Software Process

Software Engineering
Prof. Maged Elaasar

Learning Objectives

- Understand the concept of a software development process
- Learn the different software development process models
- Learn the Scrum agile process in more detail
- Learn how to practice Scrum process in Github

Software Process

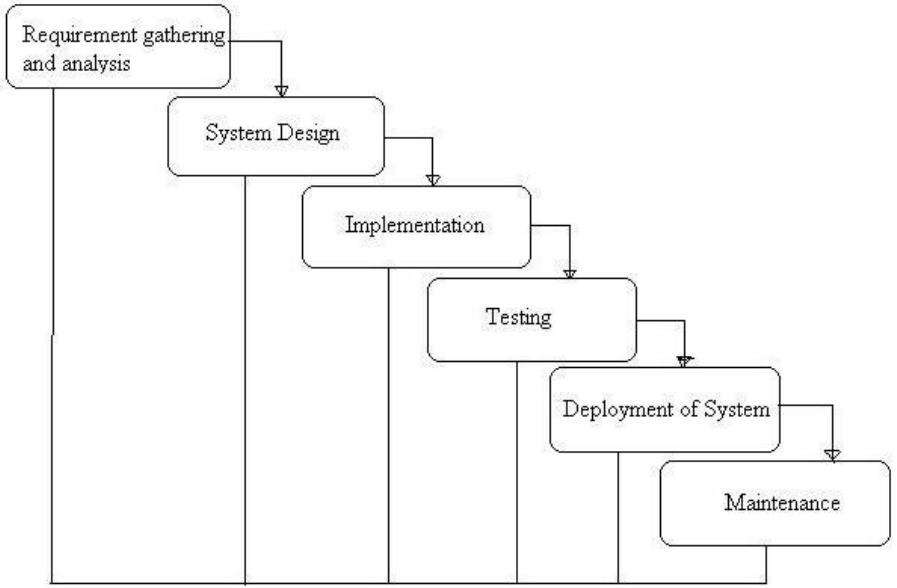
- A **software process** is a set of related activities that leads to the production of software.
- A **software process model** is an abstraction of a process that can be extended and refined to create concrete processes

Software Process Models

- Waterfall Model
  - V-Model
- Incremental Model
  - RAD Model
- Iterative Model
  - Spiral Model
- Agile Model
  - Scrum Model

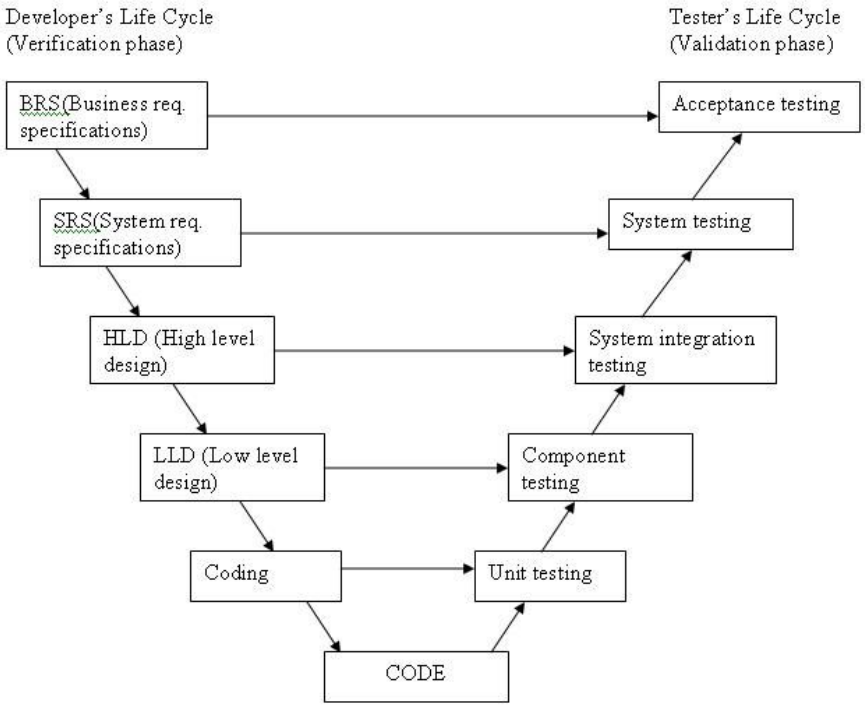
The Waterfall Model

- Each phase must be completed fully
- At the end of each phase, a review takes place as a gate to next phase
- Used only when the requirements are very clear and fixed, technology is understood and ample resources are available



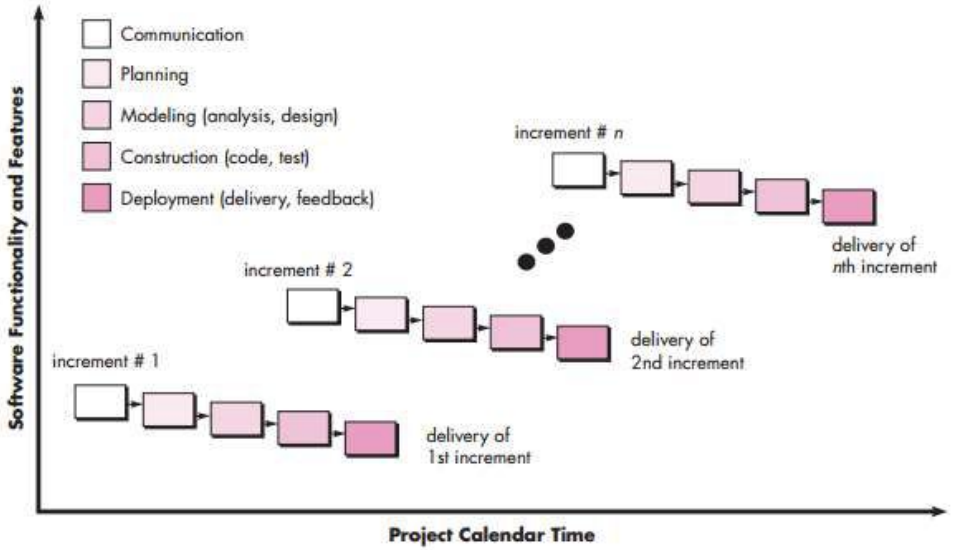
The V-Model

- A type of **waterfall** model
- Testing of the product is planned in parallel with a corresponding dev phases.
- defects are found at early stage



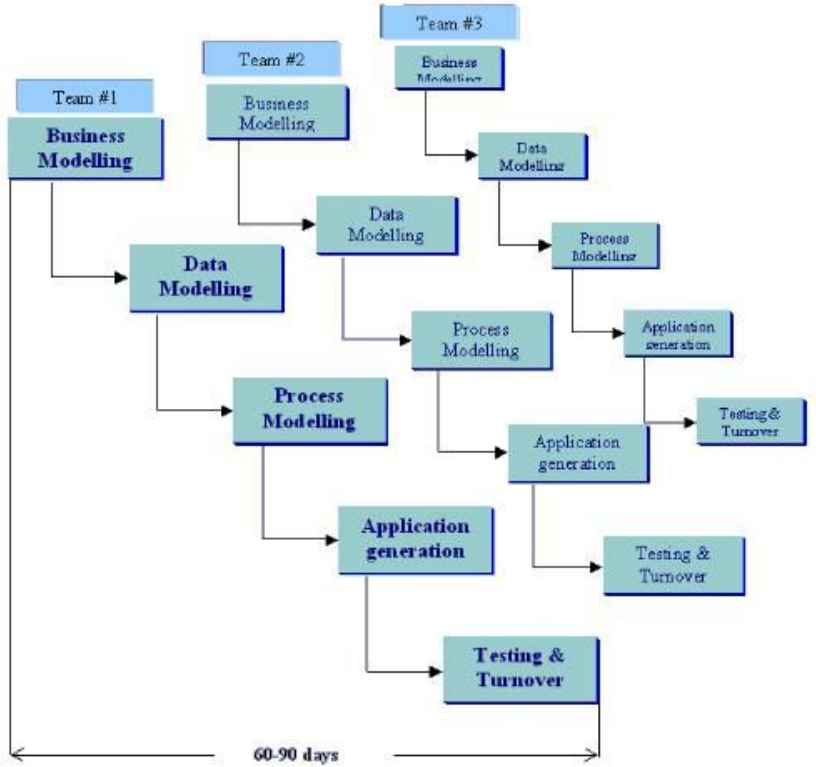
The Incremental Model

- Major requirements are broken down into multiple increments.
- Each subsequent increment adds function to the previous one.
- Used when there is a need to get a product to the market early
- Customer can respond to each increment
- Lowers the initial delivery cost
- Easier to manage risk



