

Week 3 - Monday

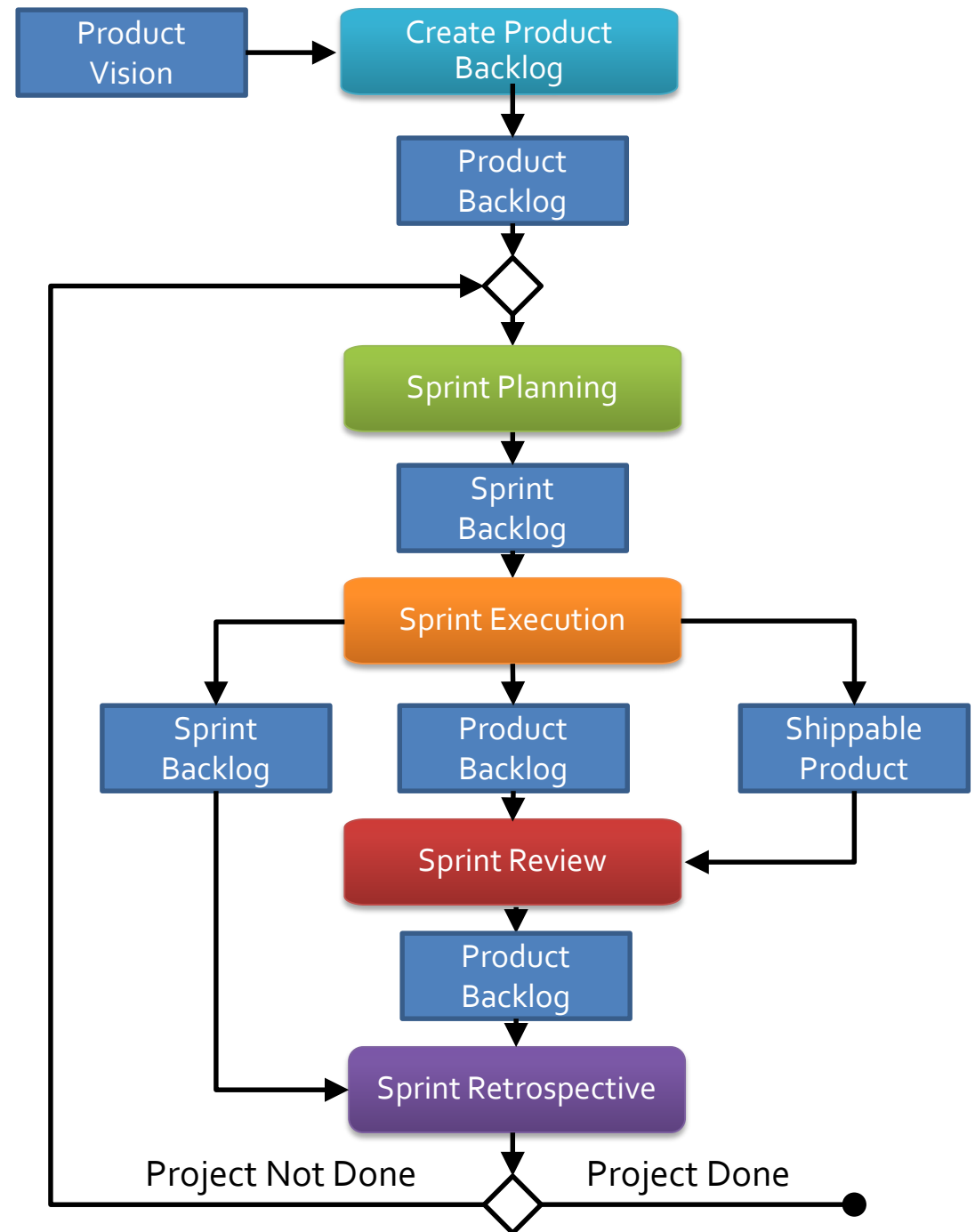
COMP 4100

Questions?

Scrum Review

Scrum process

- Scrum can be modeled with an activity diagram showing its steps
- Everything is built around a cycle called a **sprint**
- Because sprints repeat, the process is iterative
- Because each sprint produces a shippable product, the process is incremental



Sprints

- Agile methods are built around a product backlog, containing high-level descriptions of the desired features of the product
 - Items can be added to or removed from the product backlog at any time
- Some of the product backlog is chosen for a sprint
 - Making the **sprint backlog**
- The sprint backlog is implemented, making a new shippable product
- A **sprint review** allows customers to give feedback on the product
- The **sprint retrospective** is used to figure out how to do the next sprint better

Scrum roles

- **Product owner (PO)**
 - Responsible for what's in the product
 - Customer representative to the other developers
 - Updates the product backlog
- **Scrum master (SM)**
 - Guides the team through the Scrum process
 - Facilitator and coach
 - Protects the team from outside interference
- **Team members**
 - People who decide how to build the project and build it
 - Typically, everyone works on everything

