Lecture 1
Introduction to ECE 422

ECE 422: Reliable and Secure Systems Design

UNIVERSITY OF ALBERTA

Instructor: An Ran Chen
Term: 2024 Winter

Schedule for today

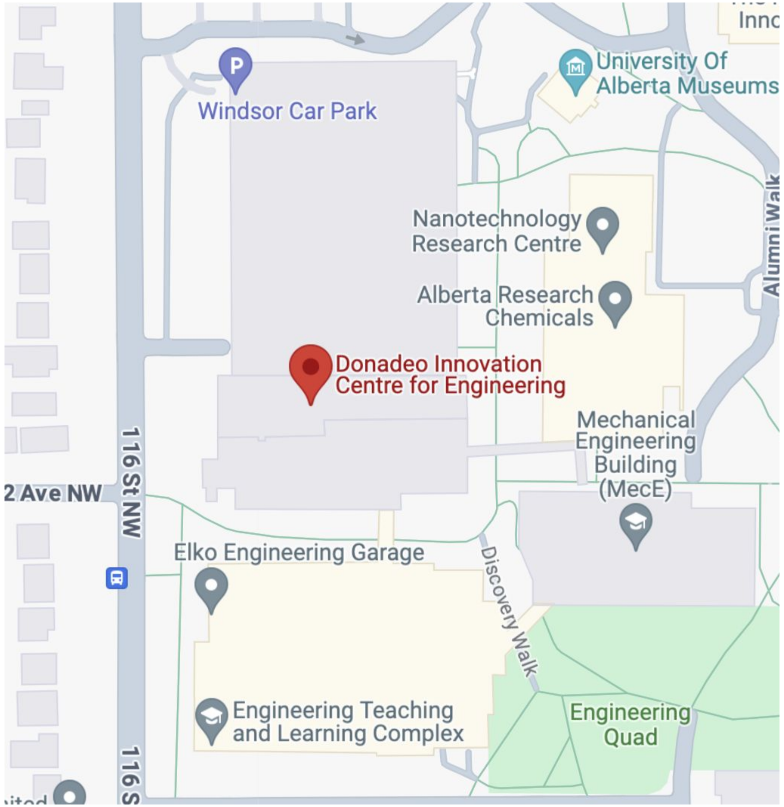
- Introduction
- Course logistics
- Agile methodology
- TODOs for upcoming classes

An Ran’s research

Software quality assurance

Mining software data

Office



Windsor Car Park
University Of Alberta Museums
Nanotechnology Research Centre
Alberta Research Chemicals
Donadeo Innovation Centre for Engineering
Mechanical Engineering Building (MecE)
Alumni Walk
116 St NW
2 Ave NW
Elko Engineering Garage
Discovery Walk
Engineering Teaching and Learning Complex
Engineering Quad

Donadeo Innovation Centre For Engineering
11-366

Email: anran6@ualberta.ca

Office hour
MWF 13:00 - 13:30 (by appointment)

Course objectives

- Determine the reliability of a system
    - Lecture slides
- Design a fault-tolerant software component
    - Project deliverables, final report
- Analyze the security of a built system
    - Lecture slides
- Design a secure software system
    - Project deliverables, final report

Course format

There are 3 main components in this course:

- Course projects (45%)
- Midterm (25%)
- Final exam (30%)

Course projects

- Done in groups of 3 people
- Programming language of your choice (e.g., Python, Java, and C++)
- Following the agile methodology, more details to be announced.

Project deliverable (3-5 pages)

- Describes design, technologies, tools, user stories, and planning

Final report (6-10 pages)

- Expands of the deliverable, based on the finished product

Demonstration (10-15 minutes)

- After the final report submission, scheduled with the TAs.

Course projects

Project 1: Reliability Auto-Scaling

An auto-scaler that actively monitors the response time and scale in or out according to the workload.

Project deliverable (5%)

- Due Wednesday, January 24

Final report and demo (15%)

- Due Wednesday, February 7

Course projects

Project 2: Secure File System

A secure file system that allows its internal users to store data on an untrusted file server.

Project deliverable (10%)

- Due Friday, March 15

Final report and demo (15%)

- Due Monday, April 8

Slack Workspace

Slack is used to help your team with the projects. Make sure to join the Slack channel, the link is provided below.

Slack link: ECE 422 Winter 2024

A starter kit, information, and tutorials on Cybera and Docker will be posted on both eClass and Slack.

Team formation deadline: Friday, Jan 12, 2024

- Send your team information on the #team channel in Slack (Find a team name and include your names)
- Setup your Cybera account

Teaching and course assistant information

No lab or tutorial is scheduled.

Ronald Unrau
Email: rcunrau@ualberta.ca

Zhijie Wang
Email: zhijie.wang@ualberta.ca

Late Policy

- Deliverable and final reports must be submitted on-time.
  - 25% per day penalty
- Deadlines:
  - Wednesday, January 24: Project 1 Deliverable
  - Wednesday, February 7: Project 1 Final Report
  - Friday, March 15: Project 2 Deliverable
  - Monday, April 8, 2024: Project 2 Final Report
- All class deadlines are due at 11:59pm

Course format

There are 3 main components in this course:

- Course Projects (45%)
    - Project 1 (20%)
    - Project 2 (25%)
    - Done in groups of 3
    - Each project is graded on the deliverable, final report and demo
- Midterm (25%)
- Final Exam (30%)

Exams

Midterm (25%)

- Monday, February 26, 2024
- In-person exam
- Covers slides from Week 1 to 6 (up to February 16)

Final exam (30%)

- ~~Friday, April 19, 2024~~ Wednesday, April 24, 2024
- In-person exam
- Covers slides from Week 1 to 14

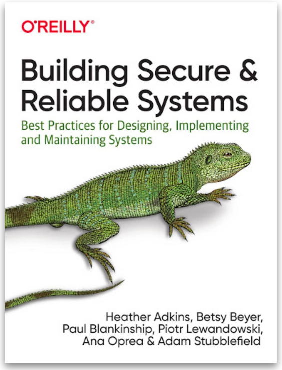
If you anticipate being unable to take the midterm for some *legitimate* reason, please inform me ahead of time. When this happens, the weight of the missed midterm will be added to the final exam (a total of *55%*).

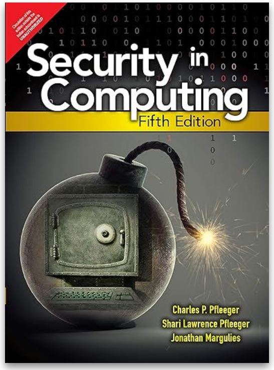
Textbooks

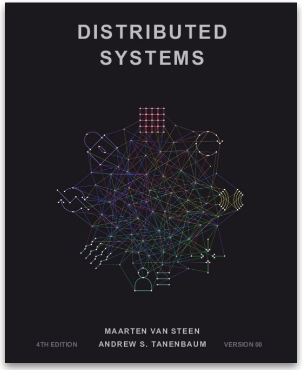
1. Marteen van Steen, Andrew S. Tanenbaum, "[Distributed Systems]", 4th edition version 01, distributed-systems.net, 2020.

