



# Project Management Newsletter

eCameron, Inc.

Industry

Issue 8  
June 2006

## In This Issue:

An Agile Focus	1
Agile Benefits Case Study	1
Technique: Remote Projects	4
Book Synopsis	7
Resources and Templates	13

## Feature Article

**Agile Benefits Case Study:** A case study in a progressive implementation of agile project management.

**Remote Projects:** Tips and techniques to consider when setting up or joining a project where team will be geographically dispersed.

**Agile Project Management, Creating Innovative Products** by Jim Highsmith. An introduction to Agile Project Management providing eighteen practices for implementation.

## Next Issue

**Discussion Topic: Risk,** a refresher on probability. This will be the first part in a multipart series on risk. The focus will be to how to manage its perception.

**Management Technique:** Executive Reporting Tools. A look at four-square reports and simple graphics to report progress.

**Book Synopsis: *The 33 Strategies of War*** by Robert Greene.

Newsletters are published monthly and issued mid-month. To receive back issues, be added or be removed from the distribution, send an email to:

[newsletter@ecaminc.com](mailto:newsletter@ecaminc.com).

Now partnering with Powell's



## An Agile Focus

Lean and Agile processes are the focus of many business restructurings. This edition of the Newsletter will feature two articles on Agile Project Management—a case study on implementing it and a synopsis of the book *“Agile Project Management, Creating Innovative Products”*. Unlike prior newsletters, these two articles are interrelated in that the case study shows how the practices outlined in the book apply in a real life situation. This is in response to your recommendations and comments. We hope you find this useful.



Looking for more information a book? More book reviews and comments are available on our website at:

<http://www.ecaminc.com/Books/BookHome.php>

Purchase them at Powell's.

**powells.com**

## Agile Benefits Case Study

Implementing Agile Project Management (APM) requires a top-down approach. The differences in reporting, team structure and customer relationship verses traditional project management methodology need buy-in at the upper echelons of the company. Educating superiors and customers on the benefits of Agile Project Management may be difficult, especially if they do not have a project management background or if they have been a classical PM for a significant length of time. Implementing a pilot project is one way to quell their fears.

This article will examine an implementation process at a company skeptical of change and compare the results, in retrospect, to the practices set forth in the book *“Agile Project Management, Creating Innovative Products”*, this month's book in review.

Getting people to change to an alternate method when they are familiar with it is difficult; getting people to change to something they have not done before is even harder. Therefore, changing to an agile methodology, which is very different from what most projects use can be very hard; there needs to be a way to demonstrate its power without making executives commit to something that might be perceived as too high of a risk. Proposing and executing an iterative approach to implementing agile

methodology can show the benefits of it more quickly than one might expect.

This case study describes what was done at a client to educate managers on the benefits of APM. First some background. The company had developed a sales application that was deployed at numerous franchised and internal sales organizations located throughout North America. The product had feature upgrades developed and deployed on 9-12 month cycle using the serial PDIM waterfall project approach. The business decision had been made to perform all bug fixes as part of the project rather than a separate release process. There were no technical constraints to inhibit distribution of releases on a much more frequent schedule—the infrastructure was in place (internet-based distribution). It was determined that APM was feasible.

Franchised dealers, sales and order processing organizations were not pleased with the turnaround time for minor enhancements and bug fixes. The IT organization was unwilling to try new methodologies since the product had been plagued by multiple maintenance and management problems (most from long lead time monolithic releases).

The Project Manager proposed implementing minor feature changes on a much

shorter cycle. To do this would require overcoming significant political pressure. The proposal involved implementing agile methodology in steps to show it could be beneficial. The approach proposed was:

1. Demonstrate that features could be fixed and released separately from the project cycle.
2. Work with the customers (franchised dealers, sales and order processing organizations) to prioritize the feature list initially restricting the changes to the simplest of enhancements and critical bug fixes.
3. Segment the project (which was on a nine month cycle) to create a small team to implement and test the features in short iterative cycles.
4. Show success in deploying minor features.
5. Improve the process to include more significant enhancements.

A number of factors had to be taken into consideration, the biggest, though, was to overcome the inertia of the team and the customer, followed by limiting the number of items to be fixed in each cycle. Many other smaller struggles were encountered as the process continued.

**Convincing the team**

Due to previous issues, the team was highly demoralized, some resources had been denied training on tools and felt incapable of meeting the request, many were skeptical of the Project Manager’s radical ideas for the project and a few members’ actions bordered on insubordination. The first task was finding allies. These were found in the developers and the Product Manager.

The Project Manager wanted a release every four weeks. This implied an average of two weeks for development and two weeks for regression. If needed the development could be allowed to overlap with regression testing by a week providing more time for development.

The Product Manager, development lead and Project Manager (core team) analyzed over 1,500 defects and enhancement requests in the tracking system. In a two day session these items were prioritized and assigned an effort-to-complete value (small, medium, large and too big for an iteration).

Next, the electronic distribution system was studied. If quick turns of product releases were going to take place, then this system had to run flawlessly and be able to report on the deployment success. The previous Subject Matter Expert had left leaving little documentation. As mentioned earlier, the system had been turned over to a junior resource who had not been provided training. The project manager assigned a developer to work with him to expedite his understanding of the product. Through this process the system was tuned, the junior employee was mentored and reports (showing deployment penetration) were generated.

The core team recruited two volunteer developers and a skeptical QA lead (to get proper application QA support). It was determined that approximately 160 hours could be spent on the first iteration by each group. This would be a two-week duration by each group (two resources each).

The iteration cycles were laid out as shown in **Table 1**.

Wk	Cycle N	Cycle N+1
1	Prioritize features to be worked (will deliver, try to deliver) Define features in both categories	
2	Developers start work.	
3	Developers continue work.	
4	Developers continue and QA come in to assist with unit test.	
5	QA finish testing, developers work with QA to fixed errors.	Prioritize features to be worked (will deliver, try to deliver) Define features in both categories
6	Deploy	Developers start work.
7		Developers continue work.
8		Developers continue and QA come in to assist with unit test.
9		QA finish testing, developers work with QA to fixed errors.
10		Deploy

**Table 1: Iteration Plan**

This six week process was interleaved to create a four week release cycle.

Since this process was new to the team, it was felt that a slow introduction was best. During the first iteration (each iteration produces a Service Pack) changes in features were strongly discouraged.

The Project Manager focused on reporting. When a service pack was deployed they would have an immediate and positive effect on the sales and order processing groups. It was imperative that the benefit be quantified in order to show managers the immediate effect. Getting the customer’s support on the speed of delivery and benefit to them would be critical in convincing IT management of the benefits. Reporting methods were tested on the closeout of the previous (inherited) project and approved by the customer and steering committees.

