DIGHEL4360:
Organizing software teams and software engineering

Johanne Thunes,
PhD candidate
Department of Informatics

Today's agenda

Part 1: Software development and software teams
Part 2: Practical tasks and discussion

Goal: Familiarize yourself with terms and concepts in the IT world

Software engineering

What is software engineering?

According to MIT: “Software engineering is the branch of computer science that deals with the design, development, testing, and maintenance of software applications. Software engineers apply engineering principles and knowledge of programming languages to build software solutions for end users.”

https://www.mtu.edu/cs/undergraduate/software/what/

System development

What is system development?

Summed up: The doctrine of developing and managing high-quality software systems within given time and cost constraints.

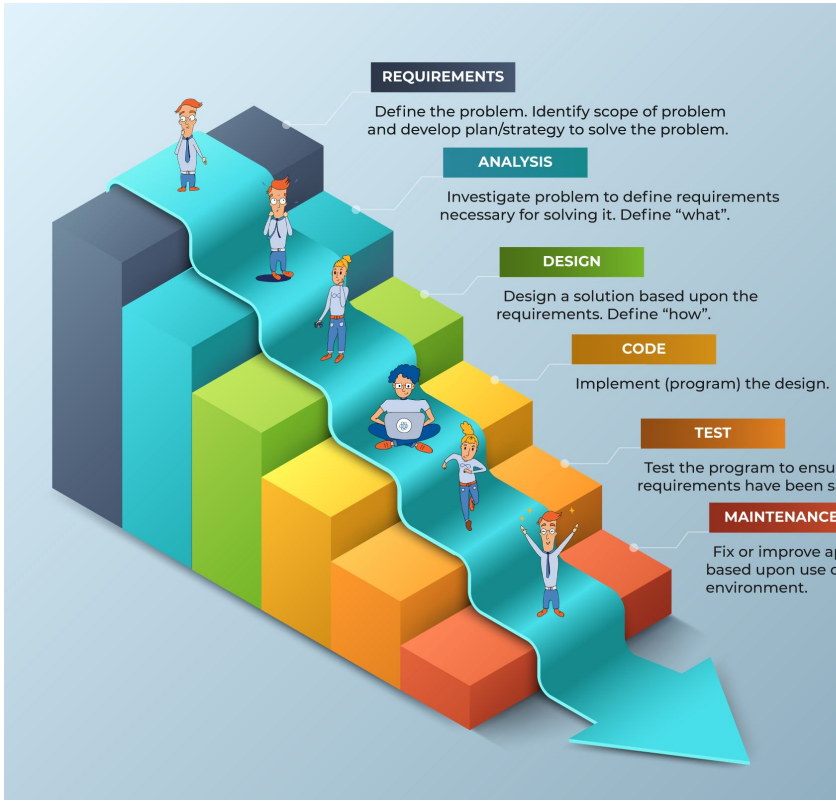
Two **main approaches** to software development:

Plan-driven development (waterfall)**:** Start - develop - finish
Agile development: Starting small, scaling in iterations

Plan-driven development

- The requirements and ideas is planned “up front”
- Clear start and finish
- Requirements are usually formalized in official documents
- When one step of the process is finished, you move on to the next (usually you don’t go back to a previous step in the process)

Can you think of any challenges with plan-driven processes?



https://kruschecompany.com/waterfall-software-development-methodology/

Staten tapte ankesak – må ut med 235 millioner kroner etter bompengefiasko

Vegvesenet og den amerikanske IT-giganten IBM har kranglet om bompengeprosjektet Grindgut siden 2015. Nå er statens anke forkastet.

