G64OOS (Spring 2014)

Lecture 07

Agile Software Development

Peer-Olaf Siebers

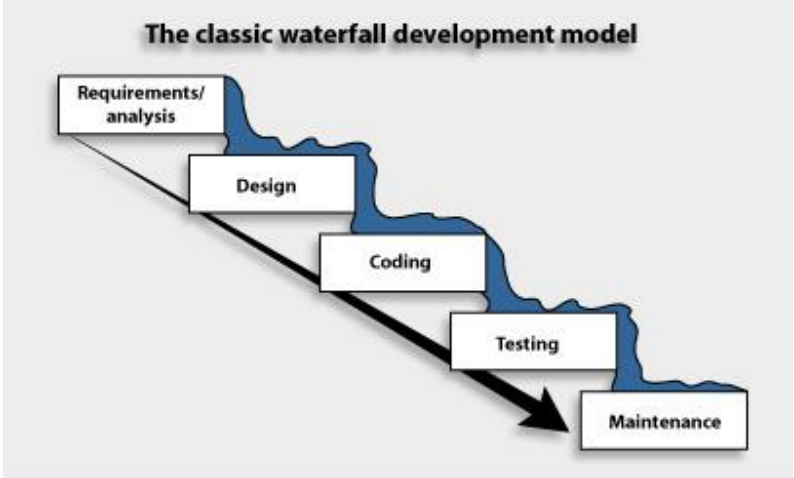
The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

Motivation

- Understand the Agile Software Development paradigm
- Know how Agile compares to Waterfall and Spiral
- Understand Scrum as an Agile Software Development method
- Know how to make a Scrum time planning

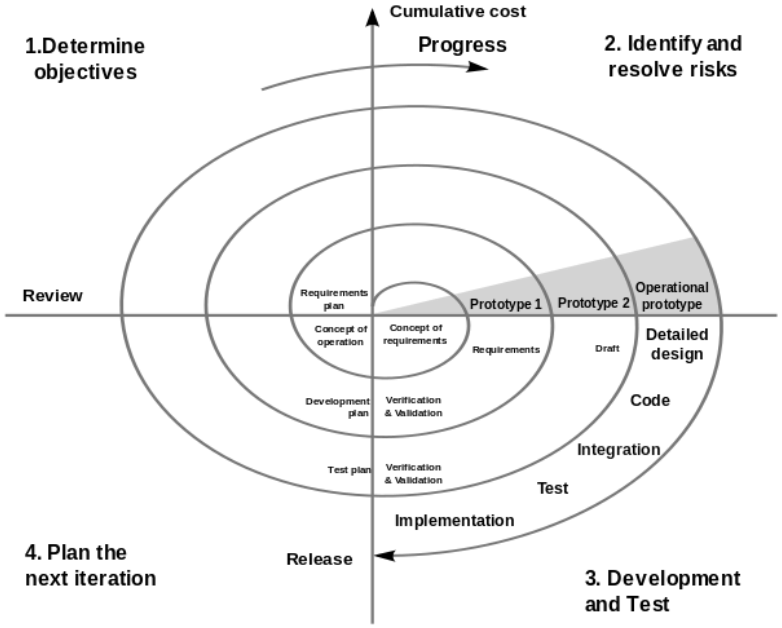
Software Development Methods

- Waterfall Software Development
  - Sequential design process in which progress is seen as flowing steadily downwards (like a waterfall) through the different phases.
  - Originates in the manufacturing and construction industries
  - Key Criticism
    - Impossible for any non-trivial project to finish a phase of a software product's lifecycle perfectly before moving to the next phases and learning from them



Software Development Methods

- Spiral Software Development
  - Risk-driven process model generator for software projects; based on the unique risk patterns of a given project, the spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or evolutionary prototyping.
  - Key Criticism
    - The spiral is simply a sequence of waterfall increments
    - All project activities follow a single spiral sequence



Software Development Methods

- There are many problems with these approaches ...
  - A lot of waiting time for developers
  - Tons of documentation
  - Can result in costly unnoticed errors and buggy code
  - Hard to incorporate new or changing customer requirements

- Surely there must be a better way ...
  - Is it not possible to generate code in a unified yet flexible manner?

