

Agile Project Management

Geoffrey Withnell
BAE Systems

Presented to ASQ511 Northern Virginia
September 14, 2016



Project Management

Initiating

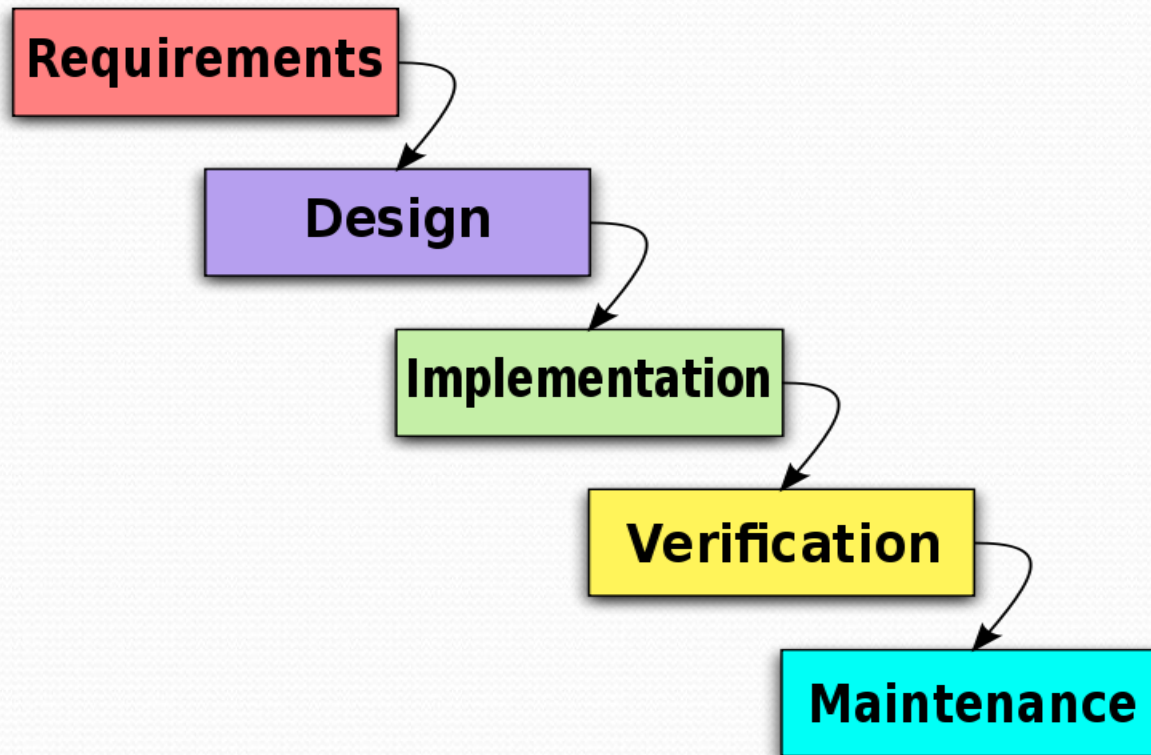
Planning

Executing

Monitoring and Controlling

Closing

Waterfall Development



Manifesto for Agile software product Development

We are uncovering better ways of developing software product by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software product over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Principles behind the Agile Manifesto

We follow these principles:

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

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

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

Principles behind the Agile Manifesto

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 software product is the primary measure of progress.

Principles behind the Agile Manifesto

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.

How Work Gets Done!

- **Plan-Driven**

- **Often phase-oriented, delivers results at the end**
 - SDLC, Waterfall, RUP, PERT, Critical Chain, Prince2

How Work Gets Done!

- **Agile with Planning**
 - Planning and execution are *epicyclic*, repeated at different scales for nested time horizons
 - Scrum, XP, CBPM

How Work Gets Done!

- **Agile without Planning**
 - **Processes unpredictable requests efficiently, for types of work where planning is not possible or required**
 - Kanban

Understand the Adaptive Spectrum!