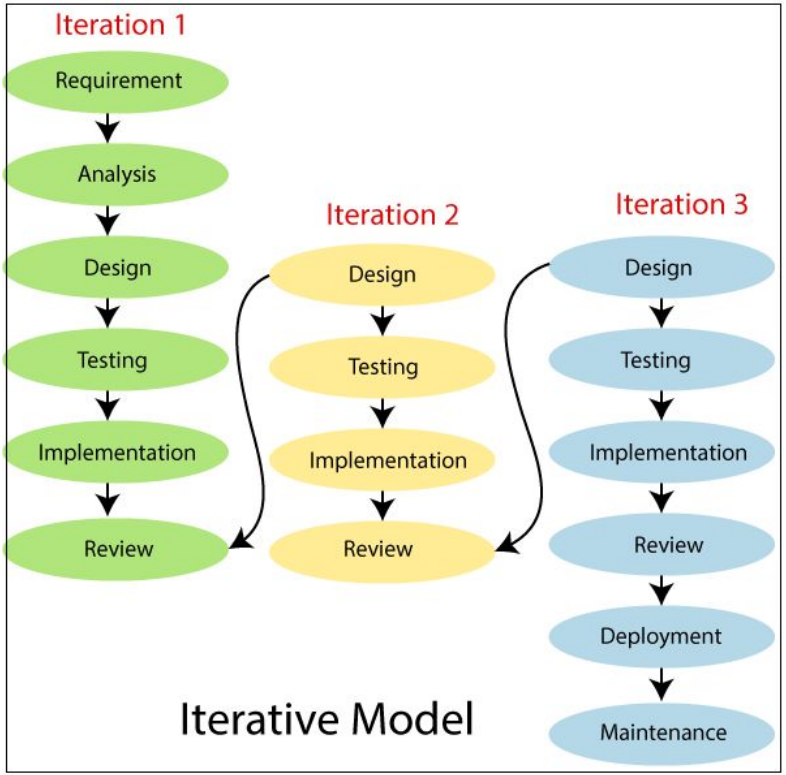
The RAD Model

- Rapid Application Development; a type of **incremental** model.
- The components or functions are developed in parallel.
- The developments are time boxed, delivered and then assembled into a working prototype
- Often code generated from models; requires highly skilled developers; and bigger budget to use modeling



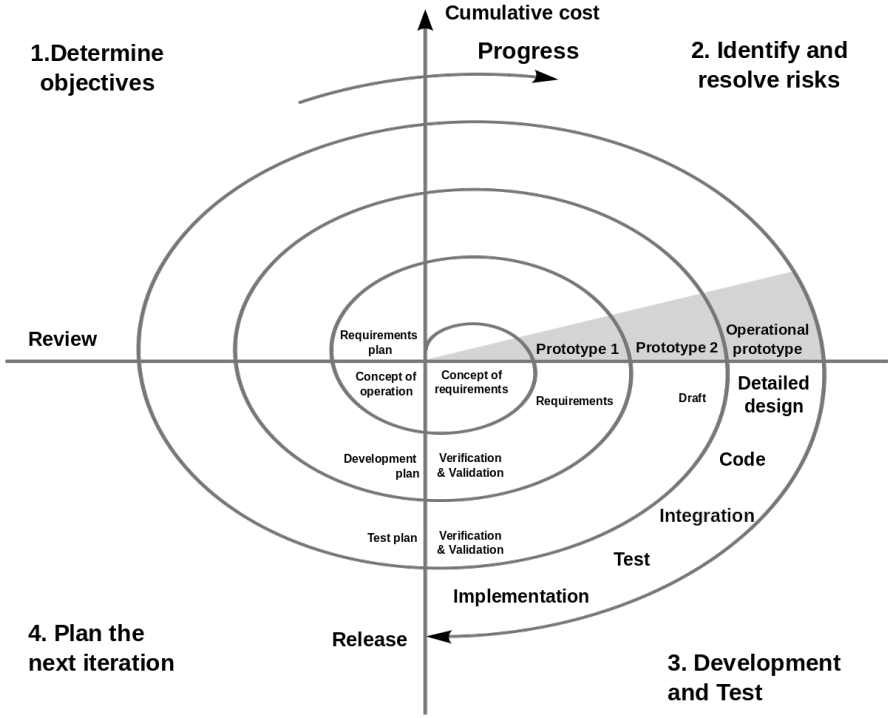
The Iterative Model

- Starts by designing and implementing part of the requirements only, which can then be reviewed before identifying further requirements.
- This process is then repeated, producing a new version of the software for each cycle of the model.
- Can get reliable user feedback early
- Each iteration is rigid with no overlaps
- May lead to costly redesign.



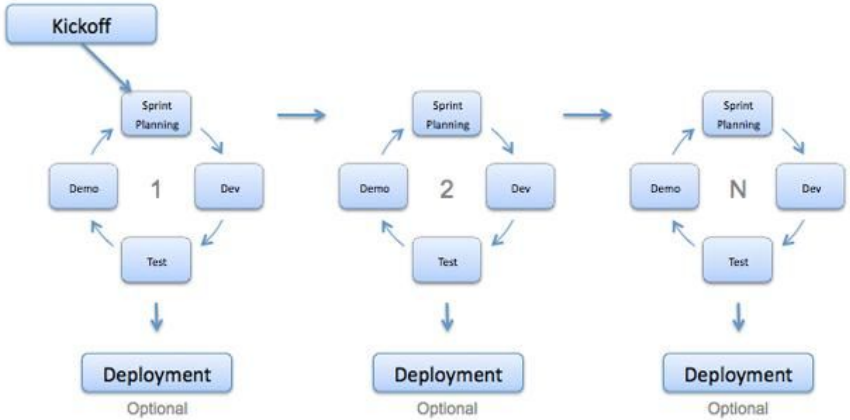
The Spiral Model

- A type of **iterative** model where iterations are called spirals with the first one called the baseline spiral
- Each spiral has four phases: Planning, Risk Analysis, Development and Evaluation
- At the end of each spiral, risk is assessed and next requirements are gathered
- Risk analysis requires highly specific expertise; good for large and mission-critical projects