The Project Manager had to achieve a “win” for the team. He worked to build confidence by:

1. Promoting the team to the customers,
2. Holding individual meetings with team members,
3. Removing specific naysayers.

eCameron, Inc. has significant experience setting up projects with processes tailored to the needs of given products and organizations. Creating the right level of processes and implementing the correct methodology to get projects done on time and within budget. For more information please contact:

Todd Williams  
 Phone: 1-360-834-7361  
 Email: [todd.williams@ecaminc.com](mailto:todd.williams@ecaminc.com)

Implementing these goals and a firm but fair management style, would improve the team's self-image and confidence. The Project Manager knew that a successful deployment would raise credibility and build much needed confidence.

### Convincing the Customer

The concept was proposed to the steering committee (composed primarily of customer representatives). As expected it was well received, but they were also skeptical of success. The reporting methods were shown and specific success criteria were outlined to allay fears of being unprepared. Reporting was critical since the committee had been told by prior management that the reports could not be generated. Showing them actual reports greatly improved the confidence in the Project Manager's skill and resourcefulness.

Next, the Project Manager requested the Steering Committee prioritize the feature/defect list. For the first iteration, a pre-prioritized list was submitted. As expected the customers rated all the items as high priority. The Project Manager had to remind the committee that there were only 160 hours of developer and 160 hours of QA time to make the changes. Since releases would be every four weeks the next set would be fixed in four weeks—the next Service Pack. Eventually this resulted in super-high, medium-high, high and low-high—whatever terminology works.

After selling the concept to the steering committee, they supported the Project Manager's ideas in front of IT, whom were less than supportive of a new concept and had been scarred into a zero-risk (and non-customer oriented) approach.

### Ground Rules for the Team

Risk was very high because the team had never followed this approach and their management (IT) was very skeptical. As a hedge the Project Manager under-committed on the team's capability and worked with the team to over deliver. In order to abate many of the political issues, the Project Manager worked with developers to include as many "no-time" fixes as possible—minor and medium items, all low risk. Although this strained the QA group it was felt this approach was worth the effort.

The team agreed on a new single-page feature definition document; it included screen mockups, behavior and validation rules. This document was designed so that QA could use the same document for testing.

Daily status was gleaned from each team member. Instead of a formal meeting the Project Manager visited each member individually; this allowed each team member to better voice their concerns without feeling pressure from the others on the team. Any questions or concerns were given the highest priority and addressed promptly.

### Iteration 1 Success

The first release resulted in thirty-five major feature changes and over 100 minor (cosmetic) and medium (non-functional, cumbersome) bugs.

The response from customer and steering committee was phenomenal. Penetration of the release to the sales force was approximately 80% in the first twenty-four hours and positive effects in the order processing organization were seen within a couple of hours.

Team self-confidence sky rocketed.

### Enhancing the Process

For each iteration the development cycle was altered. These changes included:

1. Allowing reprioritization and new requests. Trade-offs were usually required and, with facilitation, the customer adapted and followed the process.
2. More work was assigned to each iteration.
3. The development team contacted the requestor of the feature directly (very taboo in the organization) to streamline the solution proposal.

Unexpectedly, there was significant push-back from one of the allies—the developers—of the original proposals. Simple questioning determined that the team was frustrated with the condition of the code. They were instructed to fix the code as necessary following two principles:

1. The changes they were making had to be sound. They were not "hurried" changes; they were changes that could be done properly in a short amount of time.
2. Each iteration had a few days set aside for refactoring code. When they touched code that was poor quality, they were supposed to change it.

This, and the implied trust that went with it, made pride soar.

### How Does This Compare?

As part of this newsletter, focusing on agile management, there is a synopsis of the book *"Agile Project Management: Creating Innovative Products"* by Jim Highsmith. To show the usefulness of his principles, this implementation will be matched to the eighteen he describes. This implementation of agile management was completed over a year before reading this book, so this article is a retrospective look at the application.

### Envision Phase

1. **Product Vision Box, Elevator Test Statement and a product vision document:** The product existed; therefore using this practice was not practical.
2. **Product Architecture:** The product existed; therefore using this practice was not practical.
3. **Project Data Sheet:** The product existed; therefore using this practice was not practical.
4. **Get the Right People:** This process was accomplished by reassigning people inside the team and replacing others. Developers volunteered for the new approach and felt responsible for its success. Replacement of team members was done quickly in order to minimize anxiety in the team.
5. **Participant Identification:** This had already been done, but the new methodology required new roles and responsibilities. Identifying the real customer was an issue—users of the application (Sales) and users of the output (order processing) both claimed equal authority. Eventually a compromise was established based on corporate cost of an error—order processing had priority. In addition, customers had been providing only high-level input to the process; they needed to provide details and be more involved with the workings of the process.
6. **Customer/Developer Team Interface:** Customer and

team interaction needed significant work to enable this interaction in an effective manner. It took two to three iterations to make this work properly. Training the development team to not say "yes" and instead say "Sure, what would you like to trade for that?" and getting the customer to do the tradeoffs initially took facilitation.

**7. Process and Practice Tailoring:** The processes and practices were constantly changing. Just the fact that this process was being implemented in an iterative fashion made it tailored at every iteration.

#### Speculate Phase

**8. Product Feature List:** This was accomplished by compiling a variety of lists that already existed and working with the customers to prioritize and validate the requirement for the features. Many of the items were removed, some were combined and as the process continued the customers realized that even more features were not needed. It appeared that even the customer started realizing that many requirements were really insignificant.

**9. Feature Cards:** Combination Feature Cards and Performance Requirement cards were developed. Due to the type of interaction with the customer and their dispersed nature, a spreadsheet (an example may be found at [http://www.ecaminc.com/Publications/Support/IterationCandidateList\(v1.0\).xls](http://www.ecaminc.com/Publications/Support/IterationCandidateList(v1.0).xls)) was used to summarize them so they could be quickly sorted. The entries were automatically color coded based on the number of hours required for the estimated work assigned to the service pack. This allowed for the Feature Cards to be used with WebEx® and provided immediate feedback on what lower priority feature would be excluded from the iteration.

**10. Performance Requirements Cards:** Performance card data was included in the feature cards above.

**11. Release, Milestone, and Iteration Plan:** The iteration plan was artificially generated prior to the Speculate Phase. An arbitrary monthly cycle was chosen.

#### Explore Phase

**12. Workload Management:** The team involved with this process was small. Initially their work was "strongly suggested" and as confidence grew the team members would split the work between themselves.

**13. Low-Cost Change:** Due to organizational issues, the cost of this process was not isolated from other work. But the cost savings from the order processing group and helpdesk was monitored. Both showed extremely favorable results in reduction of manual processing of orders from program issues. There was also a reduction in helpdesk calls for specific usability issues.

**14. Coaching and Team Development:** Training was critical in many areas. In addition, developers requested work in areas of the product or with tools they were not familiar with to advance their skills and knowledge. This was encouraged, but monitored closely. As mentioned the process was new, therefore coaching on the processes was intensive.

