SDLs, scrum and teams

CSE 403 Software Engineering

Winter 25

Today's Outline

- Assignment 1 – Project proposal teams
- Software development lifecycles review
- Deeper dive on scrum (agile) teams
- Building a highly performing team

Reminder: see the required reading material on the course Calendar webpage

Assignment 1 - Project proposals

- Monday 1/13 - 11:59pm
  - Proposals due in Canvas (one submission per proposal-group)
  - You must define your proposal-group when submitting (one submission per group)

- Tues, Wed, Thurs
  - Pitches in class – **time-per-pitch updated to 3 minutes**
  - Staff will publish the order by noon Tues (see Ed Announcements for list)
  - You will "share" your deck over the class Zoom link – be ready

- Fri 1/17 – 12:30pm
  - Preferences survey due (see Ed Announcements for link)

Student preferences survey

1. Rank (highest to lowest) the projects you'd like to work on

   Top entry == project you'd most like to work on

2. [Optional] Identify one or two other students that you'd like to be on a team with

   Note 1: Your requests and ranking must match the other students

   Note 2: This may affect which project you'll be placed on as there will need to be space

How we form the project teams

1. Staff first select the set of projects, those:
   - That students have found most interesting (higher ranked) and
   - That we think will be successful in our quarter class and
   - That balance the types of projects done in the class, so that you can see a range of projects developed.

2. If a project is selected to go forward, then students who proposed that project have priority for it (assuming that they ranked it their top preference).

3. Next, we will place other students on the selected projects:
   1. We aim for groups of 5-6 students per project.
   2. We will try to assign you to a project with at least one student you have requested to work with, as long as that request was mutual. This will trump #2 above if needed.
   3. We will try to give you one of your top ranked projects. But, just as in the real world, you may not get your top choice.

Back to SDLC – lots of models

- Code and fix
- Waterfall model
- Prototyping
- Spiral model
- Staged delivery

Common stages

- Requirements
- Design
- Implementation
- Testing
- Release
- Maintenance

Let's try a poll in PollEV: https: //PollEv.com/cse403wi

What SDLC would you pick and why?

- A control system for anti-lock braking in a car
- A hospital accounting system that replaces an existing one
- An interactive system that allows airline passengers to quickly find replacement flights
- New innovative but tbd features for a social media app
- Your 403 class project

What SDLC would you choose?

0 surveys completed

0 surveys underway

A control system for anti-lock braking in a car

- Waterfall | Staged Delivery
- Prototyping
- Spiral
- Agile - XP | Scrum

A hospital accounting system that replaces an existing one

Waterfall | Staged Delivery

Prototyping

Spiral

Agile – XP | Scrum

An interactive system that allows airline passengers to quickly find replacement flights

- Waterfall | Staged Delivery
- Prototyping
- Spiral
- Agile - XP | Scrum

New innovative but tbd feature for a social media app

- Waterfall | Staged Delivery
- Prototyping
- Spiral
- Agile - XP | Scrum

Your 403 class project (ok to change)

Waterfall | Staged Delivery

Prototyping

Spiral

Agile - XP | Scrum

Today's Outline

- Assignment 1 – Project proposal teams
- Software development lifecycles review
- **Deeper dive on scrum (agile) teams**
- Building a highly performing team

Scrum – start with the team

Very popular model used in industry

Product Owner

- Owns the product and is responsible for defining and prioritizing features
- Signs off on the deliverables