The Agile Model

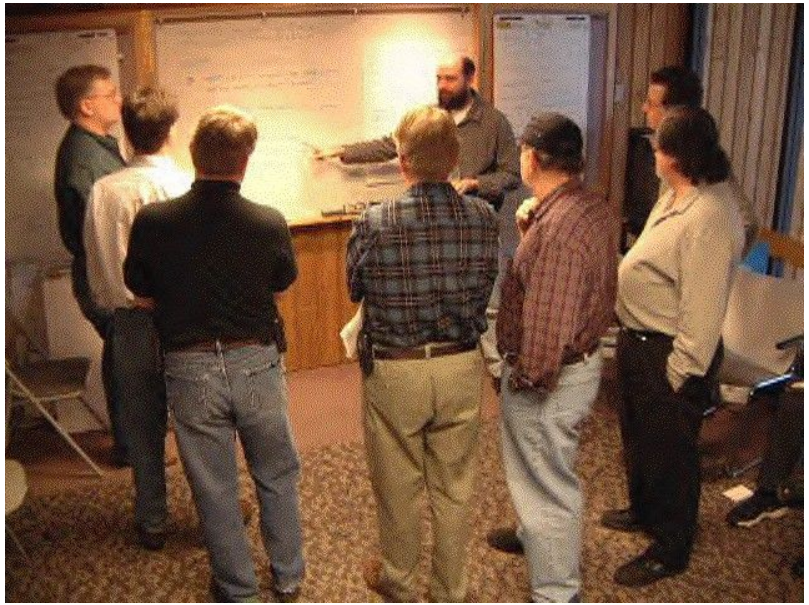
- A **mix of incremental and iterative** process where the software is built and delivered through successive refinement.
- Customers, developers and testers constantly interact with each other; adapts to changing requirements.
- Limited planning; designing and documentation



VALUE↑ + TIME↓

Agile Manifesto

Individuals and interactions >
Processes and tools



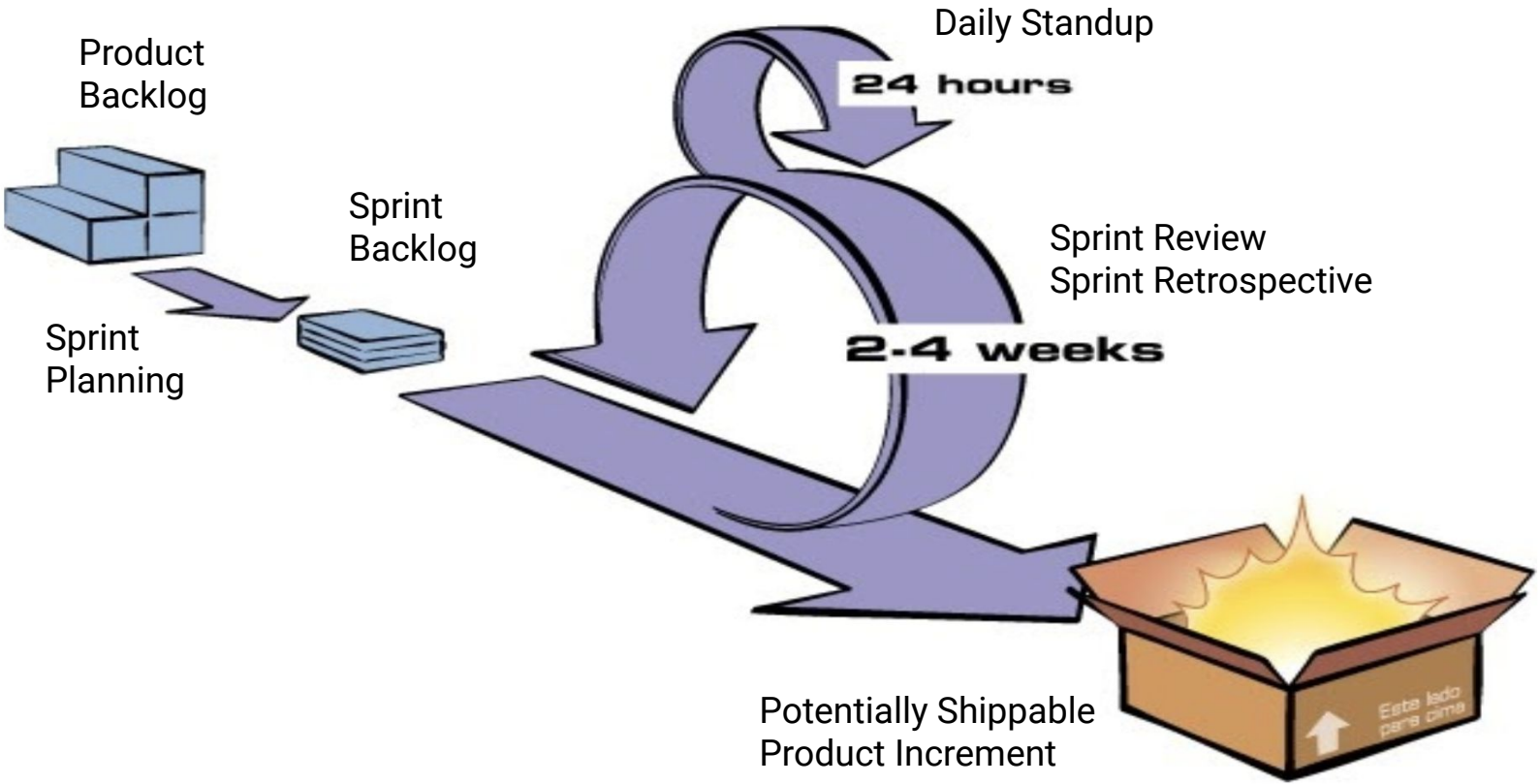
Working software >
Comprehensive documentation

Customer collaboration >
Contract negotiation

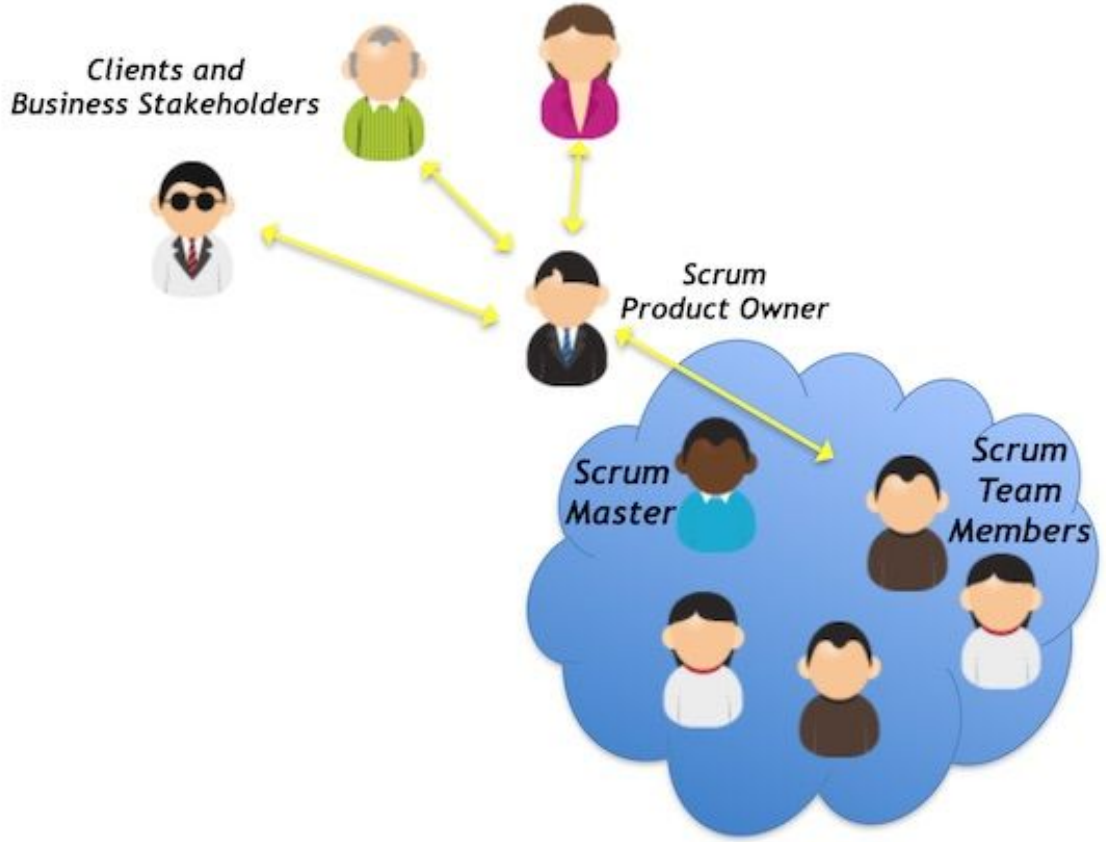
Responding to change >
Following a plan

Software Process Model Quiz

Scrum Process



Scrum Roles


Clients and
Business Stakeholders
Scrum
Product Owner
Scrum
Master
Scrum
Team
Members

Product Owner

- Represents Stakeholders
- ROI Responsible
- Accepts/Rejects Results
- Manages Product Backlog
- Defines/Prioritizes Functionalities

Product Backlog

TODO List

Product Requirements

Always Changing...