**15. Daily Team Integration Meetings:** The team met frequently but not mandated as daily. The Project Manager visited each team member at least twice a week.

**16. Participatory Decision Making:** This process could have

improved in this application. Decisions were still centrally run at the Project Management level. This was partly due to the team's inexperience with the process.

**17. Daily Interaction with the Customer Team:** Many processes were changed in communicating with the customer. Prior to implementing this approach, the customer was continually trying to get more from the developers in a very random manner. Therefore, for the first iteration all communication was pushed through the Project Manager. Before opening up the communication, three items had to be completed:

- The new process had to be defined and understood.
- The customers had to be trained on the method and the philosophy of maintaining the iteration length. Tradeoffs on features had to be instilled.
- Success had to be shown. Showing the customers that this process worked was crucial in getting them to make their tradeoffs. They had to know that what they asked for would be delivered; hence trading off an item did not mean losing it forever (as in the past).

#### Adapt Phase

**18. Product, Project, and Team Review and Adaptive Action:** The adapt phase was primarily applied to the process and not the product. The product had a relatively stable roadmap, but the process changed in every iteration to bring it closer to an agile methodology.

In summary, many of the principles Highsmith discusses were used in this implementation. There was much more to be done, but the organization had the proof it needed to show that agile methodologies not only worked but provided a less expensive development cycle and a happier customer.




---

## Technique: Remote Projects

---

As business becomes more global and outsourcing of teams more common, remote projects and virtual teams become commonplace. There are a whole new set of issues including accommodating language, time zones, tools and culture that need to be addressed. There are many steps that can be taken to avert anticipated problems.

#### Language

The first step is to standardize on a language. Although this may seem obvious, it is not trivial. Most people reading this newsletter are North American and a large majority of North Americans are mono-lingual English speaking. For this audience the only option is English; relying on the other team members to resort to English. This may result in different parts of the project being conducted in different languages. For instance, a common scenario would be a North American company running a project in China. In this case:

- The end product is used by mono-lingual Mandarin speakers. Hence, requirements are constantly filtered through a translator.
- The customer representatives for the project use English and Mandarin. They need to have the domain knowledge

for the application and excellent command of English.

- The project management is carried out in English. But significant conversations need to be done in Mandarin.

Understanding this in the beginning and setting expectations is crucial. There must be a high degree of trust between supplier and customer, especially when the customer has side discussions in a language the supplier does not know.

### English as a Second Language

Remote teams often mean some team members use English as a second language (ESL). Those of us that were raised in North America and have spoken English all our lives do not understand the difficulty in learning it.

Not only is the language difficult but the dialects are subtle and can cause considerable misunderstanding. An example comes from working on a project where the Asian team learned English in a British sponsored school. The customer was adamant that a given "requirement was very dear". This was interpreted by the American team as the requirement was needed and even carried an emotional attachment. The customer could not understand the seriousness that was associated with discussion of the requirement and why this requirement that would "cost dearly" was always being discussed and given highest priority. After nearly a month of confusion the team was at dinner and the customer complained about the menu prices being "quite dear". The connection was made.

It must be assumed that ESL vocabularies are small. Therefore attempt to use a guideline of using only two syllable words. This is difficult (impossible if you try to implement it literally), but helps when searching for words. Use "very important" not "paramount" and "list" not "enumerate". Try to talk slower, thinking about your choices of words and try to use simpler words.

Often people that have learned a second language are hesitant to say they do not understand a word. They attempt to figure out the meaning in the context it was used. This is how nearly everyone expands their vocabulary. As an example, after nearly two months on contract negotiations, an ESL client once asked what the word "caveat" meant. Without too much thought one can imagine what points did not get across in the previous discussions.

Two very important rules in an ESL environment:

- Do not use colloquialisms: "When all is said and done" may mean nothing to a person not fluent in English.
- Do not use jokes, sarcasm or cynicism. Even though it may be funny to you and your team, there probably is no translation. In some cultures the concepts of sarcasm or cynicism do not exist.

### Desktop Tools

International or not, the team's tools need to be as close in version as possible. A team member who uses a tool that has features other team members do not have can render all or parts of a file unusable. In severe cases the file can become corrupt and need to be recovered from a backup, potentially losing data. Strict configuration control must be maintained on software. Trying to determine the offending

party (or even that version compatibility is the cause) may take significant effort when the team is geographically dispersed.

In international projects the tools cannot be the same. Language specific operating systems and desktop software may be numerical equivalent versions, but their functionality may be radically different.

For example, trading documents between individuals that use different text orientation platforms can result in documents that are extremely difficult to work with. European languages use text orientation of left-to-right, where Middle East and some Asian countries use right-to-left. Editing documents transferred between these machines can cause some very strange behavior. Backspace and delete keys may be reversed (and only in certain sections of a document) and transposition of certain strings may not be correct. In some versions of Microsoft Outlook a phone number entered in one orientation, as 972-4-555-1234, will get turned around when sent to another orientation computer resulting in 1234-555-4-972.

Synchronizing desktop tools is more difficult than it looks on the surface. It includes not only word processors and spreadsheets but also email, drawing packages, instant messengers (IMs) and operating systems. Even if all of these packages can be made identical, configuration may still be an issue. The most common problem is spell-check files. The simple example for English speakers is American English versus British English. This issue can be solved (albeit with a struggle) by standardizing on a common dictionary, defining procedures and using tools that allow easy switching between dictionaries.

### Communication

Communication is a big problem with any project. When the team is separated by cities, time zones or continents, communication can consume the Project Manager. Some issues to consider:

- When setting up finances for the project, make sure there is a significant travel budget. Nothing is better than a face-to-face meeting to build a team or solve a problem.
- Try to align working times. If there are only one or two time-zones different, have the team stagger their hours to bring them more in line. A three time-zone difference may mean one team starting an hour early and another starting an hour later than normal.
- Some countries do not work Monday through Friday. Some countries work Sunday through Thursday; for instance, Israel. Trying to synchronize teams between North American Eastern time-zone and Israel illustrates the problem. When the east coast gets to work Monday at 8:00am, the Israeli team has been at work for nearly two days, it is 3:00pm on the second work day of the week—effectively a 31-hour time-zone difference, or 40% of the workweek! For North American west coast it is even worse with no time during the normal workweek when business hours overlap. In this situation the project may need volunteers in both countries to work a day on the weekend to aid communication.

### Tools to Help

Everyone—the client, the Project Manager, individual contributors and the stakeholders need visibility into the progress of the project. Make sure there is a common place they may look for the answers. Collaborative Project Management systems are available and keep schedules, tasks and documents synchronized throughout the project life. These are web-based tools and invaluable to keeping a team in touch.

Virtual meeting rooms (Lotus Notes®, eRoom®, SharePoint®, etc.) are great tools for any project. They must be budgeted into the project, procedures developed for their use and a support group assigned. They can be an excellent tool if used properly.