2. C.P. Pfleeger, S.L. Pfleeger, J. Margulies, “[Security in Computing, 5th Ed].”, Pearson Education, 2015.

3. H. Adkins, B. Beyer, P. Blankinship, P. Lewandowski, A. Oprea, A. Stubblefield, “[Building Secure and Reliable Systems]”, O'Reilly Media, 2020.


O'REILLY
Building Secure & Reliable Systems
Best Practices for Designing, Implementing and Maintaining Systems
Heather Adkins, Betsy Beyer, Paul Blankinship, Piotr Lewandowski, Ana Oprea & Adam Stubblefield





Deadlines

Activity	Due/Scheduled	Weight
Project 1 Deliverable	Wednesday, January 24, 2024	5%
Project 1 Final Report	Wednesday, February 7, 2024	15%
Project 2 Deliverable	Friday, March 15, 2024	10%
Project 2 Final Report	Monday, April 8, 2024	15%
Midterm Exam	Tentative - Monday, February 26, 2024	25%
Final Exam	Tentative - ~~Friday, April 19, 2024~~ Wednesday, April 24, 2024	30%

Academic Honesty

Deliverable and final report: You are welcome to discuss the projects with other students, but write your deliverable and final report by yourself. Avoid plagiarize, both the deliverable and report should reflect the work of your team.

Code submission: You are allowed to consult other people's code for the structure of your project, but the main code should be written by you and your teammates alone.

Schedule for today

- Introduction
- Course logistics
- Agile methodology
- TODOs for upcoming classes

Agile methodology

Agile methodology proposes principles instead of an actual process

- Proposed by developers based on their experience
- The Agile Manifesto: https://agilemanifesto.org/

Agile methodology

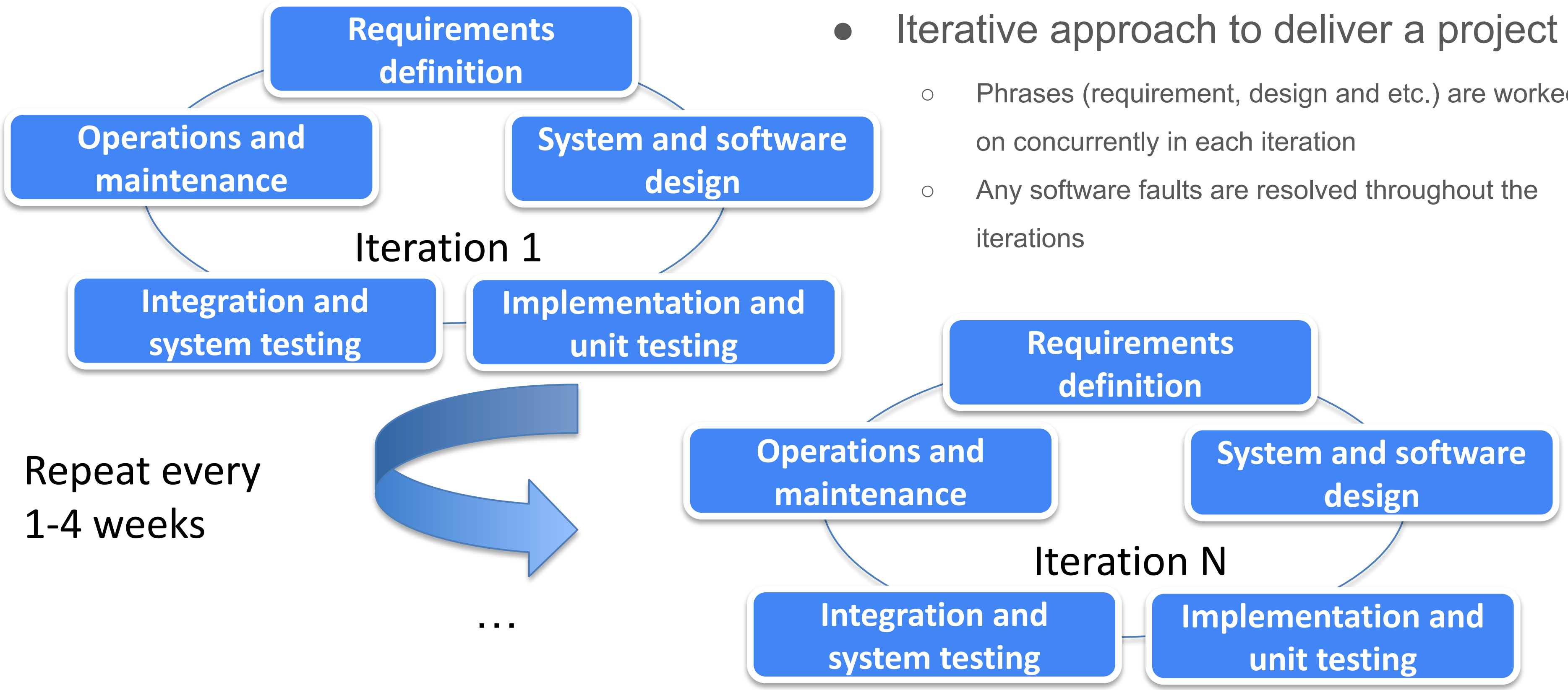
Agile methodology proposes principles instead of an actual process

- Proposed by developers based on their experience
- The Agile Manifesto: https://agilemanifesto.org/
- The Agile Manifesto contains 12 principles that focus on 4 values:
    - Working software over comprehensive documentation
    - Customer collaboration over contract negotiation
    - Responding to change over following a plan
    - Individuals and interactions over processes and tools

Agile methodology

- Suggests an iterative approach to deliver a project, different from traditional approaches (e.g., waterfall model)
  - Focuses on continuous releases
  - Flexibility to adjust and iterate during the development process
- Cornerstone of DevOps practices
  - By fostering collaboration and feedback (both with the customer and within the team)
  - Deliver earlier = detect faults earlier, less expensive to fix (developing software as building house, software faults as cracks in house’s foundation)

Agile workflow


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operations and maintenance
Iteration 1
Repeat every 1-4 weeks
...
Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operations and maintenance
Iteration N

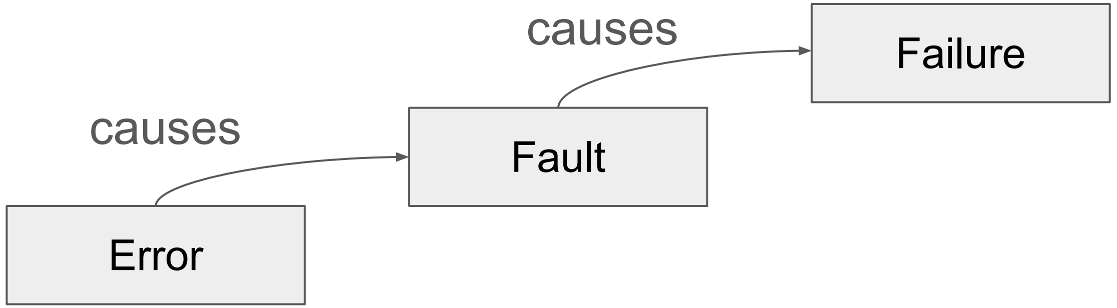
- Iterative approach to deliver a project
  - Phrases (requirement, design and etc.) are worked on concurrently in each iteration
  - Any software faults are resolved throughout the iterations

What is a software fault?