Agile development

- Agile is about working in **iterations** (you return to each step in the process several times)
- Planning is done little by little, continuous changes. You don't always know what the best solution is up front!
- Scrum, kanban & scrumban
- Devops

The Agile manifesto: 12 principles

1	Our highest priority is to satisfy the customer through the early and continuous delivery of valuable software.	7	Working software is the primary measure of progress.
2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	9	Continuous attention to technical excellence and good design enhances agility.
4	Business people and developers must work together daily throughout the project.	10	Simplicity–the art of maximizing the amount of work not done–is essential.
5	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	11	The best architectures, requirements, and designs emerge from self-organizing teams.
6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

https://agilemanifesto.org/iso/no/principles.html +
https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/

Agile processes: **Scrum**

- Choose tasks and work with them in a set time interwall
- Time intervals are called **sprints** (time boxing) and you have a **product backlog**
- **Important roles:** Scrum master, product owner, scrum team
- **Some scrum meetings:**
  Sprint planning, daily stand-up, retrospective, spring review

Questions for daily stand-ups meetings:
What did you do yesterday?
What are you going to do today?
Do you have any blockers?

https://www.nimblework.com/agile/scrum-methodology/

Agile processes: **Kanban**

- Tasks have a continuous flow, tasks are worked on until finished
- Reduces **bottlenecks** in the process
- Work in **task-boxes,** inspired by Toyota
- **WIP:** Work in progress
  - Secures flow, production stops to solve problems if needed, goals is to secure against errors
- **JIT:** Just in time
  - Tasks are added to kanban board “just in time” in order to control WIP.

https://www.nimblework.com/agile/scrum-methodology/

User stories - communicating user needs

Often included in product backlog

Format: “As a [user] I want to [some particular feature] so that [some benefit] is received.”

Example: As a student I want to know where my lectures are, so that I can participate in the lectures.

The Scrum Framework At a Glance

Version 5.0

The Product Backlog is an emergent, ordered list of what is needed to improve the product and includes the Product Goal.

Product Backlog (owned by PO)

Product Owner

Refinement

Refinement is a continuous activity used to add detail, estimates, and order to items in the Product Backlog.

Sprint Planning establishes the Sprint Goal, what can be done, and how the chosen work will be completed.

Sprint Planning

Sprint Backlog (owned by the Developers)

The Sprint Backlog is the set of Product Backlog items selected for the Sprint by the Developers, plus a plan for delivering the product Increment and realizing the Sprint Goal.

ScrumMaster

Daily Scrum

The purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work.

***Sprint of 1 month or less** (the heartbeat of Scrum)

Each Sprint should bring the product closer to the overall Product Goal.

The purpose of the Sprint Review is to inspect the outcome of the Sprint with stakeholders and determine future adaptations.

Sprint Review

Increment

A sum of usable Sprint Backlog items completed by the Developers in the Sprint that meets the Definition of Done, plus the value of all of the increments that came before.

Developers

The Sprint Retrospective is when the Scrum Team inspects how the last Sprint went with regards to individuals, interactions, processes, tools, and Definition of Done. The Team identifies improvements to make the next Sprint more effective and enjoyable. This event concludes the Sprint.

Sprint Retrospective

Who should participate and for how long?

SPRINT PLANNING:

Who: *The entire Scrum Team*

Timebox: *Maximum of 8 hours for a month-long Sprint. Shorter timebox for shorter Sprints.*

DAILY SCRUM:

Who: *The Developers, but if the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers.*

Timebox: *Maximum of 15 minutes.*

SPRINT REVIEW:

Who: *The entire Scrum Team. Stakeholders are invited to provide feedback on the increment.*

Timebox: *Maximum of 4 hours for a month-long Sprint. Shorter timebox for shorter Sprints.*

SPRINT RETROSPECTIVE:

Who: *The entire Scrum Team.*

Timebox: *Maximum of 3 hours meeting for a month-long Sprint. Shorter timebox for shorter Sprints.*

