

# Innovation & Quality in Healthcare IT: The Agile Revolution<sup>1</sup>

Jim Highsmith<sup>2</sup>

## Abstract

*From materials research to drugs to integrated circuits, companies are relentlessly driving the cost of change out of their software development processes. Why? — To increase experimentation, to increase the diversity of paths explored, to foster innovation. “Exploration” projects severely challenge traditional “production” oriented project management and software development practices that attempt to optimize, predict paths, and conform to detail plans. So a new “exploration” model has arisen and gained popularity over the past several years—an “agile” development and project management model that focuses on quick starts, iterative exploration, delivering customer value, low-cost iterations, frequent feedback, and intense collaboration. Agile methods excel on projects with high “exploration factors,” those projects in which: new, risky technologies are incorporated; requirements are volatile and evolve; time-to-market is critical; and high quality must be maintained.*

## Keywords

Agile Project Management, Agile Software Development, Exploratory Projects, Innovation, Reliable Processes

## INNOVATIVE SOFTWARE DEVELOPMENT

In mid 2002, when Alias Systems 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 teams didn’t begin with a lengthy product planning effort. The team’s marketing and product strategy work evolved over several months, but their product development effort began early and in parallel with the strategy process. They 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

---

<sup>1</sup> This article is excerpted and modified from Jim Highsmith’s new book, *Agile Project Management: Creating Innovative Products*, Addison Wesley 2004 (March). The article is copyrighted by Jim Highsmith.

<sup>2</sup> Jim Highsmith is Director, Agile Project Management Practice and Fellow, Business Technology Council at Cutter Consortium. A recognized leader in the Agile movement, he co-authored the Agile Manifesto and helped found the Agile Alliance. ([www.jimhighsmith.com](http://www.jimhighsmith.com))

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 co-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 what 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.

Healthcare information technology (IT) applications may be different from a graphics software package, but the issues of flexibility, quality, and speed (at least for the category of high exploration-factor projects) are very similar. I have worked with companies that developed mass spectrometers for the pharmaceutical industry and clinical drug trial monitoring systems. While these types of software applications require compliance activities beyond those for non-healthcare applications, the objective of delivering value to customers, through adaptable products that can weather future change, delivered on time, and possessing high-quality in a volatile business and technology environment, dictates the need to reassess traditional project management and software development practices.

While most people recognize industry volatility, they have not truly modified either their mindset or their management practices, relative to change. For example, the perceived high cost-of-change has driven methodologies to attempt to limit change (because it is expensive) by engaging in extensive front-end planning and analysis work. The problem is that no matter how thorough this early work—stuff happens. Change is going to happen. So, rather than let the high cost of change dictate process, agile developers focus on reducing costs, such that low-cost change enables us to use a process in which a minimum amount of up-front work is followed by a succession of short development iterations.

When we reduce the cost of change or experimentation enough, the entire economics of how we do software development changes—it switches from a process based on anticipation (define, design, and build) to one based upon adaptation (envision, explore, and refine). 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.

And while companies need results from their high pressure development efforts, 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 software 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.

Linear thinking, prescriptive processes, and standardized, unvarying practices are no match for today's volatile business environment. So, as software development processes swings 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

There are five key business objectives for a good exploration process such as agile development methodologies and 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; and (5) reliable results—to support business growth and profitability.

## **Continuous Innovation**

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 and 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 projects, changes in the market, technology, or specific requirements happen weekly. Timeframes are shrinking and the only way to survive is to strive for 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 development times indicate, reducing delivery schedules to meet market windows continues to be a high-priority business goal for 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 features and their prioritization in short, iterative timeboxes forces teams (customers and developers) to carefully consider both the number of features to include and the depth of those features. Constant attention reduces the overall work load by eliminating marginally beneficial features. Second, APM—like its lean development counterparts—streamline 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 encourages 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? 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. I think one reason that agile development has escalated 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. This Manifesto 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 software by doing it and helping others do it. Through this work we have come to value:**

*Individuals and interactions over processes and tools*

*Working products (software<sup>1</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>2</sup>**

These statements should not be construed as indicating that tools, process, documents, contracts, or plans are unimportant. There is a tremendous difference between one thing being unimportant and being less important than another. Tools are absolutely 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 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. 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-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 exploring into that vision rather than detailed planning and then relatively strict execution of tasks. It's not that one is right and the other wrong, but each style is more or less applicable to a particular project type.