Error: a mistake in the code

Fault: a defect in the code that cause incorrect or unexpected results

Failure: occurs when the system fails to perform its required function



A developer makes an error that causes a fault in the software, which can cause a failure.

What is a software fault?

Error: a mistake in the code

Fault: a defect in the code that can cause incorrect or unexpected results

Failure: occurs when the system fails to perform its required function

```
public static int numZero (int[] x) {
        int count = 0;
        for (int i = 1; i < x.length; i++) {
                if (x[i] == 0) {
                        count ++;
                }
        }
        return count;
}
```

Use case 1

Input: [0, 2, 3]
Output: 0

Error state
First iteration *for* where *i* = 1, skipping index 0.

Is there a/an ___?	
Error	True / False
Fault	True / False
Failure	True / False

What is a software fault?

Error: a mistake in the code

Fault: a defect in the code that can cause incorrect or unexpected results

Failure: occurs when the system fails to perform its required function

```
public static int numZero (int[] x) {
        int count = 0;
        for (int i = 1; i < x.length; i++) {
                if (x[i] == 0) {
                        count ++;
                }
        }
        return count;
}
```

Use case 1

Input: [0, 2, 3]
Output: 0

Error state
First iteration *for* where *i* = 1, skipping index 0.

Is there a/an ___?	
Error	True / False
Fault	True / False
Failure	True / False

What is a software fault?

Error: a mistake in the code

Fault: a defect in the code that can cause incorrect or unexpected results

Failure: occurs when the system fails to perform its required function

```
public static int numZero (int[] x) {
        int count = 0;
        for (int i = 1; i < x.length; i++) {
                if (x[i] == 0) {
                        count ++;
                }
        }
        return count;
}
```

Use case 2

Input: [2, 3, 0]
Output: 1

Error state
First iteration *for* where *i* = 1, skipping index 0.

Is there a/an ___?	
Error	True / False
Fault	True / False
Failure	True / False

What is a software fault?

Error: a mistake in the code

Fault: a defect in the code that can cause incorrect or unexpected results

Failure: occurs when the system fails to perform its required function

```
public static int numZero (int[] x) {
        int count = 0;
        for (int i = 1; i < x.length; i++) {
                if (x[i] == 0) {
                        count ++;
                }
        }
        return count;
}
```

Use case 2

Input: [2, 3, 0]
Output: 1

Error state
First iteration *for* where *i* = 1, skipping index 0.

Is there a/an ___?	
Error	[✓ True] / False
Fault	[✓ True] / False
Failure	True / [✓ False]

Scrum

- Scrum is an agile project management framework that helps teams structure and manage their work based on the Agile principles.
- Scrum is composed of three main aspects:
    - Roles
    - Artifacts
    - Workflow

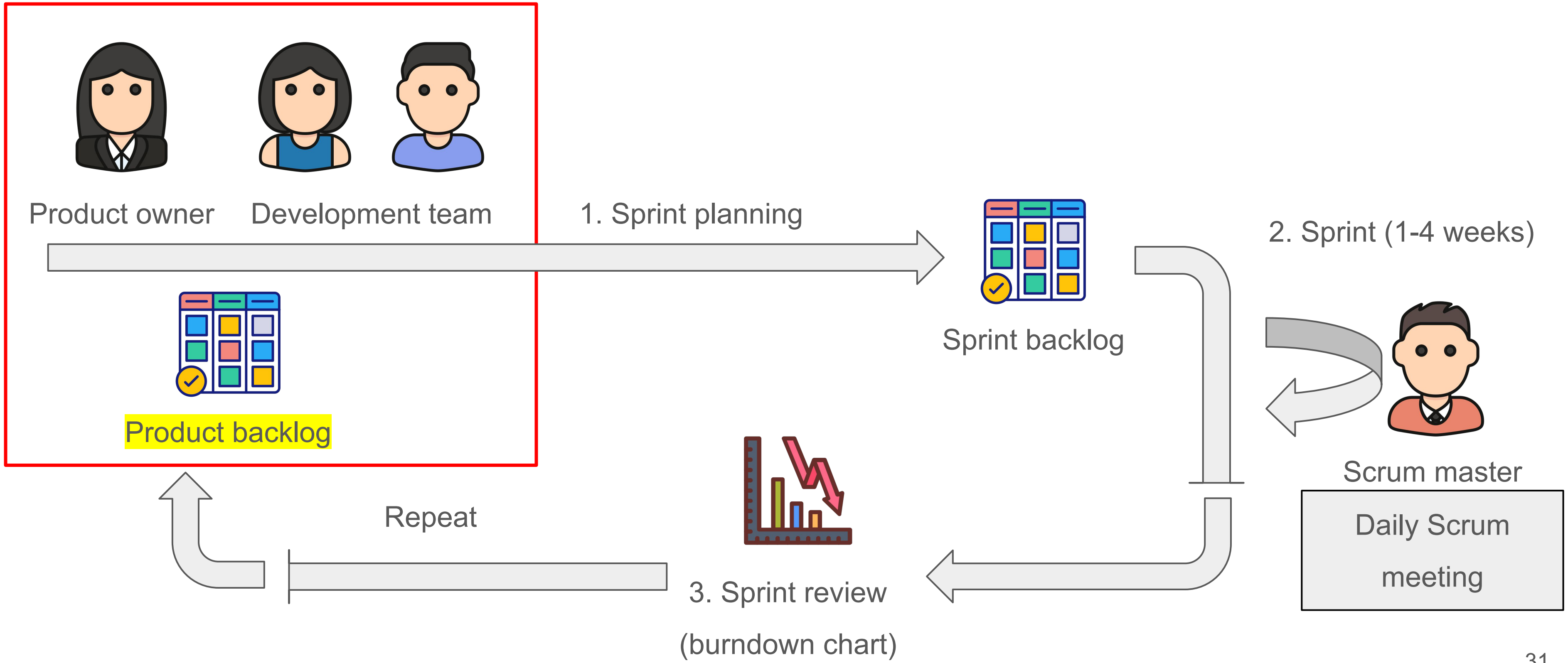
Roles

- Product owner
  - Makes sure the product is what the customer wants
  - Defines features to implement
- Scrum team
  - Individuals who are working on the software
  - Team members are cross-functional (consist of developers, testers, designers…)
- Scrum master
  - Makes sure scrum process is being followed
  - Hosts Scrum meetings
  - Removes impediments so that team members can finish their task

Artifacts

- Product backlog
  - List of work that the team needs to do to deliver a finished product (new user stories, bug fixing, etc)
- Iteration/sprint backlog
  - List of work that the team needs to do for the current iteration/sprint
- Burndown chart
  - Keeps track of the remaining tasks for the current iteration

Workflow



Icon created by Jesus Chavarria and Vectors Tank

User stories

User stories: a software feature written from the end user’s perspective.

