

B345 Internet Science and Technology

Week 5 lecture 2

Today's Lecture Learning Objective

- Get to know some popular methodologies based on the "agile" principles described in the last lecture.

Example Agile Methodologies

- eXtreme Programming (XP)
- Crystal Methods
- Scrum
- Adaptive Software Development (ASD)
- Feature Driven Development (FDD)
- Dynamic System Development Method (DSDM)

eXtreme Programming (XP)

- Kent Beck.
- A very programming-oriented methodology.
- Some key properties of XP
 - Whole team approach
 - Planning game
 - Collective code ownership
 - Pair programming
 - Test driven development
 - Refactoring

<< See XP.jpg >>

Crystal Methods

- A family of methods.
- Developed by Alistair Cockburn.
- Select which method in the family based on
 - How big is the team?
 - How much certain problems can cause damage?
 - What are the project priorities?

<< See Crystal.jpg >>

Crystal Methods

- Like XP, is very people-centred.
- Unlike XP, tries not to require high discipline.
 - Trade off between productivity and ease of execution.

Crystal Methods

- Alistair Cockburn and Crystal Methods main contributions:
 - Decision based on domains.
 - Bare sufficiency.
 - People and communication.

Scrum

- By Ken Schwaber.
- Based on rugby's scrums, to get out-of-play ball back into play.
- Has a project management flavour.
 - Compared to the programming flavour in XP.

Scrum

- Emphasis between "defined processes" and "empirical processes".
- The concept of 30-day "sprints".
 - Pre-sprint planning
 - Sprint
 - Post-sprint meeting

<< See Scrum.jpg >>

Scrum

- Ken Schwaber and Scrum's main contributions:
 - Concept of SD being "defined" vs "empirical processes."
 - Devised mechanisms to control the chaos as the basis of project management.

Adaptive Software Development (ASD)

- By Jim Highsmith.
- Adaptive environment - outcomes unpredictable - therefore, planning is a paradox!
- Focuses directly on key aspects of an adaptive process: collaboration and learning.
 - Speculate-Collaborate-Learn life cycle.

<< See ASD.jpg >>

Adaptive Software Development

- Jim Highsmith and ASD's contribution:
 - The process of collaboration and learning as the basis of SD, rather than the outcomes.
 - Initiatives to promote collaboration and learning as key.

Feature Driven Development (FDD)

- Jeff de Luca and Peter Coad.
- Focus on activities towards features of the system.
- Two-week iterations.

<< See FDD.jpg >>

Feature Driven Development

- 5 processes.
 - Develop an Overall Model
 - Build a Features List
 - Plan by Feature
 - Design by Feature
 - Build by Feature

Feature Driven Development

- FDDs special properties
 - Used as a "repeatable" process.
 - Doing much the first time at the beginning.
 - A hierarchy of people: owner, chief programmer.

Dynamic System Development Method (DSDM)

- Developed by consortium.
 - Manuals, training courses, accreditation, etc.
- Principles very similar to ones from Agile Manifesto.

<< See DSDM.jpg >>

Dynamic System Development Method (DSDM)

- Has a lot of the infrastructure of traditional methodologies, but adopts agile principles.
 - Uses prototypes rather than documentation to capture information.
 - Small number of "roles" within teams.
 - Project manager in people-oriented rather than process-oriented tasks.

Open Source

- Is this a "software development methodology"?
- The actual "approach" is not clearly defined yet.
- Geared towards geographically distributed teams.

Open Source

- Adaptivity through:
 - Distributed teams.
 - Parallel debugging.
 - Early and constant releases.
 - Multiple maintainers, easily changed.
 - » Maintainers as a point to control cohesion and quality

The Agile Manifesto

- By Agile Alliance.
- Manifesto statements:
 - **Individuals and interactions over** processes and tools
 - **Working software** over comprehensive documentation
 - **Customer collaboration** over contract negotiation
 - **Responding to change** over following a plan

Principles behind the Agile Manifesto

- 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 software 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.

Principles behind the Agile Manifesto

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

Principles behind the Agile Manifesto

- 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.

When to adopt agile methods?

- Volatile and uncertain environments.
- Motivated people. Adaptable people.
- Small to medium size teams.
- Flexible, involved client.