Predictive processes

Emphasize efficiency
Perform poorly when uncertainty is high

Plan Driven
Waterfall
SDLC

Certainty

Predictive

Adaptive processes

Emphasize adaptability to rapid change
Enable detailed planning

Scrum
XP

Reactive processes

Don't require planning
Handle unpredictable work well

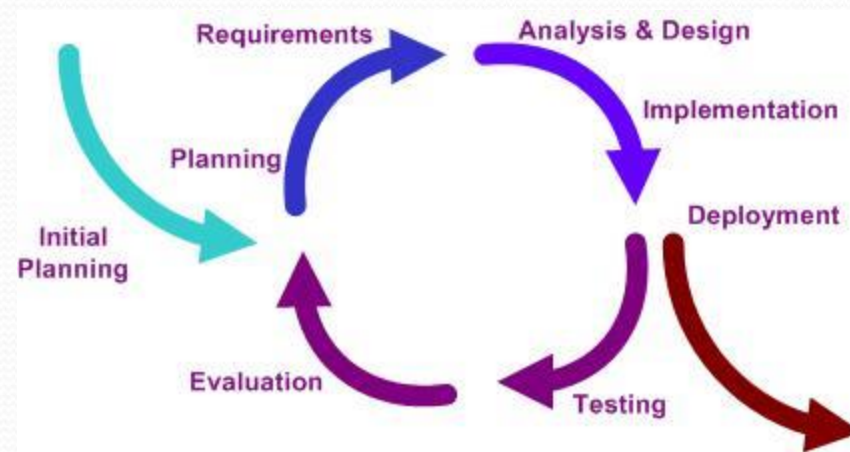
Kanban

Uncertainty

Reactive



Iterative Development



Deming Cycle



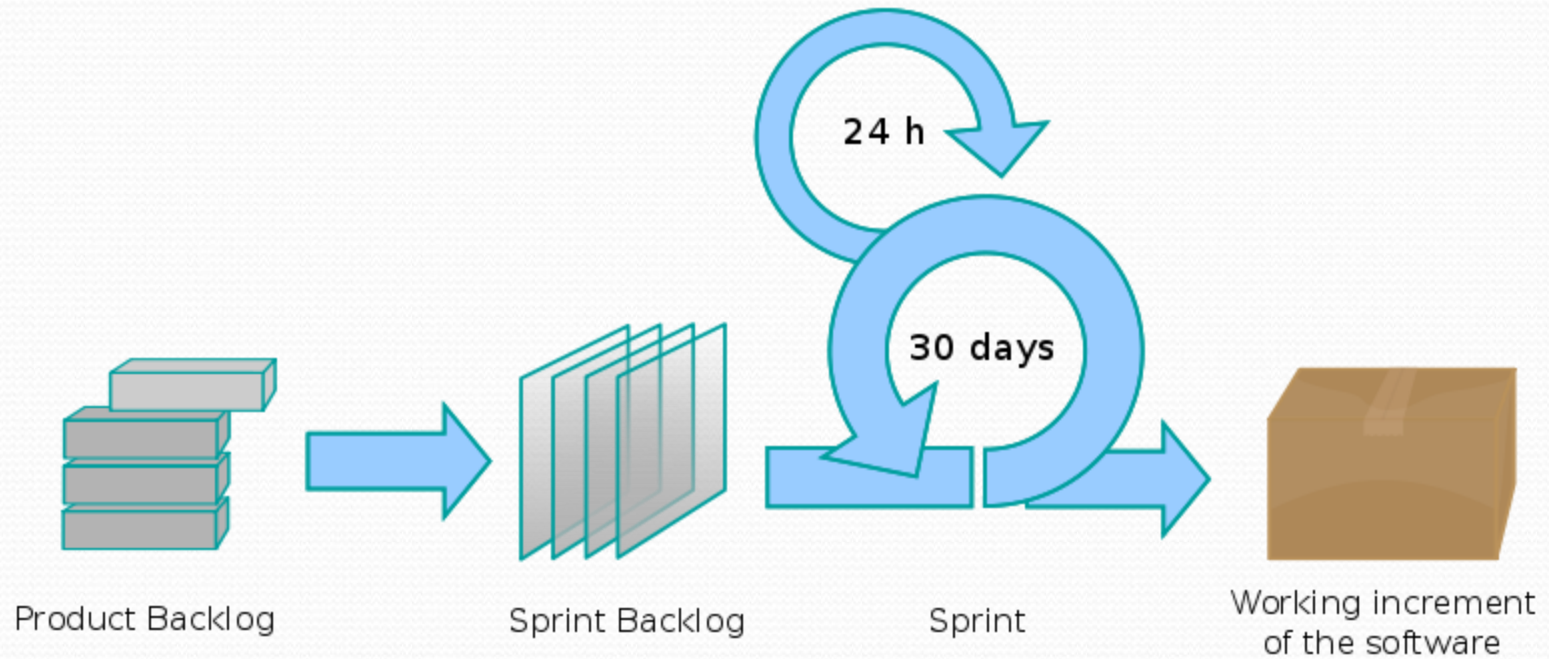
Scrum History

- **History**
- In 1986, Hiroataka Takeuchi and Ikujiro Nonaka described a new approach to commercial product development that would increase speed and flexibility, based on case studies from manufacturing firms in the automotive, computer, photocopier, and printer industries.[1] They called this the *holistic or rugby approach, as the whole process is performed by one cross-functional team across multiple overlapping phases, where the scrum (or whole team) "tries to go the distance as a unit, passing the ball back and forth"*. [1]
- In 1991, DeGrace and Stahl first referred to this as the *scrum approach*. [2] In the early 1990s, Ken Schwaber used such an approach at his company, *Advanced Development Methods*, and Jeff Sutherland, with John Scumniotales and Jeff McKenna, developed a similar approach at *Easel Corporation*, and were the first to refer to it using the single word *Scrum*. [3]

Requirements Management

- Requirements Types
- Architectural Requirements
- Structural Requirements
- Behavioral Requirements
- Functional Requirements
- Non-functional Requirements
- Performance Requirements
- Design Requirements
- Derived Requirements
- Allocated Requirements

Scrum Process



Requirements

User Stories

I want to use _____ to do _____ because _____.

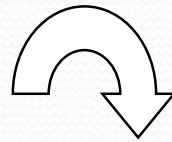
What do our customers want? Requirements should tie back to this. User stories create the feature list that becomes the Product Backlog. The portion of the Product Backlog that *the team* believes can be do in one iteration becomes the Sprint Backlog.

