

DIGHEL4360: Software engineering

02-11-2022

Magnus Li

Senior lecturer and Researcher
magl@ifi.uio.no

Department of Informatics and the HISP Center
University of Oslo



**UNIVERSITETET
I OSLO**

Today

- Last week we had a brief introduction to “IT projects”
- Today, we zoom in on how software development teams organize their work.
- The objective is not for you to know everything, but to gain some familiarity with some typical concepts.
- Relevant as:
 - you may encounter such concepts in your master’s projects
 - After the master, you may be part of, or have to interact with software teams

Today

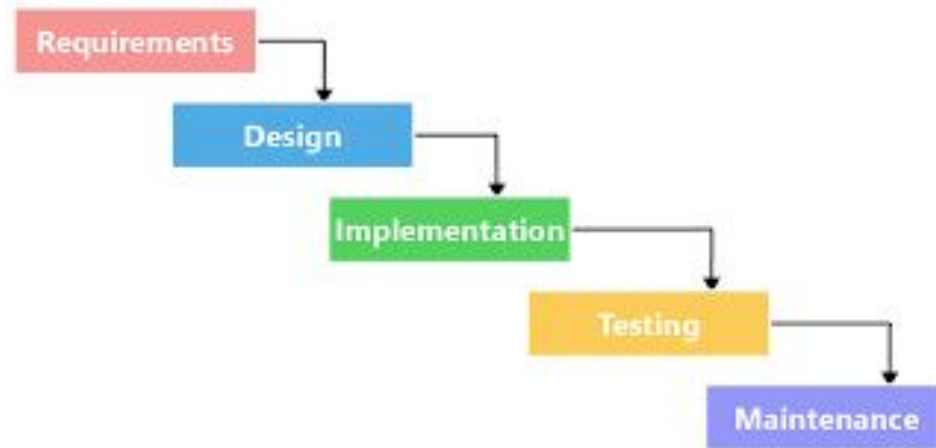
- Some fundamental concepts
 - “Software engineering”
 - Plan-driven software development vs. agile
 - DevOps, scrum and kanban
- Guests: Raymond Sollie & Malin Aandahl from Capra Consulting

Software engineering

- The process of building and maintaining software solutions.
- Not just the writing of code, but all processes and tools involved.
 - Specify what the system should do - “requirements”
 - Design of the system: architecture, data models, functionality, user interfaces, etc.
 - Development/implementation of the software (coding)
 - Validation and testing
 - Maintenance: changing the system according to new requirements (and technology)
- A whole discipline around how to best organize such processes and activities!

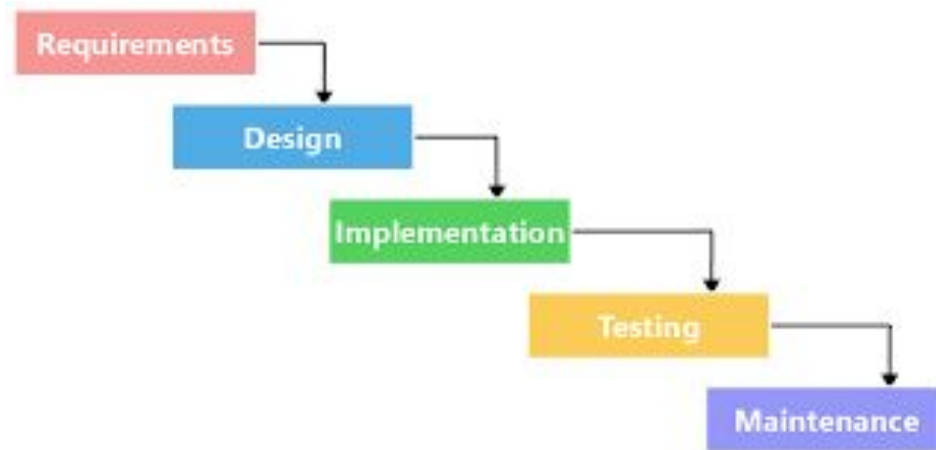
Plan-driven development

- Based on traditional engineering
- Significant detailed up-front planning



Plan-driven development

- Based on traditional engineering
- Significant detailed up-front planning
- Problem?