Software Development Methods

- Agile Software Development
  - Group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organising, cross-functional teams

- Promotes and encourages
  - Adaptive planning
  - Evolutionary development and delivery
  - Time-boxed iterative approach
  - Rapid and flexible response to change

Manifesto for Agile Software Development

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.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

Agile Software Development (Agile SD)

- Agile SD is a way of thinking about project management
  - Based on iterative and incremental development
  - Requirements as well as solutions evolve together
  - Collaboration between cross-functional, self-organising teams
  - Teams kept small (5-9 people)

Agile Principles

- The agile manifesto actually has 12 main principles, but these can be condensed to the following five:
  1. Deliver Early and Often to Satisfy Customer
  2. Welcome Changing Requirements
  3. Face to Face Communication is Best
  4. Measure Progress against Working Software
  5. Simplicity is Essential

- Agile SD is the art of maximising the amount of work not done

Agile Principles Applied

- Customer Role
  - Focus on Customer Collaboration
  - Easy to add new features
  - Late requirements changes are welcomed
  - Each team actually contains a customer representative

Agile Principles Applied

- Progress Measure
  - Primary measure of progress is in terms of working software, not lines of code
  - Agile projects measure progress by the amount of software that is currently meeting customer needs
  - They are 30% done when 30% of required functionality is working AND deployed
  - Progress is not measured in terms of phases or creating documents
  - Keeps on top of customer satisfaction and allows for more realistic and updated estimation of costs

Agile Principles Applied

- Keep it Simple
  - This refers to the art of maximizing the amount of work NOT done
  - Agile projects always take the simplest path consistent with their current goals
  - They do not try to anticipate tomorrow's problems; they only solve today's problems
  - High-quality work today should provide a simple and flexible system that will be easy to change tomorrow if the need arises

Agile Principles Applied

- Different people interpret the Agile Manifesto in different ways and have developed different Agile SD methods
  - Adaptive Software Development
  - Agile Modeling
  - Agile Unified Process
  - Crystal Methods (Crystal Clear)
  - Disciplined Agile Delivery
  - Dynamic Systems Development Method
  - Extreme Programming
  - Feature Driven Development
  - Lean software development
  - Scrum

Agile Principles Applied

- The Agile SD methods are focused on different aspects of the software development life cycle
  - Some focus on the practices
    - e.g. Extreme Programming
  - Some focus on managing the software projects
    - e.g. Scrum
  - Some prove full coverage
    - e.g. Dynamic Systems Development Method

Extreme Programming (XP)

- XP is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements through frequent releases in short development cycles
- Other elements of XP include
  - Programming in pairs [video]
  - Unit testing of all code [video]
  - Avoiding programming of features until they are actually needed
  - Simplicity and clarity in code

- More in Lecture 8 ...

Scrum

- Agile SD method for managing software projects

http://www.youtube.com/watch?v=XU0llRltyFM

Scrum [Wikipedia 2014]

- Scrum Artefacts
  - Product backlog
    - Prioritized list of high-level requirements
  - Sprint backlog
    - Prioritized list of tasks to be completed during the sprint
  - Sprint (time-box; iteration)
    - Time period (typically 1–4 weeks) in which development occurs on a set of backlog items that the team has committed to
  - Burn down chart
    - Sprint burn down chart: Daily progress for a Sprint over the sprint's length
    - Release burn down chart: Sprint level progress of completed product backlog items in the Product Backlog

Scrum [Wikipedia 2014]

