

# **Developments in Agile Project Management**

## **Mike Griffiths, PMP, Project Management Consultant, Agile Recruiting Ltd**

### **Abstract**

As agile methods are more widely used, particularly in enterprise environments, people are augmenting and morphing the core practices to ease acceptance and broaden their applicability. This paper looks at why agile methods have become popular on software development projects and identifies some emerging trends. These include the integration with existing disciplines such as Lean Production, Six Sigma and leadership best practices. In addition the alignment with “Generation Y” worker motivations and possible future trends are also covered.

### **Introduction**

Most of today’s Agile methods were created in the early 1990’s and others can trace their roots back much further. However, it has only really been the last five years that have seen their adoption and popularity explode into the mainstream enterprise world. With this expansion has come further evolution and adaptation as a wider audience is exposed to their strengths, weaknesses and grey-areas.

Along with the growth has come commercialization, proceduralisation, and other inevitable mis-directions that concern many of the original pioneers as companies try to cash-in on the demand and hype surrounding the new emerging processes. Nevertheless, many valuable tools and techniques are emerging that can be effectively used by all project managers, especially when faced with challenging projects with high rates of change.

#### **The challenges of dynamic software development projects**

Software development projects present unique challenges for traditional project management approaches. First, software is intangible and difficult to describe well; rarely is the same system built twice which makes analogy to existing functionality problematic. These issues can lead to “evaluation difficulties”, where mismatches develop between interpretations of original requirements and customer goals.

Iterative development has emerged as a technique for overcoming these evaluation difficulties by using regular review points. In addition to reviews, closer collaboration between users and development teams (and subsequent adjustments) allows the system to evolve towards the true business requirements. When users are given an opportunity to refine requirements based on evaluating early prototypes, it is common for there to be significant differences between the originally stated requirements and the true business needs. IKIWISI – “I’ll Know It When I See It” best describes the situation.

Also, the process of executing a software project can be a complex and difficult to schedule activity. Unlike many construction-based projects, writing software in today’s rapidly evolving languages is not a defined, repeatable process. Software development is often an unprecedented, research and development based process. Attempting to create detailed task-oriented plans for software developers is likely to lead to fragile, soon abandoned plans or too much project management time spent updating plans rather than managing the project.

Finally, well-structured software differs from conventional engineering by allowing certain changes to be made even late into the project. For example, validation logic could be moved from a client screen, to a middle layer, or even implemented as a trigger in the database layer with minimal impact on the remaining application. Such a change to the design is not usually possible in a physical engineering project. This “modifiability” allows software projects to add or evolve requirements to deliver maximum business value against a backdrop of changing business needs.

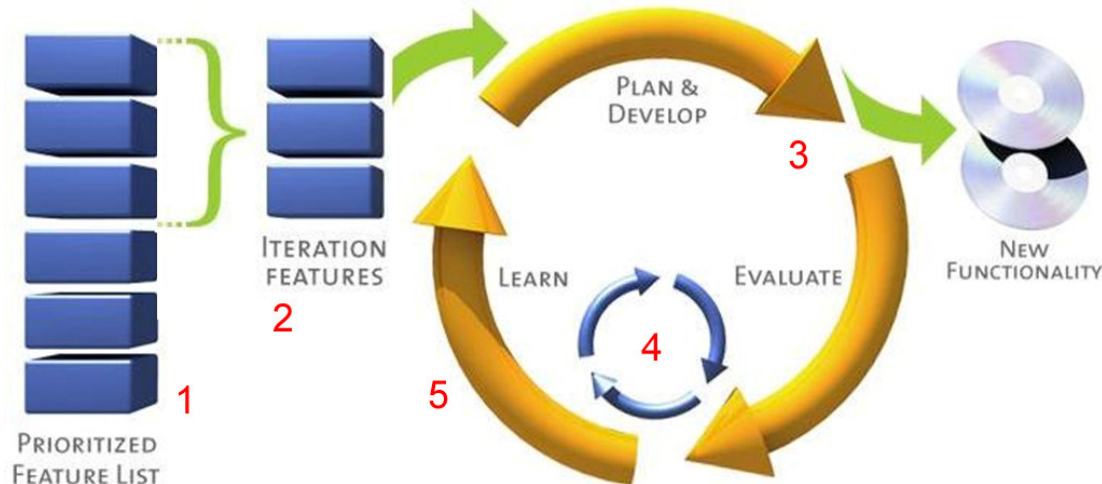
#### **Why agile methods work for software development projects**