Project status tools (e.g. Planview® and Microsoft Project Server®) are also available. These tools allow team members and management to update and monitor project progress at any time over the web. Some points to consider about their use are:

- Great for storing data, but the data is often not real time. If the data is not up to date, people will wander away from trusting this as a source. Executive sponsors are useful in promoting their use. If the team realizes that the project status, and their bosses view of their participation in the project, hinges on this data being accurate, then it is more likely to be maintained.
- Getting people to update completion percentages on tasks is difficult but critical. To keep these data stores current, the Project Manager and team leads need to build a project culture that makes using these tools second nature.
- Use of automated "stoplights" (red, yellow, green project indicators) is valuable in getting status across; not only to management but also to the team. If a portion of the project is "red" simply because data has not been updated, then the team member has a strong desire to update the system.

Meetings can be held using web tools and con-call lines—NetMeeting® and WebEx® are two such web tools. Presentations can be delivered without having to keep people coordinated on a page, the presenter runs the presentation and participants can be anywhere in the world.

### Email

Emails are a staple of the business world. But people get so many emails that it is difficult to discern the important ones. This greatly diminishes the usefulness of the tool. Additionally, a good email takes time to write, time that might be better spent on the phone carrying on a dialog and preventing misinterpretation from the start. Establish email rules:

- Do not argue in an email. If an email thread starts getting the slightest bit defensive, cut it off and pick up the phone.
- Do not expect the email will get read. If it is important, call them and tell them to read it and follow up with a call to discuss the email. Writing is a great way to organize your thoughts and create a record of what was said, but others may not read it as intended. Always

have a conversation on important emails.

- An email, although it looks like writing is not "in writing" like a Change Request. Make sure this is fully understood.

### Instant Messengers

Email is the primary communication tool of business today, probably even outranking the telephone. But a new tool's use, Instant Messenger (IM), is on the rise. Often thought of as a tool for kids and having little use in business, remote teams (or home office individuals) find they have a very valuable place in communication. Instant Messengers can provide the following benefits:

- Quick answers to simple questions from other team members.
- Creating a team environment by allowing team members to visualize your team on the screen. A form of seeing who is at their desk.
- A polite way to "Yell across the cubicle wall" without destroying one's train of thought.

Here are some things to consider prior to diving into instant messaging:

- Some companies will not allow them since they can be misused.
- There are a number of IM products—AIM®, Windows Messenger®, Yahoo! Messenger®, ICQ®, IRC® are the major ones. Many people already have their favorite. Other products like Trillian® are just User Interfaces that allow all IMs to be used in similar manner.
- It is best to use an IM (i.e. Trillian®) that has a conversation log. This allows the creation of a "paper trail" like email.
- Discourage acronyms and "short-cut spelling"; these are too easily misinterpreted.
- Do not use an IM for long conversations or for communicating with multiple people. This is best done in an email or phone call. IM is good for quick non-elaborate questions. A good one is "Is now a good time to call?"

### Face-to-Face

Nothing beats face-to-face. There still needs to be an occasional physical meeting of the team. This is important to allow team members to develop a better rapport with one another. Personal contact allows people to match the each others demeanor, body language and vocal tone with the rest of their communication. Budget this into the project and schedule them at a minimum of quarterly. ☺

eCameron, Inc. can provide presentations to your staff or organization on any subject covered in our newsletters. For more information contact:

Todd Williams

Phone: 1-360-834-7361

Email: [todd.williams@ecaminc.com](mailto:todd.williams@ecaminc.com)

## Book Synopsis

**Agile Project Management: Creating Innovative Products** by Jim Highsmith discusses the philosophy and reasons to manage projects using the Agile Project Management methodology. It also provides eighteen practices to use as a template when implementing agile management principles.

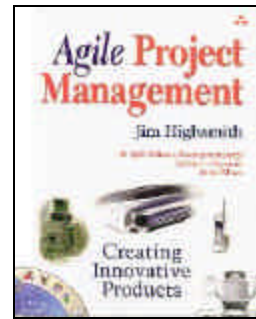
### Summary

**Agile Project Management: Creating Innovative Products** introduces the reader to agile project management methodology by defining the role of the Project Manager, and to a lesser degree defining the Product Manager's role, in relation to the development team, the customer and the stakeholders. The author spends significant time explaining the philosophy of agile projects using "lean" principles and the resulting changes in mindset and style that may be needed by a Project Manager. Counter arguments are presented for a number of myths that have developed as people have tried to refute agile as a legitimate project management process. The book explains the reasons to use this approach for projects and enumerates eighteen suggested practices and associated reports.

The book is broken into three parts—basic information (chapters 1-4), Agile Project Management (APM) methodology phases (chapters 5-8), conclusion (chapters 9 and 10). The first part explains the principles and concepts and dispels myths about Agile Development and APM. The second part of the book, describes the phases, practices and sub-practices of APM. Lastly, part three discusses scalability and concludes the discussion.

### Basics of the APM Methodology

The first chapter (The Agile Revolution) describes the genesis and need for agile projects. Highsmith describes the five business objectives of managing agile projects as being: continuous innovation, product adaptability, reduced deliv-



### Agile Project Management: Creating Innovative Products

By Jim Highsmith

Addison-Wesley, March 2004

277 Pages

ISBN: 0321219775

Buy this book at Powell's:

[powells.com](http://powells.com)

ery schedules, people and process adaptability and reliable results. He contrasts this (see Table 2) with conventional project management practices of generating and rigidly following detailed plans and designs and adhering to these documents for the life of the project.

Throughout the chapter he attempts to dispel the critic's view of Agile Project Management. He states that most of the criticism is based on exaggeration of the core APM principles. For example, he explains (multiple examples are given in the book) the differences in APM's view on documentation—no documentation (critics view), barely sufficient (APM view) and comprehensive and exhaustive (classic Project Management). He does the same for other criticism.

The primary goal of agile projects is to respond to change in order to deliver value to the customer. Compared to conventional project management, the iterative process with ever changing requirements appears chaotic. Highsmith uses two allegoric terms to help describe this attribute. He introduces the term "Chaordic" (coined by Dee Hock) which is a portmanteau of the words chaotic and ordered. He also refers to APM as an enabler for complex adaptive systems (CAS) design. Both of these terms reflect the adaptive structure capable of accommodating to change.

He describes the current problematic state of classical pro-

Agile PM	Classic PM
Respond to change	Follow a rigid plan
Envision-Explore	Plan-Do
Exploration	Production
Adapting	Anticipating
Working Products	Comprehensive documentation
Customer Collaboration	Contract Negotiation
Individual and Interactions	Processes and Tools
Envision-Explore-Adapt	Plan-Design-Build

Table 2: Agile versus Classical Project Management

### Lean Project Management; Eight Principles for Success

by Lawrence Leach

Now available as an eBook. Learn how to reduce waste that causes project delays:



- Align project stakeholders to achieve success.
- Create a project solution that really meets stakeholder's needs.
- Plan your work to complete contributions with little stress.
- Apply simple project measurements.