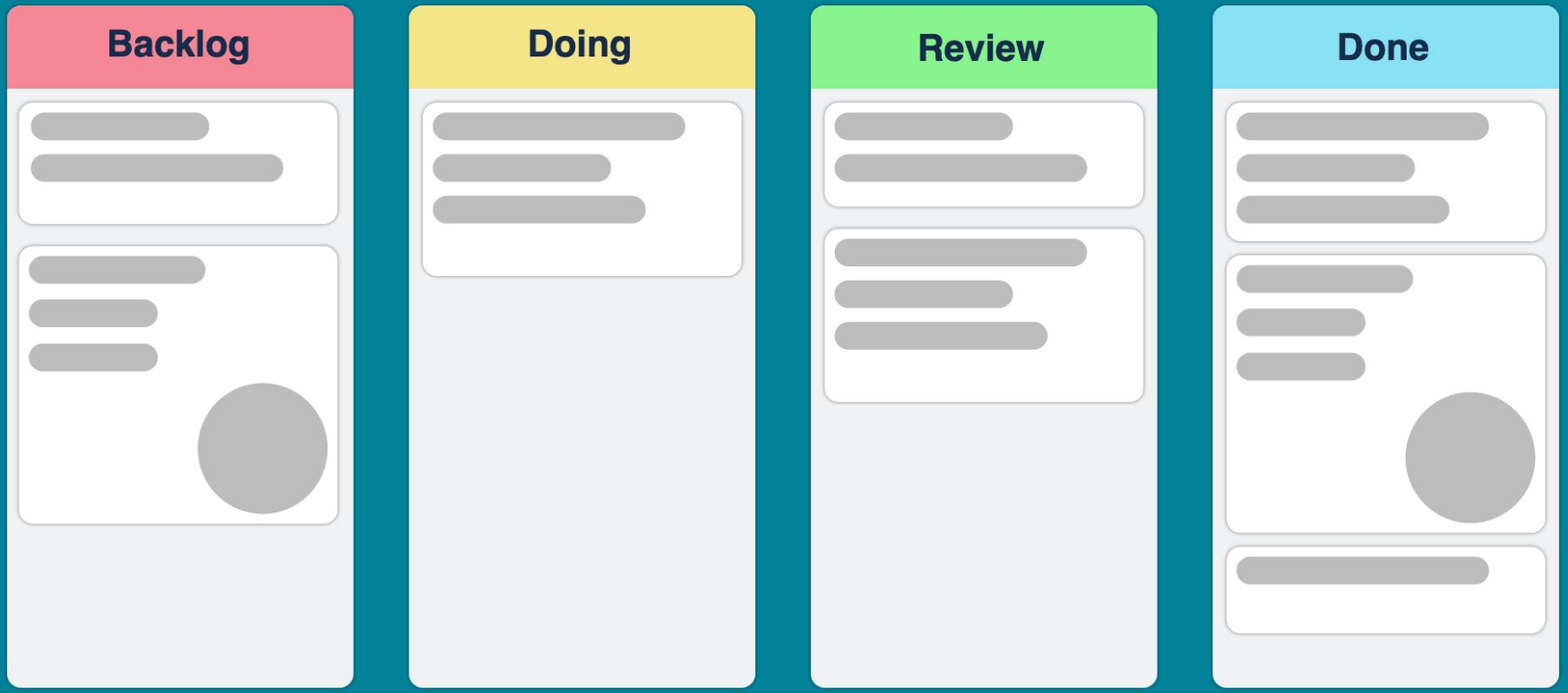
Scrum Alliance®

○ = Scrum artifacts that help manage the work

○ = Events or ceremonies that occur inside each Sprint

■ = Ongoing activity

Kanban Board



Scrum vs. Kanban

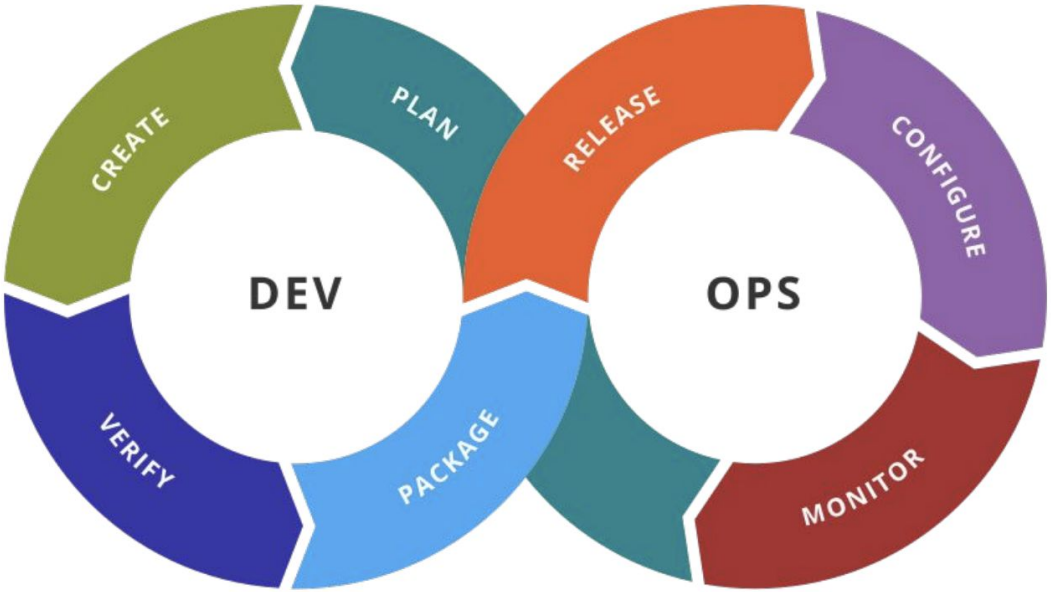
	Scrum	Kanban
Cadence	Regular fixed length sprints (ie, 2 weeks)	Continuous flow
Release methodology	At the end of each sprint	Continuous delivery
Roles	Product owner, scrum master, development team	No required roles
Key metrics	Velocity	Lead time, cycle time, WIP
Change philosophy	Teams should not make changes during the sprint.	Change can happen at any time

Agile processes: Scrumban

What is scrumban?

A combination of kanban and scrum

DevOps


CREATE
PLAN
RELEASE
CONFIGURE
DEV
OPS
VERIFY
PACKAGE
MONITOR

Devops

DevOps-prinsipp	Forklaring
Alle ansvarlige for alt	Alle i teamet har delt ansvar for utvikling, utgivelse og vedlikehold/support av programvaren.
Alt som kan bli automatisert burde bli det	All testing-, utgivelse- og supportsaktiviteter bør bli automatisert når det er mulig. Legg opp til minst mulig manuelt arbeid med utgivelsen av programvaren.
Mål først, endre etterpå	DevOps burde bli drevet av målingsprogram hvor du samler data om systemet og operasjonene. Avgjørelser som angår endring i DevOps prosessen og verktøy, bør tas med denne dataen som grunnlag.

Sommerville, kap. 10, s. 297

Interdisciplinary teams

- Developers (frontend, backend, fullstack), designers (UX, service, visual), software testers
- Project leaders, scrum master, product owner
- The user
- People from use contexts

Can you think of any challenges with interdisciplinary teams?

Think for yourself: Can you think of any challenges with agile development?

Think for yourself: Can you think of any challenges with agile development?

Group work - plan an IT project

Objective: In this group task, you will work together to plan an IT project within the health sector. The project can follow either a plan-driven or agile approach, based on your group's preference.

Case Scenario: A medical research organization has approached your team to develop an IT solution that will support their efforts in collecting and analyzing health data from multiple sources. The organization is focused on improving research outcomes by enhancing the quality and diversity of data available to researchers, and streamlining data collection and analysis workflows. They also want the solution to comply with data privacy and security regulations, and to provide scalable infrastructure to support their growing data needs.

Group Task:

- Divide yourselves into teams of 3-4 members
- Identify at least two user stories from the case scenario
- Choose a software process (plan-driven or agile) based on your group's preferences. Justify your choice with appropriate reasons.
- Create a brief project schedule with tentative timelines with prioritized tasks
- Determine the roles and responsibilities required to successfully complete the project.

Thank you!