Agile methods have emerged that acknowledge the evaluation difficulties of software, leverage modifiability and handle execution risk. They recognize that validation of requirements is best done early and often against an evolving system. Agile methods employ prioritized requirements schemes that encourage changes to be traded-off against original requirements throughout the lifecycle, thereby allowing the delivery of late changes that can provide competitive advantage to business.

Finally, agile methods do not attempt to micro-manage developers via prescriptive task plans. Instead mechanistic processes, premature decomposition and one way communications that belie the volatility of software development are swapped for more humanistic, goal directed approaches that utilize people’s ability to manage complexity.

### The Workings of an Agile Project

The basic elements of agile execution and control processes are depicted in Exhibit 1. This diagram does not show all of the project activities, but serves as a guide to processes discussed in more detail later.



**Exhibit 1 – Agile Control and Execution Processes**

In Exhibit 1, (1) represents a prioritized list of requirements or features that are to be built. Requirements are assigned a priority by a business (user or sponsor) representative to rank their value or strategic importance. A subset of these prioritized requirements or features is then selected (2) for development. The selection is based on the highest priority features remaining to be developed.

This subset of features then undergoes analysis, development, testing and evaluation during a short, fixed time iteration (3). Recommendations for the durations of these iterations vary between agile methodologies, but 1-4 weeks is common. Within these cycles, team members self-select work based on what is valuable and practical.

Inside this 1-4 week cycle there is a daily communications and risk assessment cycle (4). This is conducted as a short daily meeting where development team members briefly answer:

1. What have you been working on since the last meeting?
2. What are you working on today?
3. Do you have any problems, issues or impediments to making progress?

Via these short daily meetings, project stakeholders hear about incremental progress, team members learn what others are working on. Also, roadblocks and risks are raised quickly for removal and mitigation by the PM.

Agile methods also perform mid-project retrospectives (5) where the lessons learned questions of “What went well?”, “What did not go well?”, and “Recommendations for the future?” are captured and factored into the planning of the next iteration.

Recently it has been acknowledged that this focus on high-level objectives and autonomy for team members to self-select work; has more similarities to leadership guidelines than traditional project management guidelines.

### Development # 1: Recognizing the Link to Leadership

The table shown in Exhibit 2 illustrates how leadership focus diverges from traditional management focus and how Agile methods actually align more closely with leadership than management.

Management Focus	Leadership Focus	Agile Focus
Tasks/things	People	Business Value – business prioritized tasks
Control	Empowerment	Empowerment – team self-selects work, less command-and-control
Efficiency	Effectiveness	Effectiveness – expressed as business value
Doing things right	Doing the right things	Doing the right things - work is prioritized by the business
Speed	Direction	Direction then Speed, repeated user review to ensure development progressing in the right direction, plus tracking project progress via “Velocity” metrics
Practices	Principles	Principles / Practices – agile methods use a combination of principles and practices with the emphasis on higher level principles
Getting people to do what needs to be done	Getting people to <b>want to</b> do what needs to be done	Creating a “want to do” environment – Agile creates a better working environment by use of empowered teams and a sense of inclusion via regular retrospectives
Create Push-Work structures	Encourage Pull-work structures	Pull Structures – Agile methods borrow the pull work structures from lean production. Inventory and work in progress is kept low and requirements are elaborated at the “last-responsible-moment”

Exhibit 2 – Management / Leadership / Agile Focus Areas

While there is no single, agreed to leadership body of knowledge, experts offer remarkably similar advice. Leadership authority and author Jeffery Pinto (Pinto, 1998) defines the behaviours of effective leaders as:

- Modeling desired behaviour
- Creating and communicating a common vision
- Willingness to challenge the status quo
- Empowering others
- Encouraging others

These behaviours have strong correlation with agile project management recommendations.

