Understanding Agile at a Deeper Level

Developing an Adaptive, High-Performance Agile Approach

Agile History

Agile Fundamentals

Chuck Cobb

Twitter: @chuckcobb3
Blogsite: www.managedagile.com
Email: chuck@highimpactpm.com

Hi, with this lesson, we're going to start a new section on Agile Fundamentals

Agile Fundamentals

Agile History

Agile Manifesto Values

Agile Manifesto Principles

Scrum Overview and Roles

Scrum Methodology

Scrum Values

Here's a brief list of what we're going to cover in this section:

- First, we're going to talk about the history behind Agile and the Agile Manifesto Values
- Next, we're going to talk about Agile Manifesto Principles
- Then, we're going to discuss an overview of Scrum and Scrum Roles
- And we're going to talk in more detail about the Scrum Methodology
- And, finally, we're going to talk about Scrum Values

Agile History

Before we get into the Agile Manifesto Values and Principles, I think it's useful to quickly review some of the history that led to the development of the Agile Manifesto.

Software Development Prior to the Waterfall Model

Software Development Process Prior to the Waterfall Model:

- Unstructured,
- Ad-hoc, and
- Chaotic

Results:

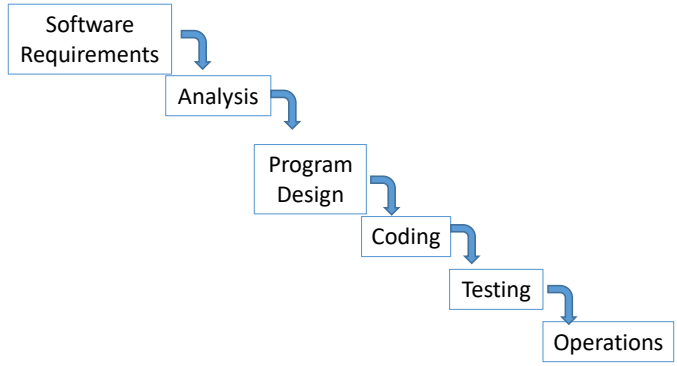
- Poor and Inconsistent Quality
- Prone to Errors
- Limited Standardization and Compatibility

Software development prior to the Waterfall Model was somewhat unstructured and chaotic. You also have to appreciate that the technology for doing software development in those days was much more primitive than it is today. In those days, software development tools were not very sophisticated, development languages were much more limited, and software development patterns and architectures were also not as well-defined as they are today.

The process for developing software in those days might be compared to the process for knitting a sweater by hand…it was an art and required a lot of craftsmanship and the process was very primitive compared to today's standards. And it wasn't just the process that was the problem, the technology and tools were primitive as well at that time.

- The nature of the technology and the process required a heavy emphasis on upfront design - can you imagine trying to knit a sweater without knowing exactly what it's going to look like before you start?
- It was also difficult to scale that process - it's like getting multiple people to try to knit the same sweater – it's a very difficult thing to do and it can be very prone to error and problematic if the efforts of multiple people weren't very carefully planned and coordinated.

Evolution of the Waterfall Model



Royce, Dr. William, Managing the Development of Large Software Systems,
http://leadinganswers.typepad.com/leading_answers/files/original_waterfall_paper_winston_royce.pdf

One of the major reasons that processes like the Waterfall model evolved was to bring structure and discipline to that development process. That was absolutely essential as projects grew larger as well as more complex and more critical.

It's important to recognize that because as we migrate to a more Agile approach, we don't want to lose sight of that and go back to a completely unstructured approach and all the problems that existed prior to the Waterfall approach.

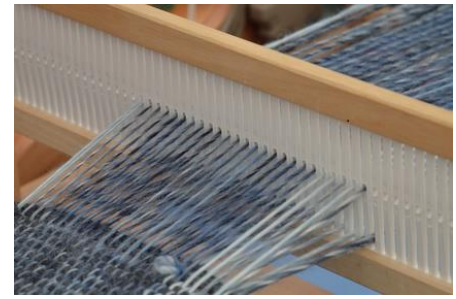
Software Development After the Waterfall Model

Software Development Process After the Waterfall Model:

- Very Structured,
- Controlled, and
- Planned and Organized

Results:

- Much higher levels of quality,
- Standardization, and
- Compatibility

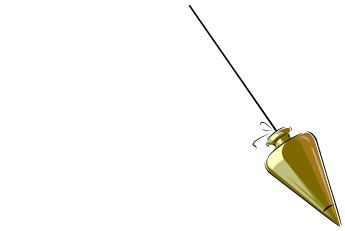


The process for developing software after the evolution of the Waterfall model might be compared to a manufacturing process for knitting a sweater in a manufacturing plant. It required a lot less individual craftsmanship and the process was much more structured, controlled, planned, and organized.

