



GÖTEBORGS
UNIVERSITET



CHALMERS

DAT257 Agile Software Project Management

Git and Agile Workflow for Software Projects

Hans-Martin Heyn, Senior Lecturer, 2023

Your teachers



Jonas Petrán

Agile Coach

HiQ



Hans-Martin Heyn

Senior Lecturer

Software Engineering Division

Your Teaching Assistants

Omar	Sulaiman
Louis	Mercier
Sogeta	Albazi
Fanny	Söderling
Mosope	Williamson
Navya	Pulikandla Satyanarayanachetty

- They will guide and help you for the project
- All of them have taken this, or a similar course, before
- From both programs (CSE and Industrial EngineerinG)

Student Representatives

- Are you already a student representative, Chalmers should have contacted you.
- If you want to be a student representative, you are more than welcome to send an e-mail too and sign up.
 - => Write an e-mail to heyn@chalmers.se
- We will have 2 meetings during the course (one in the beginning, one around mid-term).
- We will have one meeting after the course is finished.
- Your feedback is important!!

Today's agenda

1 Software Engineering

2 Course details

3 KPIs, forming teams
& team exercise

Why software engineering?

PSY - GANGNAM STYLE (강남스타일) M/V

officialpsy

Subscribe 7,600,830

-2142584554

8,761,309 1,138,933

Add to Share More

CNN BUSINESS Markets Tech Media Success Perspectives Videos

Markets →

DOW	31,927.17	0.05% ▲
S&P 500	3,867.01	0.14% ▲
NASDAQ	11,220.50	0.73% ▲

Fear & Greed Index →

Extreme Fear is driving the US market

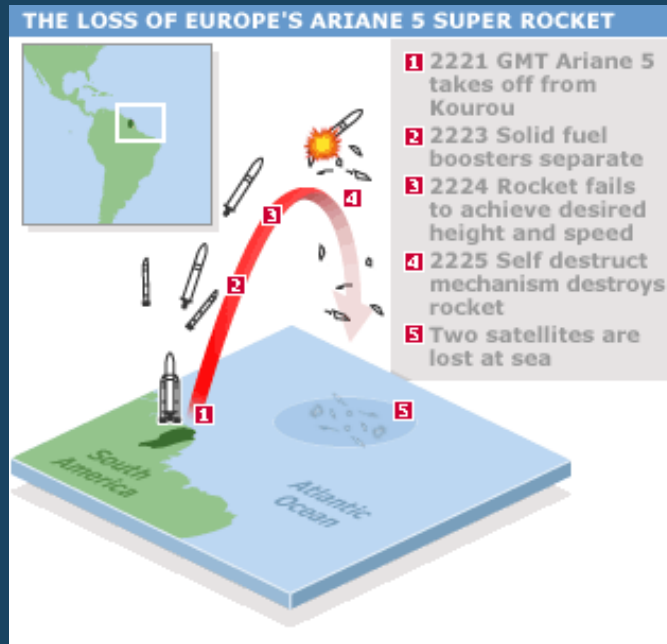
20

PayPal accidentally credits man \$92 quadrillion

Sho Wills, CNN

Updated 9:55 AM EDT, Wed July 17, 2013

f t e l



© BBC World

Software systems are complex

“The complexity of software is an essential property,
not an accidental one.” Fred Brooks, 1986

No Silver Bullet —Essence and Accident in Software Engineering

Frederick P. Brooks, Jr.
University of North Carolina at Chapel Hill

1987

There is no single development, in either technology or management technique, which by itself promises even one order-of-magnitude improvement within a decade in productivity, in reliability, in simplicity.

