



COMP30022

IT Capstone

Project

Semester 2, 2025

Lecture 2

Intro to Agile and Tools

Project Announcements

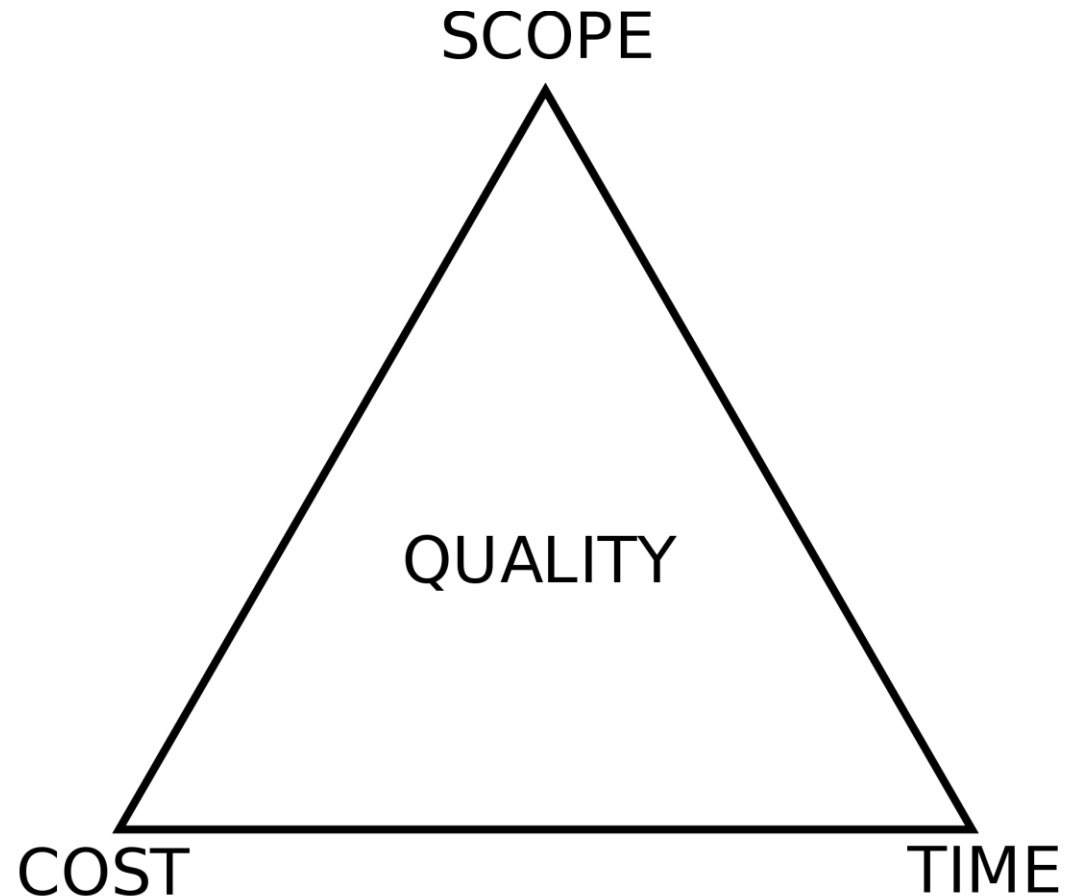
- Team formation progressing:
- Around 85% of students currently signed up to a project
- Exceptionally high demand for some workshops, and continued increase in student numbers during week 1
- Adding an additional workshop (9AM Tuesday, also focused on Forensic Tool for detecting DeepFake & Synthetic Media)
- Will be announcing a preferences form on Canvas for those not yet able to join a workshop
- **Thank you for your patience**