1. **Modeling desired behaviour** – agile project managers should demonstrate the behaviours they wish their team to exhibit. Admit your mistakes, promote candid discussion of issues and show humility. Adopt a sharing, abundant mentality to information and focus on communications.

2. **Creating and communicating a common vision** – Teams need to have a powerful, uniting vision about the project and what the system they are building should do. People need to be pulling in the same direction and a common vision helps align their effort. XP uses the concept of a system metaphor to help create a common vision, (Highsmith, 2004) offers the “*Designing the product box*” as an alignment and envisioning exercise. (Reinertson, 1997) describes the “*Designing the brochure description*” exercise to create a common vision.

3. **Willingness to challenge the status quo** – No process is perfect or optimally configured for the unique demands of every project and organization. There need to be regular reviews to ensure the team and process are on track. In Agile projects, the end of iteration retrospectives where the team is asked, “What went well?”, “What did not go well?”, and “What can we improve for the next iteration?” Is where the status quo is challenged. This is an important leadership concept, never to accept the current process just because it is established.

4. **Empowering others** – great leaders know that people work best when empowered to make local decisions and given the freedom to self organize and think for themselves. Agile teams leverage the benefits of empowered working and gain the benefits of greater commitment and accountability. By allowing team members to self-select work and sign up for tasks voluntarily, a strong internal commitment to try and deliver that work on time to a high standard is generated. Couple with this the agile practice of holding iteration planning meetings where team members select tasks in the presence of their peers, that declares a social commitment of ownership for a task and the drive to deliver work on time becomes much higher than for tasks merely handed down from a Gantt chart.

5. **Encouraging others** – Saying “Thank You” is one of the most cost effective, yet under utilized productivity tools available to leaders. It does not take long to do, but if done appropriately with sincerity it can go a long way towards ensuring continued contribution and motivating team members. Agile projects recognize this tool and use the iteration retrospectives as opportunities to recognize contributions.

## **Development # 2: Accreditation**

Where traditional project management has certifications like PMP and PRINCE2 Practitioner, agile methods are adopting more formal accreditation schemes also. Currently the agile methods: Scrum, DSDM, and FDD have accreditation schemes and recently there have been discussions about additional programs and multi-disciplinary (not just one agile method) accreditations.

However, this is not without great debate and consternation, as for many people in the agile community, certification represents the centralized control that agile methods liberate workers from. For these people creating accreditation schemes represents: commercialization, profit chasing, and rewarding the wrong behaviours. Yet for others, it provides an opportunity to demonstrate their level of knowledge and agile methods experience, it can also provide a study program for self directed learning, and perhaps a low benchmark for hiring decisions.

The Agile Alliance and the Agile Project Leadership Network (APLN) are two volunteer lead groups that promote agile methods, organize conferences and help steward application of agile methods. The boards of both groups are populated by experienced agile practitioners and have discussed the idea of endorsing certification schemes. The Agile Alliance decided not to get involved in the agile certification business and instead issued a statement that “...employers should have confidence only in certifications that are skill based and difficult to achieve.”

Within the APLN, the board has been split on whether or not to lead the development of accreditation. However, at the Salt Lake board meeting in February 2007 a motion made by Alistair Cockburn “*The APLN commits to lead and support the creation, implementation, and evolution of an accreditation program for Agile Project Leaders based on design criteria including the DOI, with a draft proposal published by August 15, 2007*” passed 10 votes to 3.

It was felt that if more agile certification was inevitable then the APLN was well positioned to do it right. Weaknesses in current schemes were examined which included:

- lack of a difficult test
- lack of peer review and endorsement of candidates in the assessment process
- a closed model to the body of knowledge

To correct these deficiencies a Learning and Recognition group was formed to create a better accreditation program that incorporated testing, case study submission, interviews, and peer review against an open and evolving framework of agile leadership competencies. This is an ambitious target and while the sub-group has made steady progress since February, other board members are now re-questioning the initiative. At the time of writing this, a survey is being drafted to canvas APLN membership on the direction this work should take.