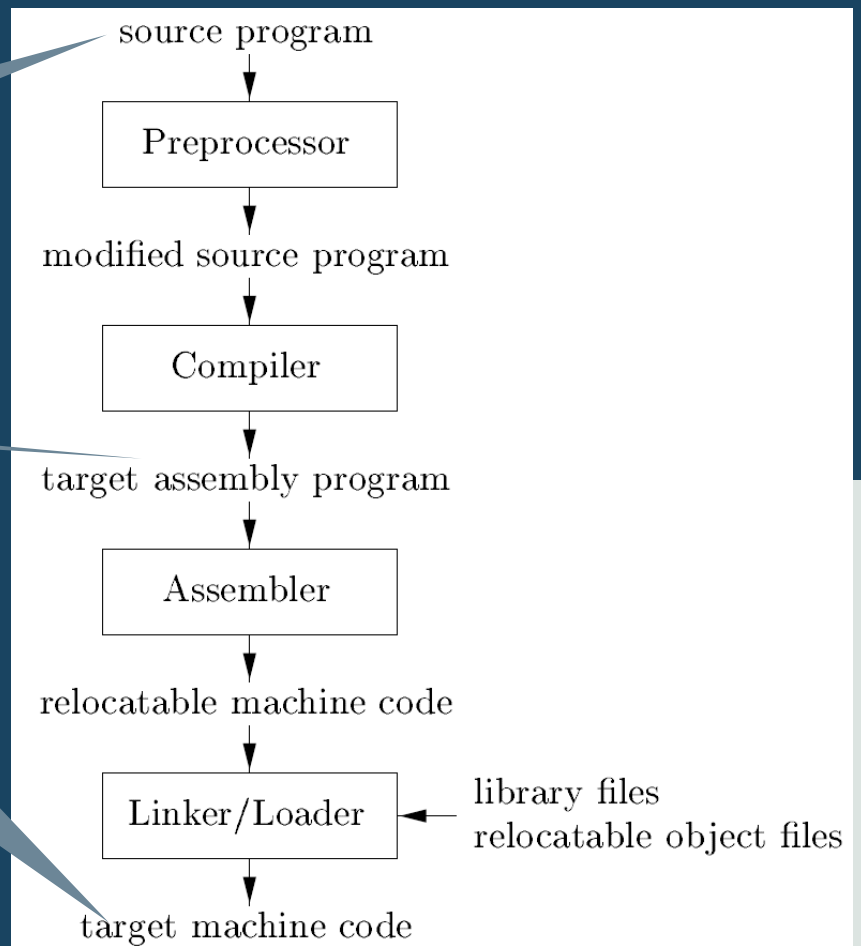
Abstraction...

3rd/4th generation language, 3/4GL

2nd generation language, 2GL

1st generation language, 1GL

**In this course, we
consider development
processes to handle
complexity!**



The software crisis

Projects running over-budget.
Projects running over-time.
Software was very inefficient.
Software was of low quality.
Software often did not meet requirements.
Projects were unmanageable
and code difficult to
maintain.



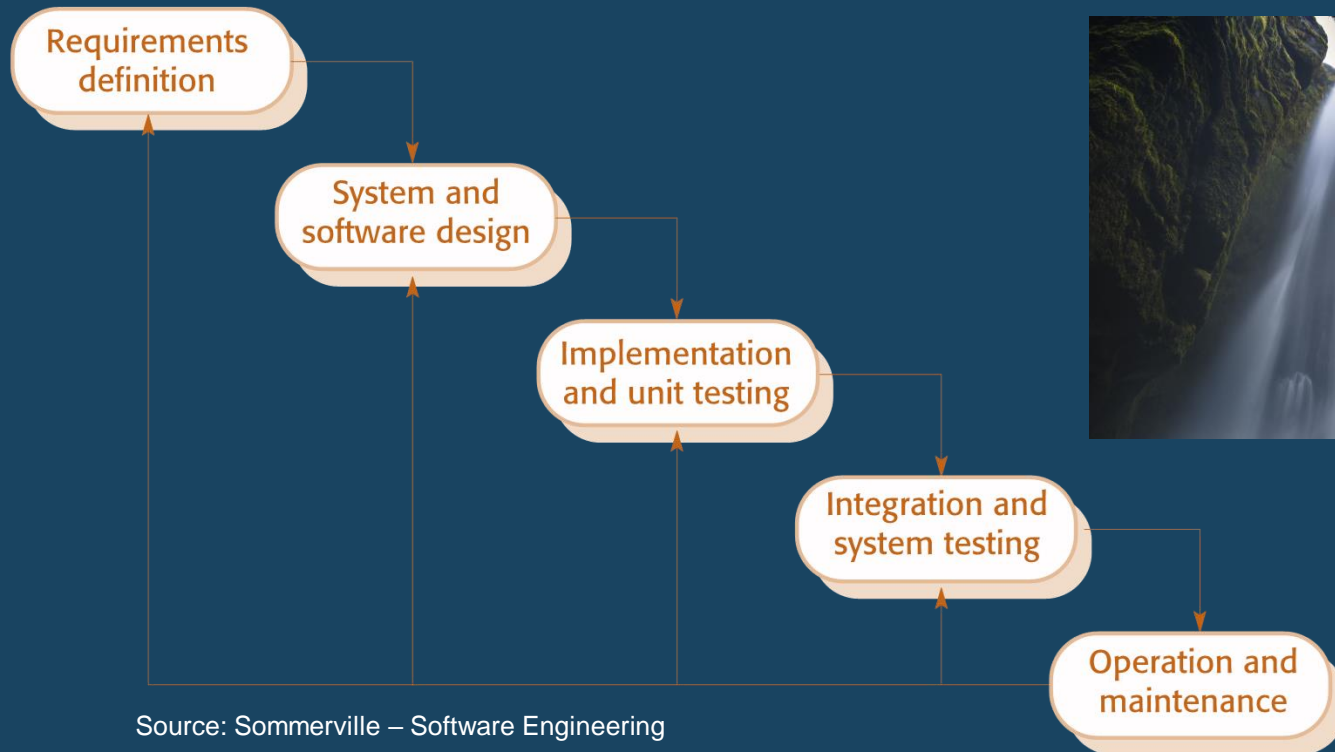
The first NATO Software Engineering Conference, Germany, 1968