- Written in non-technical language to provide context
- Why? To describe what value a software feature provides to the customer
- An user story should explain: persona + need + purpose

User stories

User stories: a software feature written from the end user’s perspective.

- Written in non-technical language to provide context
- Why? To describe what value a software feature provides to the customer
- An user story should explain: persona + need + purpose

As a [persona], I want [need], so that [purpose].

- Persona: Who is the end user?
- Need: What is the goal of the end user?
- Purpose: What problem does the end user look to solve?

Example: As a sales director, I want a sales report, so that I can monitor the sales progress.

User stories

User stories: a software feature written from the end user's perspective.

- Written in non-technical language to provide context
- Why? To describe what value a software feature provides to the customer
- An user story should explain: persona + need + purpose

As a [persona], I want [need], so that [purpose].

- Persona: Who is the end user?

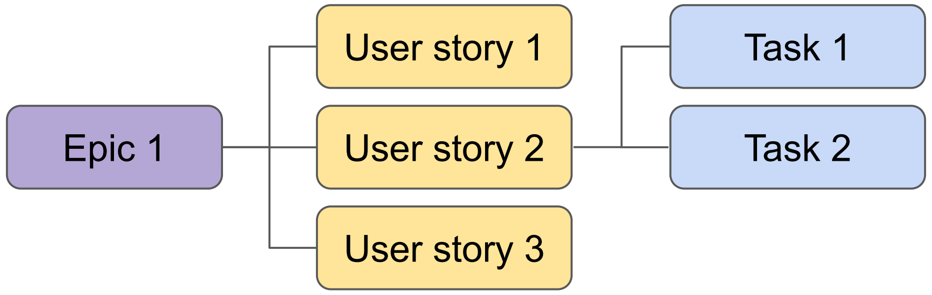
You will be asked to provide two user stories for your first project deliverable.

More details will be discussed in the next class.

Example: As a sales director, I want a sales report, so that I can monitor the sales progress.

Epics

Epics: large bodies of work that can contain a collection of user stories.



- Why? To organize the work and clarify on client priorities
- An epic cannot be completed in a single iteration

Examples of epics: “chatbot functionality” in a web app, “augmented reality features” in a mobile app, and etc.

Story points and estimation

Traditional software teams give estimates in a time format: days, weeks, months. Agile teams use story points to quantify the efforts required to carry out a task (user story).

Story points are relative to:

- Work complexity
- Amount of work and hours required to complete
- Risk and uncertainty

Planning poker

Planning poker: a consensus-based estimating technique for story points where everyone (e.g., developers, designers, testers …) is involved

- The product owner reads a user story, describes it
- Everyone picks a card to represent their estimate
    - Everyone holds a deck of planning poker cards. Each card contains a value of the Fibonacci sequence (e.g., 0, 1, 2, 3, 5, 8, 13, 20, and 40)
- Reveal their card and discuss the differences between estimates
- Re-estimate until a consensus is reached within the team

Note: usually, you should never have user stories worth more than 16 hours of work (or 20 story points)

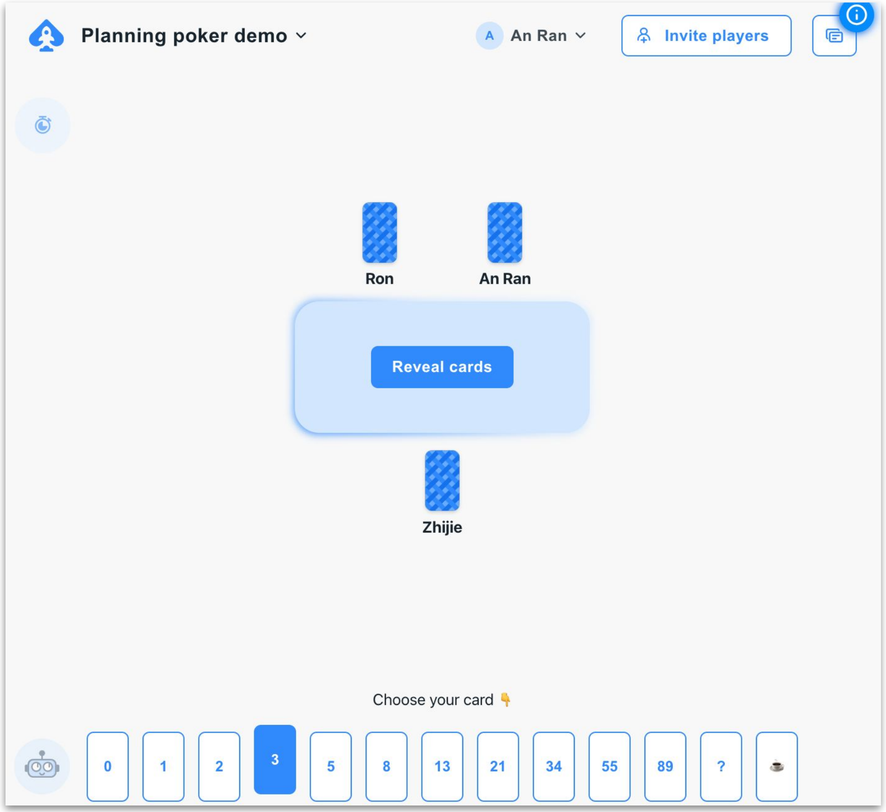
- larger the task, more difficult to have an accurate estimate