Scrum Master

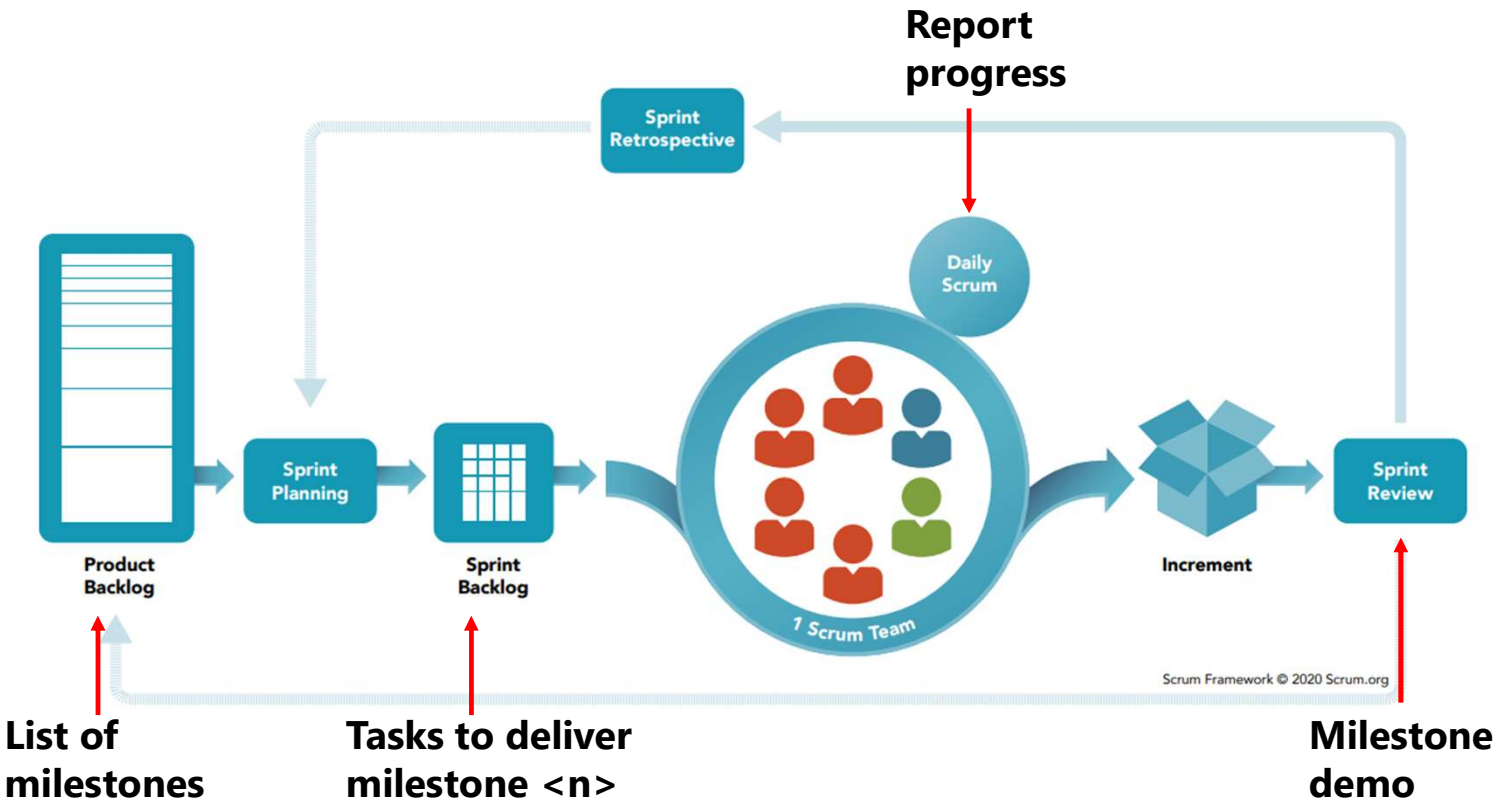
- Runs the scrum, removes blockers, and coaches the team to continuously improve

Development Team

- Develop, test, and evolve code

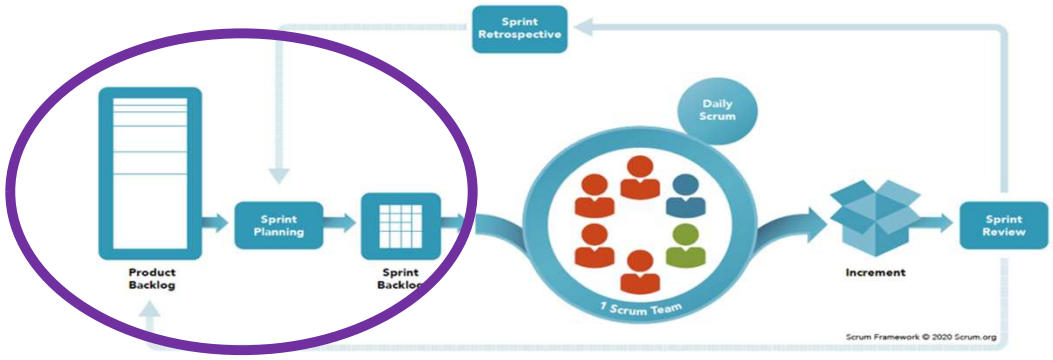
Scrum process

What aspects reflect the Agile philosophy?