Software Engineering

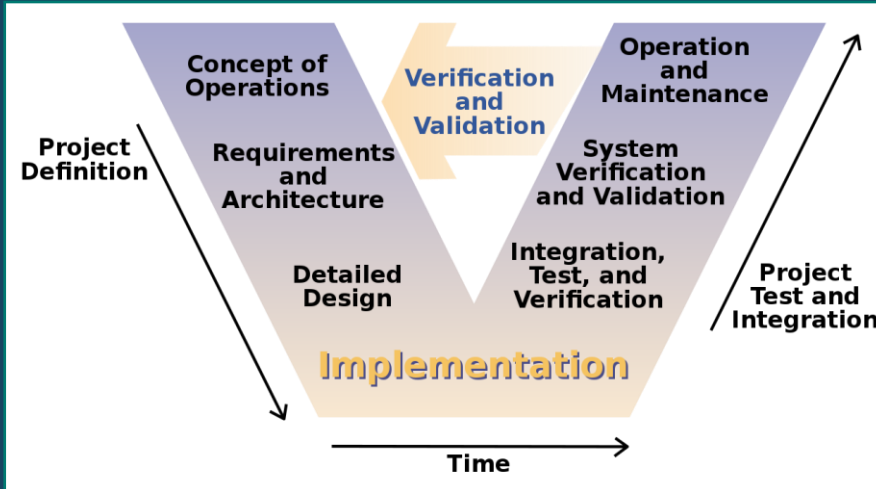
Systematic & disciplined approach
to the development and maintenance
of software
to assure quality of processes and products

Name and definition inspired by „successful“ / mature engineering
disciplines, like mechanical engineering

Waterfall project approach



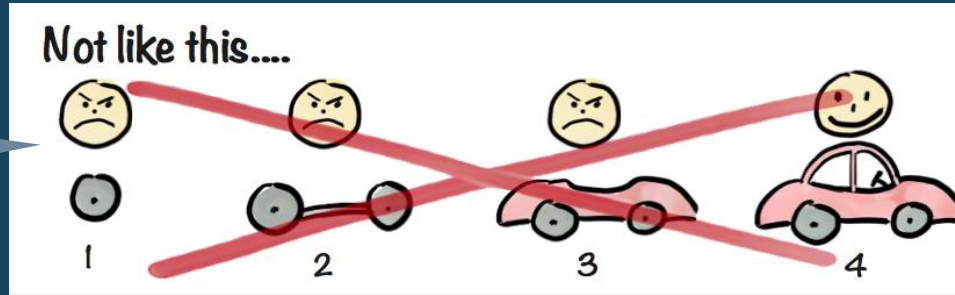
The V-Model



- The complexity of today's software systems can no longer be foreseen and understood "from the beginning" (i.e., in the concept phase and in requirements engineering).
- Since requirements and the context situation are constantly changing, the system to be developed must also be continuously adapted.
- Long development cycles lead to users who are dissatisfied with the software functionality.

Why a new development paradigm?