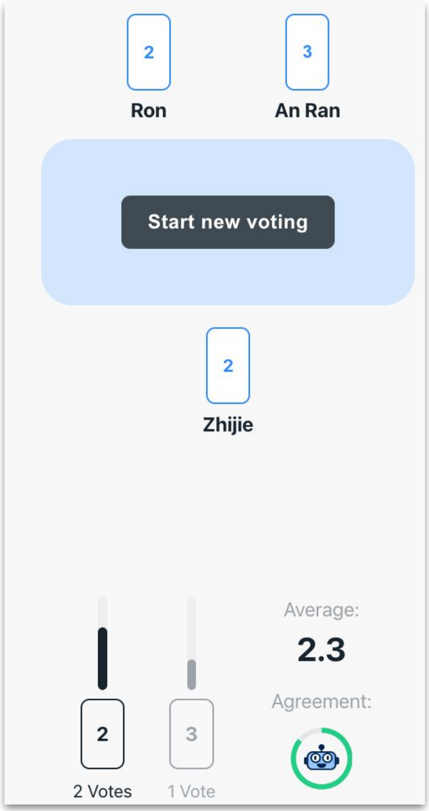
Planning poker

Common problem 1: Product uncertainty
Solution: put the story aside and research until uncertainty can be resolved

Common problem 2: Technical uncertainty
Solution: decide on the story points later, use a range as the estimate to begin with

Planning poker

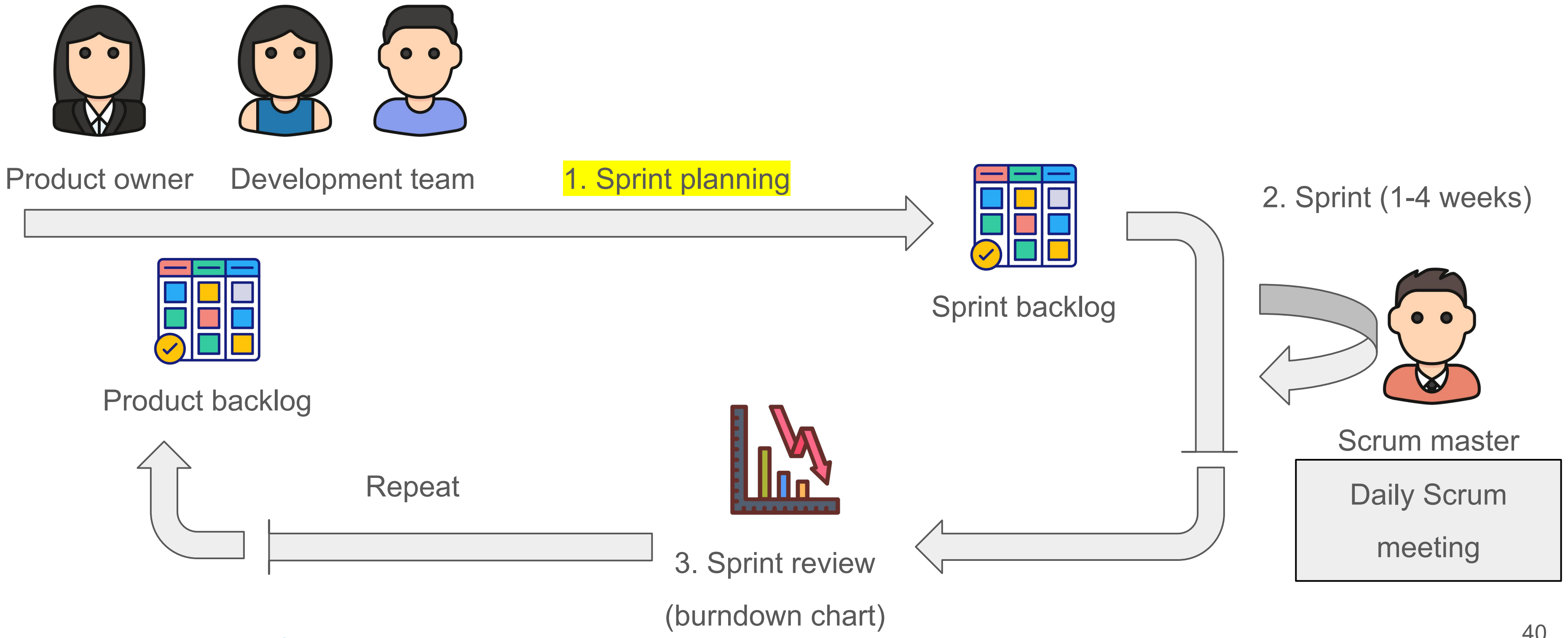




Try it yourself!

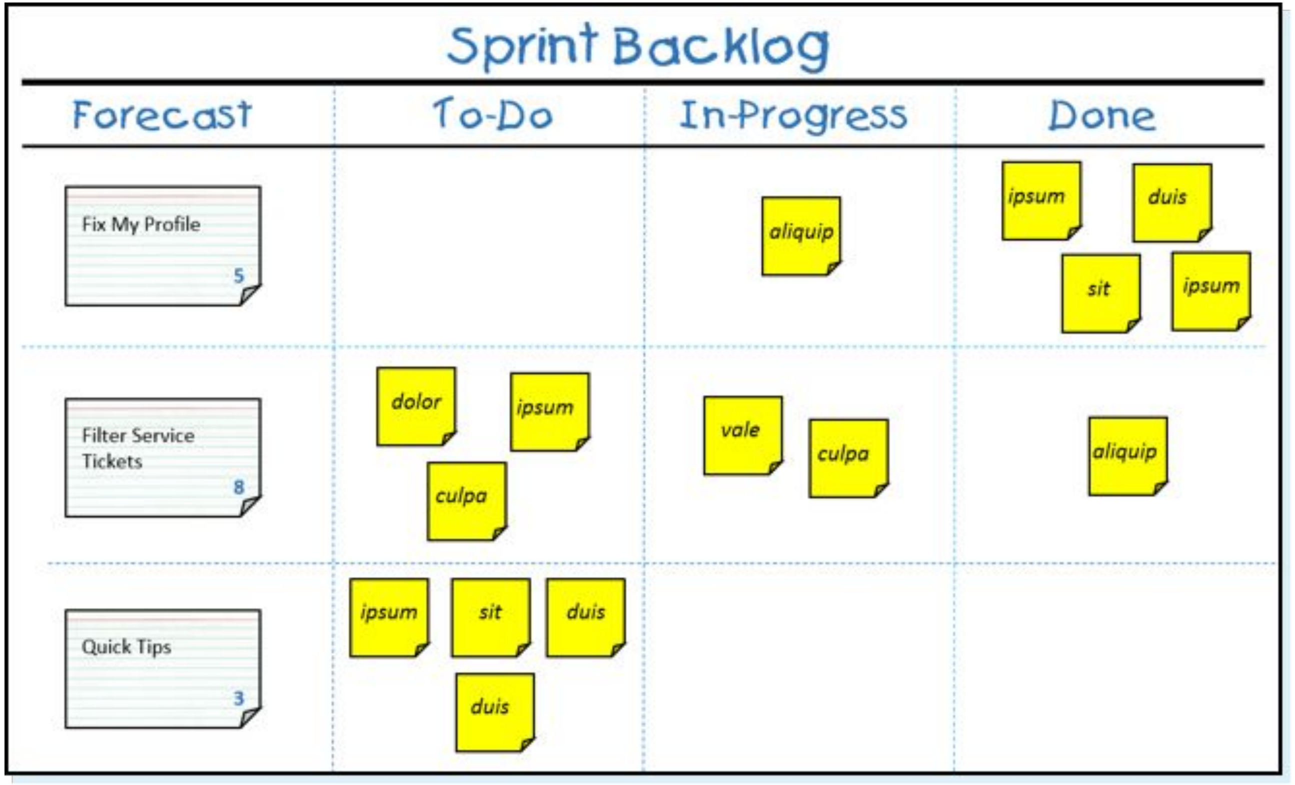
Planning poker online: https://planningpokeronline.com/