Overview of Lecture

- Brief overview on Agile development
- Important Ceremonies
- Intro to Tools

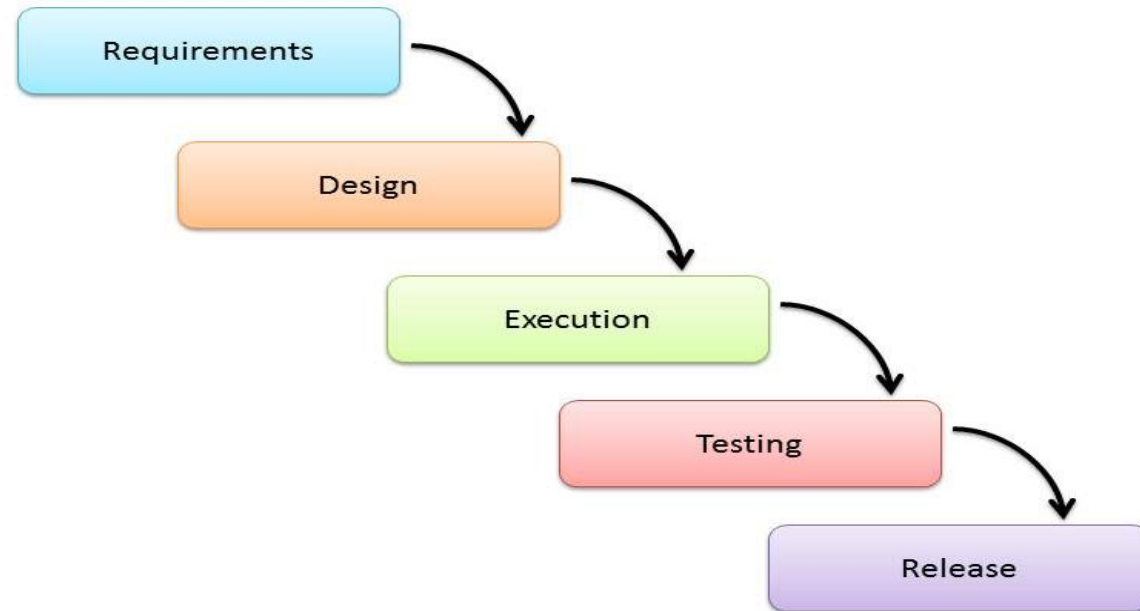
What is a project?

Temporary endeavor to create a unique product, service or result



Approaches to software development

Waterfall model –
Dates back to 1970's



Agile methods evolved around 2000s
to address project issues, change emphasis, and make less document heavy

Agile in COMP30022

Appreciate that (most of) you have little background

- Project Inception
 - Sprints
 - Consideration of roles, e.g. scrum master, product owner
 - Stand ups (weekly, starting next work)
 - Retrospectives (end of inception and each sprint)
 - Scope considered dynamically each sprint
-
- Flexible and Lightweight