Customer
requirement: “Car”



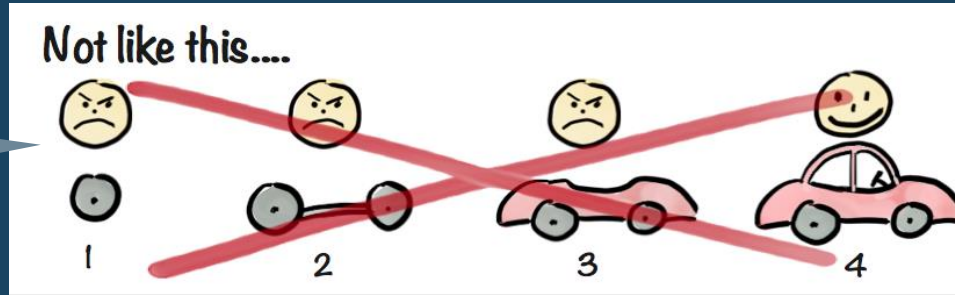
Source: <http://blog.crisp.se/2016/01/25/henrikkniberg/making-sense-of-mvp>

The one long
development
cycle leads to an
end result with
which the
customer is
(hopefully!)
satisfied

- Long Development cycles can make the product “out-dated” before it even reached the customers.
- Testing and validation can take very long in very complex (monolithic) software systems.
- Teams can change during the development.

Why a new development paradigm?

Customer
requirement: "Car"



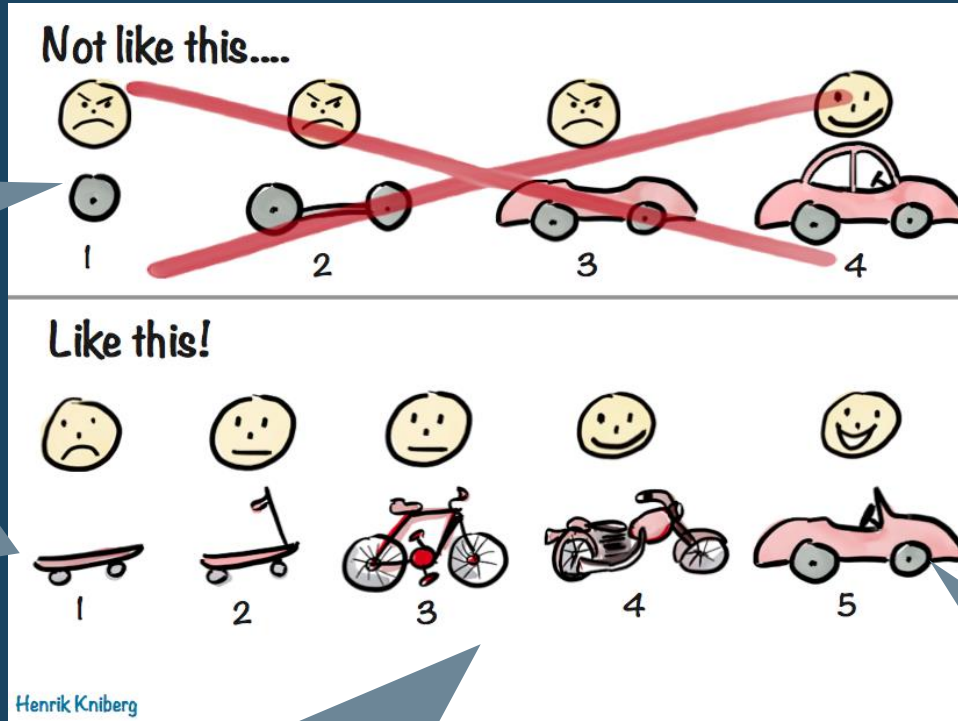
Source: <http://blog.crisp.se/2016/01/25/henrikkniberg/making-sense-of-mvp>

The one long
development
cycle leads to an
end result with
which the
customer is
(hopefully!)
satisfied

- Long Development cycles can make the product "out-dated" before it even reached the customers.
- Testing and validation can take very long in very complex (monolithic) software systems.
- Teams can change during the development.