## **Development 3: Appeal to the Generation Y workforce**

This development is not so much how agile has evolved, but instead how the workforce has evolved to embrace it. The younger members of today’s workforce (the Gen Y’s and Millennials) are characterized by a different set of motivational factors and workplace expectations than their older counterparts.

Research (Tobe, 2003) found that while older workers value characteristics such as:

- job security
- good wages
- promotion opportunities

Younger workers ranked the following motivators higher:

- Appreciation
- feeling in on things

- understanding attitude

While it is wrong to assume that all workers from a certain era are motivated by a consistent set of factors, it is surprising how often these preferences dominate.

Agile methods recommend and encourage a set of working practices that happen to align very well with the Gen Y motivations. Practices such as frequent retrospectives where team members are recognized and thanked for their contributions and asked how the process can be improved for the next iteration appeal to the younger generations needs for appreciation and feeling in on things.

Obviously, good managers of the most traditional style projects know how to manage younger workers and can add the appropriate recognition steps to their projects, but by having the steps hard-wired into the methodology, the not-so-good managers and organizations are practicing these steps also. As a result agile methods are gaining popularity with younger workers. This is occurring to the extent that young software developers are looking for agile based organizations when selecting employers and contracts.

In the future we can expect to see many of the empowered team practices promoted by agile methods becoming more widely accepted and demanded. The command-and-control direction of workers via Gantt charts and task lists are hang-overs from a workplace dominated by industrial revolution style thinking. New team engagement models have emerged that better utilize worker planning skills and tap into the motivational factors of the largest demographic group to ever enter the workforce.

#### **Development 4: Integration with Lean Production and Lean Six Sigma**

Agile methods utilize many of the same lean principles from the Toyota Production System (Duvall, 2006). The following list shows how lean practices are integrated in agile methods.

1. **Just-in-time:** reducing inventory and working closely with suppliers to ensure that parts arrive just when needed.

**Agile Integration** – in agile projects the elaboration of (non-architecturally significant) requirements is delayed until the “last responsible moment”. There is no point building up large inventories of detailed requirements and then handing them over, in large batches, to downstream activities. Since, many of the requirements may change or go away completely before they are coded. This would be pure waste (muda), plus large batch transfers are inefficient and cause queues, and increased levels of scrap and rework.

2. **Jidoka:** At every stage of the production process Lean Production recommends employing devices allowing workers to stop production to correct defects.

**Agile Integration** – Automated build and unit test systems that stop and alert the team if ever a code check-in breaks the build or unit test suite. Multi-disciplined developers pay closer attention to quality via techniques such as TDD. Team members can raise “blocking issues” at the daily standup meeting.

3. **Kaizen:** This is a system for continuous improvement. Lean organizations such as Toyota constantly look to improve its business processes by finding ways to take Muda (waste) out of the system

**Agile Integration** – At iteration retrospectives meetings the team is asked: What went well?, What did not go well? Recommendations for the next iteration? The intent is to improve the process within this project. Lessons Learned at the end of a project is too little, too late. Issues raised at the Daily Stand-up Meeting are often pointers to things that need improvement. Agile projects are always actively looking for how to improve the process during its execution.

4. **Andons:** Wherever possible, Lean encourages the use of visual controls, or Andons, such as overhead displays, plasma screens and electronic dashboards to quickly convey the state of work.

**Agile Integration** – Agile metrics use clear visual controls over percent plan complete type measures. Information radiators, cumulative flow diagrams, burn down charts, task boards divided into To-Do / In progress / Done sections. These are all great examples of Andons in practice.

5. **PokaYokes:** Toyota practices lean production and uses a range of these low-cost, highly reliable devices throughout its operations to prevent defects.

**Agile Integration** - Build status traffic lights and lava lamps, anything simple yet hard to ignore, that helps alert the team to defects that need fixing.