Workflow



Icon created by Jesus Chavarria and Vectors Tank

An example of sprint backlog

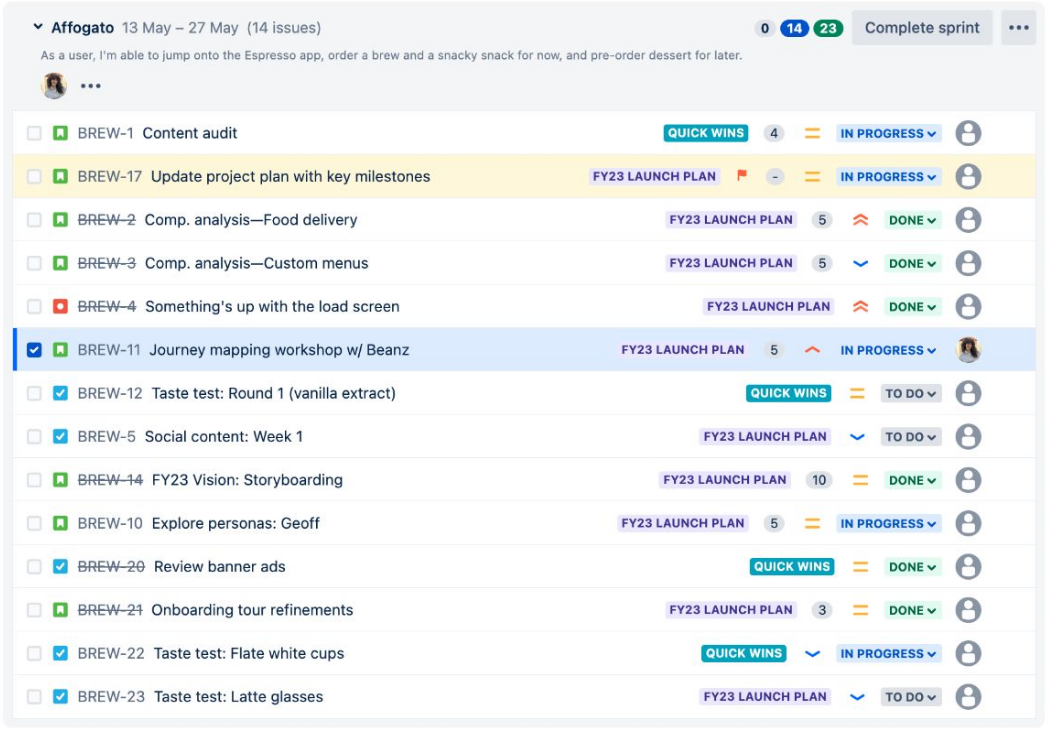


Sprint Backlog

Forecast	To-Do	In-Progress	Done
Fix My Profile 5		aliquip	ipsum duis sit ipsum
Filter Service Tickets 8	dolor ipsum culpa	vale culpa	aliquip
Quick Tips 3	ipsum sit duis duis		

Image from scrum.og

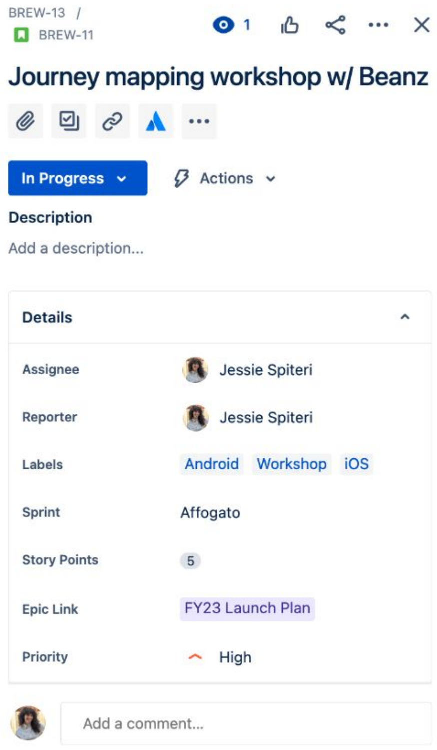
Another example of sprint backlog



Affogato 13 May – 27 May (14 issues) 0 14 23 Complete sprint

As a user, I'm able to jump onto the Espresso app, order a brew and a snacky snack for now, and pre-order dessert for later.

BREW-1	Content audit	QUICK WINS	4	=	IN PROGRESS
BREW-17	Update project plan with key milestones	FY23 LAUNCH PLAN	-	=	IN PROGRESS
~~BREW-2~~	Comp. analysis—Food delivery	FY23 LAUNCH PLAN	5	⌃⌃	DONE
~~BREW-3~~	Comp. analysis—Custom menus	FY23 LAUNCH PLAN	5	⌄	DONE
~~BREW-4~~	Something's up with the load screen	FY23 LAUNCH PLAN		⌃⌃	DONE
BREW-11	Journey mapping workshop w/ Beanz	FY23 LAUNCH PLAN	5	⌃	IN PROGRESS
BREW-12	Taste test: Round 1 (vanilla extract)	QUICK WINS		=	TO DO
BREW-5	Social content: Week 1	FY23 LAUNCH PLAN		⌄	TO DO
~~BREW-14~~	FY23 Vision: Storyboarding	FY23 LAUNCH PLAN	10	=	DONE
BREW-10	Explore personas: Geoff	FY23 LAUNCH PLAN	5	=	IN PROGRESS
~~BREW-20~~	Review banner ads	QUICK WINS		=	DONE
~~BREW-21~~	Onboarding tour refinements	FY23 LAUNCH PLAN	3	=	DONE
BREW-22	Taste test: Flate white cups	QUICK WINS		⌄	IN PROGRESS
BREW-23	Taste test: Latte glasses	FY23 LAUNCH PLAN		⌄	TO DO



