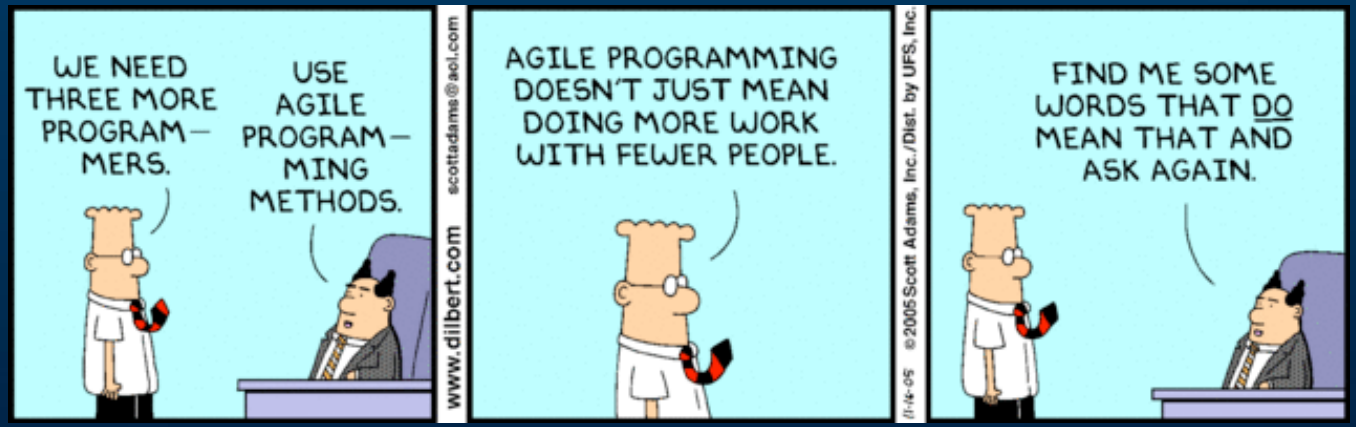
D0020E PROJECT IN COMPUTER SCIENCE 2023/2024 LECTURE 2: SCRUM



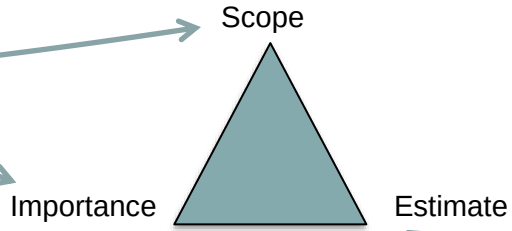
Ulf Bodin

LTU 31.10.2023

LULEÅ UNIVERSITY OF TECHNOLOGY

COMMUNICATING WITH THE PRODUCT OWNER

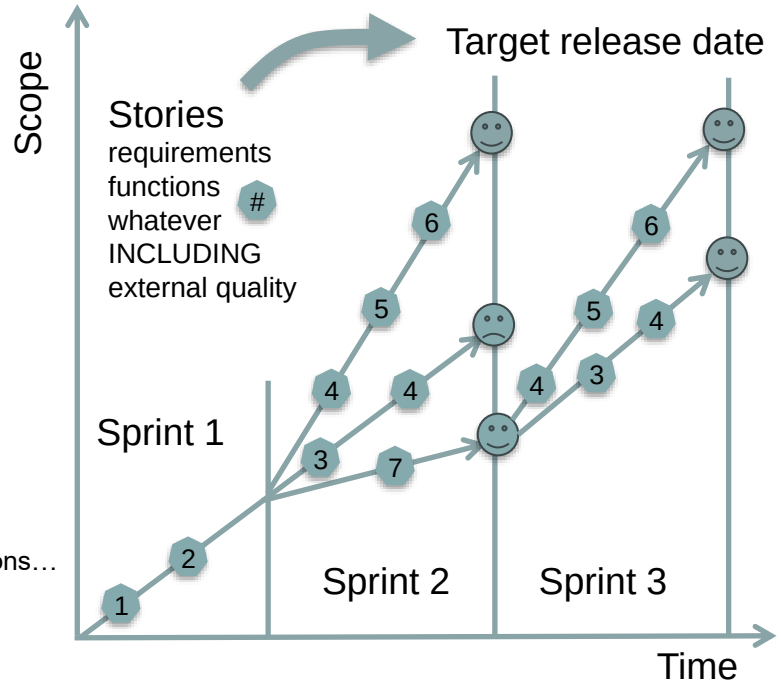
- The product owner
  - Knows customer needs (hopefully)
  - Want to make good profit for the company (and get credit for that)
- The development team
  - Needs to give clear answers to questions about cost (development and testing time)
  - Answers depend on what is to be done and at what quality (external and internal)



WHAT STORIES (REQUIREMENTS) TO SELECT

Project variables

- Calendar time (FIXED)
  - Time-boxed
- Resources (FIXED)
  - Static teams
- Quality (variable)
  - External quality, negotiable
  - Internal quality, be careful!
    - Impact time estimates
    - Trade scope against functionality and not against quality (unless you have to)
- Scope (VARIABLE)
  - Stories, requirements, functions…
  - With defined external quality

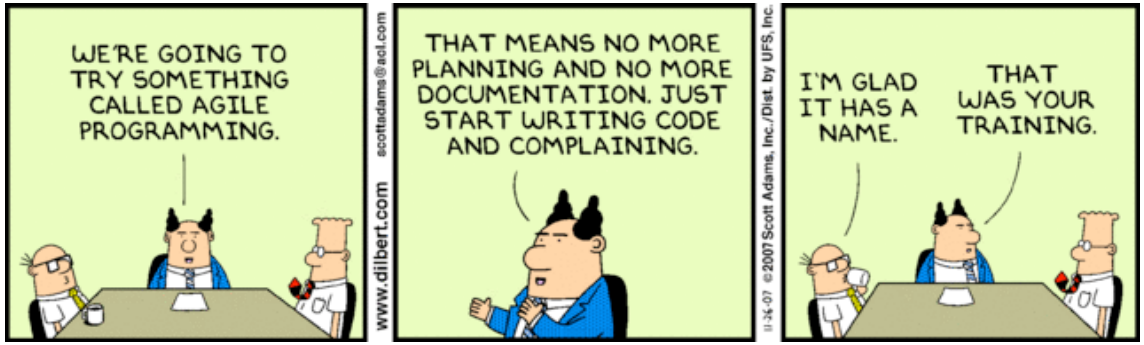


Key concepts

- Time-boxed incremental delivery
- Self-organized teams
- Prioritized product backlog
- Inspect and adapt

LULEÅ UNIVERSITY OF TECHNOLOGY

SPRINTS – AGILE BUT TIME-BOXED DEVELOPMENT



Agile does NOT mean unstructured

- Sprint goal, or theme that frame the current focus, context, etc.
  - What the product owner expect to have delivered – **main reason** for doing the sprint
  - Having the product owner stating/suggesting and/or **agreeing to a goal** is important to make sure that he/she and the team are on the same page
  - Helps the team to **understand the customer needs** (which may otherwise be hidden in stories)
  - As any other goal, it **should be measurable** (possible to determine whether or not it is reached)

Why!

- Stories must facilitate test-driven development
  - Description must come together with **defined acceptance test**
  - It must be **possible to check** whether or not a requirement (story) is met

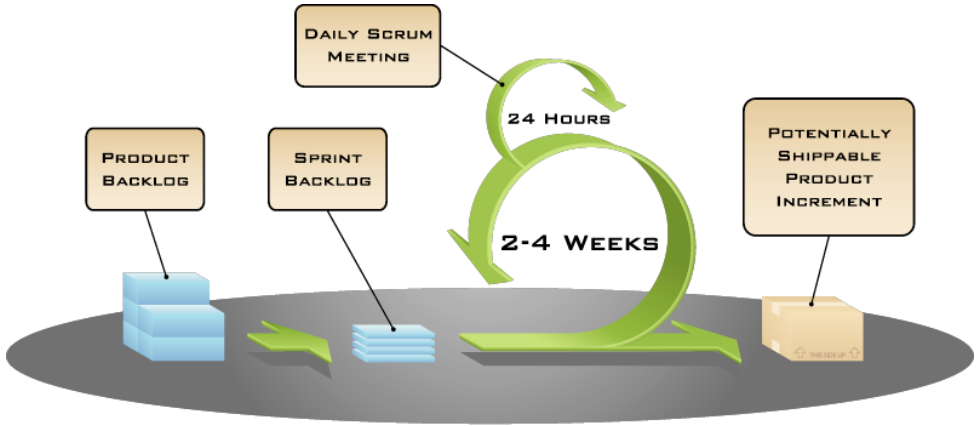
What!

SPRINTS – AGILE BUT TIME-BOXED DEVELOPMENT

- The Scrum Agile software development process
  - **Sprint planning** is done together with the product owner
  - The team decides **how many stories** from the product backlog to include in a sprint backlog
  - The product owner affects **which stories** to include through prioritization, or in dialogue with the team, split or redefined stories (larger or smaller scope)
  - At the end of each sprint, make a **sprint retrospective**

How much work!

Which work!



ESTIMATING TIME

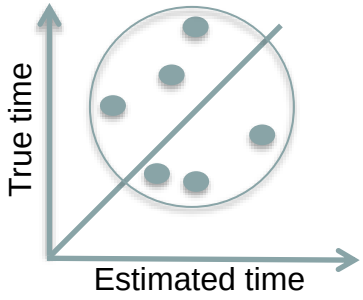
Time is hard to estimate, some advices…

- **Collect experience!**
  - Make sure following up on estimates
    - **Expected** versus **used** time
  - Task X of size S1 should take T1 hours, while task Y of size S2 should take T2 hours
  - Task Z is **larger than** X but **smaller than** Y, and should hence take between T1 and T2 hours
- Good estimates **requires knowledge**
  - Tasks requiring **new knowledge** will be estimated with **lower certainty**
  - Be open about uncertain estimates, ask others to estimate and compare
    - Discuss estimates that differs largely
- **Task size** - smaller tasks are easier to estimate
  - Not **small-enough tasks** usually means that the solution is not properly defined (or known)
    - E.g. tasks estimated to more than 2 days may not be understood properly
  - However, do not break everything down to tiny tasks
    - E.g. nothing less than 1 hours, preferably longer
- **Collaboration** - avoid having many team members involved in the same task
  - Tend to take longer time than expected, i.e., estimates are generally too low

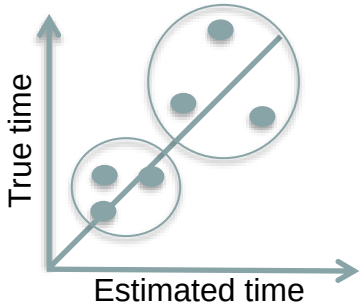
ESTIMATING TIME (CONT.) AND TEAM CAPACITY

- Time is hard to estimate, some advices…
  - So, collect **experience**, beware of how much is known (**uncertainty**) and strive for **reasonable sized tasks** for **one or few team members to execute**

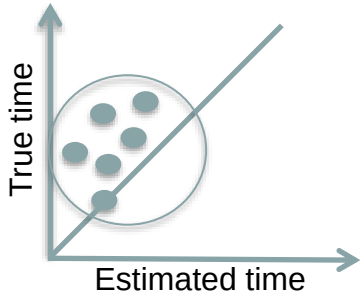
Uncertainty



Small vs. large tasks



Several involved



- **Velocity**: the availability of resources and their efficiency
  - Make sure to plan when to work and agree on efficiency, e.g., 70% of time spent
  - Still just an estimation, because unplanned things may happen, and efficiency vary
- **Capacity**: velocity integrated over time, must be > total estimated time (cost)
  - Total time estimated for all tasks for all stories in a sprint

ROLES IN THE SCRUM DEVELOPMENT PROCESS

- Product owner
  - Responsible of the **vision, and the value** of the product
  - Define (and/or decide) **functions and properties** of the product
    - Prioritize on their value (and/or agree on prioritizations)
  - Adapts **functions, properties and priorities for each iteration** (and/or agree on adaptations)
    - Often based on gradually improved understanding of customer needs and product potential

- Scrum master
  - Responsible of keeping to Scrum concepts (**keep track of the project**), facilitates meetings
  - Protects the team from contra productive impact from external people
- Team member (whereof one is also responsible of code repository and builds)
  - Responsible for **own progress and plan** (and CRM also keep track of versions etc.)

MEETINGS IN THE SCRUM DEVELOPMENT PROCESS

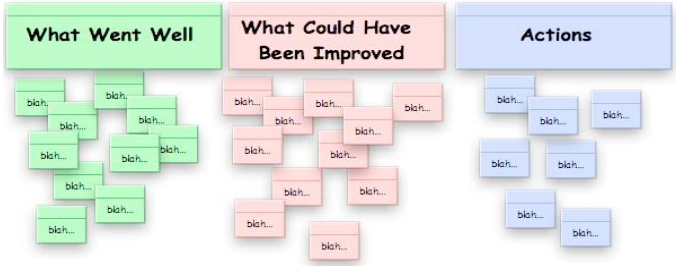
- Sprint planning – product owner, scrum master, team members
  - Decide on **theme** for the coming Sprint (important)
  - Select **stories** from product backlog for the sprint backlog
  - Agree on high level **design**

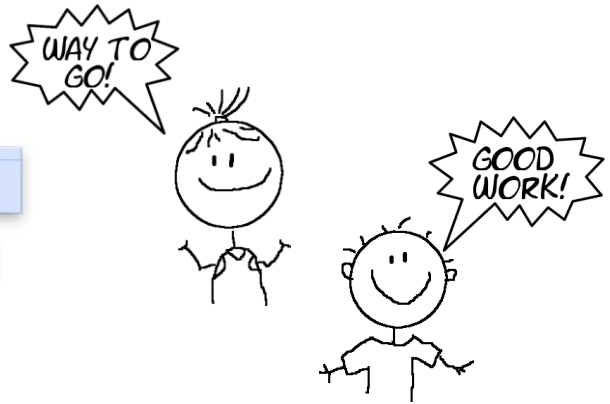
- Daly scrum – team and product owner (not required)
  - Short meetings, e.g., 15 minutes
  - Look at the **task board**!
  - Look at **task burn-down**
  - Each team member
    - What has been **done**, **problems**, **obstacles**, **help needed**, what **comes next**
  - A few bullet items can come handy to make sure the most essential things are efficiently covered

Scrum: 15 minute daily meeting.
Teams member respond to basics:
1) What did you do since last Scrum Meeting?
2) Do you have any obstacles?
3) What will you do before next meeting?

MEETINGS IN THE SCRUM DEVELOPMENT PROCESS

- Sprint review – everyone interested
  - Limited time, e.g., max 2 hours
  - Show **what has been achieved**, e.g., though demonstrating
  - No presentation, **look at what is delivered and discuss (to learn for next iteration)**
- Sprint retrospective – same as sprint planning, maybe also the customer
  - What worked out well, look at tracked issues (unexpected and unplanned work), problems
  - Team climate, is it supportive, improvements
  - Try to give positive feedback to each other
    - We need to improve, you did great





LULEÅ UNIVERSITY OF TECHNOLOGY

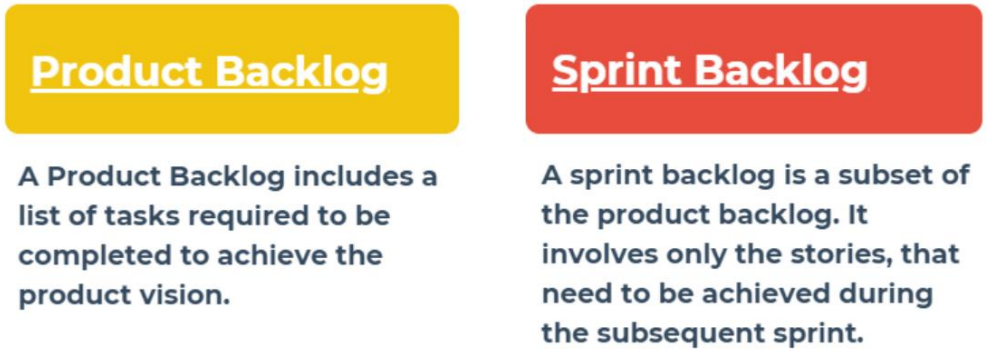
SCRUM PRODUCT/SPRINT BACKLOG

- Define stories together with acceptance test
- Test-driven development
  - Make it possible to check whether or not a requirement (story) is met
  - Makes it easier to define tasks knowing how things will be tested (or demoed)

Story ID	Story	Task ID	Task	Tidsuppskattning (story)	Använd tid (story)	Tidsuppskattning (task)	Använd tid (task)	Beroenden	Risk	Prioritet
1	***Titel***: <bra namn> ***Avsedd användning***: <t.ex. i vilket sammanhang> ***Önskade egenskaper***: <t.ex. justerbara skala på graf> ***Testfall***: <tydlig beskrivning>	1.1	***Att göra***: <A> ***Risk***: <a>	Tid = AA+ BB+ CC	TBD	AA=4	TBD	N/A	Tex. risk = floor (a+b+c/3)	Tex. hög/ medium/ låg
		1.2	***Att göra***: <B> ***Risk***: <b>			BB=12	TBD	N/A		
		1.3	***Att göra***: <C> ***Risk***: <c>			CC=8	TBD	1.1		

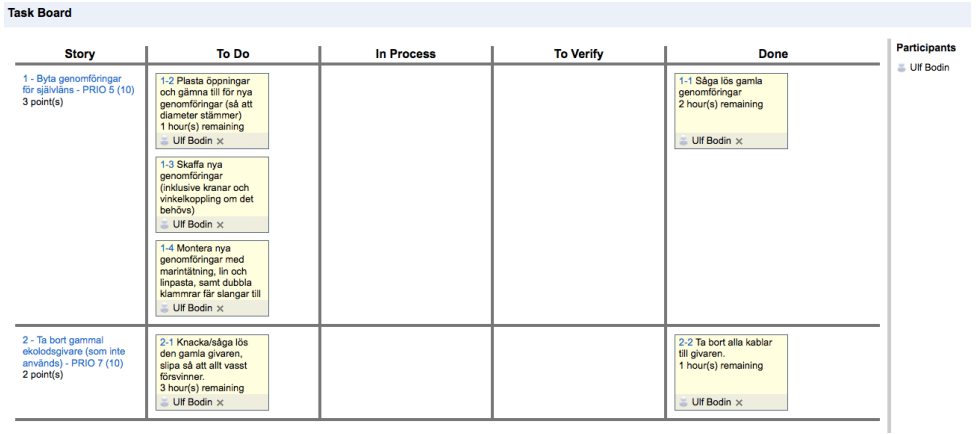
SCRUM PRODUCT/SPRINT BACKLOG, IMPORTANT

- **Track priority**, e.g., in the name field of each story
  - Some tools support this, but just need to be noted so that is fine in case not directly supported
- Select stories from **product backlog** for **sprint backlog**
  - Make sure cost does not exceed capacity, i.e. velocity integrated with agreed time for the sprint



SCRUM PRODUCT/SPRINT BACKLOG, TASK BOARD (CONT.)

- Keep track of the **task board**
  - E.g., look at it at daily scrum
  - Watch for **deviations**
    - Some tasks are done, but nothing in progress, team is idle
    - A lot in progress, nothing done, team have problems and have got stuck
    - Nothing is checked out at all, what is the team doing?!



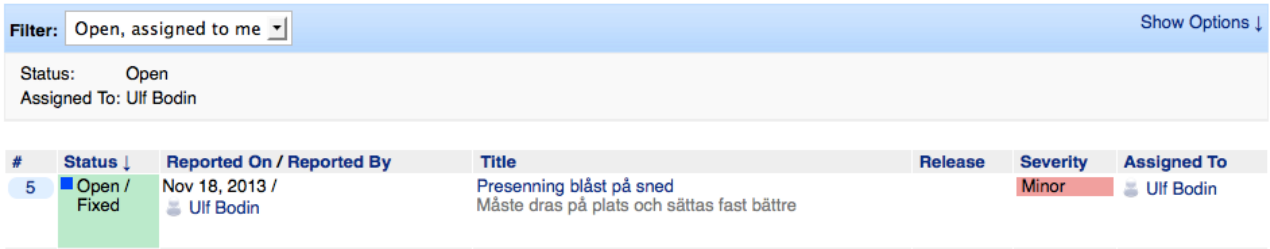
Task Board

Story	To Do	In Process	To Verify	Done
1 - Byta genomföringar för självläns - PRIO 5 (10) 3 point(s)	1-2 Plasta öppningar och gämna till för nya genomföringar (så att diameter stämmer) 1 hour(s) remaining Ulf Bodin 1-3 Skaffa nya genomföringar (inklusive kranar och vinkelkoppling om det behövs) Ulf Bodin 1-4 Montera nya genomföringar med marintätning, lin och linpasta, samt dubbla klammrar får slangar till Ulf Bodin			1-1 Såga lös gamla genomföringar 2 hour(s) remaining Ulf Bodin
2 - Ta bort gammal ekolodsgivare (som inte används) - PRIO 7 (10) 2 point(s)	2-1 Knacka/såga lös den gamla givaren, slipa så att allt vasst försvinner. 3 hour(s) remaining Ulf Bodin			2-2 Ta bort alla kablar till givaren. 1 hour(s) remaining Ulf Bodin

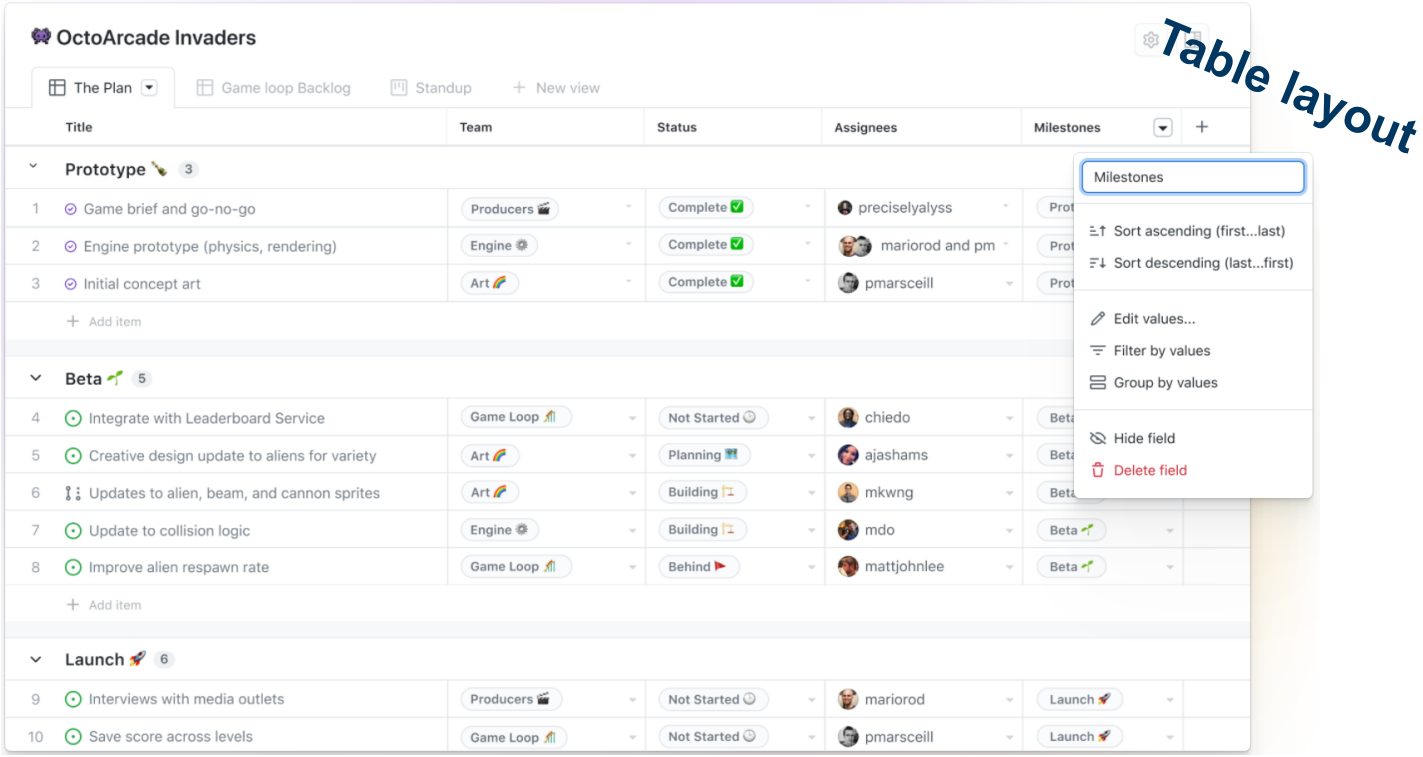
Participants
Ulf Bodin

SCRUM PRODUCT/SPRINT BACKLOG, TASK BOARD (CONT.)