Customer
requirement:
“Car”

Customer
requirement:
Initially
“mobility”,
then
incremental
refinement



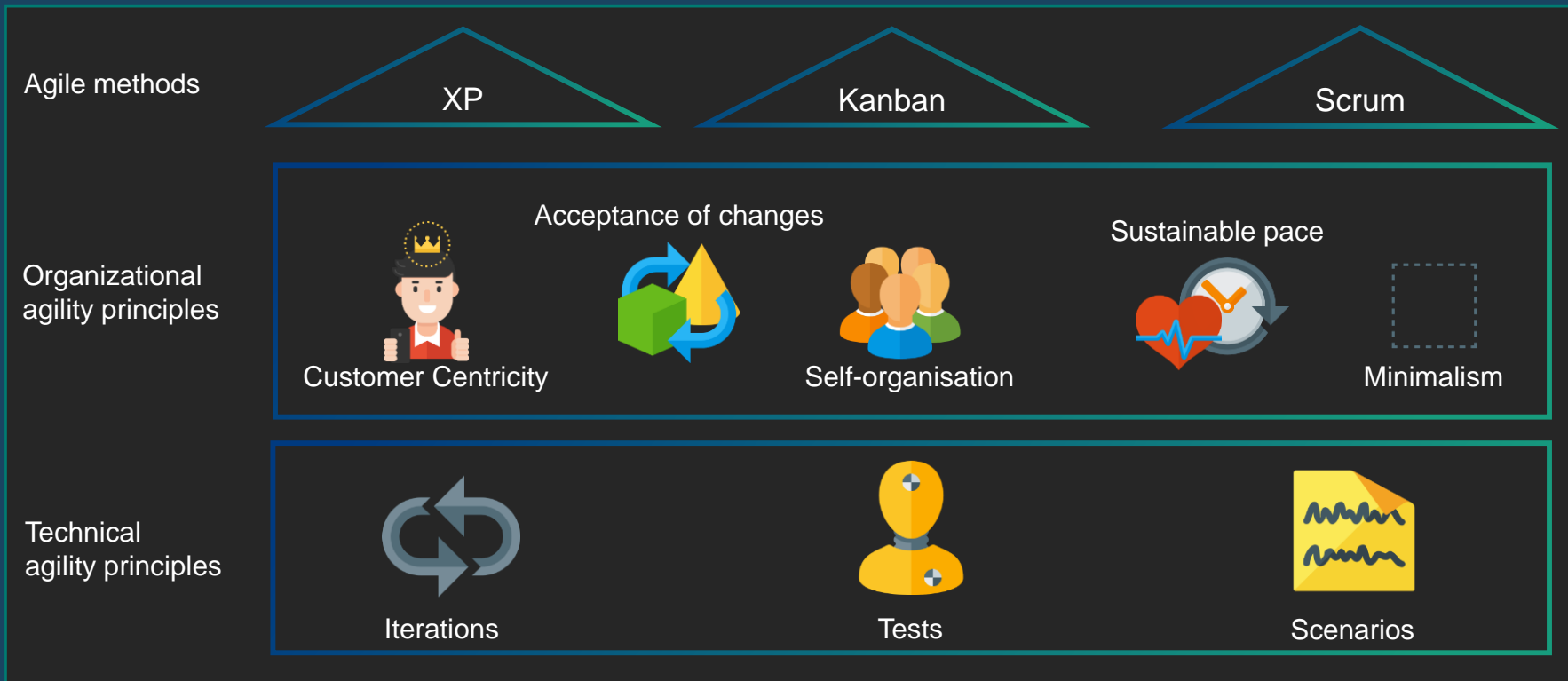
The one long
development
cycle leads
to an end
result with
which the
customer is
(hopefully!)
satisfied

Through the value of the
intermediate results and
the incremental
incorporation of customer
feedback, customer
satisfaction with the final
result is increased

Customer receives valuable and
incrementally improved prototypes at the
end of several short development cycles,
which the developers get feedback on