Dynamically Reprioritized

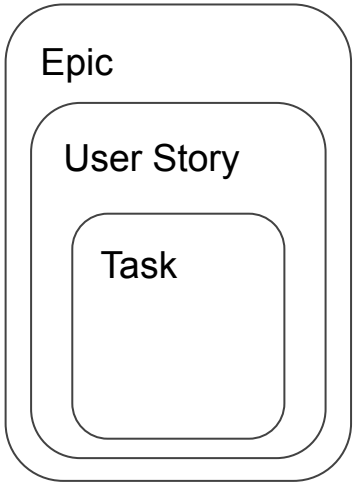
Product Backlog

- The **Product Backlog** consists of a prioritized list of user stories

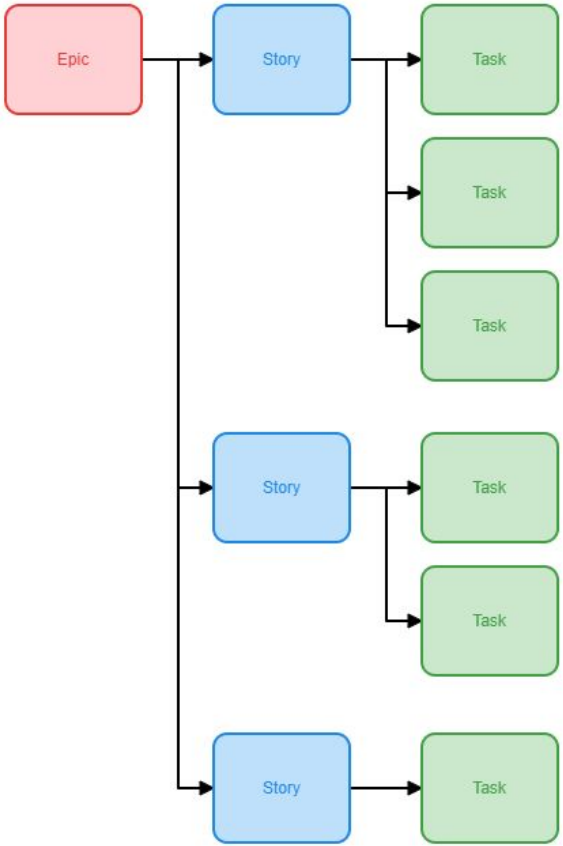
PRODUCT BACKLOG EXAMPLE						
ID	As a...	I want to be able to...	So that...	Priority	Sprint	Status
1	Administrator	see a list of all members and visitors	I can monitor site visits	Must	1	Done
2	Administrator	add new categories	I can allow members to create engaging content	Must	1	Done
3	Administrator	add new security groups	security levels are appropriate	Must	1	Done
4	Administrator	add new keywords	content is easy to group and search for	Must	1	Done
5	Administrator	delete comments	offensive content is removed	Must	1	Done
6	Administrator	block entries	competitors and offenders cannot submit content	Must	1	Done
7	Administrator	change site branding	the site is future-proofed in case brand changes	Could	1	Done
8	Member	change my password	I can keep secure	Must	1	Done
9	Member	update my contact details	I can be contacted by Administrators	Must	2	Work in Progress
10	Member	update my email preferences	I'm not bombarded with junk email	Should	2	Work in Progress
11	Member	share content to social networks	I can promote what I find interesting	Could	2	Work in Progress
12	Visitor	create an account	I can benefit from member discounts	Must		To be started
13	Visitor	login	I can post new entries	Must		To be started
14	Visitor	add comments	I can have a say	Must		To be started
15	Visitor	suggest improvements	I can contribute to the site usability	Should		To be started
16	Visitor	contact the Administrators	I can directly submit a query	Could		To be started
17	Visitor	follow a member's updates	I'm informed of updates from members I find interesting	Should		To be started
18	Visitor	view a member's profile	I can know more about a member	Must		To be started
19	Administrator	generate incoming traffic report	I can understand where traffic is coming from	Must		To be started

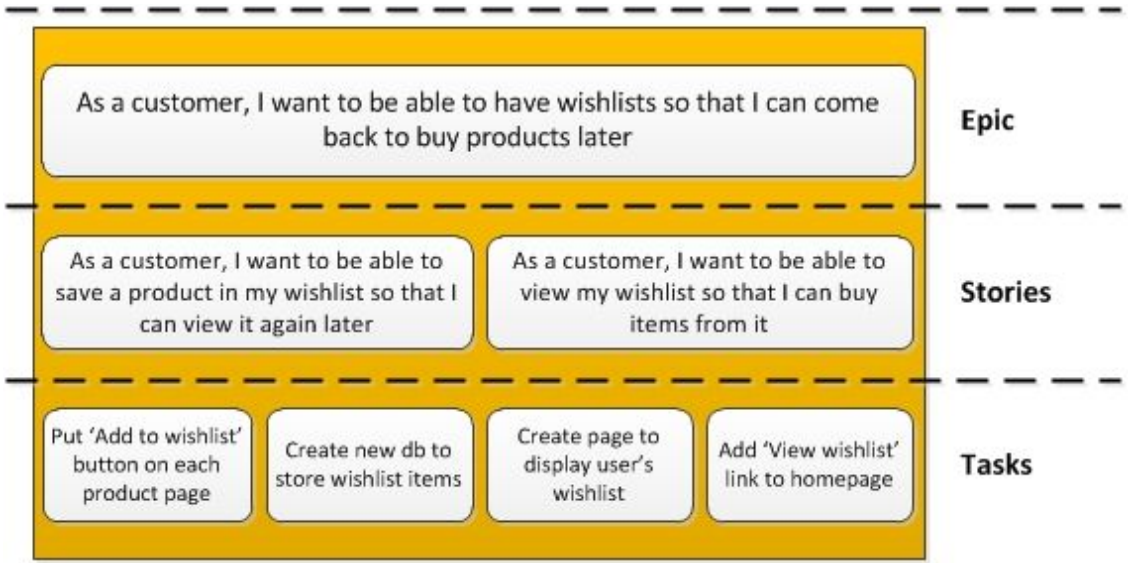
User Stories

- A **user story** is a way to specify a requirement from a user's perspective
  - Ex.: **"As a <type of user>, I want <some goal>, so that <some reason>"**
- It describes an incremental business value
- It includes a set of acceptance criteria
  - - Ex: "**Given <condition>, then <expected result>**"
- It must be doable within **one sprint**
- It can be be aggregated with related stories into an **epic**
  - An epic may span multiple sprints
- It can be decomposed into related **tasks**