Scrum artifacts

- **Product backlog**
 - A prioritized list of product features that haven't been implemented yet
 - **Product backlog items (PBIs)** are the elements of this list
 - Priorities are based on business value
- **Sprint backlog**
 - Subset of PBIs
 - Tasks needed to complete them
 - Estimates of effort needed for each one
- **Potentially shippable increment (PSI)**
 - Product that could be shipped to the customer (though maybe without all the desired features)
 - A PBI on the sprint backlog that wasn't finished goes back into the product backlog

Scrum activities

- **Product backlog creation**
 - The PO creates the product backlog for the first time, using customer input
- **Product backlog refinement**
 - The PO constantly adds and deletes PBIs from the product backlog based on feedback from stakeholders
- **Sprint planning**
 - The PO, SM, and other team members select PBIs, maybe with a particular sprint goal
 - PBIs are chosen by priority, taking into account how much can be done by estimating the work for the tasks for a PBI
- **Sprint execution**
 - Everyone performs the tasks to implement the sprint backlog PBIs
- **Sprint review**
 - A product demo where stakeholders discuss what was added and how they feel about it
 - Goal: improving the product
- **Sprint retrospective**
 - The team discusses what went well, what didn't, and how the next sprint can be better
 - Goal: improving the process

Managing the product backlog

- The product backlog is a prioritized list of PBIs
- Each PBI consists of
 - Specification
 - Priority
 - Estimate of effort
 - Acceptance criteria

PBI specifications

- PBI specifications can be less formal and more general than requirements in waterfall
- They could be traditional requirements statements, UI diagrams, use cases, user stories, bugs, design tasks, research tasks, etc.
- They start at broad levels of abstraction and are refined over time
- PBIs are refined into detailed, sprintable PBIs as needed, based on priorities
- Product backlogs should contain enough refined PBIs for two or three sprints

User stories

- **User stories** are the most popular way of specifying features in Scrum
- User story format:
 - As a *<user role>* I want to *<goal>* so that *<benefit>*.
- Examples:
 - As a course scheduler I want to determine whether students can take other sections of a course so that I can see if I can cancel a section with students already enrolled in it.
 - As a shopper I want to see whether an item is still on sale so that I can buy it more cheaply.
 - As an internet user I want to secure my devices so that I can protect my private information.
 - As an electric utility customer I want to see my usage over several years so that I can analyze it to budget my electricity costs more exactly.

User story abstraction

- User stories come at different levels of abstraction and size
 - Large, abstract stories that would take months of coding are sometimes called **epics**
 - Medium-sized stories that would take several sprints are sometimes called **features**
 - Small, detailed stories that can be done in a single sprint are **sprintable stories** (or simply stories)
- Even sprintable stories usually aren't detailed enough to implement without additional conversations with stakeholders

PBI priorities

- In addition to the specification of functionality, every PBI should have a priority
- Priorities express how important the PBI is and can be expressed as a number or a rubric (low, medium, high, critical)
- The PO sets the priorities based on stakeholder feedback
- Dependencies also determine priorities: If X is needed for Y, then the priority of X must be at least as high as Y
- High-priority PBIs should be small enough to do in a single sprint

PBI effort estimates

- Each PBI must have an effort estimate
- High-priority, sprintable PBIs need precise estimates (such as person-days), to aid in sprint planning
- Low-priority, abstract PBIs are further from sprintable status and only need rough estimates (small, medium, large, gigantic)
- As PBIs are refined, their effort estimates need to become more precise

PBI acceptance criteria

- How do we know when a PBI is done?
- Acceptance criteria are checks a user can do to see if a PBI is finished and correct
- Often, these form a test suite used by developers
- Following the same pattern of steady refinement, high-priority PBIs should have detailed acceptance criteria
 - These acceptance criteria might be further refined during the sprint

Product backlog refinement

- **Refining or grooming** the product backlog means:
 - Adding, removing, or modifying PBIs
 - Making PBIs nearing the top of the product backlog more detailed
 - Re-estimating and re-prioritizing PBIs
 - Adding acceptance criteria to PBIs
- Refinement happens during sprint review
- It should happen at least once during a sprint to make sure there are enough sprintable stories for the next sprint
- POs will be managing the product backlog through Trello

Estimating work and timeline

- Two pieces of information are needed: The size of the job and the speed of the team
- PBIs are estimated by **story points** or **ideal hours**
- One or two story points is supposed to be how much effort the smallest stories take
 - Bigger stories are estimated relative to that size
- An ideal hour or a person hour is the amount an average developer can accomplish in one uninterrupted hour of work
- Story points are more commonly used, since they're easier to estimate

Velocity

- **Velocity** is the amount of work done per sprint
- After a sprint, story points or ideal hours can be added up to see how much got done
- Past velocities can be used as a guide for how many story points can get done when planning the next sprint
- Ideally, tracking this information will help get better estimates of story points and ideal hours for other stories and also a better estimate of team velocity