Complete your projects in "half the time, all the time."

Currently on sale for only \$29.95 at:

<http://www.advanced-projects.com/Products/LPM.htm>

ject management as achieving “administrative excellence” losing focus on the product and the deliverables. This quality of project management violates the lean principle of reducing waste—it does not add value to the end product. Although some administrative functions (Highsmith uses the term “compliance functions” to imply documents and procedures that are externally required for the project) may be needed, the level attained in today’s projects is such that they stifle innovation in product development.

Having provided the general definition for APM, Highsmith proceeds to describe the core focus of agile projects and the guiding principles of customers and products. He claims that classically managed projects focusing on process, cost and control lose sight of the goal of delivering products of value to the customer (the prime directive of an agile project). To do this a number of changes are required in project management philosophy. Project methodology must be realigned so that it is adaptable to changes in the business world and customer requirements, processes must be lean and focus on delivery rather than compliance to procedures. One must look for the value chain in the project, minimizing waste. The project team needs to continually strive to meet the four tenets of lean thinking—specify value, identify the value stream, enable continuous flow and instill a pull mentality.

To achieve this he claims that APM’s iterative, feature based delivery is superior to classical project management techniques. Getting “somewhat functional” products to the customer for review, comment and change will provide a faster delivery, happier customer and lower risk project. In short, instead of being focused on administration, projects need to focus on technical excellence

Progressing toward the heart of the book, Highsmith then describes the concept of Leadership-Collaboration Management. Highsmith claims and provides an argument that projects for innovative products are not repeatable. By the nature of the fact that they are innovative, they will encounter issues not seen in the past and require changes that cannot be anticipated. Therefore, these projects need to be managed differently than conventional project methodologies define. Interactions with customers must be open to change and adaptable; the processes must be lean.

To achieve this, Project Managers must have the authority to build teams with the following qualities:

- Above average skill set,
- Adaptive to changes in requirements,
- Self-organizing,
- Self-disciplined.

When this is accomplished, the Project Manager needs to refocus his or her energy on:

- Articulating the product vision,
- Encouraging interaction,
- Following participatory decision making processes,
- Implementing a system that allows for and insists on accountability (individuals choose their work),
- Using a style of steering, as opposed to controlling, progress.

In order to create a lean project, processes must be simple

and barely adequate—eliminating processes that are wasteful. It is the Project Manager’s responsibility to facilitate this. Highsmith claims that many elaborate processes are often implemented in order to eliminate mistakes and accommodate deficiencies in individuals on the team. Therefore when reducing the number of procedures. Project Managers must be tolerate of mistakes, educate the team and not over react by implementing restrictive processes to compensate. They must concentrate on what is “barely sufficient” from a compliance and control point of view. To achieve this, they must be highly critical when selecting individuals for the team.

One of the many problems that Highsmith discusses with current management practices is the hiring process. He claims that employee selection philosophies are not based on getting the best people. In order to produce innovative products, better than average people are required. He puts great emphasis on this as the foundation of his statement that the Project Manager needs complete authority over the selection and retention of people on the project.

To conclude the high-level discussion of APM, Highsmith introduces the Agile Project Management Model. This short chapter briefly describes the five phases of an agile project (Envision, Speculate, Explore, Adapt and Close).

At this point the book provides the detail of the APM Methodology. The subsequent four chapters provide detailed descriptions of each phase and the practices suggested in those phases.

### Envision Phase

Envisioning, rather than detailed requirements, paints a high-level view of the deliverable. A number of concise project documents are generated in this phase. A possible set of documents are described, these include: a Vision Box, an Elevator Statement, a Vision Document, a Feature Breakdown Structure and a Project Data Sheet.

In addition, during this phase the team members are selected and the process of forming a team is started. Team members are also grounded in the methodology that will be used. Since agile projects are flexible by definition, this is a requirement even with an experienced team so that they understand the practices to be used in this application of APM.

There are six practices in this phase:

**1. Product Vision Box, Elevator Test Statement and a Product Vision Document.** These are developed in a one-to three-day process of working with the customer and/or Product Manager. Creating the product vision consists of developing three items:

a) Vision Box. A Vision Box is a box that would conceptually hold the product. It is designed to show the primary functions and features of the product being built. This vehicle is used since it forces the team to think of concise selling points for the product and helps define the product value.

b) Elevator Statement. This is a paragraph that describes the product. Highsmith provides a structure for an elevator statement as follows:

*(Product name) is for (target customer) who (statement of the need or opportunity) the (product name) is a (product*



category) that (key benefit, compelling reason to buy) unlike (primary competitive alternative) this product (statement of primary differences).

c) **Product Vision Document.** The aforementioned items, as well as customer satisfaction measures, key technologies, operational constraints, competitive analysis and financial indicators comprise a Product Vision Document. This document should be between two and five pages.

2. **Product Architecture.** This defines the features of the product using a form called a Feature Breakdown Structure. This is analogous to a Work Breakdown Structure but identifies features not tasks. Key concepts that Highsmith defines are:

a) **Guiding Principles.** The initial list of the guiding operating principles for the development team is generated. As opposed to classically managed projects, agile projects (even for similar products), may differ drastically. There may be a need for the addition or removal of compliance materials, communication tools or practices to satisfy unique requirements of the customer or product. This will be elaborated further in the development iterations.

b) **Project Data Sheet.** A project data sheet is developed that identifies a variety of project level parameters—customer, Project Manager, project objective statement, tradeoff matrix, exploration factor, delay cost, features, customer benefits, performance/quality attributes, architecture and issues and risk. The book provides a brief description of each data element and detailed information is given for Tradeoff Matrix and Exploration Factor.

c) **Tradeoff Matrix.** All projects have four major variables (scope, schedule, stability and resources) that influence the project. A tradeoff matrix attempts to reduce the variables by making one Fixed (i.e. must deliver by a certain date), a second item Flexible (i.e. should be about X dollars) leaving the other two variable. This establishes a foundation for limiting the types of change.

d) **Exploration Factor.** The exploration factor is a barometer to risk and uncertainty in the project. By looking at the maturity of the technology (“bleeding edge” to “well understood”) and the stability of the requirements (“fluctuating erratically” to “stable/unchanging”) a numeric gauge to the amount of exploration required is assigned.

3. **Get the Right People.** Highsmith makes it very clear, the goal of an agile project is to be adaptive and the key to making this work is having self-disciplined team members. He is insistent that the Project Manager must have complete control over team member selection and the team must have an above average skill set. Without this the chances of success are greatly diminished. As in earlier chapters, he claims that the requirement for most procedures is undisciplined, below average performers. Qualified, self-disciplined team members with a customer-first attitude require minimal procedures and are the most important component to an agile project.