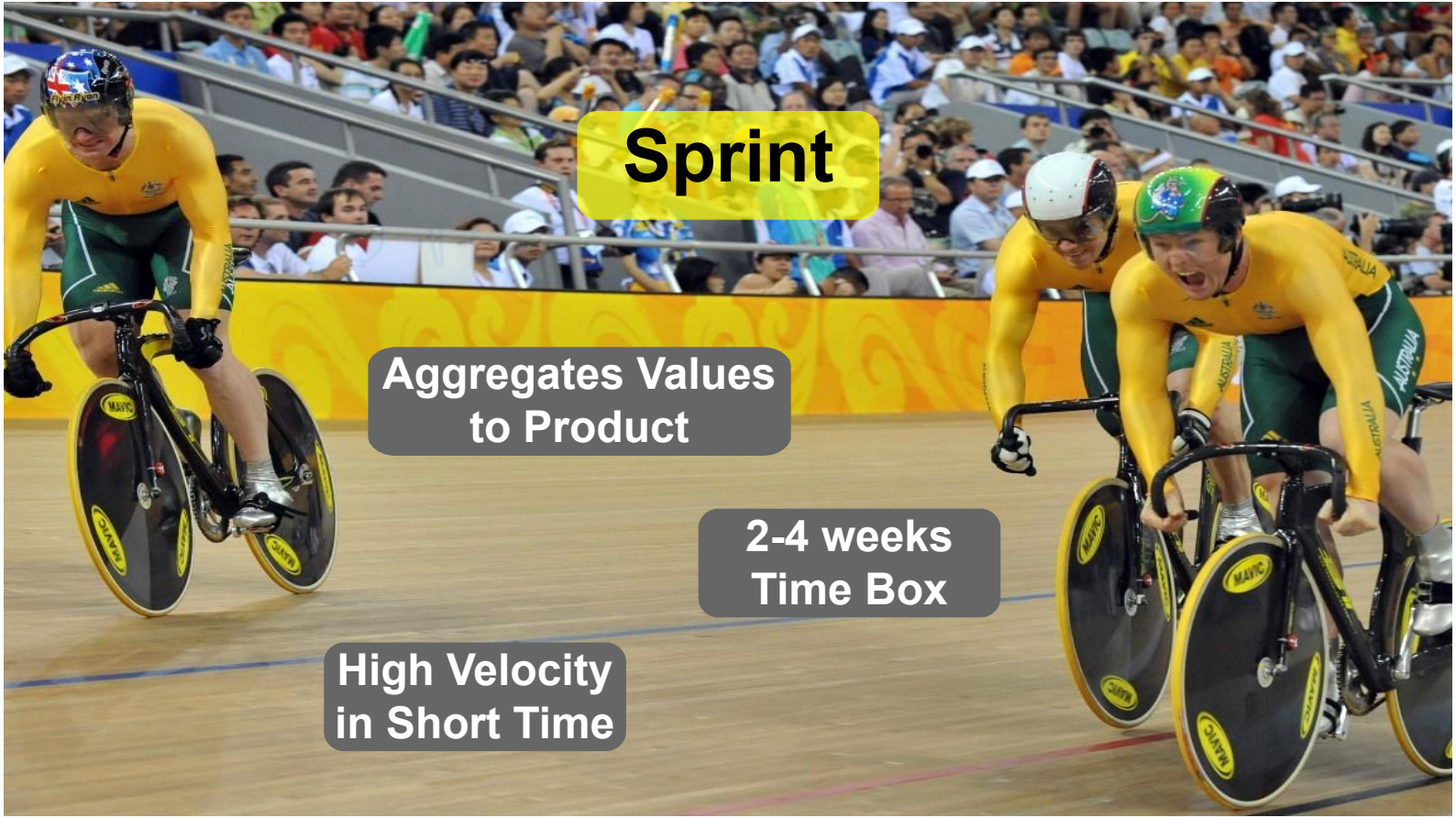


Epic → Story → Task





Sprint


Sprint
Aggregates Values to Product
2-4 weeks Time Box
High Velocity in Short Time

Sprint Planning

PO Presents High Value User Stories

Dev Team Selects Stories to Sprint, Estimate them and Splits them into Tasks

PO + Dev Team defines **Spring Goal** and a **Sprint Backlog** is Created

Dev Team


Scrum Master
I was Late!
Certified ScrumMaster
Self-managing
5 to 9 people
Multi-functional

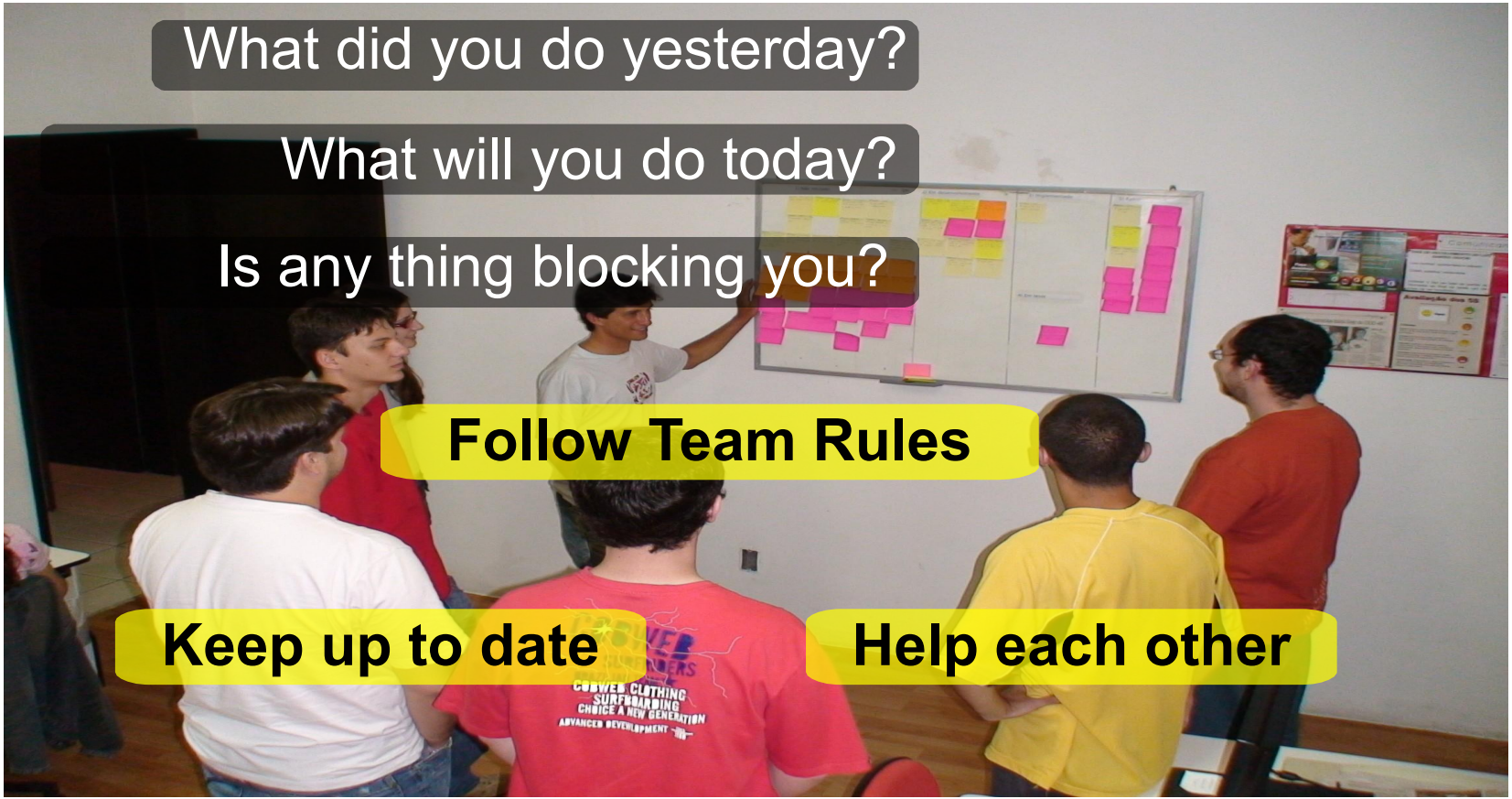
Developers


Backend developer
Software Architect
UX Engineer
Frontend developer

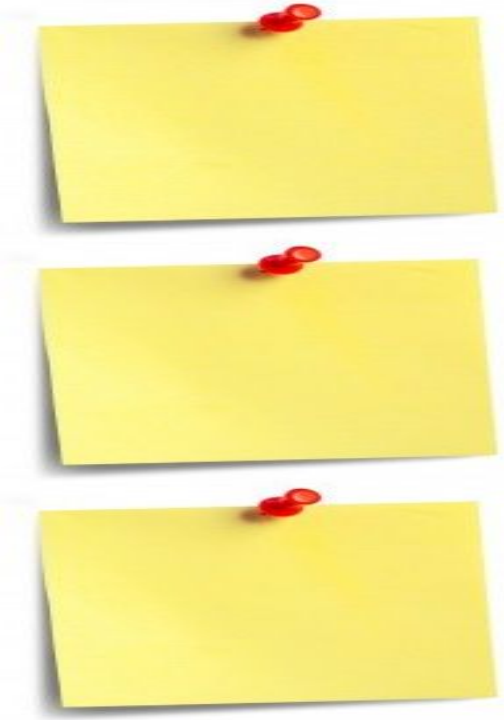
Scrum Master

- Removes Obstacles
- Avoids Interferences
- Keeps Focus on Sprint Goals
- Guarantees Collaboration
- Guards Scrum Practices