https://www.scrum.org/

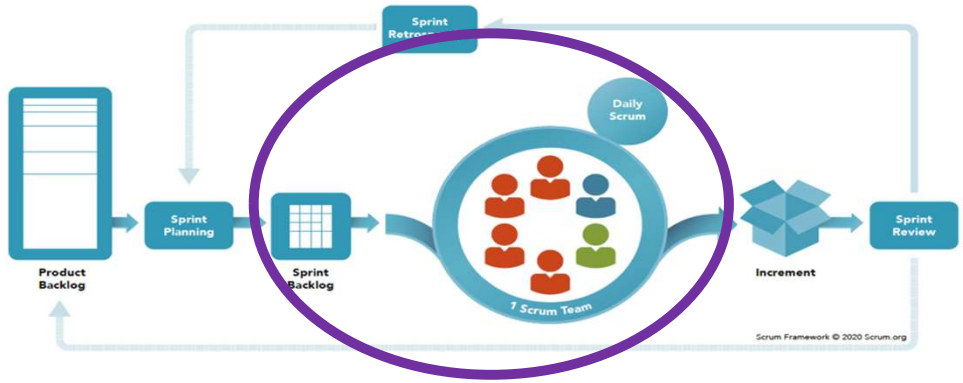
Key scrum activities



Sprint planning

- What features can be delivered this Sprint? (from product backlog)
- What tasks are needed to get these features delivered? (build sprint backlog)

Key scrum activities



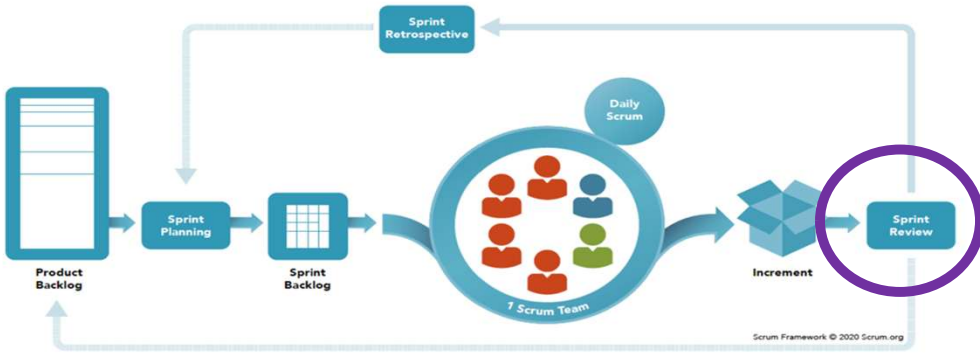
Sprint planning

- What features can be delivered this Sprint? (from product backlog)
- What tasks are needed to get these features delivered? (build sprint backlog)

Daily standup

- What did I accomplish yesterday?
- What am I planning to work on today?
- Are there any blockers that are preventing me from making progress?

Key scrum activities



Sprint planning

- What features can be delivered this Sprint? (from product backlog)
- What tasks are needed to get these features delivered? (build sprint backlog)