BREW-13 / BREW-11

1

Journey mapping workshop w/ Beanz

In Progress Actions

Description

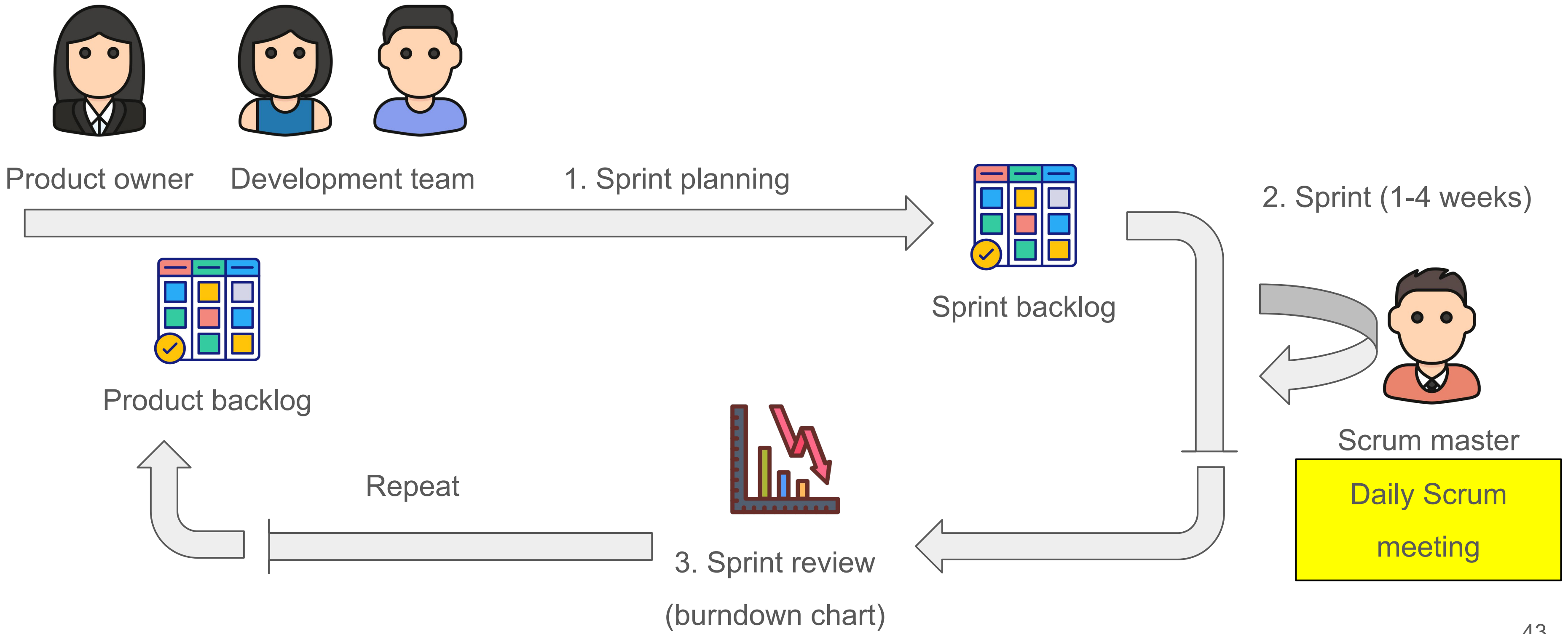
Add a description...

Details

Assignee	Jessie Spiteri
Reporter	Jessie Spiteri
Labels	Android Workshop iOS
Sprint	Affogato
Story Points	5
Epic Link	FY23 Launch Plan
Priority	High

Add a comment...

Workflow



Icon created by Jesus Chavarria and Vectors Tank

Daily Scrum meeting

15-minutes meeting held on each day of a sprint.

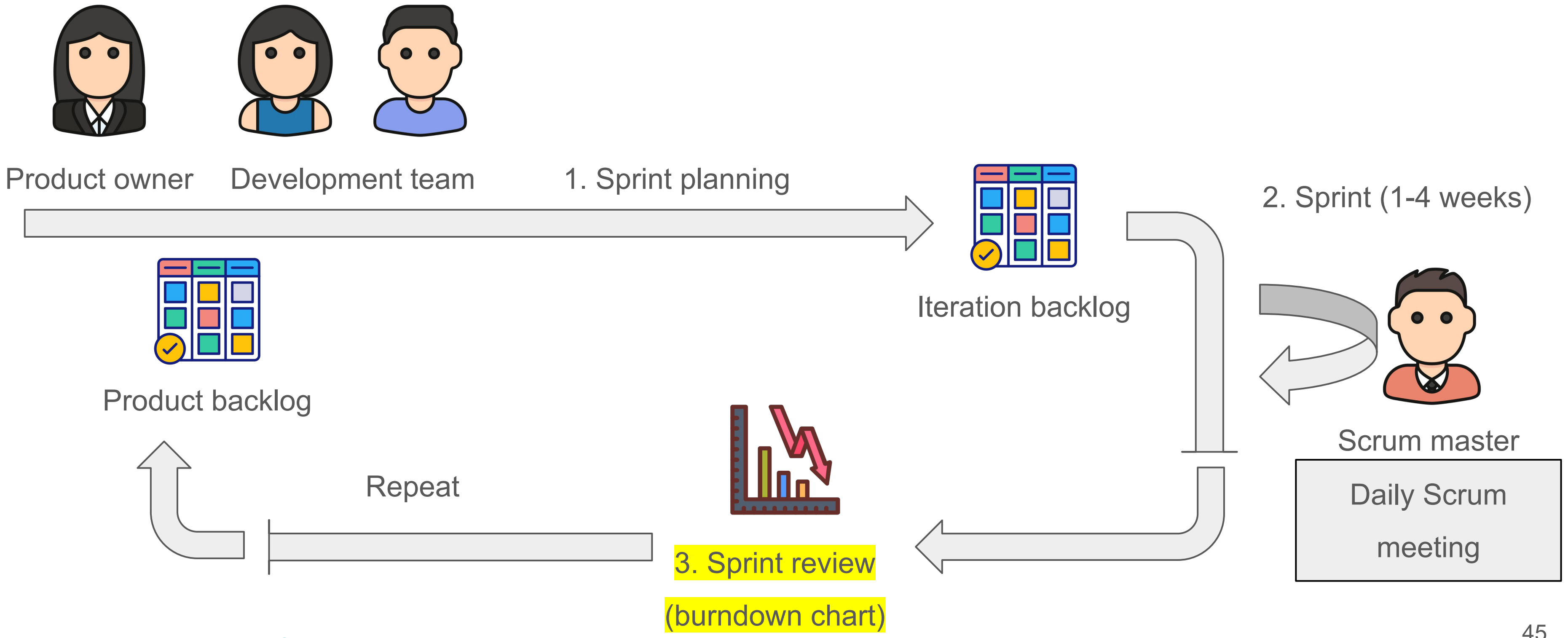
Three questions of the daily meeting:

- What did I do yesterday?
- What will I do today?
- Are there any impediments in my way?

Scrum master is responsible for resolving the impediments.