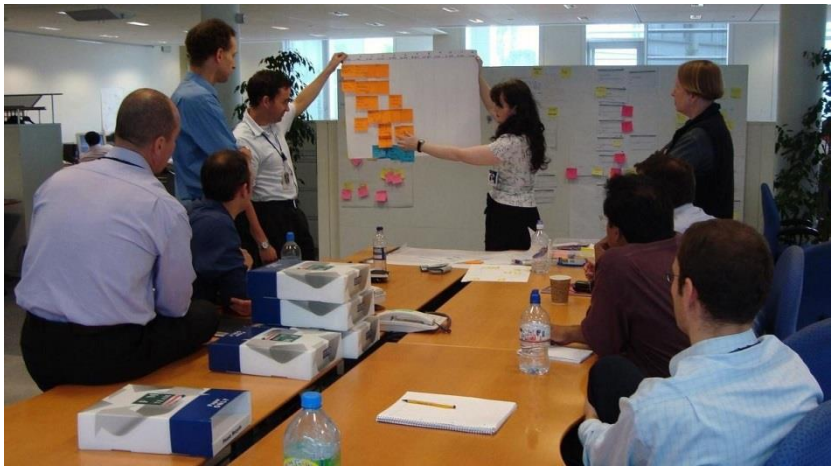
- People involved
  - Scrum Team
    - Product Owner, Scrum Master and Development Team
  - Product Owner
    - Person responsible for maintaining the Product Backlog by representing the interests of the stakeholders, and ensuring the value of the work the Development Team does
  - Scrum Master
    - Person responsible for the Scrum process, making sure it is used correctly and maximizing its benefits
  - Development Team
    - A cross-functional group of people responsible for delivering potentially shippable increments of Product at the end of every Sprint

Scrum

- Product backlog
  - Collection of all user stories
    - A kind of wish list of what would make our product great
  - Implemented user stories will be removed the product backlog
  - New ideas and bug fixes will be added to the product backlog
- Release backlog
  - List of user stories for a specific release (including prioritisation)
  - Estimate amount of work involved in each item; sum of the work involve provides the estimated release date for the specific release
  - After a software release the release backlog will be emptied and a list of user stories for the next release will be added (from the up to date product backlog)

Scrum

- Release planning
  - Team and customer pick features from product backlog
  - This creates a release backlog
  - Release backlog features are assigned priority
  - Release backlog features are assigned effort estimates
  - Release is split in a number of sprints



Scrum

- Sprint
  - Sprints last typically 1–4 weeks
  - Each release consists of 4-12 sprints
  - Duration of sprint is proportional to release interval
  - Sprint backlog lists features included per sprint

Scrum

- Monitoring progress
  - Progress is measured in terms of functionality added
  - Time to complete features in active sprints drawn in a Burndown chart
  - Burndown chart is updated daily
  - Burndown velocity can be measured from chart
  - Team estimates daily how much time is required per remaining feature

Scrum

- The Scrum: Daily Meeting (only for the team members)
  - Meeting should be held standing
    - Keeps people on their toes
    - Keeps meetings short and to the point
  - Discusses what was done yesterday
  - Discusses what will be done today
  - Excellent tool to spot issues as they arise
  - The Scrum Questionnaire
    - Every team member has to answer three question:
      - What have I done since last meeting?
      - What will I do until the next meeting?
      - What problems have I encountered?

Scrum

- Resources
  - Books
    - Rubin (2012) Essential Scrum: A Practical Guide to the Most Popular Agile Process
    - VII (2014) The Scrum Checklist: For the Agile Scrum Master, Product Owner, Stakeholder and Development Team
  - Websites
    - http://www.agile-process.org/
    - http://www.mountaingoatsoftware.com/topics/scrum

Scrum Time!

- Job: Develop a web site that lets users search, compare, and review apps for both Apple and Android platforms

- Group Activity:
  - Get together in groups of approx. 5 and assign Scrum roles
  - Create a time planning:
    - Product backlog
    - Release backlog
    - Initialise Burndown chart
  - Hold Day-1 Scrum meeting
  - Present your time planning to the other groups

Questions / Comments