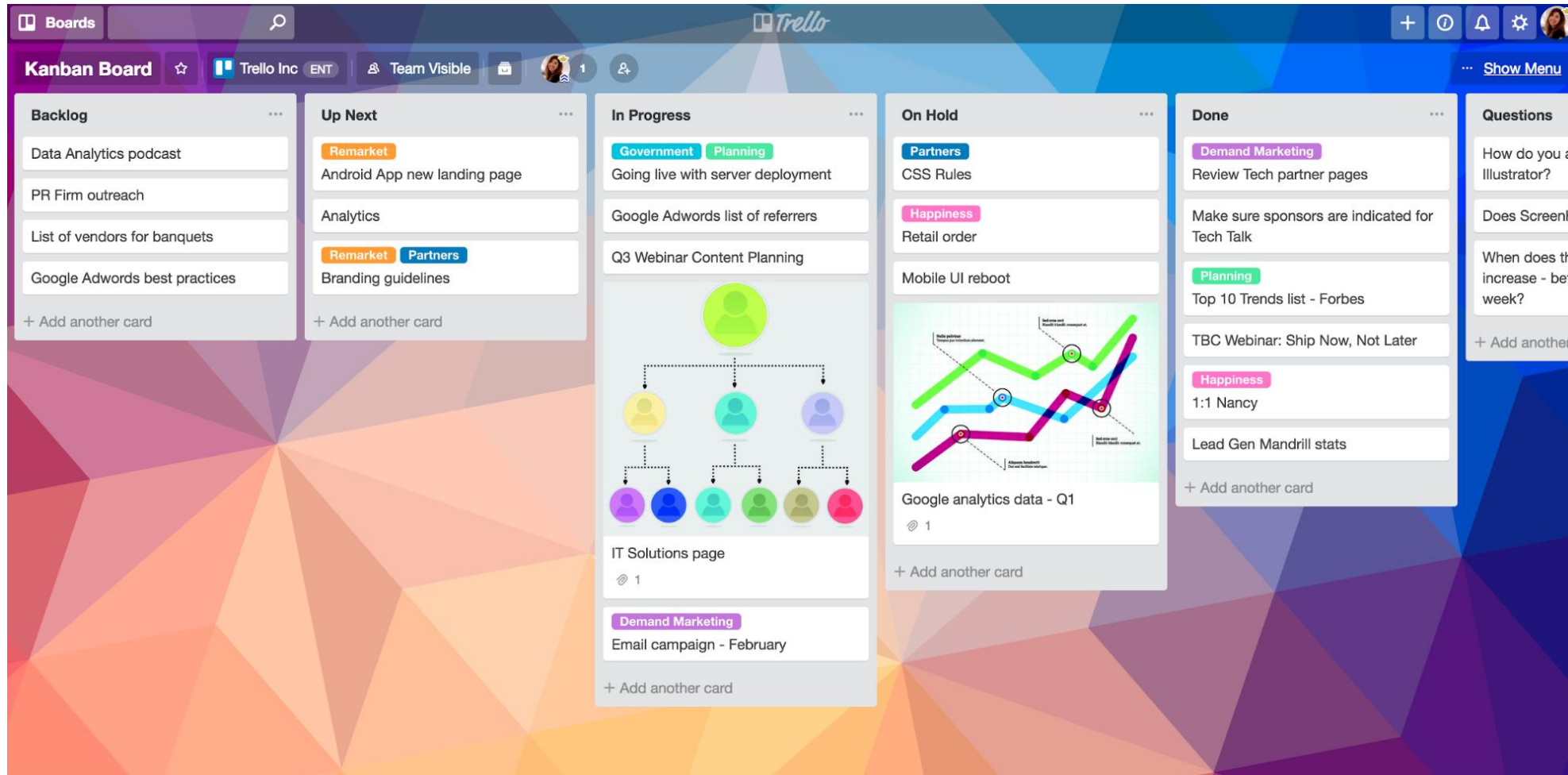
Roles to consider

- **Scrum Master:** Provides leadership & guidance, ensures team works together, facilitates discussion, promotes agile process.
- **Product Owner:** Deep knowledge of business, priorities backlog, communicates with stakeholders.
- **Developers:** Build the product, may include other sub-roles such as designers, testers, QA.
- **Stakeholders:** Those with an interest in the project, e.g. your client.

Agile Methods

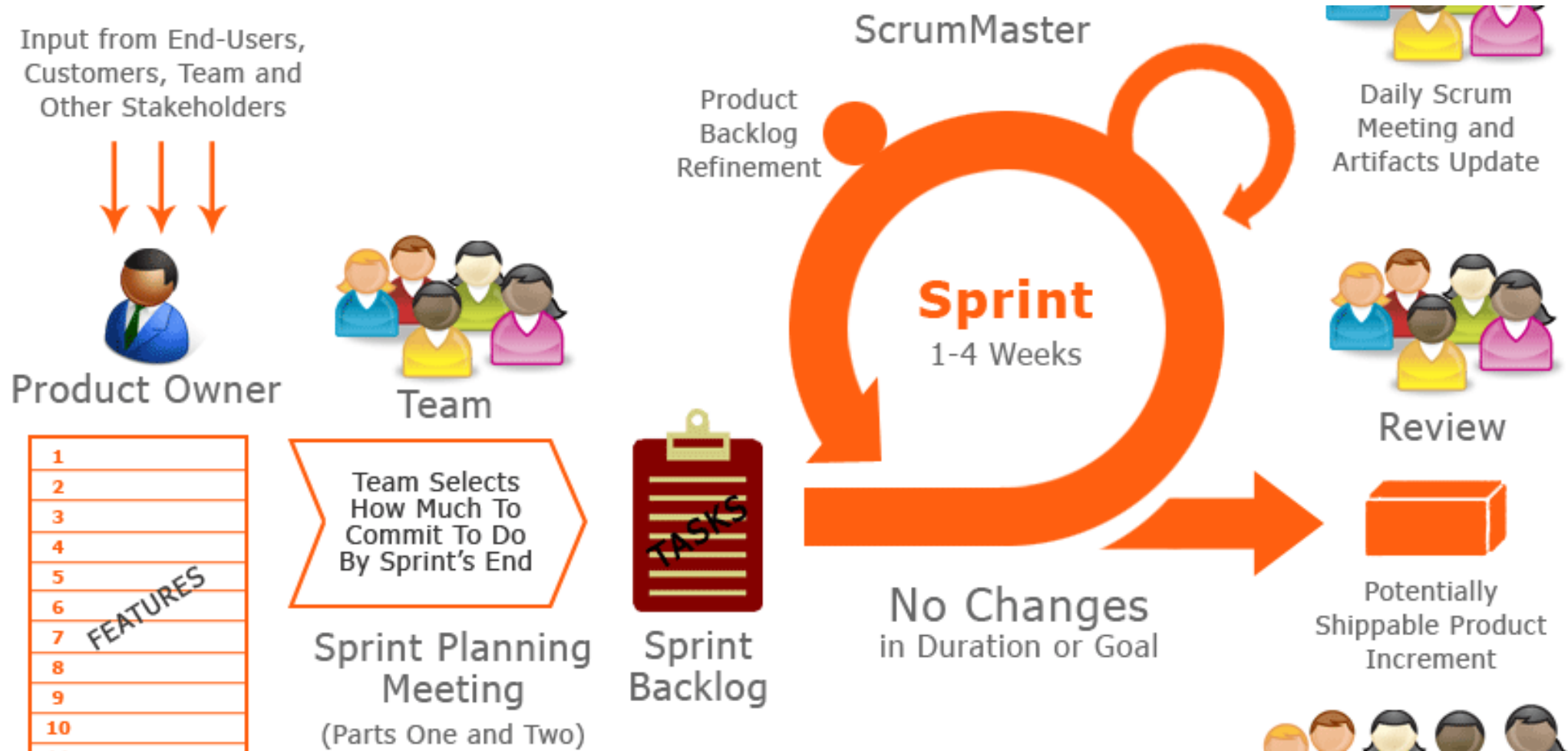
- Many methods
 - Extreme Programming (XP), Kanban Boards, Scrum
- Used broadly in society
- Outcome of subject – practical, not theoretical
 - Revisited over semester
- To be demonstrated in workshops

Kanban Boards



Scrum

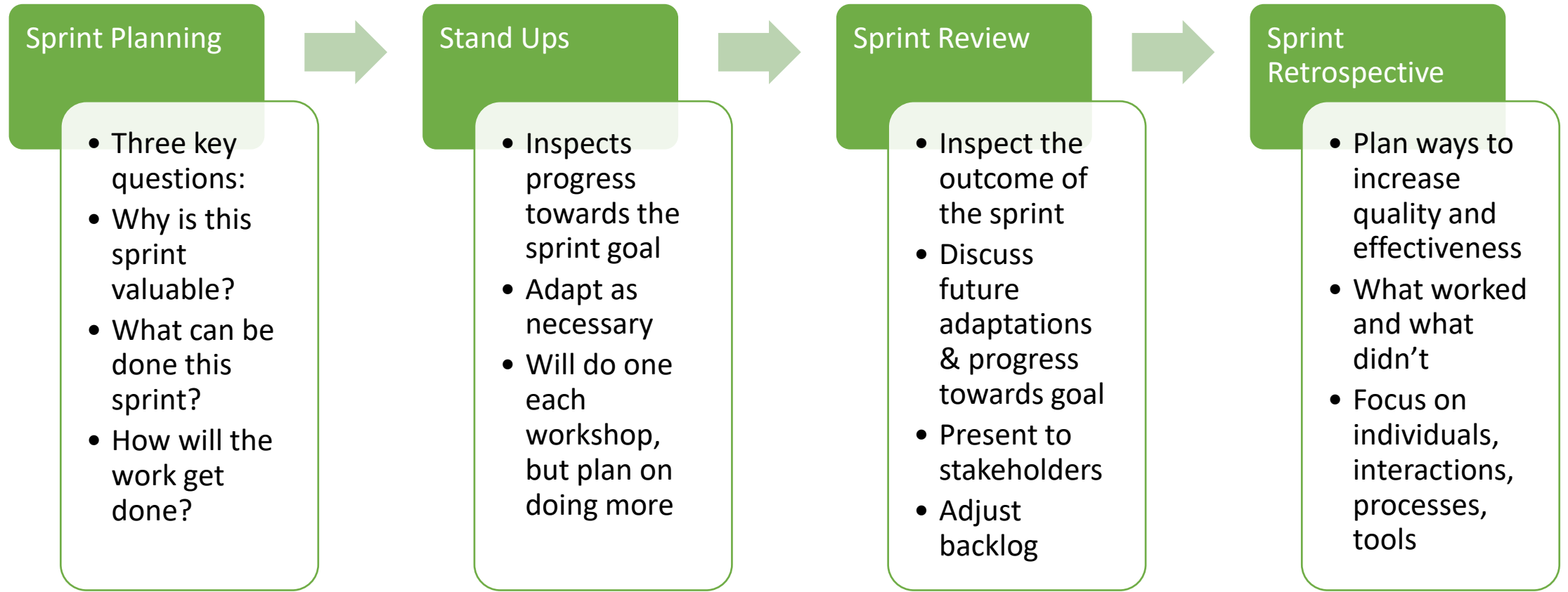
cvcedhlab.hypotheses.org



Agile ceremonies

- Stand ups – doing them
- Sprints
 - Planning
 - Estimation
 - Sprint retrospectives
 - Sprint review
- Methods
 - Scrum
 - Kanban
- Working code

Agile Ceremonies



Stand Ups

- Keep everyone on track
- Stop blockers
- Enhance communication
- Practice in workshops
- Everyone should speak – leave an impression in what you do
- Key questions
 - What did I do since the last meeting?
 - What will I do before the next meeting?
 - Are there any blockers?