Agile is a Mindset

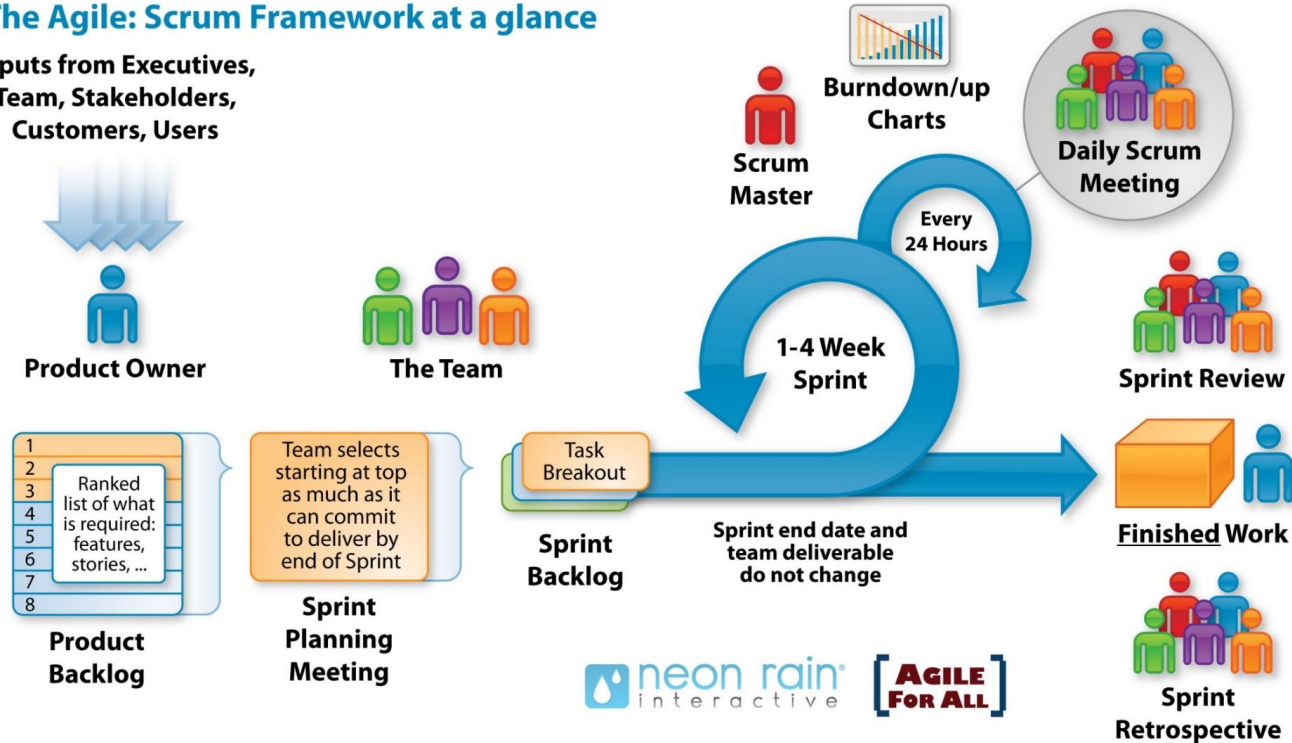
Concrete Methods Implement the Idea of Being Agile



Scrum

The Agile: Scrum Framework at a glance

Inputs from Executives,
Team, Stakeholders,
Customers, Users



Today's agenda

1 Software Engineering

2 Course details

3 KPIs, forming teams
& team exercise

Overview of the course

...this is a practical project course

Week 12 13 14 15 16 17 18 19 20 21 22

Lecture

Lectures
1-2

Lectures
3-5

Lecture 6

Exercise

Lego

Project

Team

Scope

Eastern

1st
Sprint

2nd
Sprint

3rd
Sprint

4th
Sprint

5th
Sprint

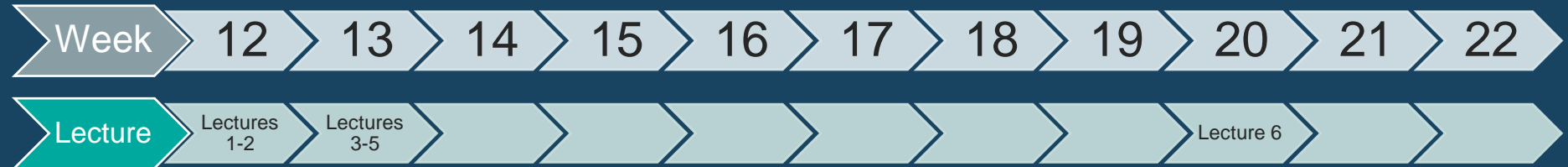
6th
Sprint

Video

Report

Overview of the course

...this is a practical project course



Lecture 2: Working in agile teams

Lecture 3: Scrum and Software Quality

Lecture 4: Building the product

Lecture 5: Q&A about the project

Lecture 6: Revisiting the course, Final Report

Sign up for the exercise session

...it starts tomorrow!