6. **Genchi Genbutsu:** The literal translation of this term is, "Go and see for yourself." Rather than hear about a problem, encourage workers, team leaders and executives to go and see a problem directly and to work collectively on a solution.

**Agile Integration** – Again, Daily Stand-up Meetings are a great way to go and hear the issues from the horse's-mouth rather than interpret trends from tracking Gantt charts. Regular releases of working software are excellent ways of assessing progress and defects; go try it and see for yourself.

### **Six Sigma Concepts**

At its core, Six Sigma revolves around a few key concepts (Wikipedia, 2007).

**Critical to Quality:** Attributes most important to the customer - customer defines feature delivery priority and defect fix sequence

**Defect:** Failing to deliver what the customer wants – focus on delivering business value

**Process Capability:** What your process can deliver – transparent metrics with velocity (rate of tested feature delivery) the core metric

**Variation:** What the customer sees and feels – automate build and release processes to remove variation while focusing on customer requests over specification

**Stable Operations:** Ensuring consistent, predictable processes to improve what the customer sees and feels - automate build and release processes to remove variation

**Design for Six Sigma:** Designing to meet customer needs and process capability - focus on emerging requirements and process improvement

### **Six Sigma Processes**

Six Sigma has two main processes DMAIC (Define, Measure, Analyse, Improve, and Control) and DMADV (Define, Measure, Analyse, Design and Verify). DMAIC is used to improve an existing business process. DMADV is used to create new product designs or process designs in such a way that it results in a more predictable, mature and defect-free performance.

Both of these concepts align closely with agile's iterative, goal seeking approach. The agile retrospective procedure that questions the current process and looks for improvement opportunities is a direct application of six sigma principles. By engaging the team in reflection on what when well, what did not go well, and recommendations for the next iteration, key Analyse and Improve work is undertaken.

The Define and Measure principles are embodied with iteration planning and velocity tracking activities. By selecting a group of features to develop in a short iteration and then tracking performance against this plan and investigating the root-cause of deviations, controls that apply six sigma concepts are effectively employed.

Of course, all of these principles can be superimposed on top of traditional, waterfall inspired methodologies, but they are "baked-into" the lifecycle of agile methods and therefore less likely to be missed or neglected when under time pressures. Recently more recommendations and tools from the lean and six sigma world have been introduced into agile methods. It is now common place to hear Andons graphs actually being called Andons and DMADV cycles supported in agile tools.

## **Development 5: Tool Support**

The increase in popularity of agile methods has generated a need and opportunity for better tool support that has quickly been filled. Previously many agile projects were managed via spreadsheets, white boards and cards and now sophisticated project management systems exists to automate the process. Adoption however has been cautionary as

agile favours low-tech, high-visibility approaches that the entire team can readily engage with over complex tools that limit who can interpret them. So it is common to see task boards with “To Do, “In Progress” and “Done” sections and many tasks written on cards posted below them even when the team has access to web based agile project management tools.

The tools encompass requirements management, estimation, planning and reporting functionality and frequently integrate defect tracking and change control also. Offering like Microsoft Visual System Team Studio also combine process guidance, configuration management and time tracking all with the same integrated environment. Beyond the core development team tasks tools are also help track supporting processes. In addition to requirements lists, tools now feature risk lists and impediment lists to be managed as they continue to evolve.

### Where next for Agile Methods?

Agile methods have “Crossed the Chasm” (Moore, 2002) and entered the majority market of the enterprise environment. While this may seem threat to agile methods as they are increasingly misapplied, watered down, and then criticised when projects fail. It is nothing new and just the logical next step in Sarah Sheard’s “*Universal Technology Adoption Lifecycle*” (Sheard, 2003).

1. Discovery
2. Successful Application
3. Publicity of success
4. First (slightly modified) replication
5. Confirmation by early adopters
6. Proceduralization and implementation by uninformed middle management
7. Insufficient funding and misapplication
8. Diminished returns due to devolution of the original idea
9. Denigration of the original idea
10. Demise and new discovery

1. **Discovery** – the new approach is pioneered to find a solution in the original setting. With Agile this happened for each methodology. The Borland Quattro Pro project was an inspiration for several agile methods.

