IIW Praktikum
Building a secure state-of-the-art web application

Riccardo Scandariato

Institute of Software Security, TUHH, Germany

scanda***to @ tuhh.de

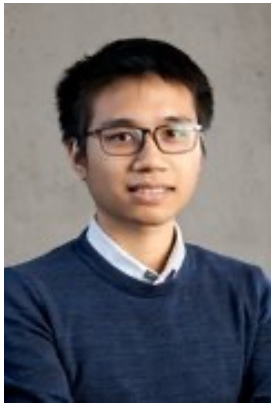
Winter Semester 2023-2024

Team members

- **Yannic** Hillers <yannic.hillers@tuhh.de>
- **Ben** Damerow <ben.damerow@tuhh.de>
- **Trang** Pham <trang.pham@tuhh.de>
- **Arsselan** Arbabzadah <arsselan.arbabzadah@tuhh.de>
- **Sarish** Raveendiran <sarish.raveendiran@tuhh.de>
- **Wael** Hourani <wael.hourani@tuhh.de>
- **Miles** Sasportas <miles.sasportas@tuhh.de>
- **Junior** Woguia <junior.noumoye.woguia@tuhh.de>

Teaching team

Riccardo Scandariato
Lecturer
scandariato@tuhh.de



Cuong Bui
Teaching assistant
cuong.bui@tuhh.de

About me

Call me Riccardo (no ~~prof~~ needed)

Head of the Institute of Software Security

Italian, 47, in HH since 2020

Roles: Teacher

- The role of the **teacher** is to ***shepherd*** you and be the ***quality gate***
  - Are we choosing the right product?
  - Do we have enough/right requirements?
  - Are we progressing as expected?
  - Is the quality where it should be?

- Different from other courses!

Roles: Teacher

- The role of the **teacher** is to solve ***team issues*** you have failed to solve internally
  - What do we do with this member not showing up for meetings?
  - What do we do with this member being bossy?
  - What do we do with this member not working enough?

Roles: Teaching assistants

- The role of the **TA** is to provide ***tech support*** and give feedback on ***quality***
  - What is the problem with this snippet of code?

- They also help with the grading
- They also help with the tutorials

- You must be pro-active in asking questions (we do not read minds, *yet*)

About you

Have you ever programmed before?

Favorite programming languages ?

What is this course about?

- Experience SW development as a team
  - **Challenges**: communication, delegation and trust, professionalism, ego
- Programming something bigger
  - **Challenge**: managing complexity, quality control with several “moving parts”

Course at a glance

- No theoretical lectures (some tutorials)
- We meet weekly for team supervision meetings
  - Show the **results** (show the backlog, show the design, show STRIDE analysis, demo the implementation…)
  - Discuss **problems** (i.e., **retrospective**)
  - Discuss the **plan** for next **sprint**
- You work happens outside of office hours

- The course results in the development of a software product (i.e., running system)

- The development is carried out in a team

- The development follows a software process

What product?

- We suggest you develop a web shop
  - E.g., Amazon clone
- Open to other suggestions
  - Instagram-clone
  - Dating site…
- Big enough to keep 8 ppl busy for 14 weeks

- Use an Javascript (Vue JS) and Python (Django) to create the system
- The system you develop will have a UIs and DBs
- The system must run (demo-able)

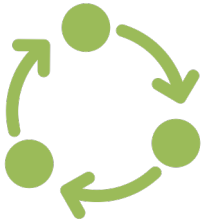
What product?