PMBOK Waterfall

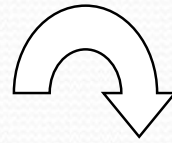
Initiating



Planning



Executing

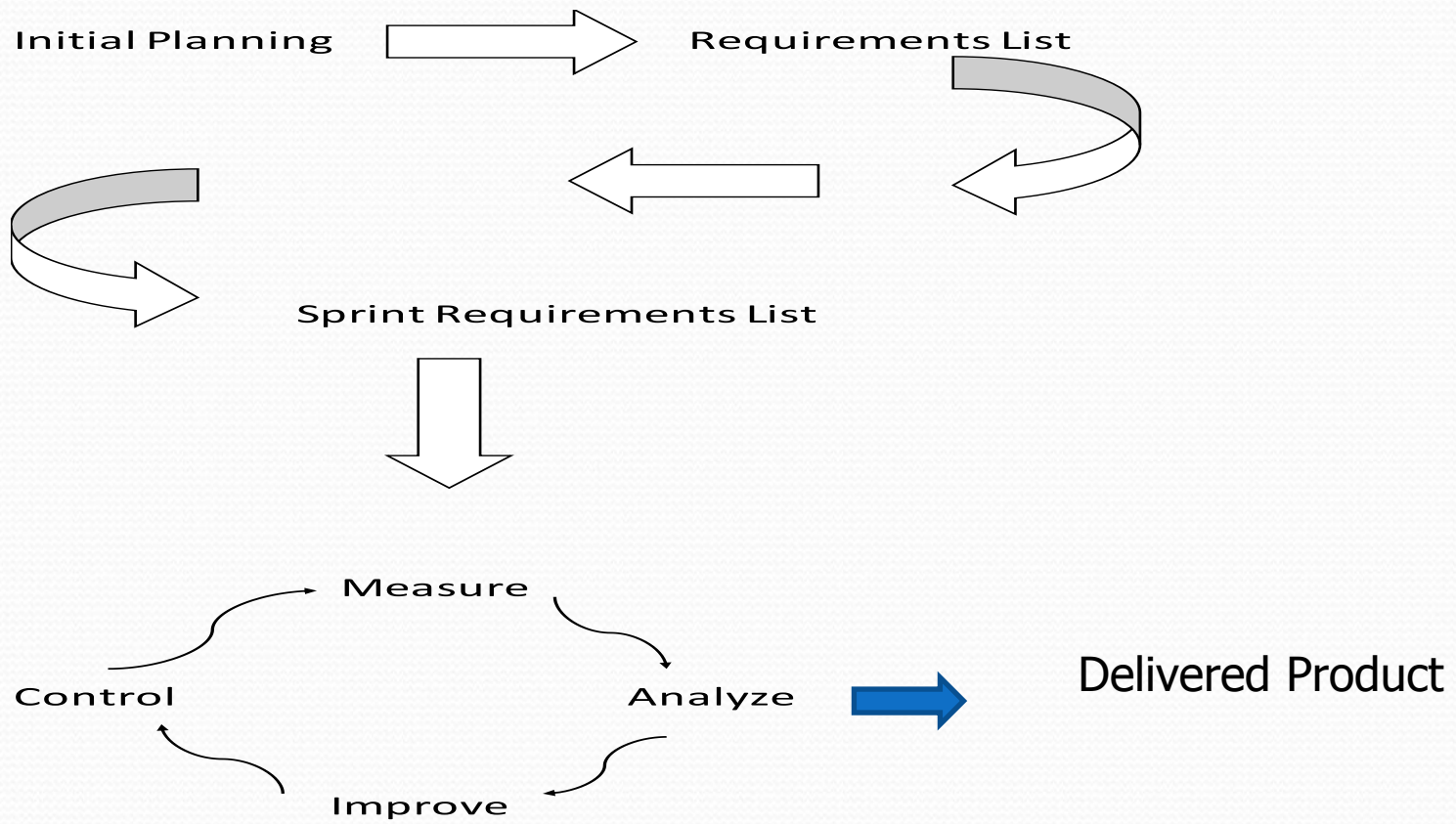


Monitoring/Controlling



Closing

Agile



Manifesto for Agile Product Development

Manifesto for Agile Product Development

We are uncovering better ways of developing product by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Processes and tools that enable individuals and interactions are to be encouraged.

Working product over comprehensive documentation

Customers (and future developers) have a right to enough documentation to understand the product

Customer collaboration over contract negotiation

Collaboration is easier when the expectations for Agile Development are negotiated in the contract

Responding to change over following a plan

Responding to change is easier when one has planned how to do so.

We value the items on the left, and we value the items on the right as they contribute to the success of the items on the left.

Agile Project Principles

- *We follow these principles:*
- Our highest priority is to satisfy the customer through early and continuous delivery of valuable product.
- 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.

Agile Project Principles

- 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/process improvements are the primary measure of progress.

Agile Project Principles

- Agile processes promote sustainable improvements. 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 most effective improvements emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly