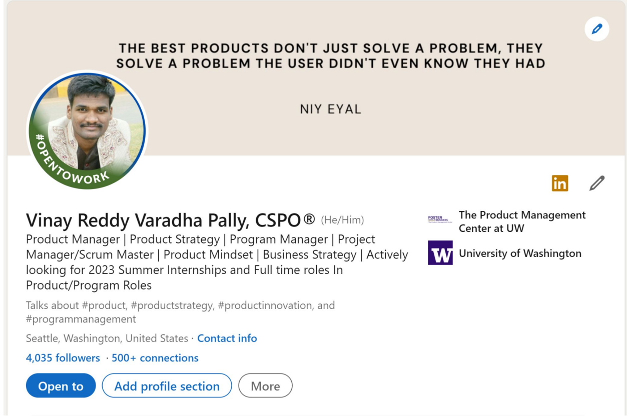
CSE 403

Software Engineering

Spring 2023

#7: Scrum and Teams

About Me: Vinay Reddy


THE BEST PRODUCTS DON'T JUST SOLVE A PROBLEM, THEY SOLVE A PROBLEM THE USER DIDN'T EVEN KNOW THEY HAD

NIY EYAL

Vinay Reddy Varadha Pally, CSPO® (He/Him)

Product Manager | Product Strategy | Program Manager | Project Manager/Scrum Master | Product Mindset | Business Strategy | Actively looking for 2023 Summer Internships and Full time roles In Product/Program Roles

Talks about #product, #productstrategy, #productinnovation, and #programmanagement

Seattle, Washington, United States · Contact info

4,035 followers · 500+ connections

The Product Management Center at UW

University of Washington

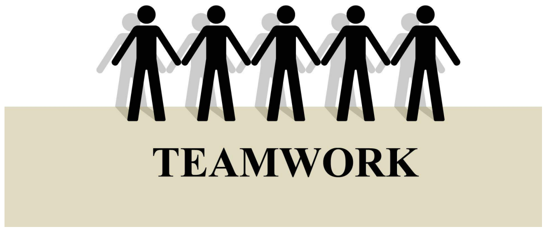
- Incoming Senior Product Manager Intern at IBM, San jose, CA for Summer 2023
- 3+ years of work experience, started as a developer and pivoted into Product Manager at few startups in India
- Certified Scrum Product Owner (CSPO).
- Master's in Information Management at the University of Washington, Information School
- Bachelors in Computer Science and Engineering

Linkedin: https://www.linkedin.com/in/varadhapallyvinay/

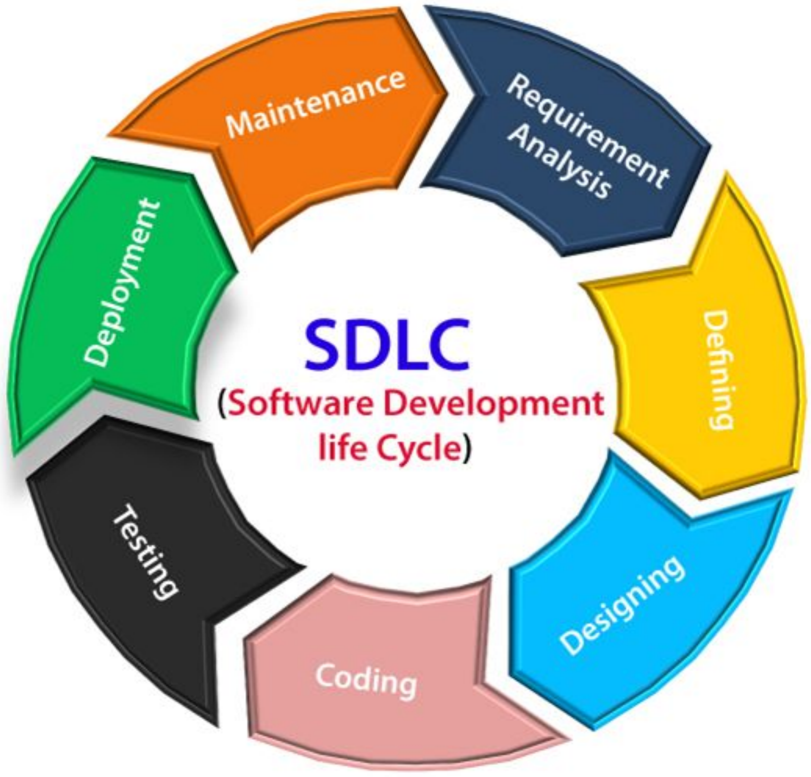
Today

- Project Development Methodologies (why? what?)
- Agile Manifesto
- Scrum

- Working in Teams



Software Development Lifecycle



Why do we need project development methodology?

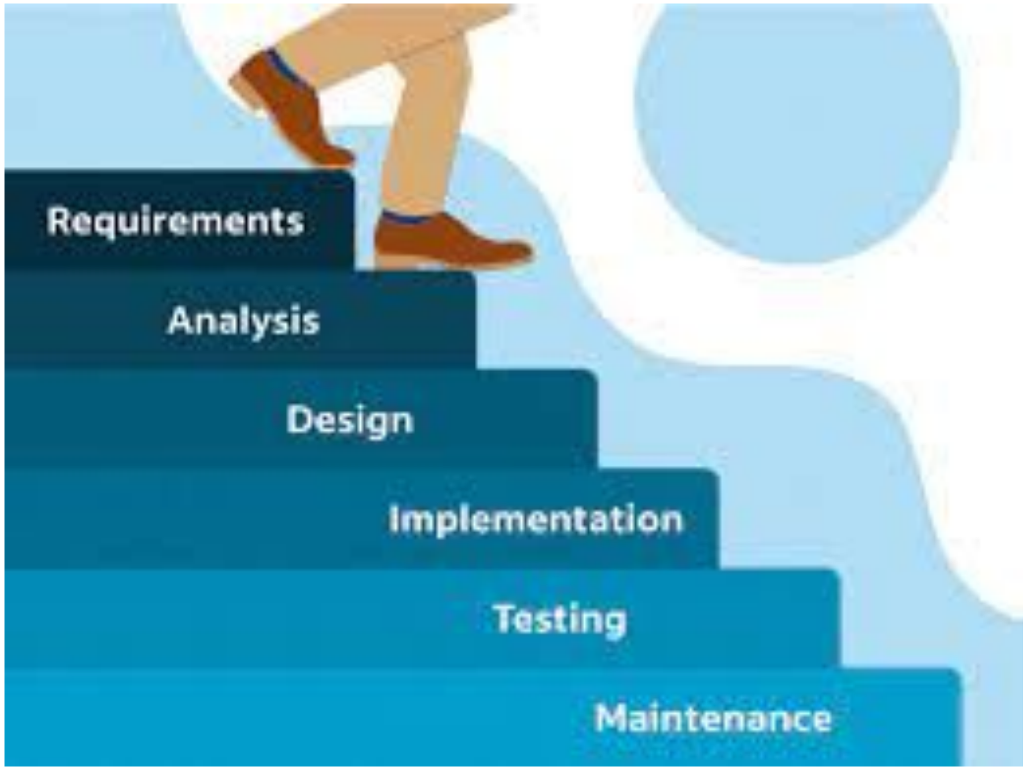
Project development methodologies provide a structured approach to software development that helps ensure that projects are completed on time, within budget, and to the required quality.

Project Development Methodologies

- Water Fall
- Agile Software Development
  - Scrum
  - Lean
  - Kanban

Waterfall Method

The Waterfall model is a sequential, linear approach to software development that divides the process into distinct phases, with each phase being completed before moving on to the next.



Problems in Waterfall models

- Rigid and Inflexible
- Limited Customer Involvement
- Increased Risk
- Limited Testing
- Long Delivery Time
- Lack of Adaptability

Agile Manifesto

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

  That is, while there is value in the items on the right, we value the items on the left more.

https://agilemanifesto.org/

Why Agile?

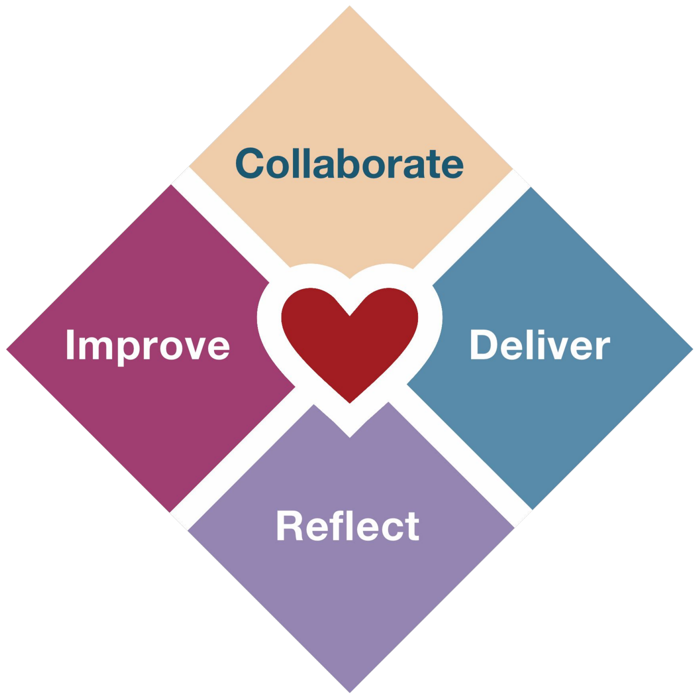
- Flexibility
- Customer Focus
- Collaboration
- Faster Time-to-Market
- Quality Control
- Continuous Improvement

What is Agile?

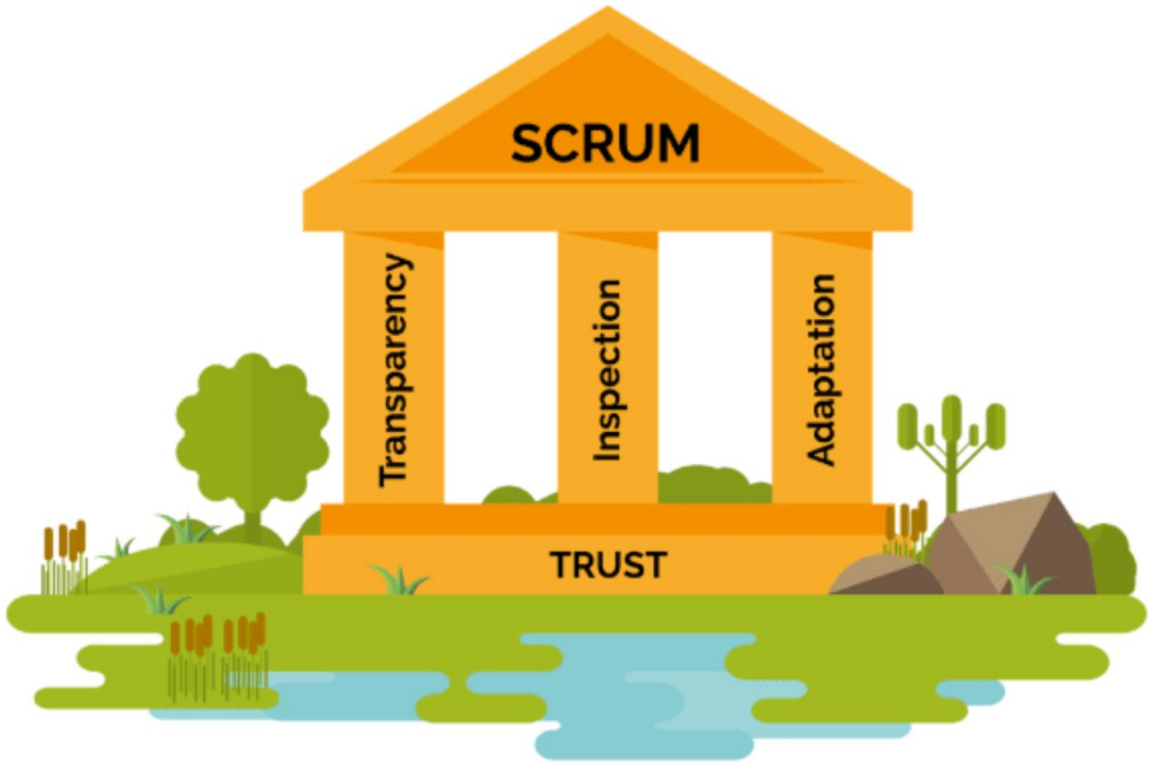
Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches

Scrum: overview

Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems (Scrum Guide)



Scrum Values



COURAGE
Scrum Team members have courage to do the right thing and work on tough problems

FOCUS
Everyone focuses on the work of the Sprint and the goals of the Scrum Team

COMMITMENT
People personally commit to achieving the goals of the Scrum Team

RESPECT
Scrum Team members respect each other to be capable, independent people

OPENNESS
The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work

Credit: ABN AMRO Bank N.V.

Scrum Team

Development Team: Responsible for building the product increment during each sprint. It includes software developers, testers, and other specialists who are needed to build the product.

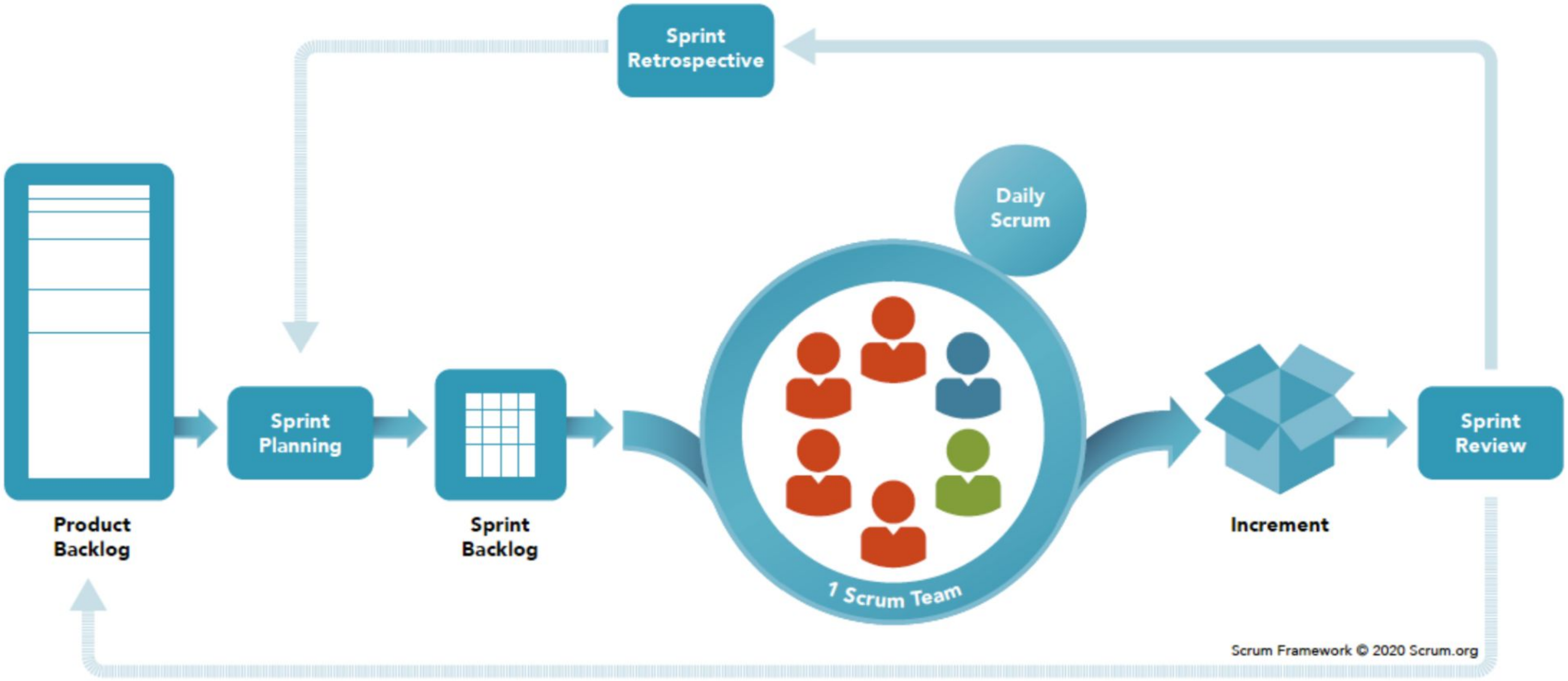
Product Owner: Product Owner owns the product backlog and is responsible for prioritizing the items on it, writing clear and concise user stories, and making sure the product increment meets the Definition of Done.

Scrum Master: The Scrum Master is a facilitator who helps the team to understand and implement Scrum practices, removes impediments to progress, and coaches the team to continuously improve.

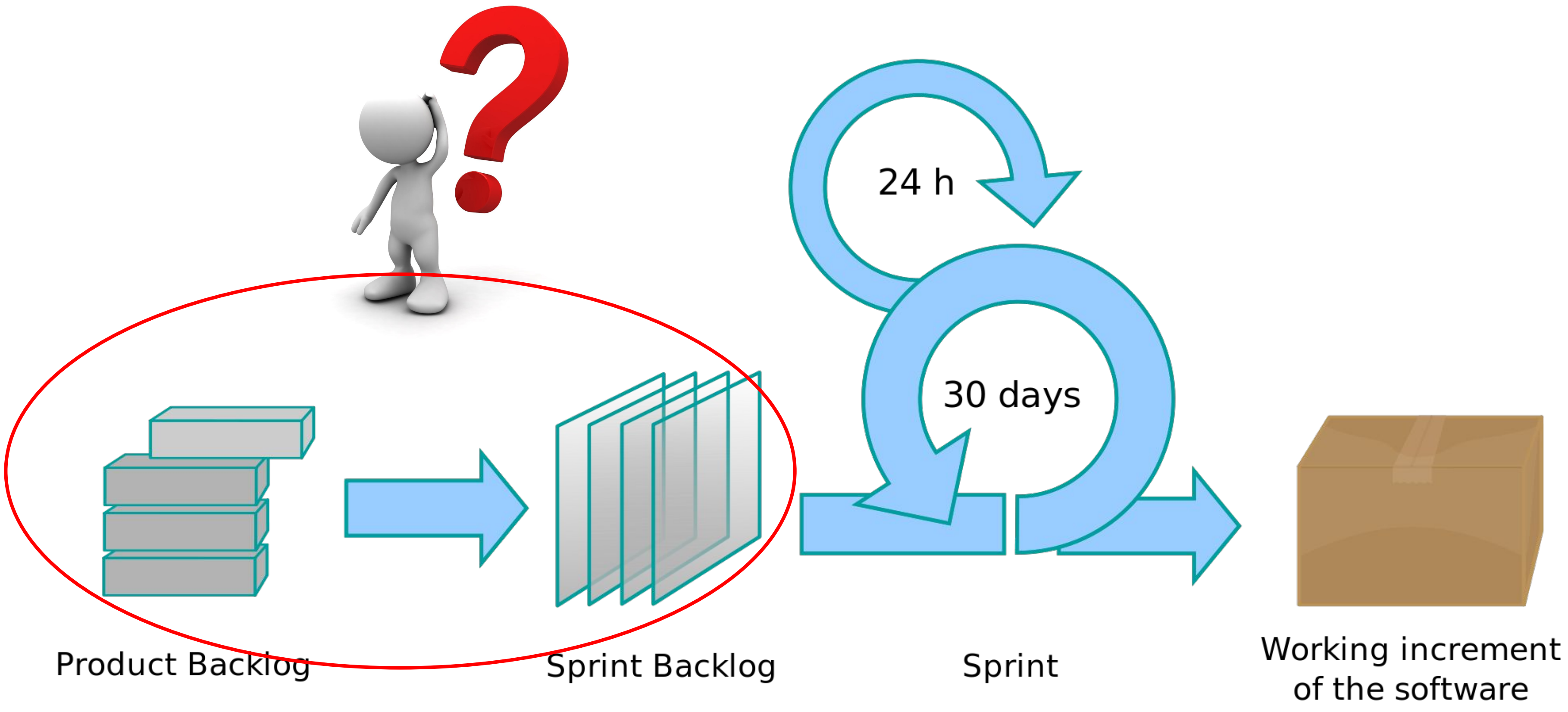
Scrum Ceremonies

1. Sprint Planning
2. Daily Standup
3. Sprint Review
4. Sprint Retrospective

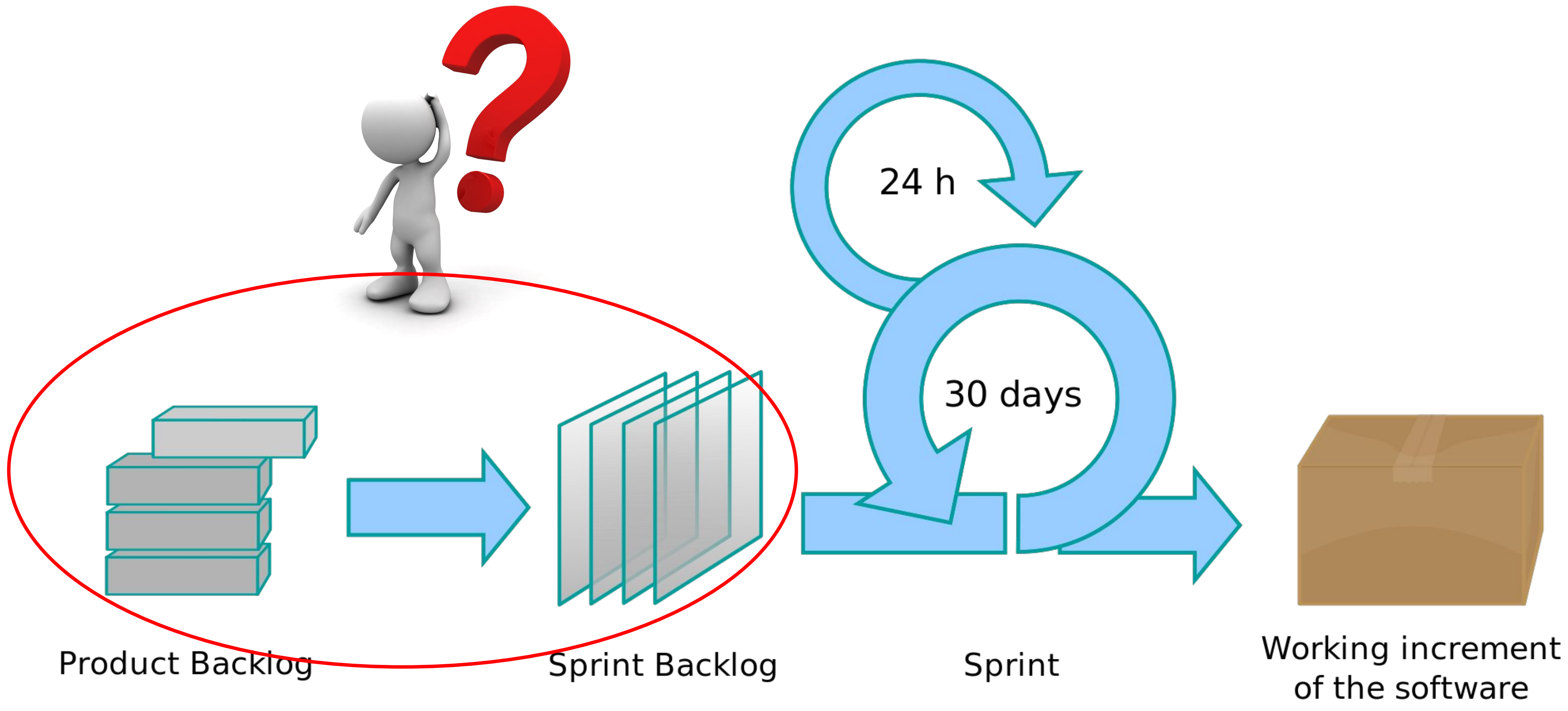
Scrum: overview



Scrum: overview



Scrum: overview



Prioritization:

- Must have vs. Should have vs. Could have vs. Won’t have

Product and Sprint Goal

Product Goal: It describes a future state of the product which can serve as a target for the Scrum Team to plan against. The Product Goal is in the Product Backlog.

Product Goal: Our goal is to make the customer onboarding experience easy and simple..

Sprint Goal - The Sprint Goal is the single objective for the Sprint.

Sprint Goal: In this two-week sprint, our goal is to integrate the social logins such as google and facebook login for customers to complete a purchase."

Use case example

Use case: User would like to go login to the platform

Actor: User

Flow: User needs to enter the username and password and check/read the terms and conditions.

Alternative Flow: If the user does not enter the password with the specified way such as 8 digits, special characters and others, we need to throw an error message.

Sprint planning

Sprint Planning addresses the following topics:

Why is this Sprint valuable?

What can be Done this Sprint?

How will the chosen work get done?

Attendees: Development team, scrum master, product owner

When: At the beginning of a sprint.

Duration: Usually around one hour per week of iteration. e.g. a two-week sprint kicks off with a two-hour planning meeting.

Daily Standups

The Daily Scrum is a 15-minute event for the Developers of the Scrum Team

What did I accomplish yesterday?

What am I planning to work on today?

Are there any obstacles or roadblocks that are preventing me from making progress?

Sprint Review

The Developers discuss what went well during the Sprint, what problems it ran into, and how those problems were solved

The Developers demonstrate the work that it has “Done” and answers questions about the Increment;

Attendees: Development team, scrum master, product owner

When: At the end of a sprint.

Duration: Typically 45 minutes per week of iteration - e.g. a 90-minute retrospective after a two-week sprint.

Sprint Retrospective

The team reflects on what went well during the sprint and what could have gone better, focusing on their processes and teamwork.

The team identifies specific actions they can take to improve their processes and teamwork in the next sprint.

The team creates a plan for implementing these improvements and assigns responsibilities for carrying them out.

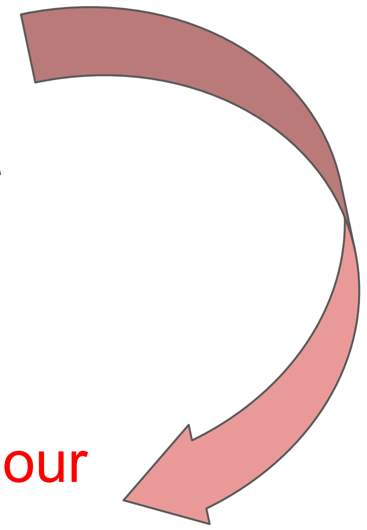
The team reviews and updates their Definition of Done to ensure that it reflects the changes they have made.

Scrum: overview

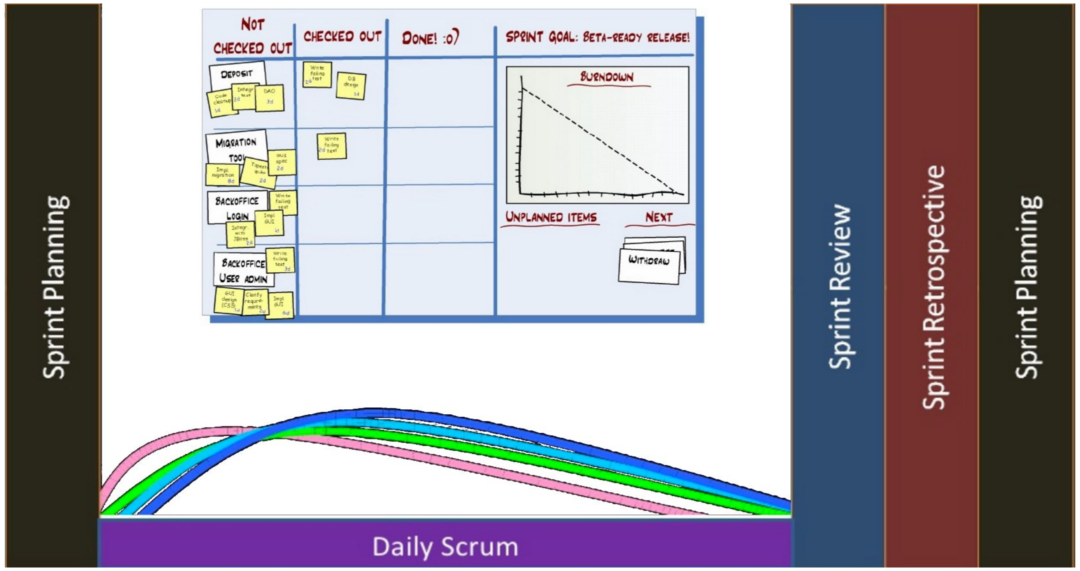
Small number of team members: 6 (+/- 2)

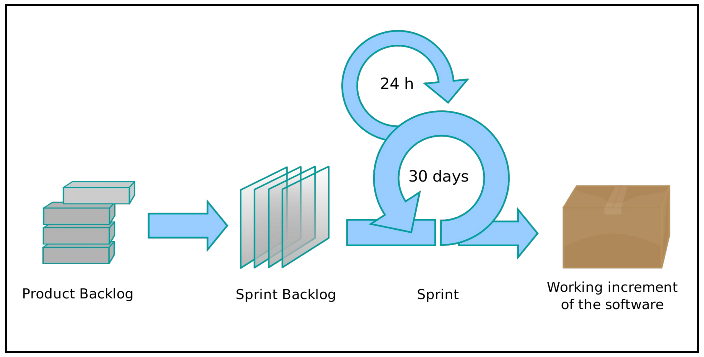
A time-boxed model:

- Each Sprint (time box): max 30 days
- Fixed number of tasks for each Sprint
- Daily Scrum meeting: 15 min max
- Each sprint results in a
  - Sprint review (product demo): 0.5-1 hour
  - Sprint retrospective (post-mortem): 1-3 hours



Scrum: activities and planning





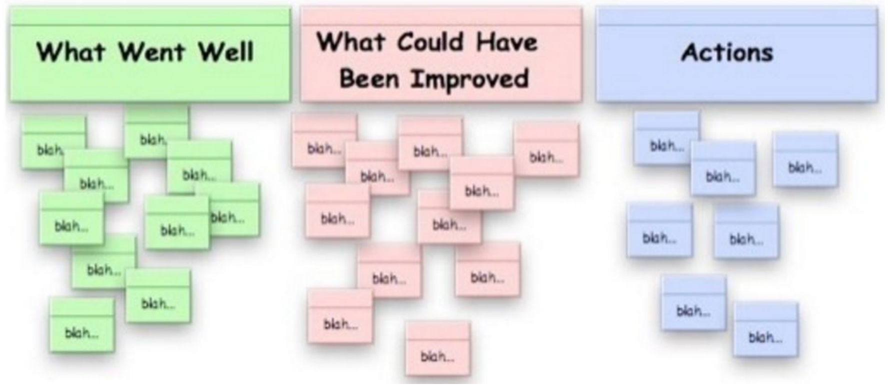
Daily scrum meeting (15min):

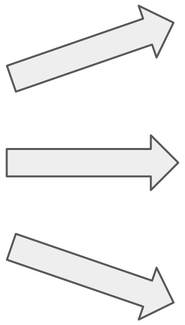
- What did I do since the last meeting?
- Any obstacles or blocking issues?
- What will I do until the next meeting?

Scrum: sprint retrospective

Who and what?

- Product owner, scrum master, and scrum team.
- Reflect, change, improve



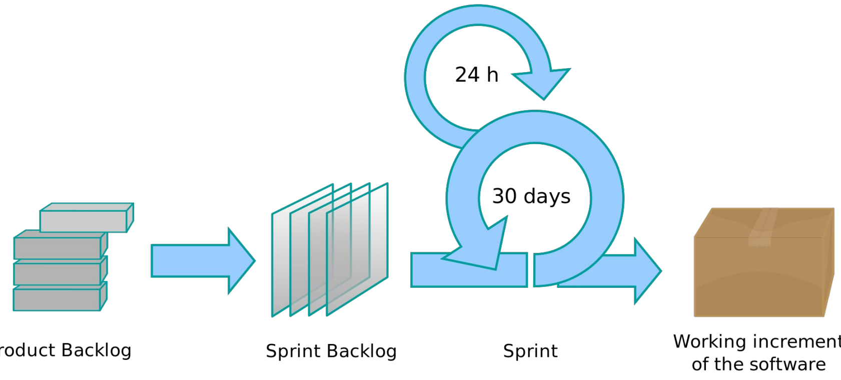


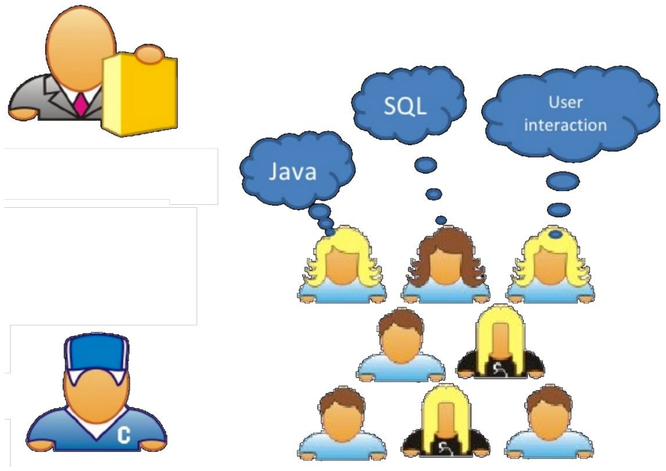
Stop doing

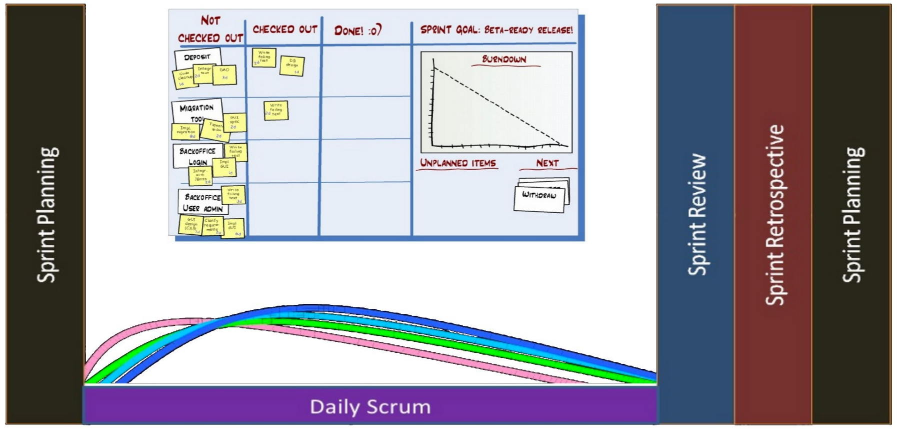
Continue doing

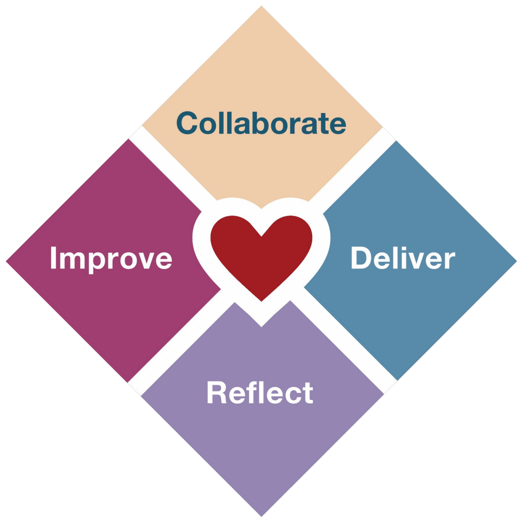
Start doing

Scrum: summary









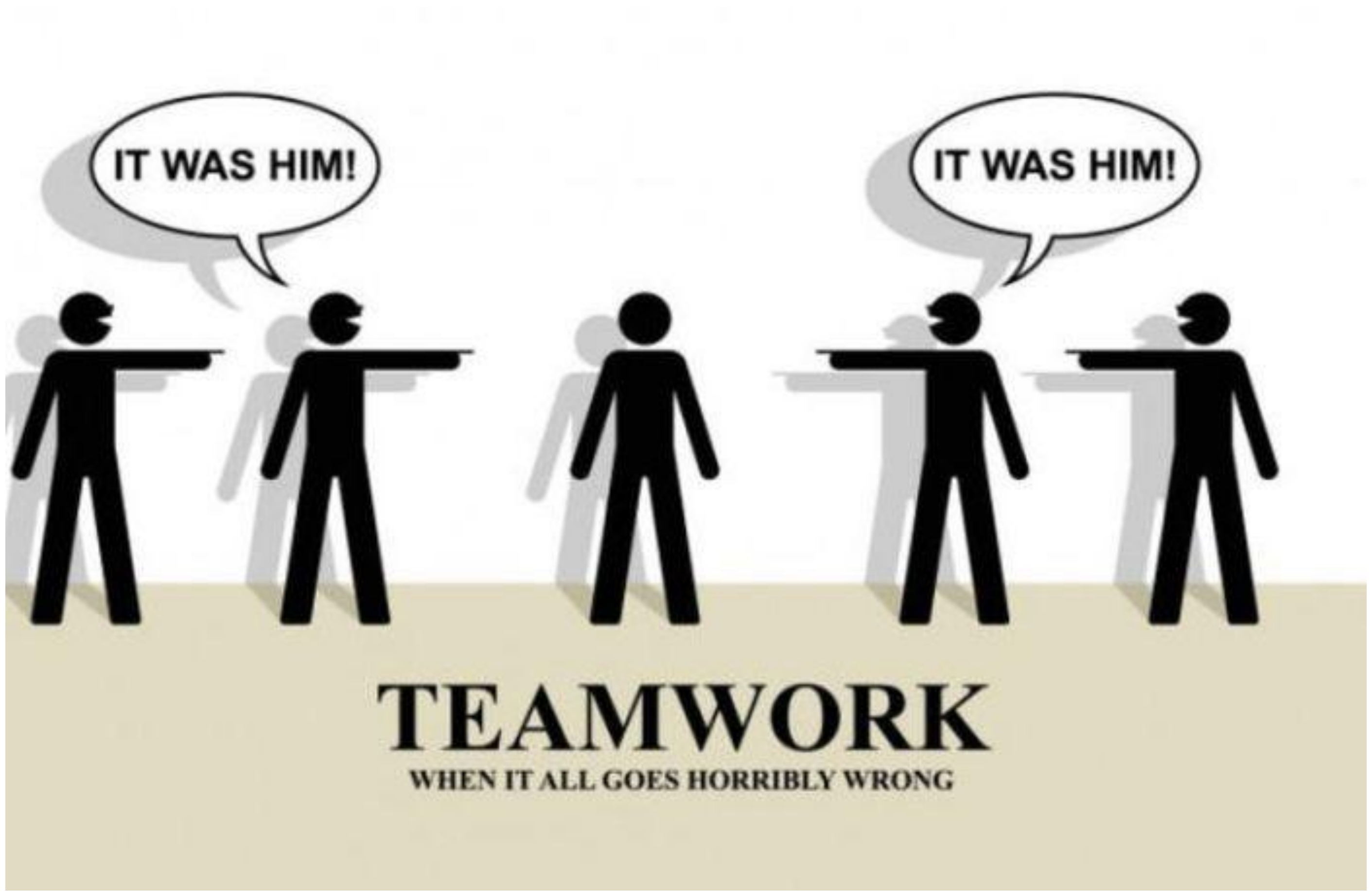
Scrum: discussion

Will you use Scrum for your project?

- **Yes** (describe why)
- **A variant** (describe your variant)
- **Need more infos** (state 1-3 specific questions)
- **No** (describe why not)

Working in Teams

Working in teams is great



Seriously, working in teams can be great!

Benefits

- Attack bigger problems in a short period of time
- Utilize the collective experience of everyone

Risks

- Communication and coordination issues
- Lack of planning, reflection, improvement
- Conflict or mistrust between team members

Big questions

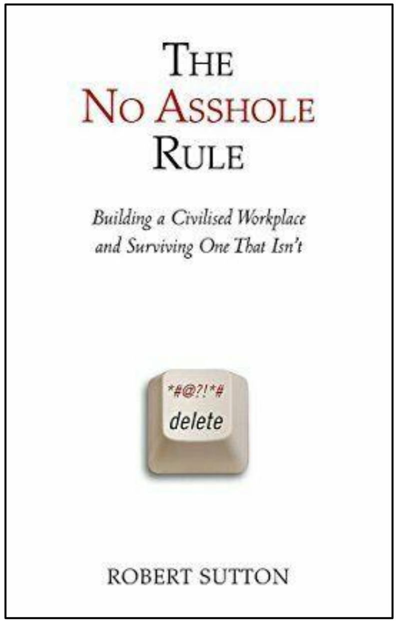
- **Communication:** How will everyone communicate?
- **Decisions:** How will your team make decisions?
- **Structure:** How do you **divide** your team **into subgroups**?

Big questions

- **Communication:** How will everyone communicate?
- **Decisions:** How will your team make decisions?
- **Structure:** How do you **divide** your team **into subgroups**?

Communication: powerful but maybe costly

- **Communication** requirements increase with increasing numbers of people (**everybody to everybody: quadratic cost**)
- Every attempt to communicate is a **chance to miscommunicate**
- **Not communicating will guarantee miscommunication**



Communication: example

"Hey *X*, I was wondering whether you finished the *Y* feature you were assigned? Since we were late on some features last time, I thought I'd check. When you have time, can you please tell me when *Y* is done. Thanks, *Z*."

What do you think about this email?

Communication: example

"Hey *X*, I was wondering whether you finished the *Y* feature you were assigned? Since we were late on some features last time, I thought I'd check. When you have time, can you please tell me when *Y* is done. Thanks, *Z*."

Be quantitative and specific:

- Use **specific, incremental goals**, not just things must be "done".
- List **specific dates** for when results are expected.
- **State requests** in a communication **explicitly**.
- **State an expected date/time** for a reply to a communication.
- **Remind** about upcoming **deadlines**, meetings, key points.
- **Don't be accusatory**; offer support and gratitude as appropriate.

Communication: example

"Hey *X*, I was wondering whether you finished the *Y* feature you were assigned? Since we were late on some features last time, I thought I'd check. When you have time, can you please tell me when *Y* is done. Thanks, *Z*."

A possibly better email:
"Hey *X*, how is your work on *Y* going?

It's due a week from Friday. Like we talked about at our last meeting, we are hoping to have the first 2 (out of 3) features designed by Sunday so we can review them together.

Please let me know by tomorrow night how much progress you made on *Y*. If you have any questions or need some help along the way, please let me know.

We'll all meet Saturday in person and you can give us another update at that time. Thanks, *Z*."

Big questions

- **Communication:** How will everyone communicate?
- **Decisions:** How will your team make decisions?
- **Structure:** How do you **divide** your team **into subgroups**?

Leadership and high-impact decisions

Who makes important product-wide decisions?

- One person?
- All by unanimous consent?
- Other options?

- Is this an **unspoken or** an **explicit** agreement?

Making decisions

- Delegate to subteams when possible
- Let everyone give their input (even if some is off-track)
- Write down pros/cons of alternatives
  - Evaluate cost/benefit/risks
  - How long will it take? How much to learn? etc.
- Have a clear procedure for resolving disagreement
  - Strive for consensus, but if it cannot be achieved, ...
  - Majority vote and PM decides on a tie, etc.
- Pareto: find 20% of work that solves 80% of a problem
  - Know what the real problem is!
- Document, Plan, Prioritize

Most importantly: compromise, compromise, compromise

Big questions

- **Communication:** How will everyone communicate?
- **Decisions:** How will your team make decisions?
- **Structure:** How do you **divide** your team **into subgroups**?

Common SW team responsibilities

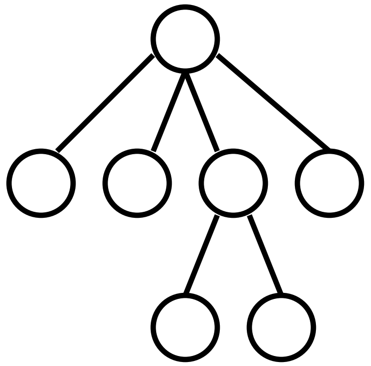
These following could be all different team members, or some members could span multiple roles:

- Project management
- Functional management
- Designers/architects
- Developers: programmers, testers, integrators
- Lead developer (“tech lead”)

Key: Identify and stress roles and responsibilities

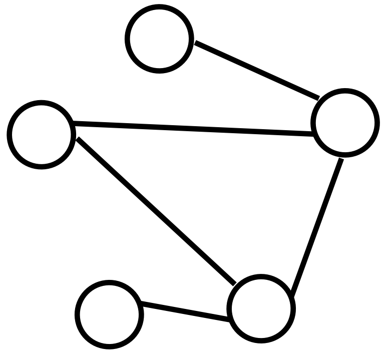
Team structure models

Dominion model



- Pros:
  - clear chain of responsibility
  - people are used to it
- Cons:
  - single point of failure at the top
  - little or no sense of ownership by everyone

Communion model



- Pros:
  - a community of leaders, each in their own domain
  - inherent sense of ownership
- Cons:
  - miscommunication, competing visions, dropped responsibilities
  - many points of partial failure