- Keep track of the task board
  - Nothing is checked out at all, what is the team doing?!
  - Are all time spent on **issues/bugs**…
    - Make sure reporting and keeping track of issues/bugs
    - Many issues/bugs is a clear warning that may need to be discussed with product owner
  - Bad (neglected) internal quality is likely to eventually cause bugs
  - Design mistakes may cause unplanned work, i.e., issues
  - Testing usually reveals both new and old bugs

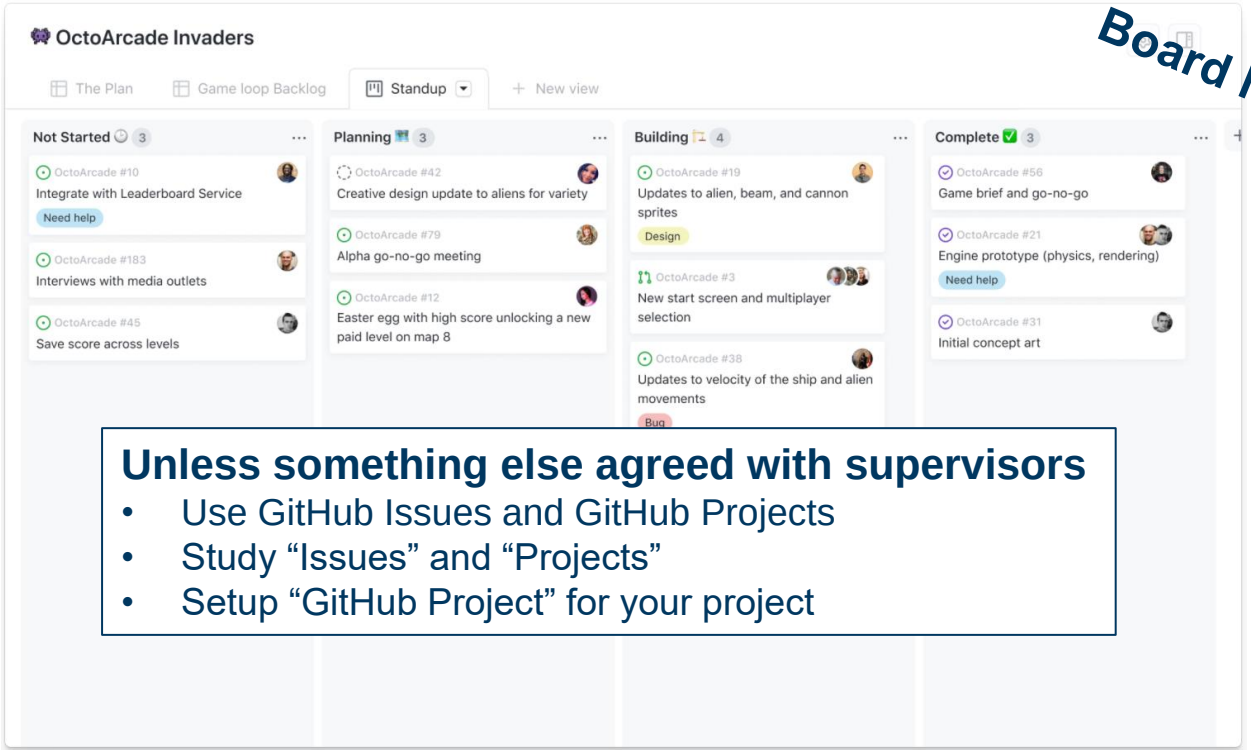


GITHUB ISSUES/PROJECTS, BACKLOG AND BUGS



LULEÅ UNIVERSITY OF TECHNOLOGY

GITHUB ISSUES/PROJECTS, TASK BOARD



Unless something else agreed with supervisors

- Use GitHub Issues and GitHub Projects
- Study “Issues” and “Projects”
- Setup “GitHub Project” for your project

LULEÅ UNIVERSITY OF TECHNOLOGY

LULEÅ
UNIVERSITY
OF TECHNOLOGY