Agile Software Development Project

CSM2020 (18/19)

Department of
Computer Science

Alexandros Giagkos (alg25, LL-C42)

Aims and Objectives

- To introduce to agile software development practices and processes
- To provide a strong foundation in agile software engineering and to prepare to work in the software industry

Learning Outcomes

- “Make an effective contribution to an agile software development team”
- “Employ advanced software development skills to create a software product”
- “Communicate the decisions, methods, and practices that resulted in a software product.”

Assessment

- Delivery:
  - 10 x 2 Hour Lectures in CIS
  - 10 x 2 Hour Practicals in MP-3.03
- Assessment:
  - 100 Hours Group Project (70%)
    - product + small individual reports with your personal reflection
  - .5 Hours Group Presentation (30%)

Term dates

- Week 1 (28/1)  You are here!
- Week 2 (4/2)
- Week 3 (11/2)
- Week 4 (18/2)
- Week 5 (25/2)
- Week 6 (4/3)
- Week 7 (11/03)
- Week 8 (18/3)
- Week 9 (25/3)
- Week 10 (1/4)
- Another teaching week (29/4)
- Reading week (6/5)

Term dates

- Week 1 (28/1)
- Week 2 (4/2)
- Week 3 (11/2)
- Week 4 (18/2)
- Week 5 (25/2)
- Week 6 (4/3)
- Week 7 (11/03)
- Week 8 (18/3)
- Week 9 (25/3)
- Week 10 (1/4)

Presentation and reports →

- Another teaching week (29/4)
- Reading week (6/5)

← **Product delivery and demo**

General plan

- I will deliver something new every Monday for 30-40 mins
- We will then “go agile” for the rest of the time

- We will “go agile” every Thursday for 2 hours :)
  - Agile meetings, discussions, demos

Overview

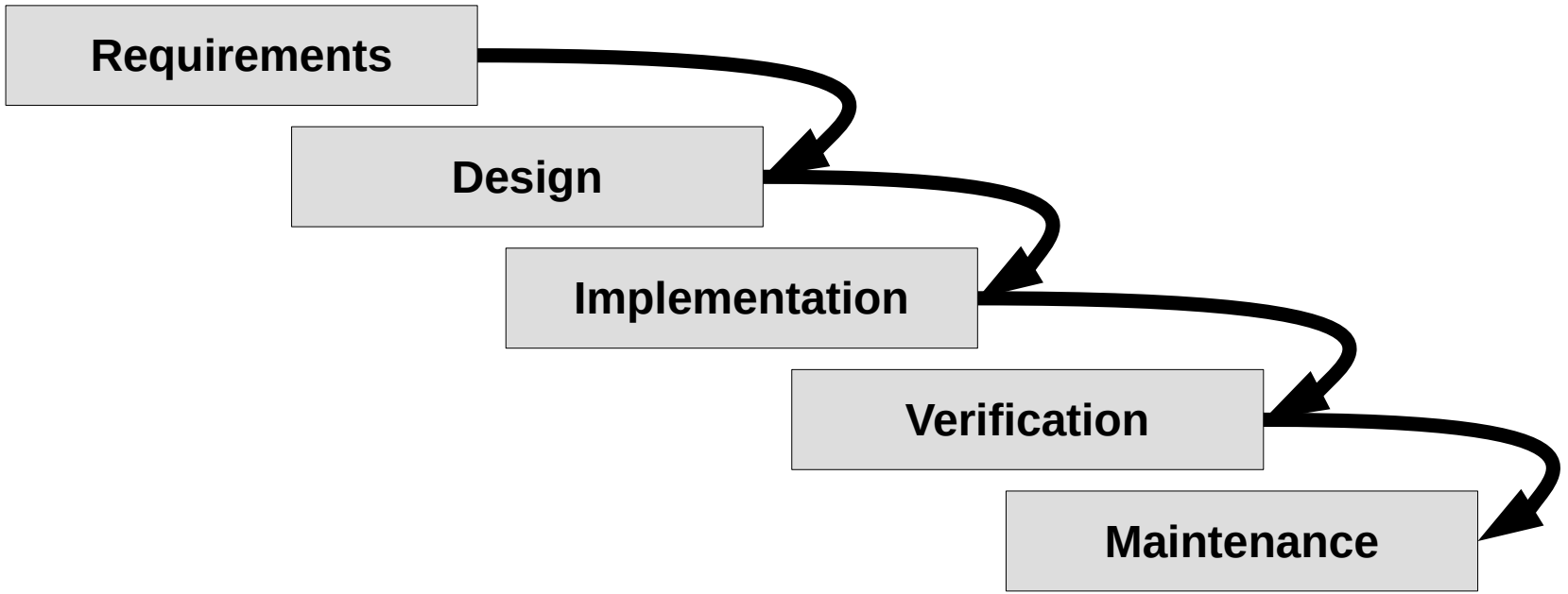
- Introduction to agile software development
- Introduction to Scrum
- Preparation before the project is released

What is agile?

