Everything you wanted to know about Scrum… but were too afraid too ask

A Brief Overview…

Tobias Mayer
tobias@agilethinking.net

Victor Szalvay
victor@danube.com

Part 1

Agile History & Context

Phased/Defined Approaches

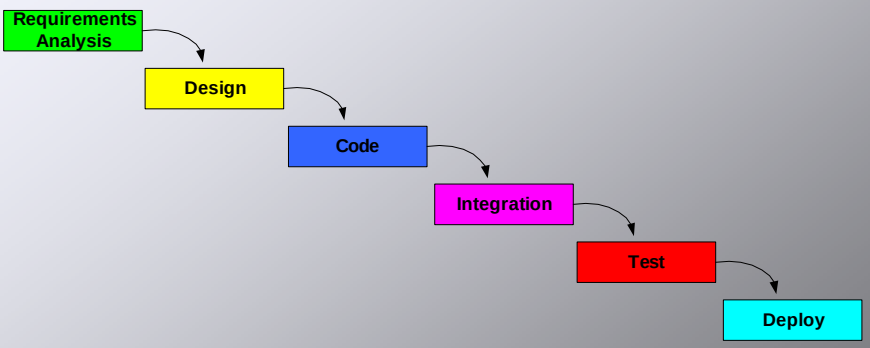
- The majority of Software Development organizations have been using phased & defined approaches to developing software since the early 1970's
- Based on a Tayloresque[1] factory model: rule by compliance; strictly authoritarian and hierarchical.
- Attempts to mitigate all risk upfront
- Relies on documentation and other "hand offs" to convey essential information
- Creates functional divisions for better control
- Resists change

[1] "The Principles of Scientific Management" by Frederick W. Taylor (1911)

The Waterfall

- The best-known and most-used phased approach
- The original Waterfall paper was written by Winston Royce in 1970
- Royce recommended the use of a "waterfall" model as an *ideal* way of developing software
- But Royce actually discredited the concept himself… *in the same paper*

The Phased Approach »



The Agile Manifesto (2001)

Not an “ideal” way - this comes from real experience

We are uncovering better ways of developing software 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*** over comprehensive documentation
- ***Customer collaboration*** over contract negotiation
- ***Responding to change*** over following a plan

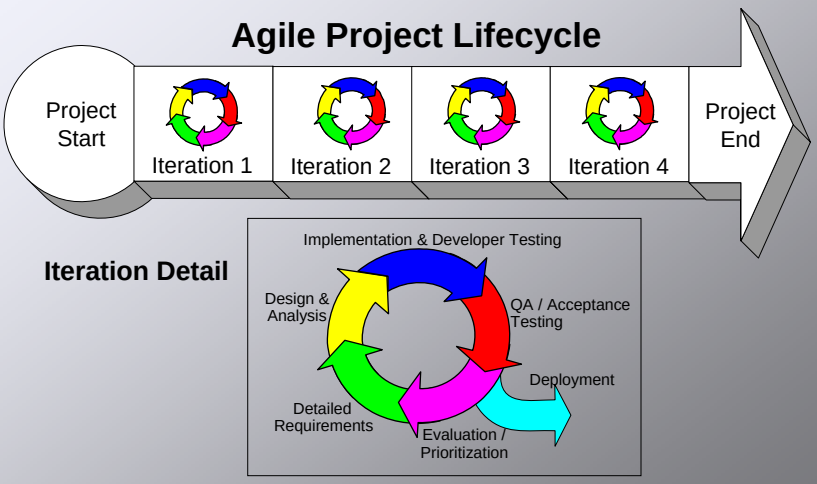
That is, while there is value in the items on the right, we value the items on the left more.

http://agilemanifesto.org

Characteristics of an Agile Process

- Empirical (relies on observation and experience)
- Lightweight
- Adaptive
- Emergent
- Fast – but never hurried
- Exposes wastefulness
- Customer-centric
- Pushes decision making to lower levels
- Fosters trust, honesty and courage
- Encourages self-organization

The Agile Approach



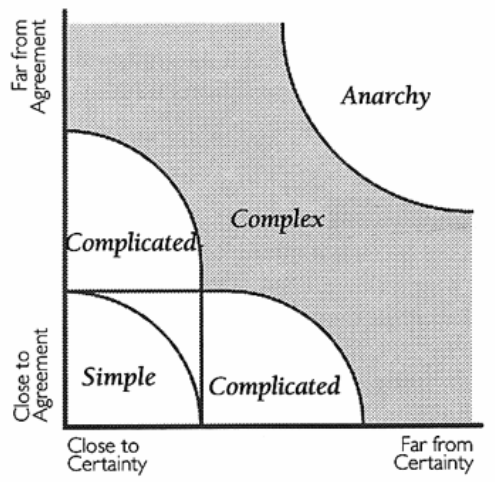
Part 2

Scrum Overview

What is Scrum?