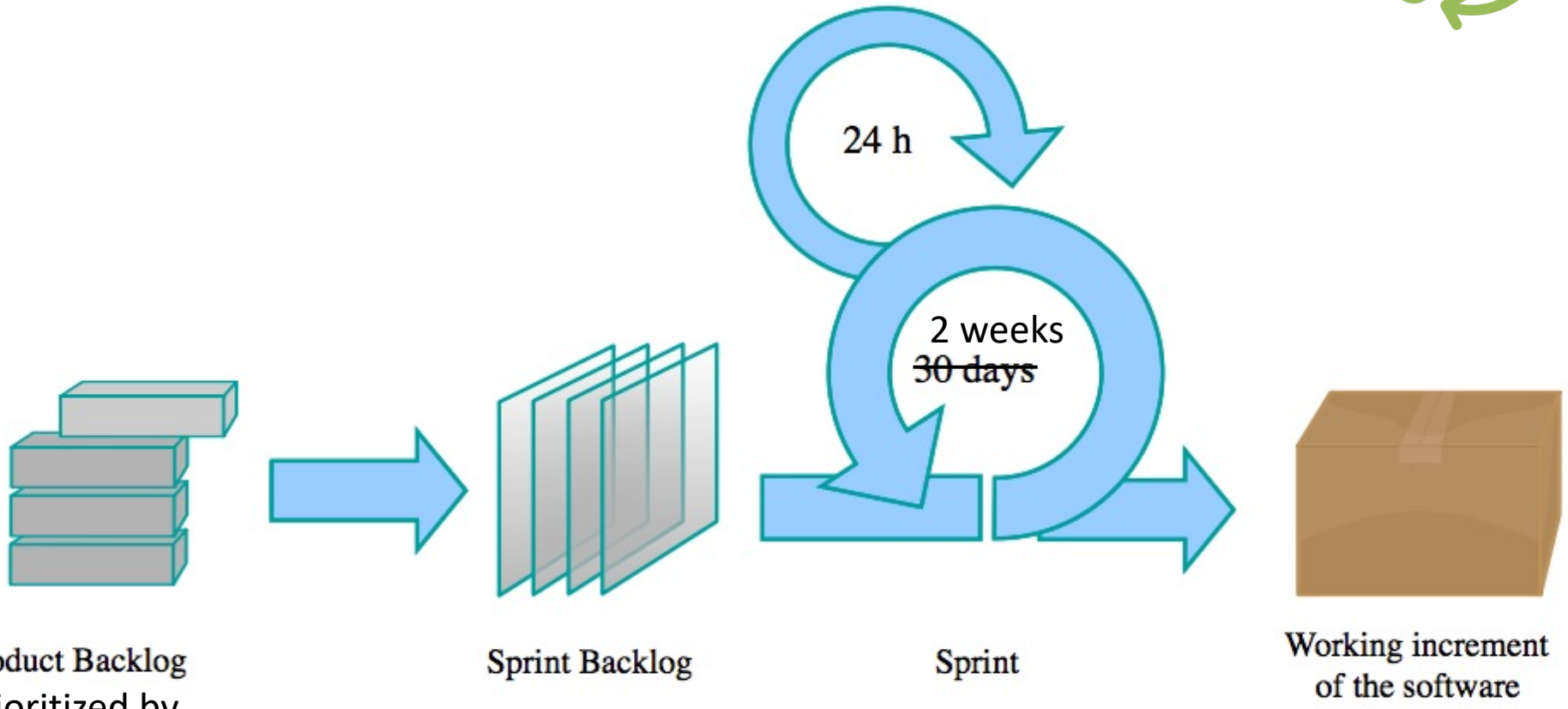
- You (as a team) define the *specific system* you want to build
  - Specify what the system is expect to do… i.e., its requirements

IMPORTANT SLIDE

- Everyone must be able to demo on their laptop all the time
- Keep it simple for everybody !
- When we meet, I want a different “demo person” every time

Agile process (à la SCRUM)

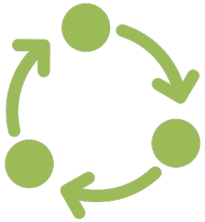




Part 1
(analysis and design)

Part 2
(implementation)

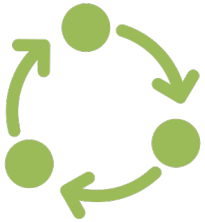
Backlog



- **Product backlog**: Prioritized list of all product requirements (stories)
  - Can be continuously updated by the Product Owner

- **Sprint backlog**: List of requirements selected for the current sprint
  - Contains more detailed information: **tasks**
  - Only updated by Scrum Team

User stories

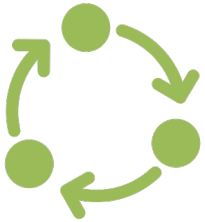


- Short, simple description of a feature told from the perspective of the person who desires the new capability
  - Usually a customer of the system
  - Avoid *developer* stories (tasks?)

Template

- As a *[role]*, I want to *[do something]* so that *[reason/benefit]*

User stories

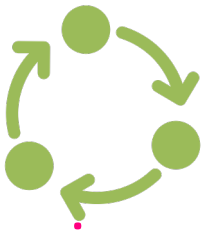


- Short, simple description of a feature told from the perspective of the person who desires the new capability
  - Usually a customer of the system
  - Avoid *developer* stories (tasks?)

Example

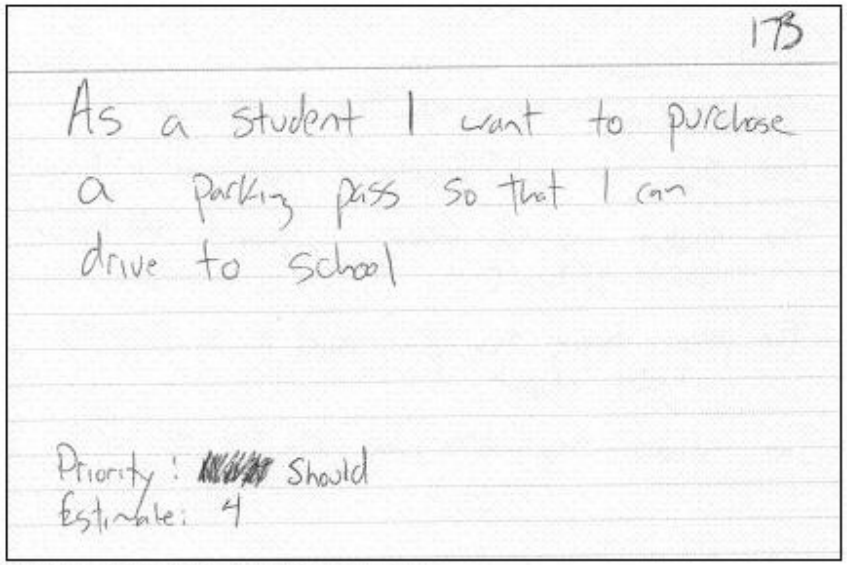
- As a user, I can indicate folders not to backup so that my backup drive isn't filled up with things I don't need saved

User Story Cards

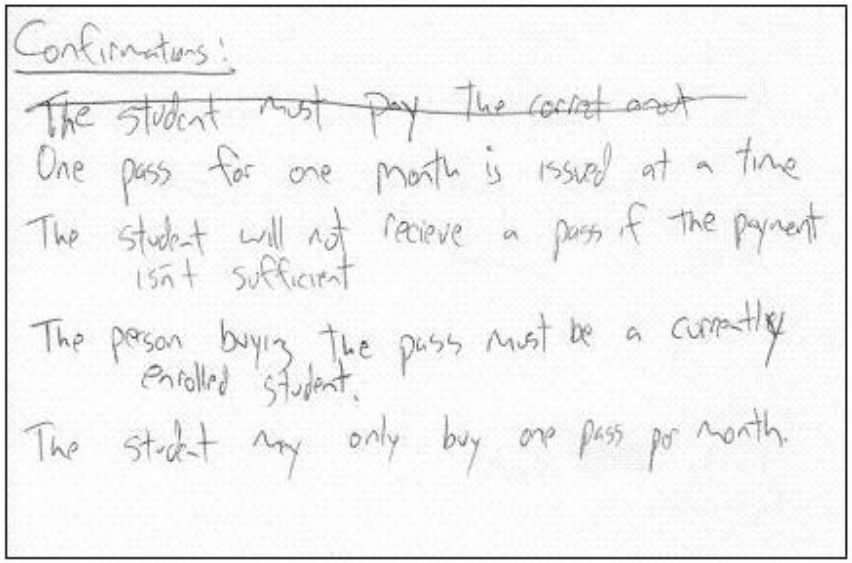


- Describes the requirements and the acceptance criteria
- Can also hold information about the the priority (from Product Owner) and effort estimate (from Scrum Team)

Front of Card

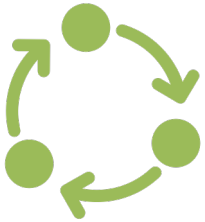


Back of Card



http://www.agilemodeling.com/artifacts/userStory.htm

User Story – More material

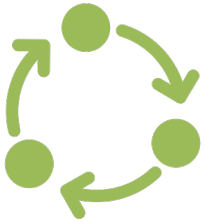


- Some do and don't about user stories in a specific presentation
- Also including info on security stories

Not shown today, but you should read the slides

Tasks

- User stories are usually “too big” to tackle as one

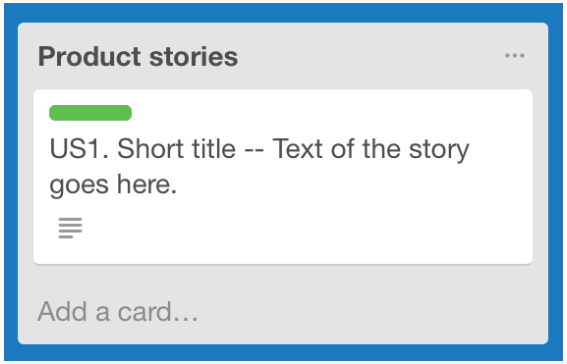


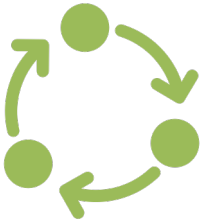
STORIES	TODO
As a student, I want to purchase a parking pass so that I can drive to school.	Coding, Test, Build, Demo
As a student, I want to purchase a parking pass so that I can drive to school.	Integrate parking pass into datamodel; Create connection to payment service; Design GUI for purchase process; Define parking pass validation scheme

https://en.wikipedia.org/wiki/File:Scrum_task_board.jpg

Trello

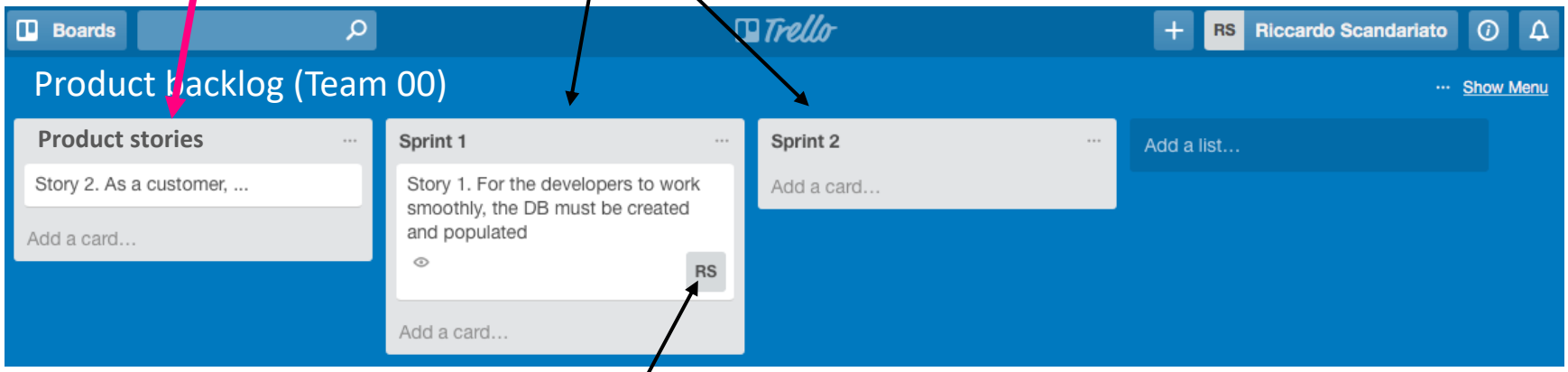
Number, title, and story





At the beginning, you populate this list

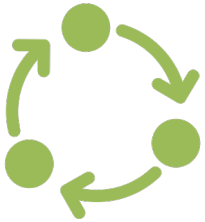
Progress



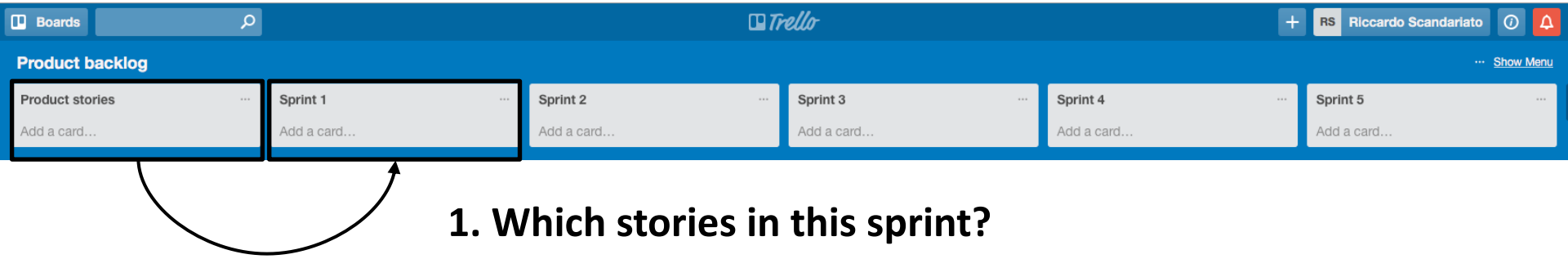
Clear responsibilities

Also a board for the current sprints (sprint backlog, TODO tasks, done tasks)

Sprints
Planning

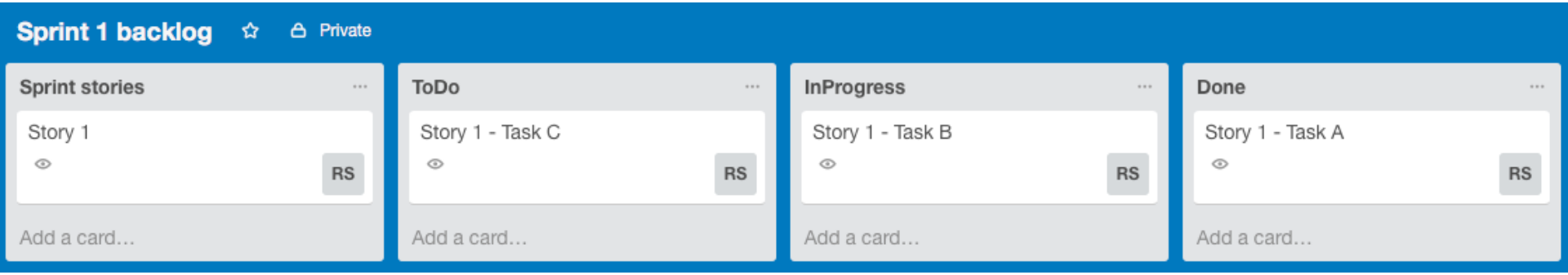


Product board (Trello)



1. Which stories in this sprint?

Sprint board (Trello)



2. Stories detailed into tasks

3. Who does what

IMPORTANT SLIDE

- Clear, agreed-upon responsibilities is the key
  - Visible in Trello
- Meet to distribute tasks
- YES: Give tasks to a single person as much as possible (e.g., ownership of features)
  - NO: 2-3 ppl working on 1 task, all the time

IMPORTANT SLIDE

- NO: “here is the pile of work, take what you can”
  - YES: Divide the work beforehand, with clear responsibilities
  - YES: Divide the work in a participatory way
  - NO: avoid dictatorships (“I tell you what to do”)

Focus on security as a key quality

- Perform a threat analysis (STRIDE) early-on (as soon as the initial architecture has emerged)
- Derive security mechanisms you have to use (security requirements)
- Use static application security testing (SAST) to avoid security vulnerabilities
  - Your code → Bandit
  - Docker image → KICS

Security requirements

- Derived from STRIDE analysis (threats)

- About security requirements as user stories
  - SAFECode guide (focus on Section 2a)
  - Source of inspiration (on git!)