Daily Standup

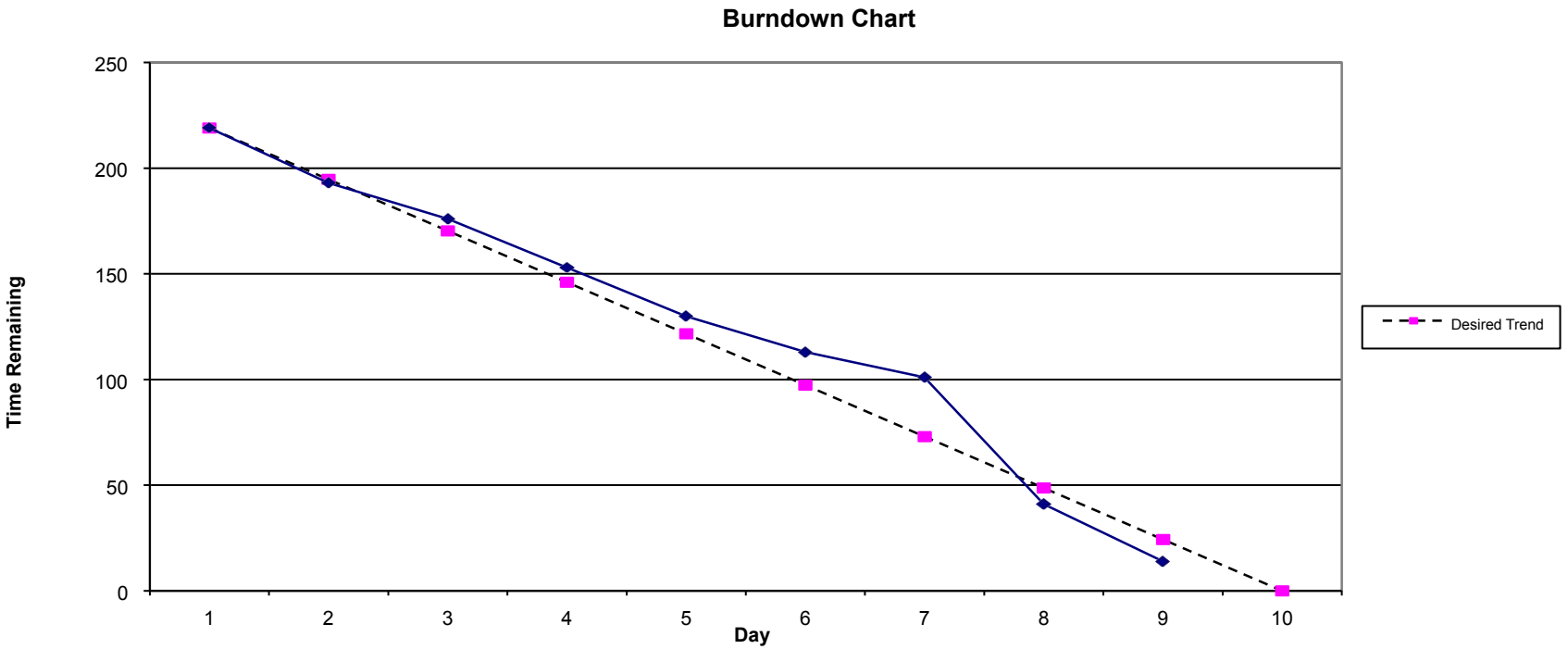


Sprint Backlog (Kanban Board)

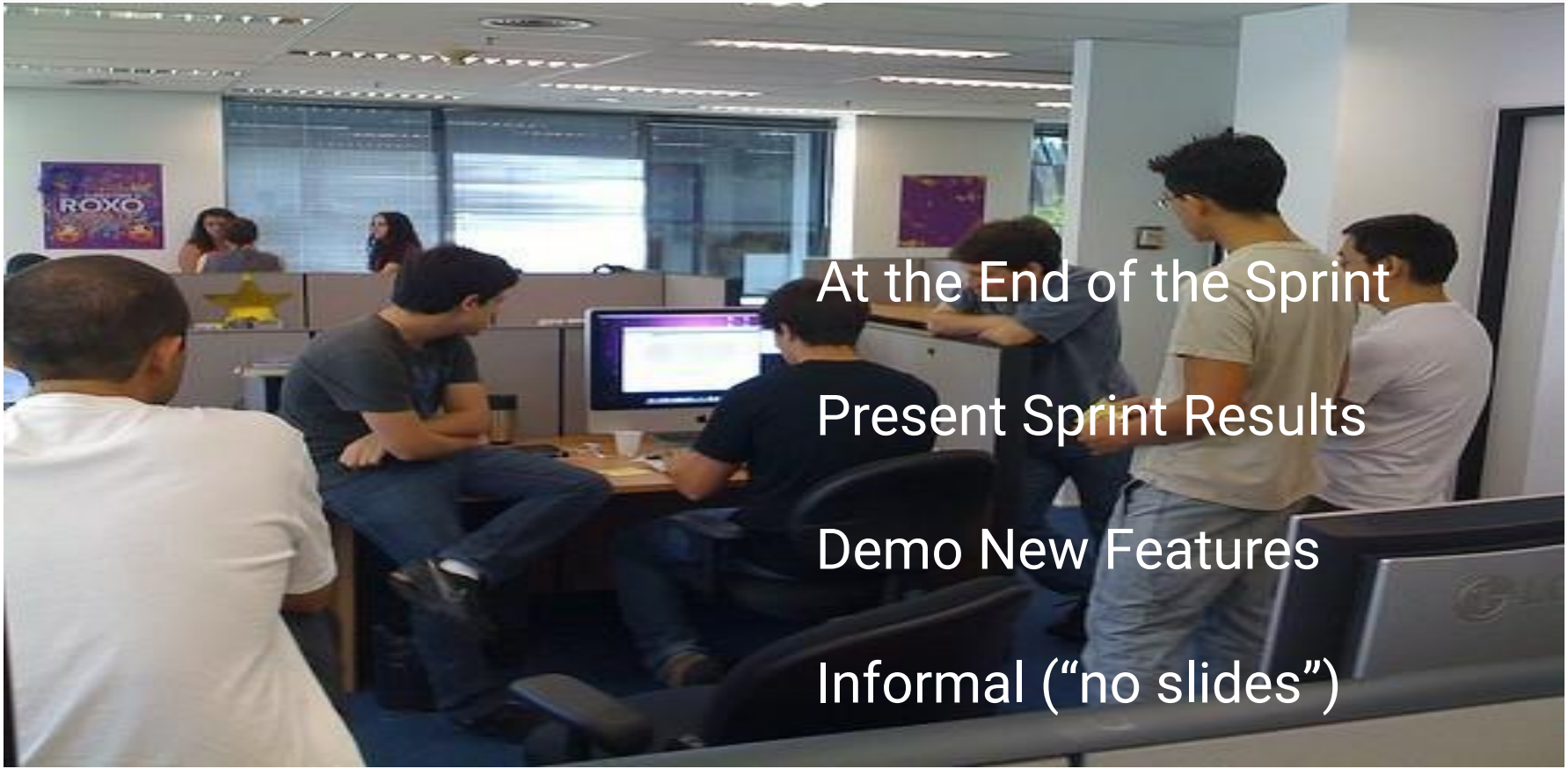
To Do	In Progress	Done
		

How do you determine that the work is done?