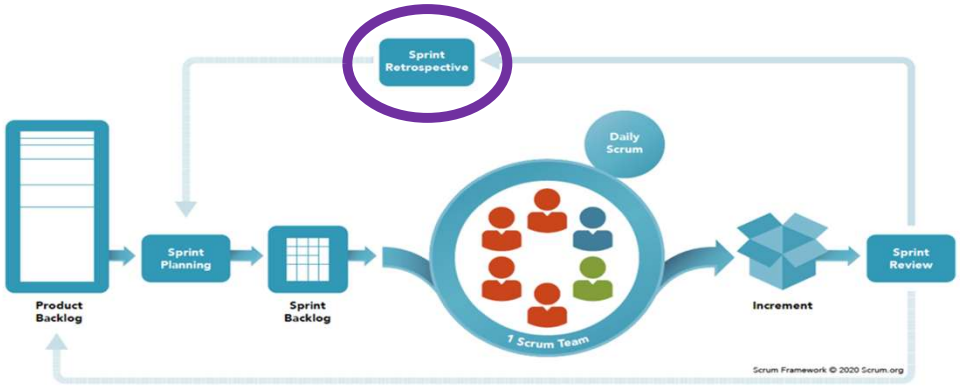
Daily standup

- What did I accomplish yesterday?
- What am I planning to work on today?
- Are there any blockers that are preventing me from making progress?

Sprint review

- Demo of working software to product owner

Key scrum activities



Sprint planning
- What features can be delivered this Sprint? (from product backlog)
- What tasks are needed to get these features delivered? (build sprint backlog)

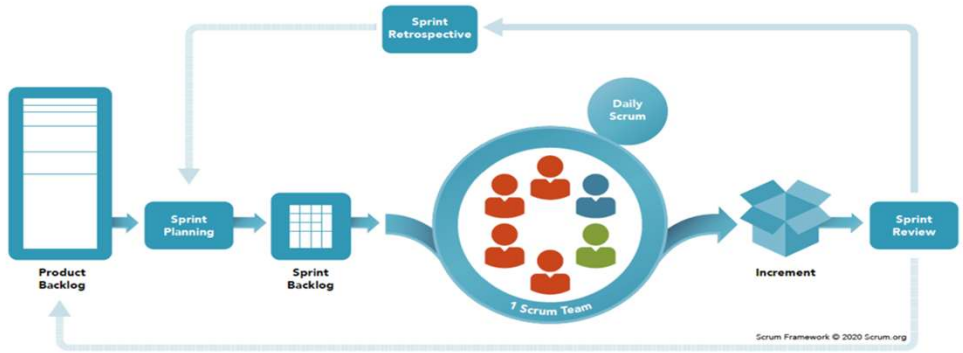
Daily standup
- What did I accomplish yesterday?
- What am I planning to work on today?
- Are there any blockers that are preventing me from making progress?

Sprint review
- Demo of working software to product owner

Sprint retrospective
- Reflect on what went well and what could have gone better
- Identify specific actions to improve processes and teamwork in the next sprint

Key scrum activities

(time boxed - e.g., biweekly sprints)



Sprint planning (30-60 min, biweekly)

- What features can be delivered this Sprint? (from product backlog)
- What tasks are needed to get these features delivered? (build sprint backlog)

Daily standup (15 min, daily)

- What did I accomplish yesterday?
- What am I planning to work on today?
- Are there any blockers that are preventing me from making progress?

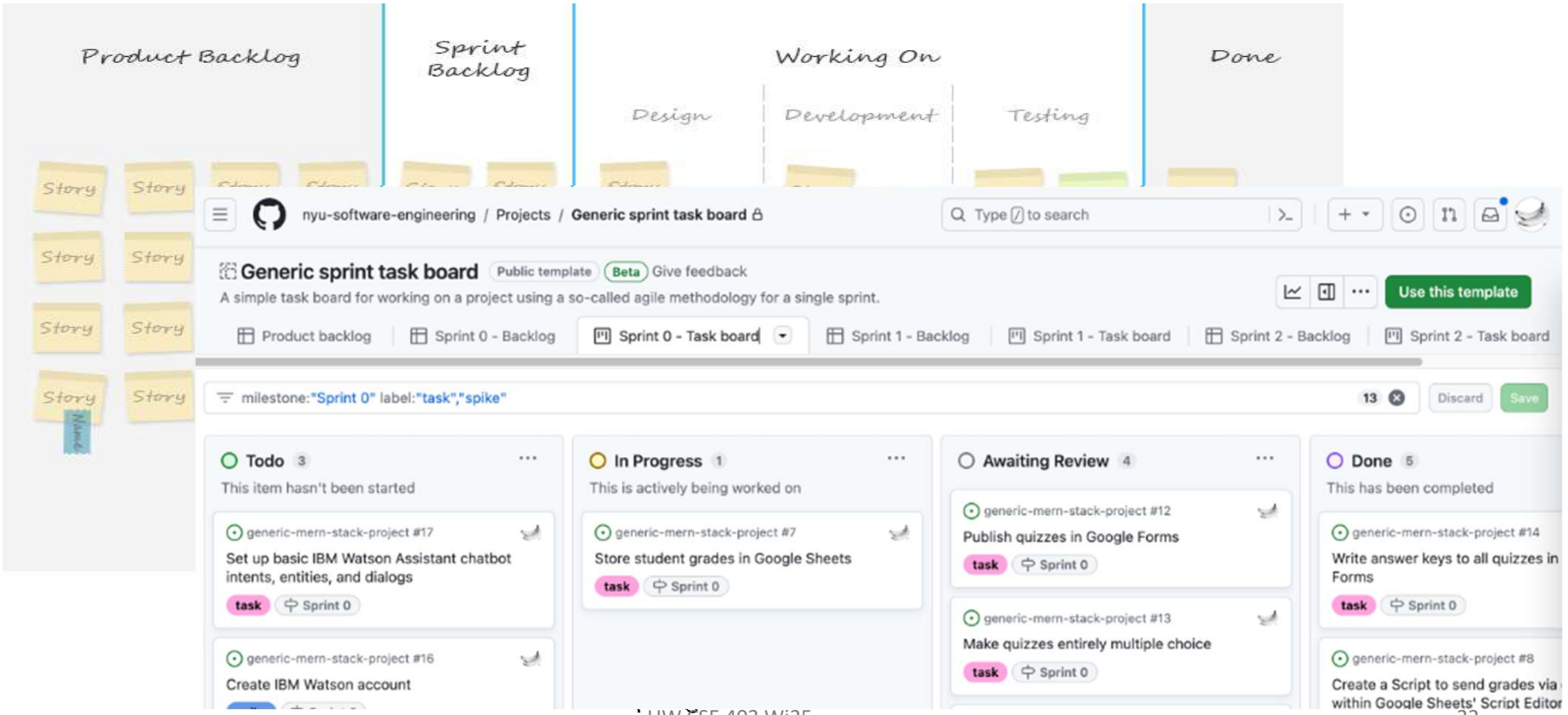
Sprint review (30-60 min, biweekly)

- Demo of working software to product owner

Sprint retrospective (30-60 min, biweekly)

- Reflect on what went well and what could have gone better
- Identify specific actions to improve processes and teamwork in the next sprint

Leverage scrum planning tools



See: https://knowledge.kitchen/content/courses/agile-development-and-devops/scrum/github-project-management

Shall we try it out?

Project Backlog	Sprint Backlog	In Progress	Done

• Proposal
• Requirements
• Architecture
• Beta release
  • Feature a
  • Feature b
  • Feature c
• Final release
  • Feature d
  • Feature e			

Sprint goal:
Proposal

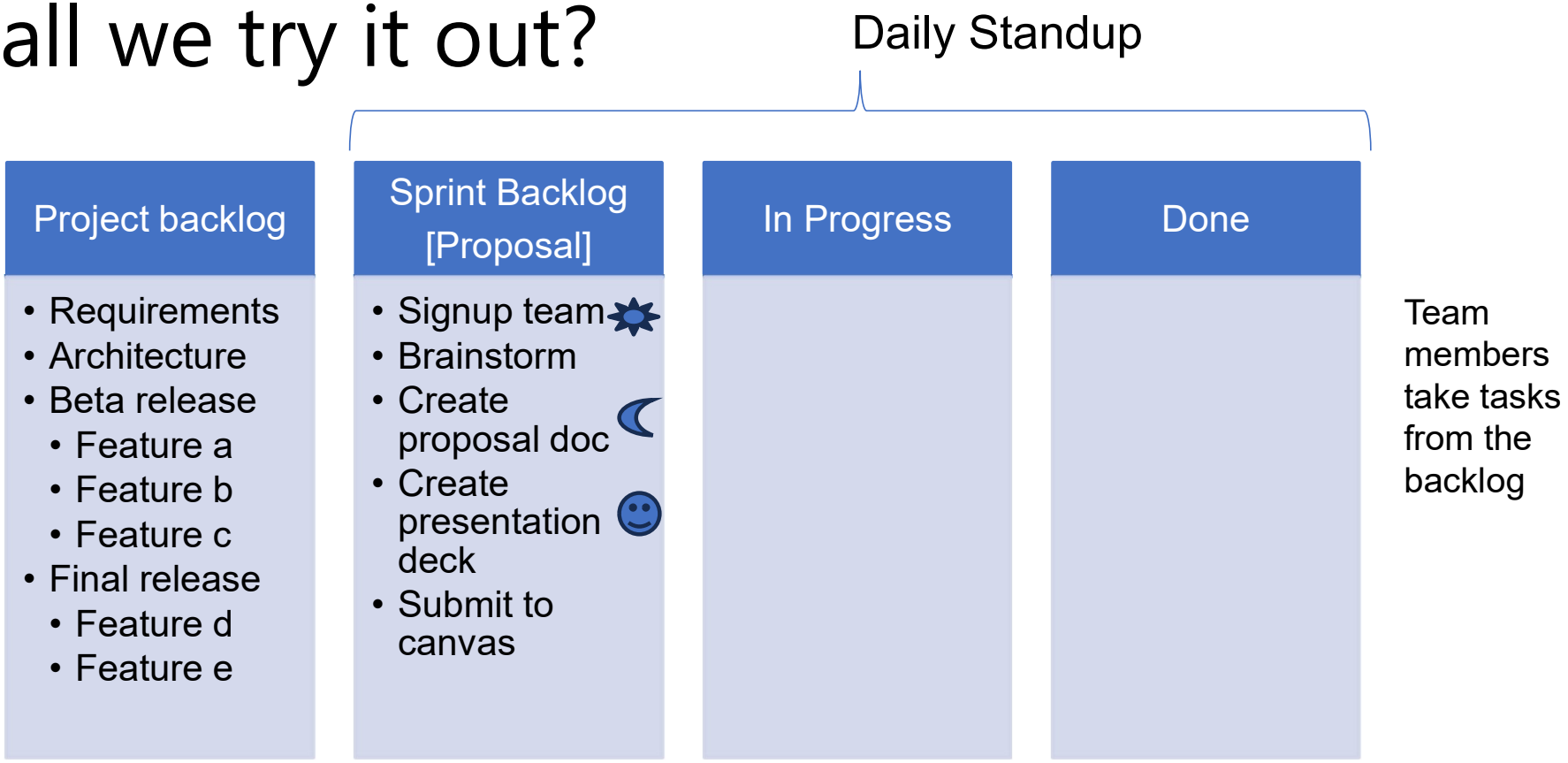
What tasks should we put in the Sprint backlog?

Shall we try it out?

Project backlog	Sprint Backlog **[Proposal]**	In Progress	Done

• Requirements
• Architecture
• Beta release
  • Feature a
  • Feature b
  • Feature c
• Final release
  • Feature d
  • Feature e
	• Signup team
	• Brainstorm
	• Create proposal doc
	• Create presentation deck
	• Submit to canvas		

Shall we try it out?



Shall we try it out?

Project backlog	Sprint Backlog	In Progress	Done

• Requirements
• Architecture
• Beta release
  • Feature a
  • Feature b
  • Feature c
• Final release
  • Feature d
  • Feature e
	• Submit to canvas	• Create proposal doc
		• Create presentation deck
			• Signup team
			• Brainstorm

Update board as part of the daily standup

Today's Outline

- Assignment 1 – Project proposal teams
- Software development lifecycles review
- Deeper dive on scrum (agile) teams
- **Building a highly performing team**

> *Talent wins games, but teamwork and intelligence wins championships.*
>
> *- Michael Jordan*

First, what is a team?

How does a **team** differ from a **group**?

First, what is a team?

How does a **team** differ from a **group**?