We have a list of security requirements for you, but you should give it a try first ;)

Tutorials

- Vue JS (YouTube video)
- Django (YouTube video)
- Trello (online article)

- Docker (attend lab – one of the options)
  - 26.10.2023 – 09h45 (room D - 1.025)
  - ~~26.10.2023 – 11h30 (room D - 0.010)~~

- STRIDE (own video + book chapters + Q&A)
- Bandit to check your code (in-person tutorial)
- KICS to check Dockerfile ~~and Snyk to check Dockerimage (in person tutorial)~~

Team communication

- We use Mattermost (it’s like Slack…)
- TA is on it

IMPORTANT SLIDE

- YES: Agree on a set of fixed meeting every week to discuss
  - Suggestion is 2 meetings per week
  - Meet in person

No messages on Stud.IP !!!

- Send an **email** !!!

1. Say that this mail is about IIW Praktikum (in subject)
2. I need to see who you are (name is clear from sender or signature)
3. Always add Cuong in CC

Git

- Team contract (see later)
- Code
- Diagrams
- Meeting minutes
- Tutorial materials
- …

Questions ?

Sprint 1 (1 week) – Concept

- Write a contract – Agree on a way of working
  - When / where to meets (at least a couple)
  - Conflict resolution (e.g., ppl not delivering, discording opinions: vote on issues?)

- Develop the concept for your product (e.g., web shop)
  - Start writing the (functional) stories on the backlog (Trello)

- Next supervision meeting: we discuss/agree upon the concept, as well as its size and scope

S2 (2w) – Stories and Secure Design

- Finish the backlog (user stories)

- Create an architectural sketch
  - Components of your system
  - Storage strategies (what/where is the data …)
  - Interactions among components and service interfaces
  - Deployment (e.g, Docker images)

- Perform threat analysis (STRIDE)
  - Define the security controls (security stories)

S3 S4 S5 S6 (2w each) – Implementation

- Select the stories for the sprint
- Define and assign the tasks
- Implement the stories
- Run security tools (**Bandit, KICS, Snyk**)
- Produce a demoable version (**Dockerized**)

- Supervision meeting: Show the demo, discuss progress, discuss plan, etc

Sprint 7 (3 weeks)

- Wrap-up
- Prepare individual report
- Prepare group presentation/demo

- Presentation for the other teams in the IIW Praktikum

Individual performance

- At the end: Short individual report
  1. Part 1: your contribution, design choices, challenges, things that could be revised or added (5 pages)
  2. Appendix: Contribution tables (one per sprint)
     - Tasks you are responsible for (copy from backlog)
     - Delivered? If not, why?
     - Integrated in the demo? If not, why?
- Label the code that you write (as comments on classes, methods)

IMPORTANT SLIDE

- Covering up for “freeloaders” is unethical
  (cf. passing the solution of an exam to someone)

- Don’t do it
  - You will get overloaded, frustrated, etc.

- Talk to me early on
  - Sorry, I cannot fix things 2 weeks before final delivery

IMPORTANT SLIDE

- Same story for character/personal issues
  - Member is bossy
  - Member is disrespectful…
- Talk to me as early as possible
- Avoid escalation and team falling apart!

Questions ?

To be scheduled

- Meeting W44 (end of S1) – Nov 2, 11-12h
- Meeting W45 (mid of S2) – Nov 7, 8h30
- Meeting in W46 (end of S2) – Nov 14, 8h30

- Meeting in W47 (tutorial on security tools) – Nov 20, 10h

- Meetings at end of each sprint – To be scheduled

TODOs

- With TA
  - Set-up GitLab (uni account)
  - Set-up MatterMost (uni account)
  - Set-up Trello (need free account)
- You
  - Attend Docker lab 26.10.2023
  - Write team contract
  - Decide on product
  - Start writing backlog of stories