- An approach to software development
  - Set of methods and methodologies
  - Sets of values, principles and practices
- A movement, a mindset

“We do not act rightly because we have virtue or excellence, but we rather have those because we have acted rightly. We are what we repeatedly do. Excellence, then, is not an act but a habit.”
– Aristotle, Nichomachean Ethics

The waterfall model



Original paper by W.W.Royce in 1970

The waterfall model (cont.)

- BRUF: Big requirements up front
- Poor communication
- Overly rigid documentation
- Inability to handle change
  - Bugs
  - Dirty – spaghetti code
  - Misery

A fractured perspective

- “Software crisis” dates back to the 70s
  - Waterfall model was widely adopted, yet project failed
- Different views of both requirements and practices
  - Understanding the elephant metaphor
- Better-than-not-doing-it gives results

Manifesto of 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 terms 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 Principles

- *"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 preference to the shorter timescale.*

Better delivery

Agile Principles (cont.)

- *"The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**."*
- *"**Businesspeople 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."*

Better communication

Agile Principles (cont.)

- *"**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."*

Better execution

Agile Principles (cont.)

- *"**Simplicity** – the art of maximising the amount of work not done – is essential."*
- *"The best architecture, requirements, and designs emerge from **self-organising teams**."*
- *"At regular intervals, the **team reflects** on how to become more effective, **then tunes and adjusts** its behaviour accordingly."*

Better working environment

Agile methodologies

- Frameworks consisting of agile values, principles & practices
- Require you have an agile mindset and discipline
- No silver bullet
  - Good communication
  - Good practices
  - Useful documentation

The Scrum

- Roots in a 1986 paper by Takeuchi and Nonaka, *“The new new product development game”*. It identifies:
  - Built-in instability
  - Self-organising project teams
  - Overlapping development phases
  - Multi-learning
  - Subtle control
  - Organisational transfer of learning

The Scrum (cont.)

- Developed in 1990
- Formalised by Sutherland and Schwaber in 1995
- Scrum (n): A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.
- Lightweight, simple to understand, difficult to master

The Scrum Guide overview

- Pillars and values
- Team roles – Product Manager, Development Team and Scrum Master
- Events – Meetings and Sprints
- Artifacts – Work or value that needs to be tracked and completed

Scrum pillars

- **Transparency** – everybody involved in delivering the project should understand what is expected and how the process works
- **Inspection** – review the progress at regular intervals
- **Adaptation** – change a process if it is not working, or is not delivering a sensible result

Scrum values

- Pillars come to life and build trust for everyone only when **commitment, courage, focus, openness** and **respect** are embodied and lived by the team

Team roles

- **Product Owner** – responsible for maximising the value of the product and work of the Development Team
  - One person, not a committee
- **Development Team** – necessary skills to successfully deliver the product
  - Self-organising and cross-functional
- **Scrum Master** – support for the Product Owner, the Development Team, and the organisation
  - *This is not about management*

Events

- **Sprint** – time-box in which the team produces a “done” product increment
- **Meetings:**
  - Sprint planning
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospective

Artifacts

- **Backlog** – ordered list of everything that is needed in the project, i.e. features and fixes
  - Product backlog
  - Sprint backlog
- **Artifact transparency**

Product Owner

- The sole person responsible for managing the Product Backlog:
  - Clearly express Product Backlog items;
  - Order the items in the Product Backlog to best archive goals and missions;
  - Optimise the value of the work the team performs;
  - Ensure transparency of backlog
- May cancel a sprint

Development Team

- Self-organising
- Cross-functional
- No titles for members
- No sub-teams
- Accountability belongs to the team as a whole

Scrum Master

- Ensures that agile methods and Scrum are employed
- Ensures transparency of artifacts
- Facilitates scrum events as requested or needed
- Coaches the development team in self-organisation and cross-functionality
- Removes impediments to the progress
- Helps stakeholders understand Scrum

Back to Term dates

- Week 1 (28/1)
- Week 2 (4/2)
- Week 3 (11/2)
- Week 4 (18/2)
- Week 5 (25/2)
- Week 6 (4/3)
- Week 7 (11/03)
- Week 8 (18/3)
- Week 9 (25/3)
- Week 10 (1/4)

- You will decide about the roles
- Project to be introduced on Thursday (Backlog to be initiated)
- 1 or 2 week sprints
- **Retrospective** before every new **sprint planning** (Mondays)
- **Daily scrum** (Thursdays)
- **Sprint review** (possibly a demo) at the end of each sprint (Mondays)

Summary

- Introduced agile software development
- Introduced to Scrum
- Some preparation before the project is released