Week

12

13

14

15

16

17

18

19

20

21

22

Exercise

Lego

The agile exercise

- Three similar sessions
 - You need to sign-up for one of the three sessions:
 - Session 1: Tuesday, 21.04 08:45 – 11:30
 - Session 2: Tuesday, 21.04 13:15 – 16:00
 - Session 3: Friday, 24.04 13:15 – 16:00

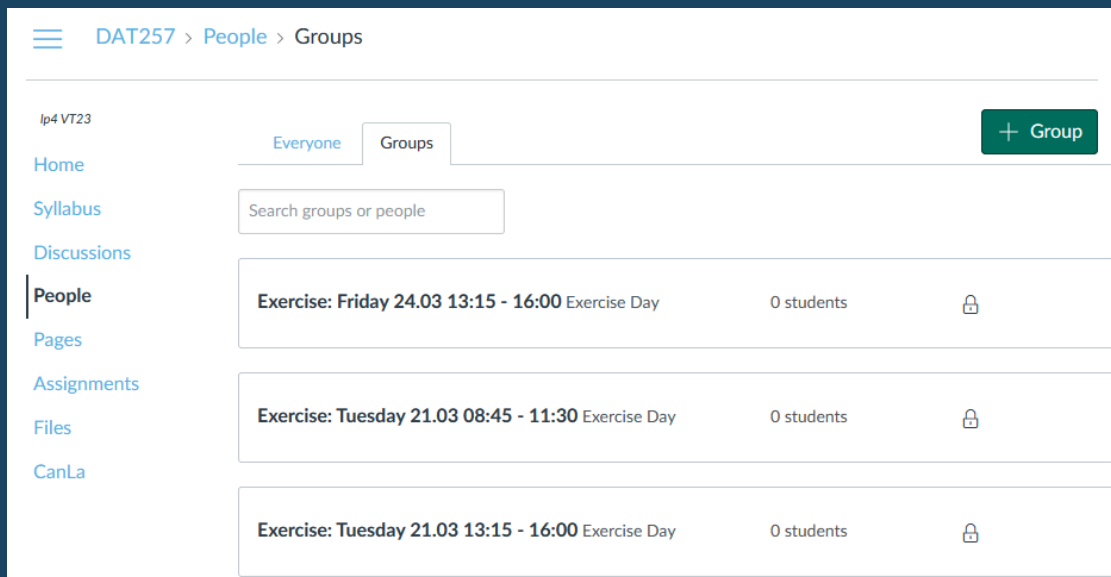
Sign up for the exercise session

...now!

Week 12 13 14 15 16 17 18 19 20 21 22

Exercise Lego

Sign-up by
choosing one of
the exercise
groups in canvas

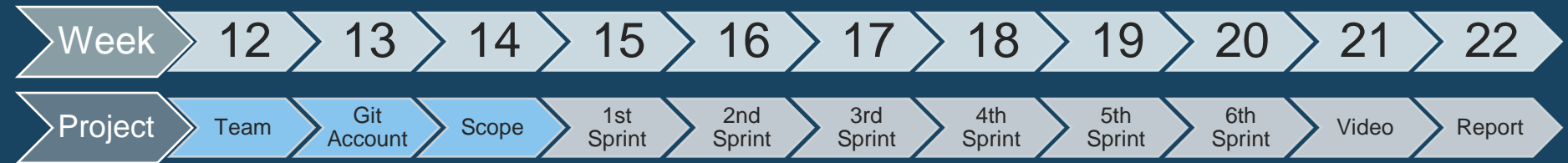


Canvas LMS interface showing the 'Groups' page for course DAT257. The page displays a list of exercise sessions, each with 0 students and a lock icon.

Exercise Session	Students	Status
Exercise: Friday 24.03 13:15 - 16:00 Exercise Day	0 students	Locked
Exercise: Tuesday 21.03 08:45 - 11:30 Exercise Day	0 students	Locked
Exercise: Tuesday 21.03 13:15 - 16:00 Exercise Day	0 students	Locked

How to pass the project...

