Agile in Project Management

A Brief Overview

By Laurence Nicholson, PMP, MBCS, FPMA, ACQI

1. History

Incremental Development Methodologies

Incremental software development methods have been traced back to 1957at IBM's ServiceBureau Corporation

In 1974, a paper by E. A. Edmonds introduced an adaptive software development process.[3]. Concurrently and independently the same methods were developed and deployed by the New York Telephone Company's Systems Development Center under the direction of Dan Gielan.

During the mid to late 1970s Mr. Gielan lectured extensively throughout the U.S. on this methodology, its practices, and its benefits.

So-called lightweight software development methods evolved in the mid-1990s as a reaction against heavyweight methods, which were characterized by their critics as a heavily regulated, regimented, micromanaged, waterfall model of development. Proponents of lightweight methods (and now agile methods) contend that they are a return to development practices from early in the history of software development.

Early implementations of lightweight methods include Scrum (1995), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995). These are now typically referred to as agile methodologies, after the Agile Manifesto published in 2001

Waterfall vs Agile

When project management became popular in the 1980's, the traditional approach of most project managers was what was known as a waterfall. A strict, rigid stepping from one stage to another in the process for gathering requirements, building the solution, testing the solution built and then putting this live. Whilst this was a successful method for delivering benefit to the business, it soon came under scrutiny as businesses became more dynamic, reacting rapidly to changing needs. The limitations of the waterfall design were based ironically in its very strength; The strict control of the stages of a project.

What was needed was a way of 'trying out' solutions on the fly and making changes throughout the process. Prototyping became the way forward for many, but this was difficult for the traditional project methodology to manage.

Prototyping became a series of small waterfalls, which worked for a while but still had the inherent limitations associated with it.

Something new had to be designed, and the profession turned to the software development and engineering industries, and discovered Agile. A cyclical set of activities called Sprints, during which an agreed set of requirements were designed, built and tested in a short time frame.

Agile methods break tasks into small increments with minimal planning and do not directly involve long-term planning. Iterations are short time frames (timeboxes) that typically last from one to four weeks. Each iteration involves a team working through a full software development cycle, including planning, requirements analysis, design, coding, unit testing, and acceptance testing when a working product is demonstrated to stakeholders. This minimizes overall risk and allows the project to adapt to changes quickly.

Stakeholders produce documentation as required. An iteration might not add enough functionality to warrant a market release, but the goal is to have an available release (with minimal bugs) at the end of each iteration. Multiple iterations might be required to release a product or new features

2. Working with Agile

How much can be used and when?

Often the best way for the use of Agile techniques in the project and programme management areas is as a kind of hybrid approach. This will depend on the project being delivered and the nature of the products. When working with large organizations, implementing business software solutions, the need to define the requirements sets up front is more borne out of travel minimization and business unit segmentation, allowing the centre to dictate a global standard set of requirements.

A typical project plan for delivering business benefits in the form of services or applications would include the following elements:

1. Discovery and Elaboration
2. Design, Build and Test in Sprints
3. Performance Assurance
4. Deployment & Transition
5. Post Go-Live Support

The 'Design, Build and Test' element is the focus for the Agile methodology, being a collaboration between the Business Consultant who gathers the requirements, the Development Lead who translates these into technical specifications and the Quality Assurance Lead who co-ordinates the Quality team in developing the Test Cases to be executed against the resultant code.

It is often a good opportunity for the Testing team to define the test scripts and Use Cases for each requirement to be used as an input to the design of each work package for the development team. This broadens the information available to the development team for each requirement.

3. Conclusion

Practical application of Agile within different commercial constructs

The overwhelming power of Agile appears in the way the requirements are evolved through a series of Sprints which take a user story and develop it within the software using feedback from the customer at the Sprint closing sessions. This approach is often frightening for classically trained PM's who like to know the boundaries of scope to be working within from the beginning.

If you are brave enough or experienced in Agile already, delivering projects in an Agile way is a rewarding experience for all parties. There is however a limitation or at least a serious consideration relating to the commercial construct of the delivery contract that could impact the way in which Agile is used in delivery projects.

If the contract is a fixed fee construct, the concept of ever changing requirements (although we should think of these as not so much changing, but evolving to meet the business needs) conflicts with a fixed scope or amount of effort by a predetermined set of resources. It is often unlikely that a change in the requirements within a fixed fee project or programme is welcomed by either party as a set of negotiations then has to happen in order to either descope other parts of the project, or increase the Fee through a Change Order process.

But consider this; in a T&M based construct, any changes in requirement would be met with an increase in duration, resource or both, which will lead to an increased cost. The other alternative open to both Fixed Fee and T&M based projects is to negotiate the content of the scope and take out some requirements in order to add in the changes, so in reality it becomes a negotiation on either price, duration, resources or scope content.

In essence Agile can and does work very well for those in a position to try it, so long as the customer fully understands their role in the process and can see the benefits of dedicating more resource and time to the process.

About the Author

Laurence Nicholson

UK

Laurence Nicholson has over 25 years Project and Programme Managment experience across a wide variety of industries and all sectors, including private, public and voluntary. He has penned many articles over the years and had some translated into numerous languages.

He has been responsible for leading change and improvement through strategic reviews and the introduction of methods and processes in project management and quality systems, including Programme management office processes and governance. He has also managed risk and communications management of project portfolios and the development of training and coaching in project management techniques for a number of leading world class organisations.

Has previously been divisional head with WarnerBros IT and with ProcServe, operating at the CxO level both internally and externally. Passionate about continuous improvement, constantly strives for process and operational efficiencies. Laurence comes from a PA Consulting background that drives a desire for efficiency, innovation and best practice through alignment of strategic objectives.

Specialties: Project and Programme Management, Change Management, Quality Assurance, Software Development Managment, Development Programme Management, Procurement, Resource Management and Divisional Management.

Contact email: laurencenicholson@gmail.com