That resulted in higher levels of quality and standardization as well as compatibility; but, of course, the downside of that was because the products were more standardized and controlled, the process was naturally much less flexible and required defining the requirements upfront.

You might compare it to the evolution that manufacturing processes went through to go from the age of craftsmanship to highly controlled manufacturing processes that came about with the industrial revolution and manufacturing assembly lines.

Dr. Winston Royce – Waterfall Model 1970



Software Development Process Prior to the Waterfall Model:

- Unstructured,
- Ad-hoc, and
- Chaotic

Results:

- Poor and Inconsistent Quality
- Prone to Errors
- Limited Standardization and Compatibility

Software Development Process After the Waterfall Model:

- Very Structured,
- Controlled, and
- Planned and Organized

Results:

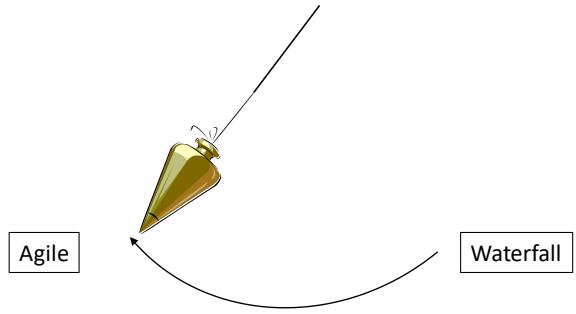
- Much higher levels of quality, standardization and compatibility

8

The Waterfall process was an improvement over what came before it but it certainly wasn't optimal. The pendulum really swung pretty far between two extremes from a highly unstructured approach based on individual craftsmanship to a very highly structured approach that was heavily controlled, planned and organized.

The problem is that designing and developing software isn't really totally like a manufacturing process at all because the requirements aren't necessarily well-defined before you start. As a result, it isn't as simple as building the same product over-and-over again on a manufacturing assembly line.

Early Agile Development Methods (1980's and 1990's)



"Agile methods were direct spinoffs of software methods from the 1980s, namely:

- Joint Application Design (1986),
- Rapid Systems Development (1987), and
- Rapid Application Development (1991)."

Rico, Dr. David F, The History, Evolution and Emergence of Agile Project Management Frameworks, http://davidfrico.com/rico-apm-frame.pdf

In reaction to the problems caused by an excessive emphasis on control in the Waterfall model, the pendulum started to swing back in the other direction in the 1980's and 1990's with a variety of iterative and incremental approaches such as the Rational Unified Process (RUP).

Agile methods really had their roots in the 1980's and 1990's in some methodologies that became popular at that time such as Joint Application Development in 1986 and Rapid Application Development in 1991. However the roots of Agile really can be traced into a number of manufacturing advances that came about at that time such as Total Quality Management which we will discuss in a later lecture.

Continued Agile Development (1990's)

"Agile methods formally began in the 1990s with:

- Crystal (1991),
- Scrum (1993),
- Dynamic Systems Development (1994),
- Synch-n-Stabilize (1995),
- Feature Driven Development (1996),
- Judo Strategy (1997), and
- Internet Time (1998).

Other agile methods included:

- New Development Rhythm (1989),
- Adaptive Software Development (1999),
- Open Source Software Development (1999),
- Lean Development (2003), and
- Agile Unified Process (2005).

However, the popularity of Extreme Programming (1999) was the singular event leading to the unprecedented success of agile methods by the early 2000s."

Rico, Dr. David F, The History, Evolution and Emergence of Agile Project Management Frameworks, http://davidfrico.com/rico-apm-frame.pdf

There was a broad proliferation of Agile methodologies that evolved in the 1990's; however the popularity of Extreme Programming was the singular event in 1999 that led to the unprecedented success of Agile methods by the early 2000's.

NEXT LECTURE…
AGILE MANIFESTO VALUES

In the next lecture, we're going to discuss the Agile Manifesto Values