A group is a collection of individuals with a common interest

whereas

A team is a cohesive coalition of individuals working together towards a common goal

Another definition

A team is a set of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they hold themselves mutually accountable

Katzenbach and Smith

Why do we need to work in teams?

Common software team roles

These could be all different team members, or some members could span multiple roles, and/or it they can change over the life of the project:

- Project managers
- Designers/architects
- Dev leads ("tech leads")
- Devs: programmers, testers, integrators

- Project mgr vs product mgr vs program mgr?
- Group mgr vs dev lead?

Tip: definitions can vary with the company – learn the language early

Why do we need to work in teams?

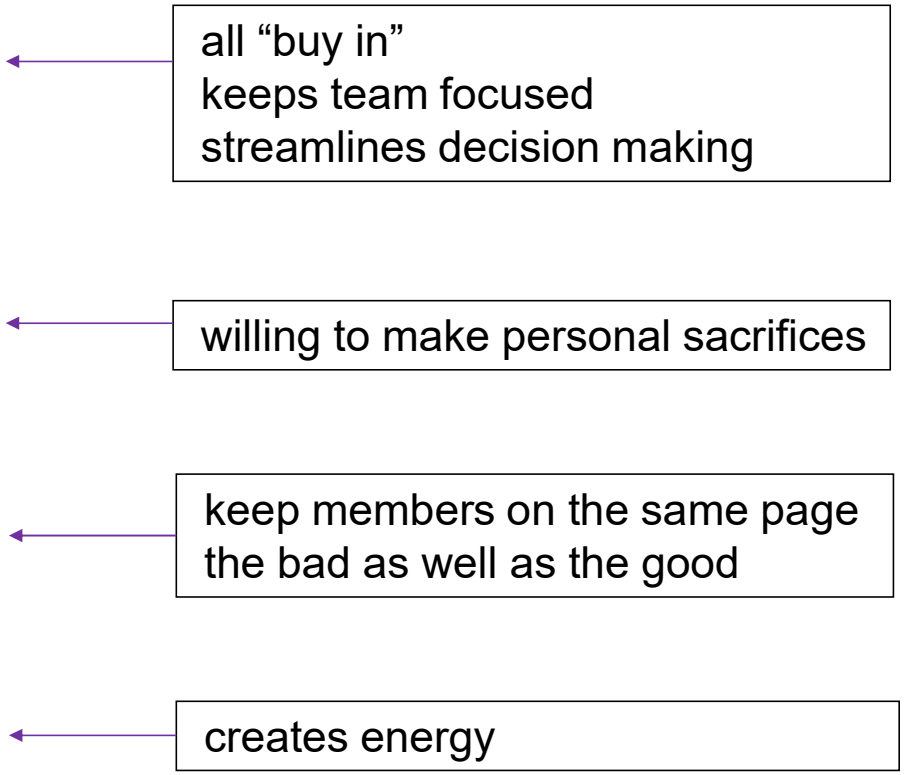
How do we become a high performing team?

According to research, high performing teams are 20% more productive and profitable, with 10% higher customer satisfaction.

Characteristics of a high performing team

The team has:

- A shared elevating vision or goal
- A sense of team identity
- A results-driven structure
- Competent team members
- A commitment to the team
- Mutual trust
- Interdependence among members
- Effective communication
- A sense of autonomy
- A sense of empowerment
- A high level of enjoyment



High performing 403 teams

Based on past classes, the high performing teams have:

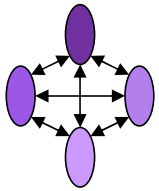
1. A clear shared vision (result) that they're excited about
2. Excellent and effective organization and communication
3. Strong collaboration and trust between members, open to ideas and adaptable
4. Individual and group responsibility/accountability to deliver

One more word on communication

Do not underestimate the power of communication!

Communication requirements increase with increasing numbers of people

- Everybody to everybody → $\frac{n(n-1)}{2}$
- Even just somebody to everybody → *n*-1



Better to over-communicate than under-communicate

- But be sure to communicate clearly

Summary: working in teams can be great!

Benefits

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

Challenges

- Communication and coordination
- Planning, reflection, improvement
- Trust and conflict resolution between team members