- A simple framework that can be understood and implemented in a few days.
- An approach to managing complex problems.
- An environment to support self-organization, creativity and emergence.
- A collaborative effort involving developers and customers in ongoing dialog.
- *"Scrum is not a methodology – it is a pathway"* -- Ken Schwaber (Boulder, Co, Nov. 2005)

The Stacey Diagram



- Scrum works best when the problems to be solved lie in the Complex Space.
- New Product Development Work and Knowledge Work both tend to exist in the Complex Space.
- When work exists in the Anarchy Space we need to seek patterns.

»

Scrum Management Roles

- Product Owner
  - Manages the Vision, and the Requirements
- Scrum Master
  - Manages the Process
- The Team
  - Manages themselves

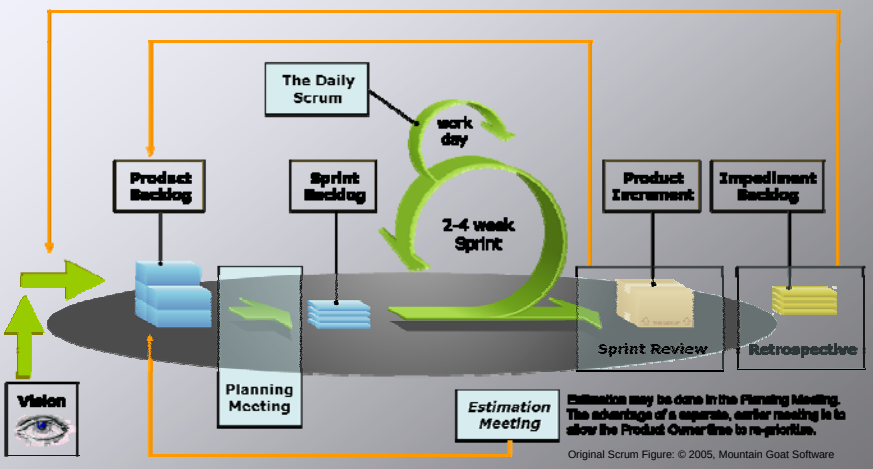
Scrum Meetings

- **Planning Meeting**
  - Occurs at the start of each sprint. Team and Product Owner negotiate the sprint.
- **Daily Scrum**
  - Every day. Maximum 15 minutes. Team meets to report to each other on progress, intention and impediments.
- **Review**
  - Inspect and Adapt the *product*. Team demonstrate the working software from the sprint.
- **Retrospective**
  - Inspect and Adapt the *process*. Team looks at what went well and what can be improved.

The Product Backlog

- Scrum begins with a Product Vision represented by a well-formed Product Backlog
- If the Backlog is ill-defined the resulting product will be shoddy and will not meet the customer's needs
- "Garbage in, garbage out"

The Scrum Framework



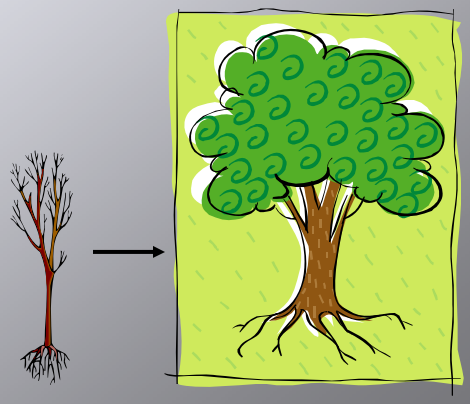
Original Scrum Figure: © 2005, Mountain Goat Software

Key Principles of Scrum

- Empirical Process
  - Detailed up-front planning and defined processes are replaced by just-in-time inspect and adapt cycles.
- Self-Organization
  - The team is self-managing and organizes itself around goals, given clear constraints.

Emergent Design

- Deliver complete "slices" of the system at regular intervals
- Design and Architecture are so important that they must happen ***all the time***



Moving Forward…

- Scrum is simple – but difficult
  - An understanding and appreciation of Agile values and principles is essential
  - Scrum alone will not create better products.
    - Engineering, design and testing practices need to become more Agile
    - Interactions and communication need to become clearer, more personal and more transparent
    - The qualities of honesty, trust and courage need to be developed
    - When the going gets tough it is easy to slip back into the old way of doing things. Courage is essential.

The End

Thank You

Tobias Mayer
tobias@agilethinking.net

Victor Szalvay
victor@danube.com