Plan-driven development

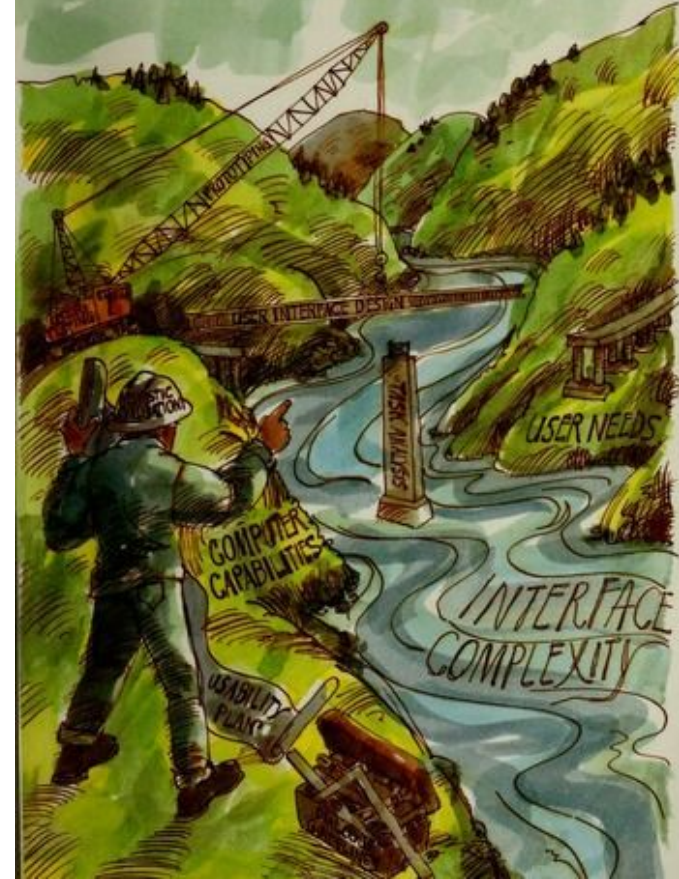
Software engineering - same as other domains of engineering?

- Organizations are “sociotechnically” complex
- Organizations are evolving
- Organizations are heterogeneous
- Digital technology is rapidly evolving

→ Difficult to plan and anticipate outcomes

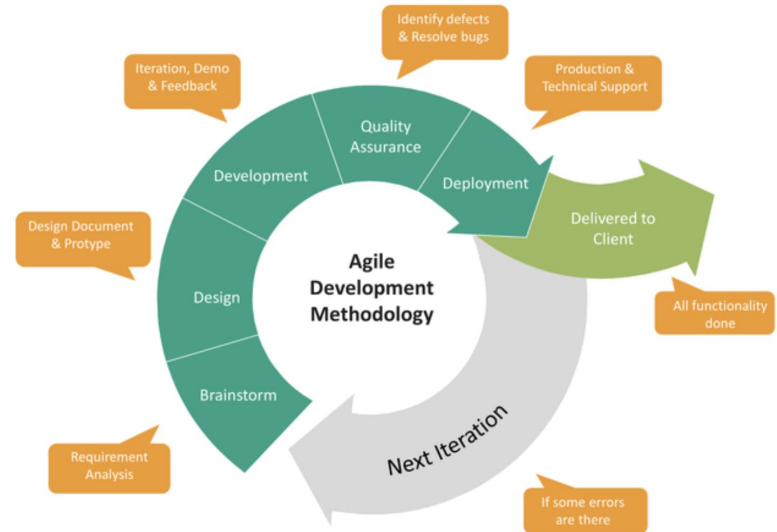
Usability Engineering

Jakob Nielsen



Agile development

- Less up-front planning
- Project as a learning system
- Iterative planning, development, implementation - rapid learning cycles
- Begin with minimum viable product (MVP)
- Each cycle produces an “increment”