Sprint planning

- Adjust the backlog on the basis of the previous sprint
- Decide what tasks are to be done this sprint
- Check with client (where appropriate)

- Try to ensure equal workload
- To allocate work, need estimate of size of tasks
 - Strongly suggest teams undertake estimation in their workshop
 - No guess is a bad guess

Estimating workload

- Measures
 - T-shirt sizes: S, M, L, XL
 - Fibonacci numbers: 1, 2, 3, 5, 8, 13, ...
- Tools
 - Plenty online

We don't expect accurate estimates – try and improve

Planning poker

- Gamifying estimation
- <https://www.mountangoatsoftware.com/agile/planning-poker>
- Tools
 - <https://www.scrumpoker-online.org/en/>
 - <https://planningpokeronline.com/>

Continuous improvement

- Continuous Improvement is an ongoing, long-term approach to improving processes, products and services. It is also called Continual Improvement or CI, and is one of those terms which we often think we fully understand, but can actually mean many different things to many different people.
 - From <https://blog.triaster.co.uk/blog/what-is-continuous-improvement>
- An important aspect of quality in an organisation
- Good message for your software careers
- Need to accept and respond to feedback
- Advocate reading

Sprint retrospective

- Focus on process
 - What did we achieve
 - What did we not achieve
 - What did we do well
 - What could we have done better
-
- Can be done in brainstorming style, or by people answering independently and combining answers

Suggestion: Try both and see which works better

Sprint review focus on product and often done with client

Example retrospective from SWEN90013

Goal review

Goals that we achieved in the Sprint 2:

- Modify table based tools into a format that is friendly to sensory users.
- Better process to ensure the quality → Pull request checklist is constructed.
- GraphQL connection and authentication (partial complete)
- CI/Cd setup.

Goals that we didn't achieve in the Sprint2:

- Refine product accessibility regarding all the feedback from the Sprint 1 review session.
 - WHY:
- Equipment database construction
 - WHY: We have continually communicated with the client but we haven't gotten the reply of the structure of the database yet. As NED is a huge database and we don't know how many levels of information we need to scrape.
- Search function
 - WHY: The search function requires the database. But since we are blocked by constructing the DB, we can't work on this task yet.

Retrospective

What did we do well?

- Continue the effective communication.

What should we have done better?

- A better way to keep track of our own tasks → Do it early.

Retrospective

What did we do well?

- Continue the effective communication.
- Ask and offer help when needed.
- Front end and backend team start working together and have more communications.
- Team members' understand better of their roles and duties.
- The team is performing efficient.
- complete the tasks independently.
- Would like to try sth new.
- Team cohesion
- Scrum master has put into place multiple effective measures to reduce problems faced in previous sprints.

What should we have done better?

- A better way to keep track of our own tasks → Do it early.
- Well constructed JIRA managing by Product Owner and Scrum Master.
- Engagement is dropping → what are the other commitments and how we can have better time management.
- Backend team to increase velocity (potentially more team members)
- Do it earlier.
- Remain the consistent effort
- Tasks explanation to avoid doing it redundantly.
- Pull request & code review process could be done more efficiently.
- Keep track of the Pull request comments.
- Client communication
- Notice the email.

Tools for Retrospectives

- Metro Retro
- 300+ templates in Miro
- <https://echometerapp.com/en/retrospective-tools-online/>
 - 1.Echometer: Mindset Retros
 - 2.Teamretro: Easy retros
 - 3.Easyretro: Remo. collaboration
 - 4.Metro Retro: Engaging retros
 - 5.Parabol: Agile meetings
 - 6.Retrium: Enabling conversations
 - 7.Neatro: Effective retros

Tools (many more options)

- Communication: Slack, Teams, Discord
- Teamwork: Confluence, Notion
- Project Management: Trello, JIRA
- Requirements: Motivational Modelling Tool, Draw.IO, Visio
- Architecture Design: Draw.IO, Visio
- Frontend Design: Figma, Canva, Marvel
- Repository: GitHub, BitBucket
- Linting: ESLint, JSLint, Pylint
- Testing: Selenium
- CI/CD: GitHub Actions, Jenkins
- Deployment: Local, Heroku, AWS, Google Cloud, Client Provided