2. **Successful Application** – Methods are used with enough success to convince the originators that it is a reliable and applicable framework. For instance the Chrysler C3 project for XP, the Easel project for Scrum, the Singapore project for FDD, and the ECGD project for DSDM, etc.

3. **Publicity of success** - The technology or approach is publicized by the organizations and technology creators internally and externally. For instance the first XP and Scrum books emerge.

4. **First (slightly modified) replication** – Authors and fast-followers try the approaches, modifying them if necessary, but crucially, paying close attention to the original intent of the approach and often working in close collaboration with the initial team members.

5. **Confirmation by early adopters** – The early adopters succeed and rightly trumpet the benefits of the new technology/process. For agile this has been a long stage as many thousands of companies and projects have adopted agile and found benefits.

6. **Proceduralization** and implementation by uninformed middle management – In some organizations the process is adopted and swallowed up by mis-aligned procedures. What was meant to be a flexible, light touch, becomes a cookie-cutter process applied with little regard to the context it was intended for.

7. **Insufficient funding and misapplication** – Just because a process or technology can save money when used well, it does not automatically follow that it will somehow miraculously rescue doomed projects, impossible deadlines, or under estimated confusions. Unfortunately when faced with a tough problem, the promise of the latest great-thing can become an appealing solution. However, no process can bend the physics of time or budget, some endeavours are just poorly conceived.

8. **Diminished returns due to devolution of the original idea** – As a wider population of majority companies begin to adopt new technologies/approaches they do so with less regard and consultation to the original creators and guidelines. An “everyone seems to be doing it so how hard can it be?” mentality seems to appear in the majority groups that is not as prevalent in the earlier adopters.

9. **Denigration of the original idea** – As more and more of these misapplications of the approach fail, criticism of the approach rises. Failed instances are held up as “proof” of the technology/approach’s shortcomings by competitors and sceptics. The good name and original successes are called into question and mixed messages spread throughout the community.

10. **Demise and new discovery** – Like fashions, most new approaches decline in popularity only to be revised with a slight twist a few years down the line. There will be some die-hards who continue to follow the approach and the encouraging thing is that these are usually the people who adopted it more completely and convinced themselves by its benefits.

Obviously, the software industry is a complicated ecosystem that can not be readily reduced to a 10 step model. Where models can help though is by creating an easier to understand view of trends and help identify patterns that can be useful to us. So while we can not point to an individual phase step and say that, as on July, 2007, software project management is at step X, we can identify movements in the market and predict what may happen next.

### Conclusion

Agile project management has crossed the chasm and entered mainstream usage. Along with this transition we have gained many useful additions to the original theories and some valuable integration points to complementary approaches. In the future we should expect to see more widespread use of these approaches as they are demanded from a younger generation of workers and identified as effective mechanisms for dealing with high change projects.

### References:

Anonymous (2007) Lean Manufacturing Retrieved July 2007 from Wikipedia:

[http://en.wikipedia.org/wiki/Lean\\_manufacturing](http://en.wikipedia.org/wiki/Lean_manufacturing)

Duvall M (2006, Sept) What’s Driving Toyota. *Baseline Magazine*, Vol 2007/09, Pages 15-21

Tobe G, Hyken S (2003) *Do They Really Want What We Think They Want?* Retrieved July 2007, from

[http://www.hyken.com/Article\\_14.html](http://www.hyken.com/Article_14.html)

Sheard S, (2003, July) *Life Cycle of a Silver Bullet* STSC CrossTalk Magazine Retrieved July 2007, from

<http://www.stsc.hill.af.mil/Crosstalk/2003/07/sheard.html>

Pinto J (1998) *Project Leadership: From Theory to Practice*. Newtown Square, PA: Project Management Institute

Highsmith J. (2004) *Agile Project Management*. Boston, MA: Addison-Wesley

Reinertsen D. (1997) *Managing the Design Factory*. New York, NY: Free Press

Moore G. (2002) *Crossing the Chasm*. New York, NY: Collins