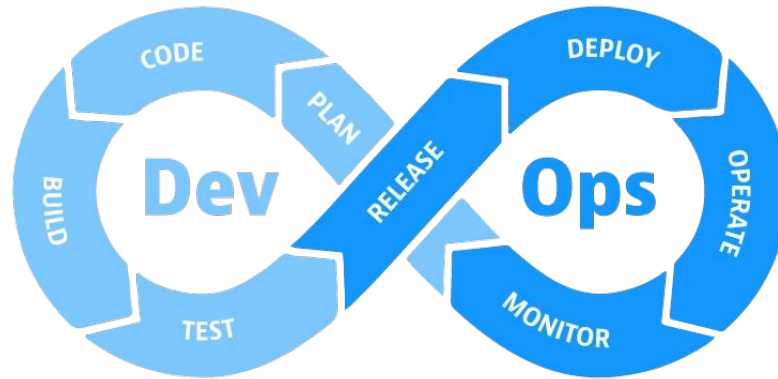


The “agile manifesto”

- 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.
- 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 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.
- 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**.

DevOps

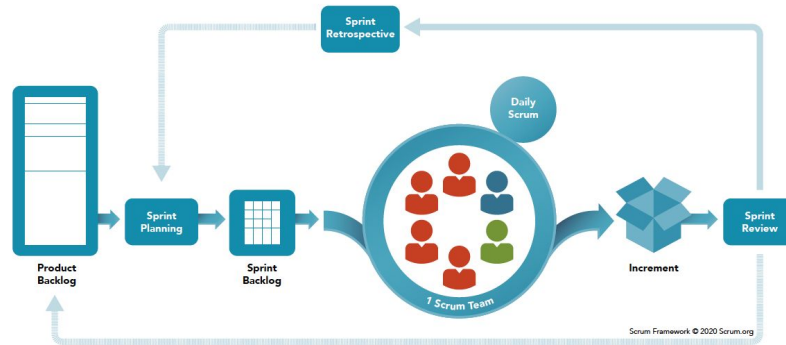
- As extension of agile - integrating the software development phase and operations phase



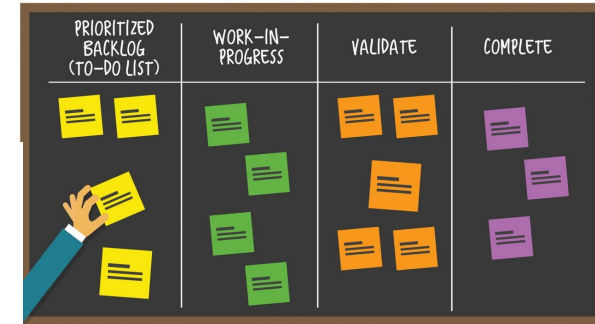
Task and time boxing

- A key question: how to organize work within the project teams?
- Time boxing: SCRUM
- Task boxing: Kanban

SCRUM FRAMEWORK



KANBAN BOARD



Example: DHIS2

- For each part of DHIS2: e.g., “analytics team”, “tracker team”:
- Requirements reported to “jira” from several sources
- Used as basis for “prioritization meetings”
- Results in a list of requirements, representing tasks (a “backlog”)
- On a regular basis, “sprints” are planned where certain tasks are assigned to developers within the team.
- Amounts to a release of new version.

[DHIS2 Jira](#)

Think and note

- Can you imagine any challenges with using agile in larger IT projects in the health sector?

Write some quick bullet points for 1 min...

Some challenges

- “Scaling” agile to larger projects can be difficult
- Planning: control and predictability versus flexibility
- What is an MVP, for instance in a electronic patient journal system?

→ Some promote “hybrid” approaches (agile + elements of plan-driven)