Agile Project Management

It is now becoming clear that our conventional project management processes are inadequate when managing complex projects. Research is underway by the Project Management Institute, and others determine what makes projects complex and to learn how to manage project complexity. Many thought leaders in the field of project management are presenting alternative approaches to managing complex projects. We are now realizing that new approaches are desperately needed to manage complex projects in the ever-changing global economic environment. – *Managing Complex Projects: A New Model*

As we saw when we are operating within a stable and predictable environment we can break projects down into stages and use a linear process of stepping though them stage by stage until completion at the end. Complex projects take place within dynamic environments that may be subject to almost continuous change. The level of uncertainty and unpredictability is high and we are not sure of what the end goal is. This necessitates an alternative approach to project management – one that is more flexible, adaptive and evolutionary. This new approach finds its clearest expression within what is called Agile project development.

Agile was first employed within software development and we can't contrast it with its predecessor called The Waterfall Model.

The Waterfall Model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance. The waterfall development model originates in the manufacturing and construction industries.

In contrast within an Agile project development, requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible response to change. It is a conceptual framework that focuses on frequently delivering small increments of working software.

Agile is Holistic

Agile project development has been compared to building an airplane in the sky. You just get enough done to get it into the sky and then you continue to build upon it from there. Instead of breaking a project down into discreet states before being completed at the end, Agile creates the whole project in one and then builds upon that.

Rapid

Thus, we can think of Agile development as very much like rapid prototyping. With Agile project management, projects are managed within small chucks of work. It is designed to deliver value to the organization through frequent small improvement to the product called features.

Evolutionary
These rapid prototypes allow us to probe the environment, gaining feedback to see where we are and where the end point might be. Feedback is a key part of Agile development, with the response to each sprint feeding back into the next one, thus allowing for the constant interaction between the problem and the solution as they co-evolve over time.

Self-organizing

Management commits to guiding the evolution of behaviors that emerge from the interaction of independent agents instead of specifying in advance what the effective behavior is. The team figures out for itself how to turn the backlog of work into a finished product.

Agile teams are often small, typically under 15 people. They are co-located for face to face interaction and they favor individuals and interactions over processes and tools.

Iterative

Agile works through small iterations called sprints, which typically happen over the course of just a few weeks. A backlog of work is created and each sprint completes a small set of new features, and then taking end user experience as feedback to be added to the backlog.

Agile Life Cycle

There are different approaches taken to managing each sprint's lifecycle but a common one is to break it down into a set of states.

Envision – Overall scope for the project; building the team structure
Speculate – Outline the features and requirements
Explore – Develop the product
Adapt –Receive feedback, pause and reflect
Close – Ensure all of your deliverables are completed

Interdisciplinary

Each iteration involves a cross-functional team working in all functions: planning, requirements analysis, design, coding, unit testing, and acceptance testing. Peer review and communication are key.

Summary

The 12 Agile Principles

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working product frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working product is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity — the art of maximizing the amount of work not done — is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Only 35 percent of projects are successful (delivered on time, on budget and with full scope of features and functions), 46 percent are challenged (the project is completed and the new solution is operational, but it was late and over budget, with reduced features and functions), and 19 percent fail (do not deliver anything of value) – *the Standish Group's 2007 first quarter research report*