4. **Participant Identification.** This practice defines the individuals involved in the project and their contribution. Broadly categorized, customers define features, project

team members build the product and stakeholders apply constraints. Classification of these participants should be expanded with enough detail to understand their interaction with the team and each other. This will also outline the Project Manager's responsibility to the customer, the team and the stakeholders.

5. **Customer/Developer Team Interface.** The importance of a Product Manager (especially in IT projects, which often lack one) is pointed out. The Product Manager manages the customer where the Project Manager manages the team. The Product Manager is responsible for ensuring that information flows to and from the development team completely and that the product vision is not lost.

6. **Process and Practice Tailoring.** Due to the premise that innovative products require a lean, non-predictive development cycle, minimal documentation and processes are key to the agile philosophy. To that end, agile projects need to be tuned to the situation in which they are applied. Three key areas need to be defined:

a) **Self-Organization Strategy.** Collaboration and communication guidelines for the organization need to be established. In order to minimize the command and control aspects of a project and promote self-organization principles, the Project Manager must guide the team in understanding what communication is required.

b) **Process Framework Tailoring.** The milestones, gates and project artifacts must be defined. This must be a minimal set that adds value to the product and provides stakeholders and the customers with a gauge of project progress.

c) **Practice Selection and Tailoring.** Some projects need more processes than those that are defined in this book and some need less. This practice will define the minimal number of process steps in the project.

d) **Early Planning.** Planning for items with long lead times (customer focus groups, acquisition cycles for capital items, etc.) may need to be started early to guarantee acceptable results.

### Speculate Phase

Speculation, rather than detailed planning, follows envisioning (project definition). Since agile projects focus on features rather than tasks, this phase defines what the features will be and the order in which they are developed and released. This phase should assist the project team by:

- Determining how the product and feature-set evolves,
- Balancing customer anticipation with product adaptation,
- Identifying high value deliverables or lynch pin features early,
- Helping understand business goals and customer expectations,
- Framing the budget and schedule requirements,
- Providing tradeoff data to evaluate requests for change
- Coordinating feature/activity dependencies,
- Defining the adaptive process in the context of this project,
- Explaining how to analyze events that occur during the project,

Recommended practices in the speculate phase are:

1. **Product Feature List.** This process expands the list of features formulated in the first three practices in the Envisioning Phase. It will add the detail required by the development team to complete the feature.

2. **Feature Cards.** For each feature listed in the prior practice, a Feature Card must be developed. Feature Cards are 4"x6" cards (easily moved around on a table) that contain the feature ID, name and description, planned iteration, feature type (customer or technology), estimated work effort, uncertainty of requirements, dependencies and acceptance tests. These will be used in the release, milestone, and iteration planning practice to follow.

3. **Performance Requirements Cards.** Performance Requirements Cards define the performance characteristics of a set of features. They are assigned to a given iteration or multiple iterations and the acceptance tests are identified. Performance Requirements cards contain the performance ID, name, description, anticipated difficulty to achieve and the acceptance tests.

4. **Release, Milestone, and Iteration Plan.** This practice is the equivalent of developing a high level schedule. Features cards are assigned to iterations based on level of risk (the higher the earlier), the value of the feature to the customer (the greater the earlier) and inter-feature dependency. Numerous tools and examples are given in this section to help with laying out the project structure.

Iterations may take on various forms. Highsmith provides three examples: complete plan, all features assigned to an iteration; two iteration plan, only the current iteration has features assigned, all other features are unassigned; iteration-by-iteration, all features are assigned to the current iteration, other features will be determined after completion of the iteration. Goals of the planning phase are:

- i) Identify Risk that effects planning. Identify any unmitigated risks as early as possible to avert additional costs and minimize project cancellation cost.
- ii) Identify schedule target.
- iii) Establish project milestones and iteration periods.
- iv) Develop a theme for each iteration. This helps constrain adaptation within an iteration.
- v) Assign feature cards to iterations, focusing on delivering working product (not necessarily feature rich) to the customer as soon as possible.
- vi) Summarize the plan in a easy to disseminate manner.
- vii) Calculate the schedule and staffing requirements.
- viii) Adjust the plan as necessary.

a) **Iteration 0.** In this iteration customer features are not delivered. Since many innovative products need new technology, this would be the step to do the proof of technology. Iteration 0 is the initialization step.

b) **Iterations I through N.** All iterations should focus on delivering customer value and minimizing risk. To help restrict changes, iterations are fixed time periods. In order to complete an iteration on time, tradeoffs must be made. Each iteration should have a defined theme that will identify the type of features that will be delivered in the iteration. Rework and contingency should be fac-

tored into each iteration to account for issues remaining from or uncovered in previous iterations.

c) **Next Iteration Plan.** Prior to starting each iteration, team members sign up for work they will do in that iteration. Two critical points that Highsmith makes are that the team members should choose what they work on (key to accountability) and the assignments are done immediately prior to building the feature, allowing for maximum flexibility.

d) **First Feasible Deployment (FFD).** Throughout the planning process the team should be looking for the first opportunity in which the product can be deployed to the customer—increasing ROI and improving customer input resulting in increased value and lower cost.

e) **Estimating.** Highsmith points out that when estimating time on agile projects one must consider that they are estimating for a feature, not a task, hence one needs to account for all work to make the feature deployable.

f) **Scope Evolution.** As opposed to myth, scope is not allowed to change indiscriminately in an agile project. Scope is allowed to change within the bounds of the vision and iteration time span. The heart and soul of agile projects is to allow the customer to change their minds, for each iteration.

g) **Risk Analysis and Mitigation.** Agile projects are better designed to handle certain types of risk than classically managed projects. This is accomplished by:

- i) Heavy team involvement in planning and estimating,
- ii) Early feedback based on delivery velocity,
- iii) Constant pressure to balance the number and depth of features with the schedule constraints,
- iv) Close interactions between engineering and customer teams,
- v) Early error detection/correction to keep a clean and functional product.

### Explore Phase

The goal of the Explore phase is to develop and deliver tested features in a short timeframe. Explore and adapt iterations allow for repeated additions/alteration of features and functions, evaluation of progress and adjustment of subsequent iterations to attain better results. The Project Manager's role in this phase is to facilitate the creation of a self-organized, self-disciplined team. To do this the Project Manager must transform the individuals into a team by:

- Focusing the team on delivering results,
- Developing each individual's capabilities,
- Removing roadblocks from the project,
- Providing the required resources,
- Coaching the customer,
- Creating and orchestrating the team's workflow and interactions.

Recommended practices for the Explore phase are:

1. **Workload Management.** Workload Management is maintained by the team and enabled by the Project Manager. Self-organizing, self-disciplined teams need to manage their own workload in order to maintain their accountability. The Project Manager needs to monitor their effec-

tiveness and assist by providing resources required to meet the team's assumptions made when planning, each iteration.