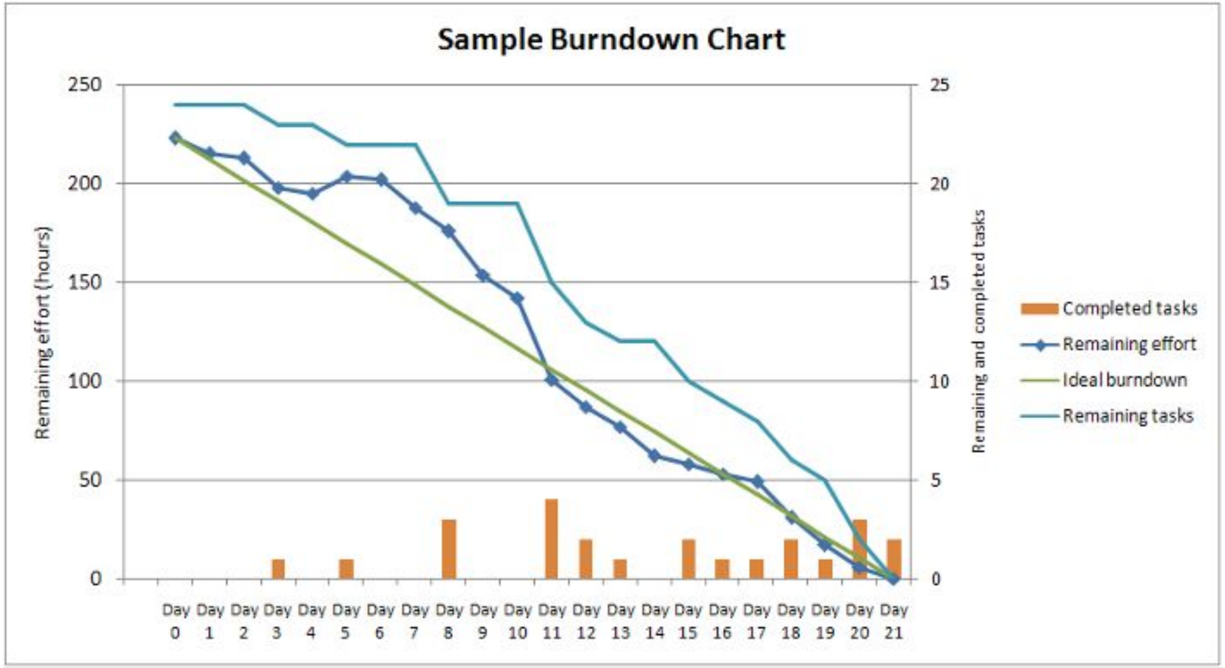
- Examples of impediments: “I need help debugging a database problem.”
- Solution: “Assigning a team member for pair programming”

Workflow



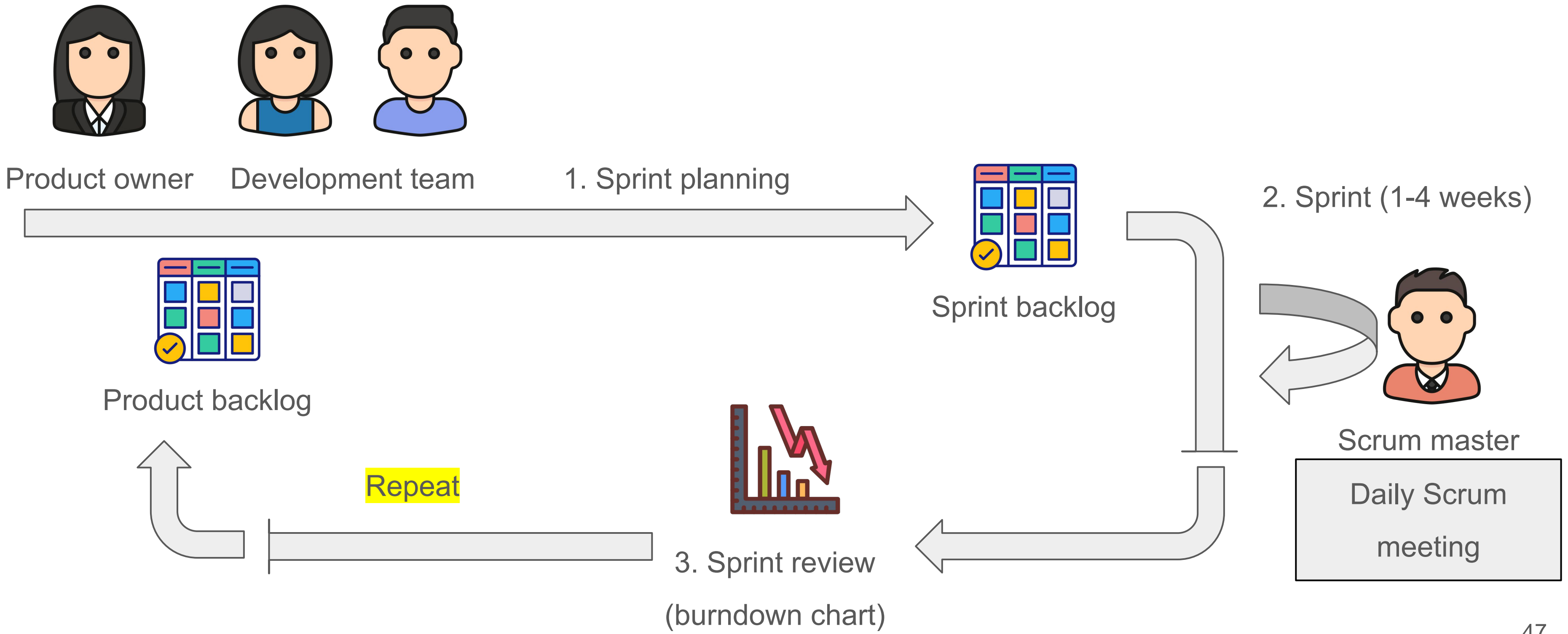
Icon created by Jesus Chavarria and Vectors Tank

An example burndown chart



By Pablo Straub - Own work, Public Domain,
https://commons.wikimedia.org/w/index.php?curid=7132232

Workflow



Icon created by Jesus Chavarria and Vectors Tank

Scrum

Sillicon Valley Season 1 Episode 5

Agile methodology

- Pros
  - Strong flexibility
    - Able to change requirements efficiently
  - Close customer collaboration
    - Make sure the things we do are what the customers want
  - Promote teamwork
    - Help each other when one encounters roadblocks
- Cons
  - Lack of documentation
  - Require continuous planning

TODOs

- Join our Slack workspace
- In Slack #team channel: post your team information before Friday 23:59 pm
  - Team Name: Team ECE422
  - Name: An Ran Chen, Ron Unrau, Zhijie Wang
- Setup your Cybera account
  - In this course, you will be using Cybera, which is a tool to help deploy your projects.