...without getting lost in details



The initial project phase

- Step 1: Setup a Team
- Step 2: Create a Git account
- Step 3: Define a project (more next week)

Step 1: Setup a Team

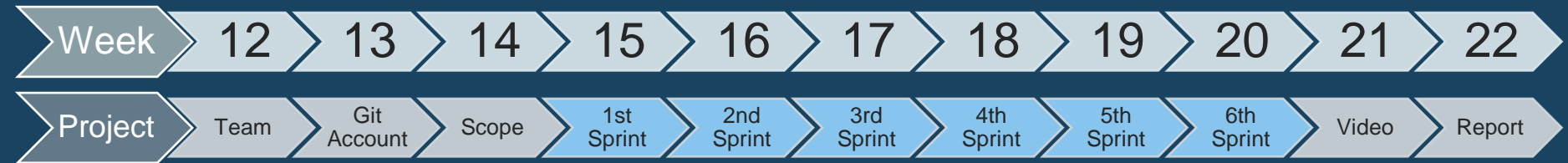
- Team shall be 6-8 students
- Two options:
 - Sign-up with friends / follow students in the same Project Team OR
 - Sign-up in the group *“I_want_to_be_auto_assigned”*

The screenshot shows the 'Groups' page for the course 'DAT257'. The sidebar on the left contains navigation links: Account, Dashboard, Courses (highlighted), Calendar, Inbox, History, and Help. The main content area has a breadcrumb 'DAT257 > People > Groups' and tabs for 'Everyone' and 'Groups'. A search bar labeled 'Search groups or people' is located below the tabs. The groups list includes:

Group Name	Students	Lock Icon
Exercise: Friday 24.03 13:15 - 16:00 Exercise Day	0 students	🔒
Exercise: Tuesday 21.03 08:45 - 11:30 Exercise Day	0 students	🔒
Exercise: Tuesday 21.03 13:15 - 16:00 Exercise Day	0 students	🔒
_I_want_to_be_auto_assigned Project Team	0 students	🔒
Project Team 1 Project Team	0 students	🔒
Project Team 2 Project Team	0 students	🔒
Project Team 3 Project Team	0 students	🔒
Project Team 4 Project Team	0 students	🔒

How to pass the project...

...without getting lost in details

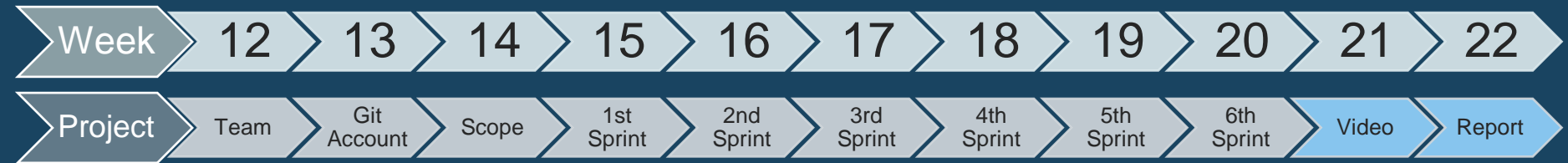


The development phase

- Step 4: for sprint $n=1 \dots 6$
 - Step 4a: Quickly develop something working
 - Step 4b: Test it
 - Step 4c: Refine the idea, add new features

How to pass the project...


...without getting lost in details



The wrap-up phase

- Step 5: Create a presentation video
- Step 6: Write a report
- Step 7: Do peer-assessment individually

Find all information on canvas!


[DAT257](#) > [Syllabus](#)

ip4 VT23

Home

Syllabus

Discussions

People

Pages

Assignments

Files

CanLa

DAT257 Agile software project management

[Jump to today](#)

Hello and Welcome to DAT257 - Agile Software Project Management.

The aim of this course is to give you an introduction into Agile Development Methodologies.

This is a practical project course, so you will apply agile methods to develop a software project in a small group.

Course Elements

The course consists of three elements:

Lectures: The lectures will provide you with a general overview and knowledge about Agile Methods for Project Management.

Exercise: We will have one exercise to practise Scrum in a practical setting.

Project: This is the core of this course: In a team of 6-8 students, you will develop a software that solves a real-world(tm) problem related to the [Sustainability Goals of the UN](#).

You will incrementally master agile software development throughout your project.

Teachers and TAs

[Here](#) you find an overview of the teaching staff for this course.

Learning Outcomes

Knowledge and understanding you should



GÖTEBORGS
UNIVERSITET



CHALMERS