Burndown Chart



Sprint Review



At the End of the Sprint

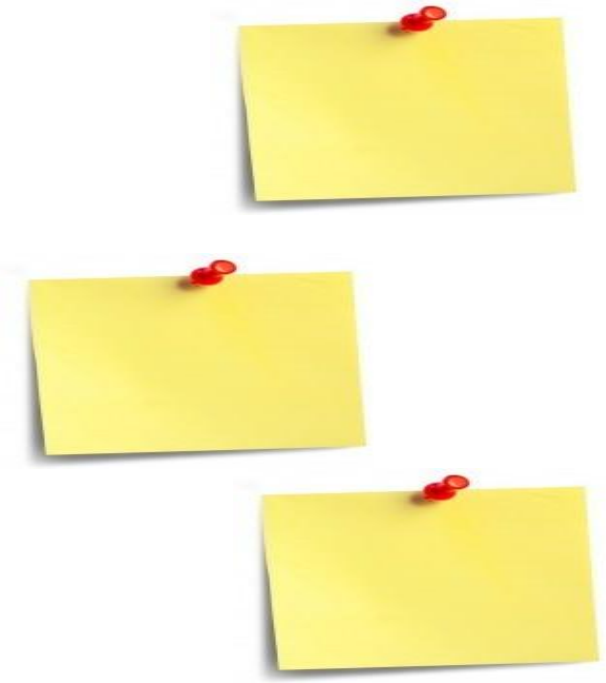
Present Sprint Results

Demo New Features

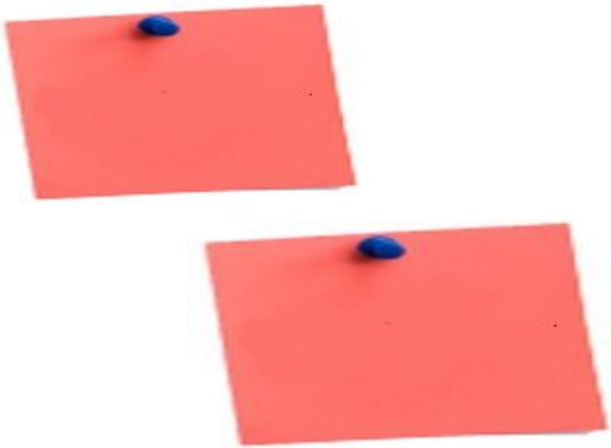
Informal (“no slides”)

Sprint Retrospective

What Worked Well



What Can be Better



Scrum Summary

Roles

Product Owner
Scrum Master
Dev Team

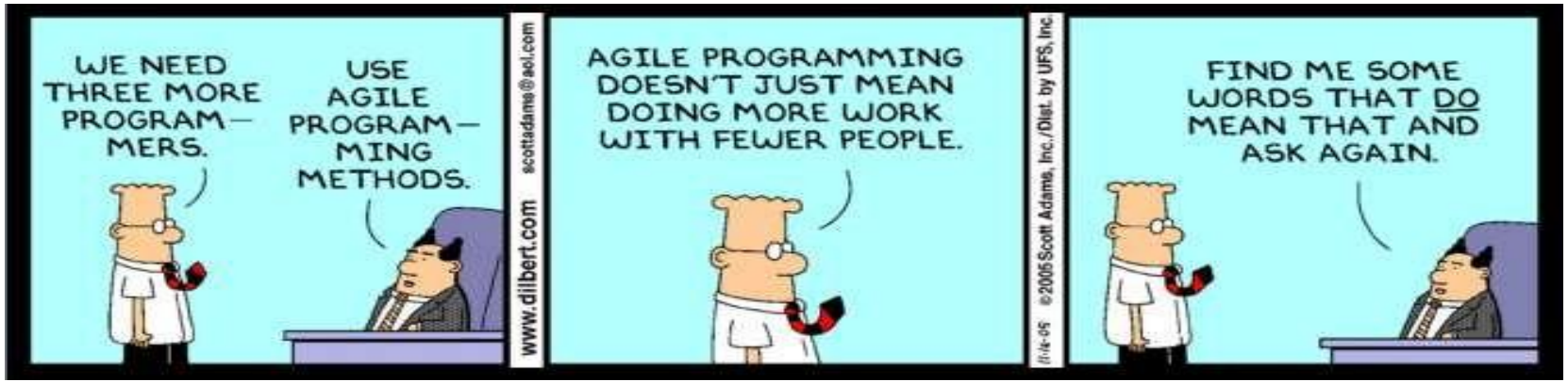
Artifacts

Product Backlog
Sprint Backlog
Burndown Chart

Meetings

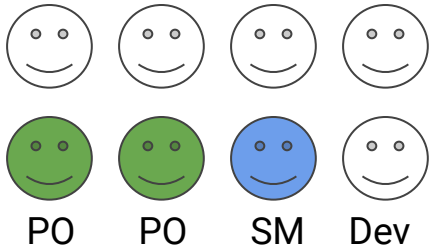
Sprint Planning
Daily Standup
Sprint Review
Sprint Retrospective