2. **Low-Cost Change.** Two points should be kept in mind 1) the iterative development cycle is required for and enables change, and 2) the goal is to deliver future value in the delivered product by making it easy to adapt to new situations. To do this five factors should be watched:

- **Technical Debt.** Technical Debt is the difference between doing something the "right-way" versus getting it released quickly. This must be minimized. Minimizing technical debt is the single most important factor in delivering a maintainable system. Keeping technical debt to a minimum and refactoring (below) creates a system that is easier to change and is responsive to customer needs.

- **Simple Design.** Keeping the design simple is a key concept enabled by the iterative style of agile projects. Since the design will evolve during the development process there is no need to build in complex solutions for problems that are anticipated and not yet encountered. If a problem is not immediate or fully defined, its solution should not be implemented until a later iteration. Actual versus anticipated problems can be addressed more completely.

- **Frequent Integration.** Frequent integration within an iteration, builds in quality and highlights potential issues in the design. Each iteration should have frequent integration.

- **Ruthless Testing.** Ruthless Testing, a term coined by Kevin Tate, is a variant of the "test-first development"—quality being built in—a continual objective of each developer.

- **Opportunistic Refactoring.** Keeping the design clean and up to date, must be a guiding philosophy of the management team as well as the developers. Any product that undergoes modifications will have inadvertent shortcuts taken. In addition, new technologies may be developed rendering parts of the product more difficult to maintain than need be. Refactoring the product will address both of these and improve the responsiveness to the customer, increase the longevity of the design and increase the profit.

3. **Coaching and Team Development.** Next to getting the right people, attending to the improvement of the individuals in the team is the most important task in making sure the project team is the best it can be. Coaching is not limited to the development team itself, but must be extended to the customer and stakeholders.

- **Focusing the Team on Delivering Results.** Due to the nature of agile projects, maintaining focus on each iteration and its relation to the final project, as a whole, is a difficult task. Individuals must be coached to maintain focus on what is needed and to only add additional functionality that is essential to the iteration. Since there is a fine line between what is essential and what would be nice to have, the team needs assistance in understanding added-value and "nice to have" features.

- **Developing Each Individual's Capabilities.** Team members must develop both technically and interperson-

ally. Everyone has room for improvement. Individuals should be fostered in adopting behaviors that include:

- Accepting accountability for results,
- Analyzing situations to determine root causes,
- Engaging in intense debate and interaction,
- Willingness to work in a self-organized environment,
- Respect for one's colleagues.

- **Molding a Group of Individuals into a Team.** By developing each individual's capabilities one goes a long way in building a team. But this must be furthered by the manager exhibiting trust in his or her group and facilitating interaction and debate to achieve technical excellence. Teams will form without a manager's influence; they will form much faster, though, with proper managerial guidance. Teams must be able to aggressively debate issues and solutions and come to an agreement on a decision. Consensus is not required, agreement is. To help, a set of guidelines should be formulated, these may include:

- Everyone having an equal voice,
- Everyone's contribution being considered valuable,
- Attacks focus on issues, not other people,
- Team privacy,
- Respect for each other's differences,
- Participation of all members.

- **Providing the Team with Required Resources and Removing Roadblocks.** The Project Manager must provide the team with the needed managerial support by supplying resources and eliminating roadblocks to keep the team focused on their task at hand and not distracted or slowed down.

- **Coaching the Customers.** Agile projects not only change the method in which the team interacts amongst themselves, but also how they interact with the customer. In agile projects, customers are much closer to the project and team. This is vital to adding value. Just like the team needing to be educated by the Project Manager on the ways of agile development, the customer must be educated by the Product Manager.

- **Orchestrating Team Rhythm.** The Project Manager needs to assist the team in finding a rhythm, a smooth interaction process, to keep them in sync.

4. **Daily Team Integration Meetings.** Due to the speed of development in an agile project, daily meetings are required. Short cycle times make problems surface quickly and resolution critical. These meetings should "inform on" not "discuss" progress and impediments. The goal of these short, efficient daily meetings should be held to:

- Review the prior day's accomplishments,
- Enumerate plans for the day,
- Highlight issues, obstacles and needs.

5. **Participatory Decision Making.** There are many changes that are required to make individuals and teams more accountable. One of the most effective is to make them part of the decision process. Participatory Decision Making is a technique that involves the team, stakeholders and customers in the decision process. It insures they are informed and their concerns have been heard and, to a degree, ad-

dressed. Participatory decision making is NOT consensus; it is a process to ensure that all are in agreement to support the decision.

• **Decision Framing.** Involving the correct people in a decision is decision framing. The decision group is selected by a predetermined set of criteria, typically:

- People impacted,
- People providing input,
- People required for discussion (i.e. subject matter experts, etc.),
- Decision makers,
- Reviewers and approvers,
- People needing to know the results of the decision.

• **Decision Making.** Highsmith illustrates a decision gradient tool to use in determining if a decision has been accepted. Since this is not a consensus process, the decision gradient can help ensure that the people involved in the process will support (not necessarily agree with) the decision.

• **Decision Retrospection.** In order to achieve continuous improvement, all decisions should be evaluated for their effectiveness. A retrospective look at each decision should be performed and what could have made that decision better should be noted.

• **Leadership and Decision Making.** Teams need leaders, not managers. Some attributes of a leader include: supporting the team, stepping in to break a stalemate in a debate, absorbing the ambiguity in the project and being accountable and responsible for problems with decisions.

• **Set- and Delay-Based Decision Making.** Design and decision sets, as used in Toyota, are critical to responsiveness. This technique entails looking at a set of solutions for a problem and delaying the decision as long as possible until a well informed decision can be made. This accomplishes coming to a best answer as late as possible, allowing the maximum amount of data to be involved in the decision, hence reducing the risk

**6. Daily Interaction with the Customer Team.** As with team meetings, the customer must be apprised of the progress on a daily basis. This requirement is more important in agile projects due to the rapid development process and the need for tradeoff-decisions to meet the iteration milestones.

• **Stakeholder Coordination.** As mentioned earlier in the book, stakeholders can be generalized as the ones that apply constraints to the project. To that end, they must be kept informed and made aware of decisions and changes that may affect them. The Project Manager has the responsibility to keep them supportive of the project.

### Adapt Phase

After each Explore phase the team needs to review the results of the previous iterations and adjusts the plans as necessary to achieve better results. This is aptly called the Adapt phase. This step is a feedback process to ensure that the project is achieving the correct results from everyone's perspective. Three questions need to be answered:

1. Is value being delivered to the customers?

2. Is the project team satisfactorily meeting its milestones, cost, feature and quality objectives?

3. Is the project team adapting effectively to changes imposed by management, customers and technology?

Many aspects of the project need to be evaluated to determine what adaptations in the project are required. These include:

- Product functionality,
- Product quality,
- Team performance,
- Project Status.

Reflecting on these items and feeding the answers back into the project processes will determine the course of the next iteration.