### **Working Products**

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. Software does.

Agile development and project management stress delivery of versions of the actual software. Finishing a requirements document verifies that a team has successfully gathered a set of requirements. Completing and demonstrating a set of working features verifies that the development team can actually deliver something tangible to the customer. Working features provide dependable feedback into the development process.

### **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.

### **Individuals and Interactions**

*Individuals and interaction over process 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 with the right technical and behavioral skills—people with the right knowledge and capacity to learn—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 to overcome obstacles.

## **The Agile Project Management Framework**

Process may not be as important as people, but it's far from unimportant. Process has gotten a bad rap in agile circles (much of it deserved) as being static, prescriptive, and difficult to change. But process, per se, doesn't have to be unhelpful. Process, like anything else, must be tied to business objectives. If the business objective is repeatable manufacturing, then a prescriptive process may be completely justified. However, if the business objective is reliable innovation, then the process framework must be organic, flexible, and easy to adapt. The APM process framework, as shown in Figure 1, supports this second business objective through the five phases of: Envision, Speculate, Explore, Adapt, and Close.

### **Envision**

The Envision phase creates a vision for the customers and the project team that covers what, who, and how. Absent a vision, the remaining activities in getting a project off the ground are wasted effort. In business-speak, vision is the “critical success factor” early in a project. First, we need to envision *what* to deliver—a vision of the product and the scope of the project. Second, we need to envision *who* will be involved—the community of customers, product managers, project team members, and stakeholders. And, third, the project team members must envision *how* they intend to work together.

Specific practices in the Envision phase include: A Product Vision Box, Product Architecture and Guiding Principles, Project Data Sheet, Get the Right People, Customer-Development Team Interface, and Process and Practice Tailoring.

### **Speculate**

The word “speculate” first calls to mind an image of reckless risk taking, but actually the dictionary definition is “to conjecture something based on incomplete facts or information,” which is exactly what happens during this phase.<sup>3</sup> The word “plan” has come to connote certainty and prediction, while the more useful definition of plan, for exploratory projects at least, is speculating or hypothesizing based on incomplete information. APM consists more of envisioning and exploring rather than planning and doing—it forces us to confront the reality of today’s precarious business environments.

There are two critical characteristics to the Release, Milestone, and Iteration planning practice used in this phase—short timeboxed iterations and feature-based planning. The iterations are generally 2-6 weeks to which features are assigned. The planning is by feature, which customers understand, not the traditional activities of requirements gathering, design, etc. Task planning is subordinate to feature planning—that is, tasks are planned by feature, not as independent activities.

Specific practices in this phase include: Product Feature List, Feature (story) Cards, and Release, Milestone, Iteration Plan.

### **Explore**

The Explore phase delivers product features. From a project management perspective there are three critical activity areas during this phase. The first is delivering planned features by managing the workload and using appropriate technical practices and risk mitigation strategies. The second is creating a collaborative, self-organizing project community, which is everyone’s responsibility, but is facilitated by the project manager. The third activity is managing the interactions among customers, product management, and other stakeholders.

Specific practices in the Explore phase include: Workload Management (the team manages the distribution of its own work), Low-cost Change (technical practices such as ruthless testing and refactoring), Coaching and Team Development, and Participatory Decision Making.

## **Adapt**

Control and correction are common terms applied to this lifecycle phase. Plans are made, results are monitored, and corrections are made—implying that the plans were right and the actual results, if different from the plan, are wrong. “Adapt” implies modification or change rather than success or failure. In projects guided by the philosophy that responding to change is more important than following a plan, attributing failure to variation from the plan isn’t productive. A purely ad hoc process fails to learn from its mistakes, whereas the incorporation and retention of lessons learned are a key piece of APM.

