project management books with dozens, if not hundreds, of practices? Is something missing here? One of the key concepts of agile project management and development is that the practices, when driven by guiding principles, are generative, not prescriptive. Prescriptive methodologies attempt to describe every activity a project team should do. The problem with prescriptive approaches is that people get lost. They have much to choose from and so little guidance as to applicability that they have trouble eliminating extraneous practices from their projects.

A set of generative practices is a minimal set that works well together as a system. It doesn't prescribe everything a team needs to do, but it identifies

those practices that are of extremely high value and should be used on nearly every project. By employing these practices, project teams will “generate” other necessary supporting practices as part of their tailoring and adapting the set to fit their unique needs. Starting with a minimal set of practices and judiciously adding others as needed has proven to be more effective than starting with comprehensive prescriptive practices and attempting to streamline them down to something usable (Boehm and Turner 2003).

So is APM new? Well, yes and no. Complexity theory tells us that biological agents evolve by recombining existing building blocks until a different organism emerges. APM involves carefully selecting existing building blocks—practices that have proven useful to project teams in the past—and linking these practices to core values, a set of guiding principles, and a conceptual framework that draws on CAS theory as its foundation. The “combination” of all these building blocks—practices, values, principles, and conceptual framework—results in Agile Project Management. APM draws on a rich project management legacy, but it is very selective about which parts of that legacy it incorporates. APM also draws on a rich legacy of management, manufacturing, and software development literature and practice that incorporates a worldview and ideological foundation better suited to mobility and speed.

APM isn’t for everyone or every project; it is *not* a universal best practice. APM works well for certain problem types, in certain types of organizations, with people who have a particular cultural perspective, and for managers who have a certain worldview. It thrives in innovative cultures and on projects in which speed, mobility, and quality are all key to success, such as in product development. APM is not defined by a small set of practices and techniques. It defines a strategic capability to create and respond to change, to balance flexibility and structure, to draw creativity and innovation out of a development team, and to lead organizations through turbulence and uncertainty.