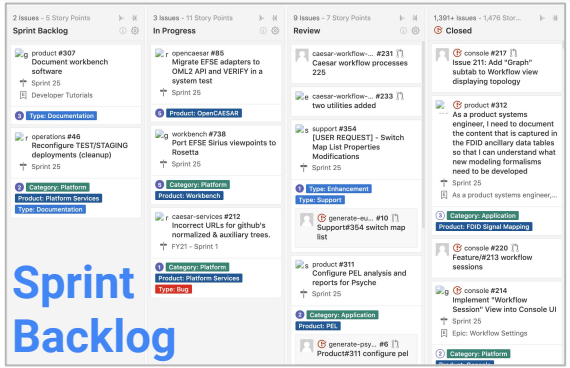
No Silver Bullet

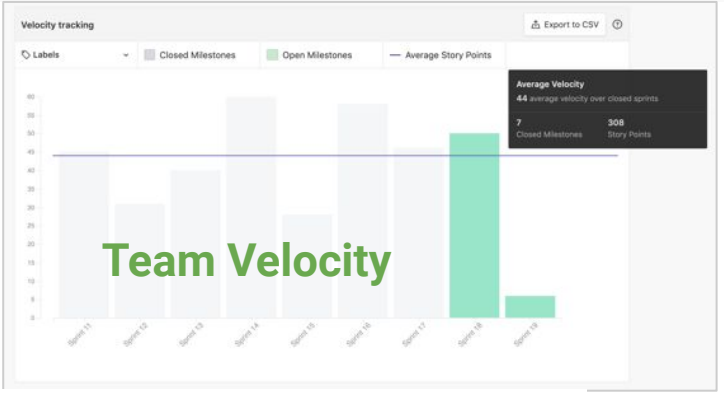


Scrum Process Quiz

Case Study: IMCE Scrum Team





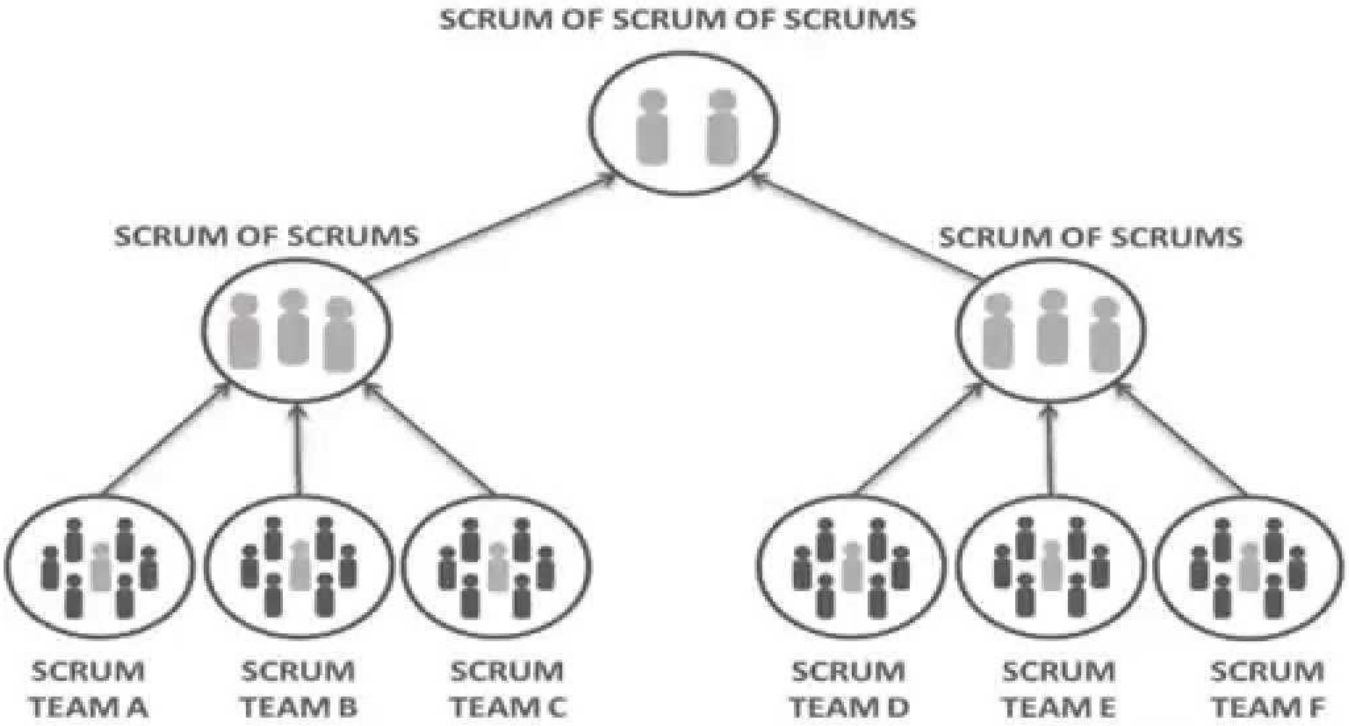


2-Week Sprint

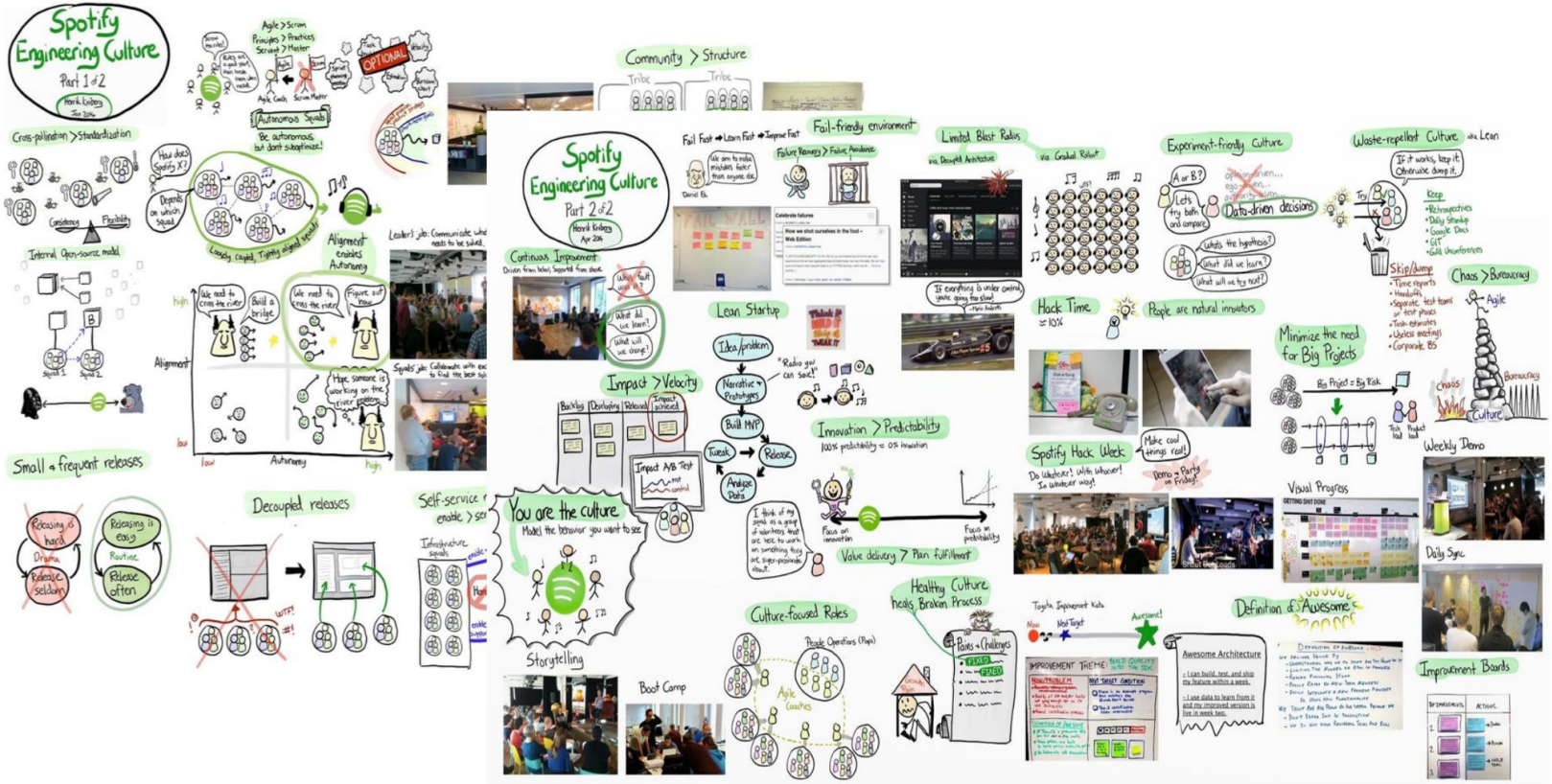
- Strategic Planning Meeting (day 1)
- Tactical Planning Meeting (day 1)
- Daily Standup Meeting (every morning)
- Sprint Review & Retro Meeting (day 10)

Sprint Metrics

Scaling Up


SCRUM OF SCRUM OF SCRUMS
SCRUM OF SCRUMS
SCRUM OF SCRUMS
SCRUM TEAM A
SCRUM TEAM B
SCRUM TEAM C
SCRUM TEAM D
SCRUM TEAM E
SCRUM TEAM F

Case Study: Agile Process at Spotify



References

- 8 Software Development Models
- Agile Manifesto
- Software Process Models:
- Scrum Process
- Agile at Spotify 1
- Agile at Spotify 2
- How to follow scrum on Github (Important)

Before Next Class

- Read up on UML diagrams (https://www.uml-diagrams.org/)
  - Focus mostly on:
    - Use Case Diagram
    - Activity Diagram
    - Class Diagram
    - Sequence Diagram
    - State Machine Diagram
  - Get familiar with PlantUML (http://www.plantuml.com/)