In the Adapt phase the results are reviewed from a customer, technical, project status, and people and process performance perspectives. The analysis looks at actual versus planned, but even more importantly, it considers actual versus a revised outlook on the project given up to the minute information. The results of adaptation are fed into a re-planning effort to begin the next iteration. This phase includes product, project, team review and adaptive action activities.

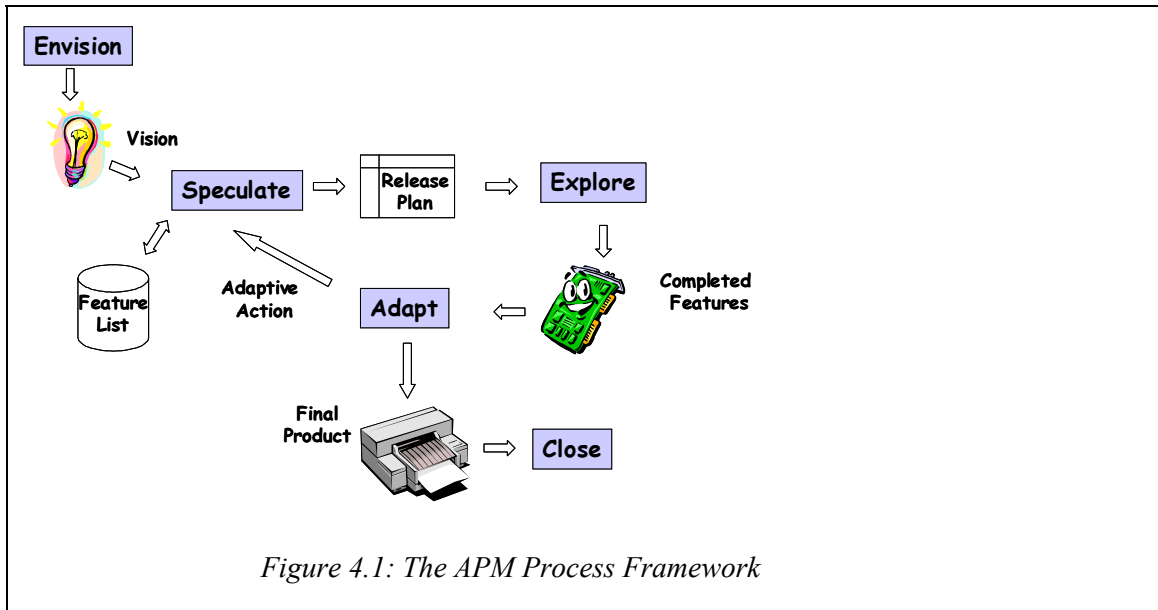
## **Close**

Projects are partially defined by the presence of both a beginning, and an end. Many organizations fail to identify a project’s end point, often causing perception problems among customers. Projects should end—with a celebration. The key objective of the Close phase, and the “mini” close at the end of each iteration, is learning and incorporating that learning into the work of the next iteration or passing it on to the next project team.

## **Reliable Innovation**

In the end, the positive answers to two questions form the essence of APM and agile software development: Are you delivering innovative products to your customers? Are you excited about going to work every day? Agileists want to build innovative products—products that test the limits of our abilities as individuals and project teams—to create a work environment in which people, as individuals and as teams, can thrive.

Just like Broadway plays always deliver on opening night, APM delivers on time and to the customer’s vision—more reliably than any other approach for high exploration factor projects. Given the high degree of uncertainty on many new product efforts, given the changes in technology, given the ebb and flow of staff, reliable results are still obtainable. Given all the “maybes,” agile methods still deliver—a tribute to the passion, drive, persistence, and ingenuity of project team members.



<sup>1</sup> The manifesto wording is “software.” I use the word “product” as a more general term.

<sup>2</sup> The Agile Manifesto is copyrighted by the authors ([www.agilemanifesto.org](http://www.agilemanifesto.org)). For an in-depth explanation of the Agile Software Manifesto, see Martin Fowler and Jim Highsmith, “The Agile Manifesto,” *Software Development* magazine, August 2001, (<http://www.sdmagazine.com/documents/s=844/sdm0108a/0108a.htm>).

<sup>3</sup> Encarta® World English Dictionary © & (P) 1999,2000 Microsoft Corporation.