The adaptation phase contains only one recommended practices:

### 1. Product, Project, and Team Review and Adaptive Action

a) **Customer Focus Groups.** A set of customer representatives different than the set that has been working with the team, is used to provide input to the product. This should be a larger group of customers with a wide range of duties. The engineering team may be present but is primarily there to listen and should not do anything that might stifle customer input. The facilitator should capture suggestions and report back to the group at a later time.

b) **Technical Reviews.** Internal, facilitated meetings are performed to review the technical aspects of the project. They are attended by technically competent individuals that can assess the product, compliance material (documentation), defect levels and other project statistics to assess the product's state.

c) **Team Performance Evaluations.** The team must be reviewed and that data should feed the practice of having the right people. Team and individual performance needs to be evaluated and corrective actions developed for improving the team. This must be done regardless of the level of success of the iteration—success can be for many reasons including low goals, and hence can be improved upon.

d) **Project Status Reports.** Highsmith provides a number of examples of reports to indicate the project progress. These include Parking Lot Report (percent of features delivered to a customer group), Delivery Performance (plan-to-actual features by iteration), Feature Value Delivery (cumulative value of features delivered by iteration).

e) **Adaptive Action.** As the name implies, the result of this phase is adaptation of the processes to ensure proper progress in delivering value in the product.

### Close Phase

The Closing Phase is conducted at the end of the project. In this phase the team focuses on tying up loose ends, completing any documentation or fixing the last few punch list items.

In addition, it provides input to future projects through [retrospection](#). Looking at what went right and wrong, what

behavior needs to be modified, what processes need to be improved (minimizing waste while providing barely sufficient process and documentation).

Last, but not least, there is a celebration of the completion to maintain individual's spirit and morale.

### Conclusion

The book closes with two general discussion chapters—large projects and a summary.

Highsmith outlines some of the challenges and proposes solutions in scaling agile philosophy to large projects. He claims the major hurdle is that agile methodology is a philosophy and mindset, not a written set of process and procedures. The managers and leads need to “think agile” and be capable of instilling this in the entire team. The challenge is exacerbated by the fact that the larger the project the greater the overhead simply due to the requirement for more teams to maintain communication. He recommends implementing a hub organizational structure (as opposed to the standard hierarchical structure). This structure creates functional area teams that would interact as peers. Each team would employ the practices as discussed in the book as if separate projects. This requires:

- Getting the right leaders,
- Articulating the work breakdown and integration strategies,
- Facilitating interaction between teams,
- Framing project-wide decision making.

To do this the Project Manager must focus on interactions rather than activities.

He introduces a concept of ‘Commitment-Accountability Protocol’ (CAP). CAP is commitment between teams on a given feature delivery. This practice extends the previously discussed individual accountability to teams. He proposes using a Commitment-Accountability Protocol Card, similar to a feature card, for coordinating teams on their deliveries.

The book closes discussing Reliable Innovation summarizing the book by reinforcing multiple points underscoring the point that agile project management is the process of creating a “value-adding Project Manager”.

\* \* \*

If a Project Manager works on new product development or software development, *Agile Project Management: Creating Innovative Products* is certainly worth adding to their library. There are only a few areas where the book tends to drag or repeat points. The eighteen practices are well described and, as pointed out in the lead article, sound advice. The primary distraction to the book is the fictional dialog that prefaces most chapters, but these may be skipped and no content of the book is lost.

\* \* \*

(This article was the basis for the Wikipedia article on book “Agile Project Management”. More discussion may be found there:

[http://en.wikipedia.org/wiki/Agile\\_Project\\_Management\\_\(book\)](http://en.wikipedia.org/wiki/Agile_Project_Management_(book)))

## Resources and Templates

eCameron's website contains a large variety of reference materials on Project Management subjects. These include templates, processes and further discussions on a variety of topics. Please feel free to browse our site at <http://www.ecaminc.com>

Or contact: Todd C. Williams, Phone: 1.360.834.7361 e-mail: [todd.williams@ecaminc.com](mailto:todd.williams@ecaminc.com)

Subject	Description	Format	Location
Templates Homepage	Home page for the items listed below	Various	<a href="http://www.ecaminc.com/Templates/Templates.html">http://www.ecaminc.com/Templates/Templates.html</a>
Change Management Process	A complete Change Management process document.	Word	<a href="http://www.ecaminc.com/Templates/CRProcessWord.html">http://www.ecaminc.com/Templates/CRProcessWord.html</a>
Change Log Template	An Excel template for a change log.	Excel	<a href="http://www.ecaminc.com/Templates/CRLogXL.html">http://www.ecaminc.com/Templates/CRLogXL.html</a>
Change Request Template	A Word template change request form.	Word	<a href="http://www.ecaminc.com/Templates/CRFormWordDot.html">http://www.ecaminc.com/Templates/CRFormWordDot.html</a>
Estimation Template	An Excel template for estimating project changes.	Excel	<a href="http://www.ecaminc.com/Templates/EstimateXL.html">http://www.ecaminc.com/Templates/EstimateXL.html</a>
Executive Presentation Material	Various presentation ideas and templates for concisely expressing complex data to executives.	Power-Point	<a href="http://www.ecaminc.com/Templates/ExecSummary.html">http://www.ecaminc.com/Templates/ExecSummary.html</a>
Generic Project Document Template	General project template to be used for specification and other control documents.	Word	<a href="http://www.ecaminc.com/Templates/ProjectDocDot.html">http://www.ecaminc.com/Templates/ProjectDocDot.html</a>
Meeting Minutes Template	A Word Template for Meeting minutes.	Word	<a href="http://www.ecaminc.com/Templates/MinutesWordDot.html">http://www.ecaminc.com/Templates/MinutesWordDot.html</a>
Risk Tool	A risk probability accumulator. Aggregates risk into a project level impact.	Excel	<a href="http://www.ecaminc.com/Templates/RiskToolXL.html">http://www.ecaminc.com/Templates/RiskToolXL.html</a>
Time Reporting	Excel Template for tracking time reporting	Excel	<a href="http://www.ecaminc.com/Templates/TimesheetXL.html">http://www.ecaminc.com/Templates/TimesheetXL.html</a>
Previous Newsletters published by eCameron.		PDF	<a href="http://www.ecaminc.com/ColProjMgmt/CPMIndex.html">http://www.ecaminc.com/ColProjMgmt/CPMIndex.html</a>
Recommended Reading		HTML	<a href="http://www.ecaminc.com/Books/BookHome.php">http://www.ecaminc.com/Books/BookHome.php</a>
General Discussion Topics		HTML	<a href="http://www.ecaminc.com/ColProjMgmt/ColProjMgmt.html">http://www.ecaminc.com/ColProjMgmt/ColProjMgmt.html</a>

**Disclaimer:** The product names are the copyrights of their respective vendors. The data provided in this newsletter carries no warranty, is the opinion of the author and has not been reviewed or approved by the product vendors. This newsletter is free of charge and has not been funded by product vendors or clients or customers of eCameron, Inc.