Creating the sprint backlog

- In sprint planning, teams refine their estimates of high-priority PBIs before finalizing the sprint backlog
- For each PBI, they estimate the tasks involved in ideal hours
 - Story points can be used, but ideal hours are more precise
- The tasks can include:
 - Coding
 - Unit testing
 - Integration testing
 - Acceptance testing
 - Code inspection
 - Updating user documentation
- The final sprint backlog includes PBIs, their constituent tasks, and effort estimates for all tasks

Sprinting

- **Sprinting** is actually doing the implementation
- Sprinting is considered a **time-boxing** technique, where the amount of work done is based on the time available
 - Rather than letting time expand as needed to finish a task
- For a given project (and at a given company) sprints are usually the same length, somewhere between a week and a month
- Short, consistent sprints are easier to plan and track and give rapid feedback
- If PBIs can't be finished during a sprint, they go back on the product backlog
- If a team finishes all PBIs before the sprint is over, they can get another one from the PO

Definition of done

- What does *done* mean?
- Team have their own versions of done, often with the following items:
 - Design is complete and reviewed
 - Code is formatted and commented
 - Code has passed inspection
 - Code has passed PBI acceptance criteria (tests)
 - Code has passed all unit tests and regression tests
 - User documentation has been updated
 - Code has been integrated and passed all integration and systems tests
- When a PBI is truly done, it's removed from the product backlog

Effort estimation in Scrum

- Units of effort in Scrum are called **story points**
 - Story points are relative units
 - They're based on some of the smallest tasks, using them as a baseline of 1 story point
 - Everything is estimated relative to those
- Story points aren't used for epics since they're too big and abstract
- As PBIs get refined, their effort estimate gets refined too
- By the time they're sprintable, they need a relatively accurate story point estimate
- This means that there are good estimates for sprintable stories but no estimates for how much work the whole project will take

Detailed estimation in Scrum

- What if members of the team disagree on the story points needed for several stories?
- Agreement is needed for the sake of fairness and to plan how much work can actually get done in a sprint
- **Planning poker** is a way to bring the team to consensus about the relative difficulty of user stories
- Its goal is accuracy (ranking the stories by true difficulty) rather than precision (getting true estimates of how long things will take)
 - It's really hard to get true estimates, but it's good to know which stories take more work

Planning poker

- First, the team decides what numbers to use as estimates
 - Our cards: 1, 2, 3, 5, 8, 13, 21, ∞ , ?
- Planning poker has rounds
 - Each round estimates the effort for one PBI
 - Each team member throws in one card to show her effort estimation
 - If all cards match, the value is the estimate
 - If they don't match, the team discusses their estimates, focusing on the highest and lowest estimators
 - Repeat the round until consensus is reached
- It usually only takes a couple of rounds to reach consensus
- Estimates are usually pretty good because of discussion

Sprint review

- At the end of a sprint, there's a sprint review to reflect on how the product is changing
- All stakeholders are invited
- Sprint review outline:
 - Starts with the overall sprint goal and the PBIs in the sprint backlog
 - Team lists the PBIs completed and explains why some didn't get done
 - New aspects of the product are demonstrated
 - Everyone discusses how to make the product better
- Results of the review are used for planning the next sprint

Sprint retrospective

- At the end of a sprint, there's also a sprint retrospective
- Only the development team, including the PO and the SM, are invited
- The retrospective is for analyzing how the team is working and how to improve
- Improvements tend to be clear when a new team is working on a new product
 - It may still take several sprints for an improvement to get fully integrated into the process
- Over time, the team can become comfortable with the process, but finding improvement opportunities is still important

Other Scrum practices

- **Daily scrum:** Short daily meeting, often called a stand-up (having no chairs encourages brevity)
 - What did I do since the last meeting
 - What will I do today
 - What is impeding my progress
- **Story time:** Groom the product backlog
- **Cross-functional teams:** Get non-specialists to help with specialized tasks, to get the job done and expand skills
- **Sustainable pace:** Don't overwork
- **Planning poker:** Have team members contribute their time/work estimate for a PBI
- **Bidding:** Team members bid on tasks with ideal hours
- **Pair programming:** Two people sit together to code, with one typing (the driver) and the other checking (the navigator), switching off frequently

Trello recommendations

- At least have categories for:
 - Product Backlog
 - Sprint Backlog
 - Assigned Stories
 - In-Progress Stories
 - In-Testing Stories
 - Done
- Cards in the Product Backlog will often be broken down into smaller tasks (new cards)
- Cards should have a priority, an effort estimate, and note if they require another card to be done first

Web recommendations

- Most groups are making a webpage of one kind or another
- Webpages often have backends
 - Servers that generate the actual HTML, CSS, and JavaScript that web browsers view
 - Often this requires interacting with a database
- Webpages also have frontends
 - This is the art of designing the HTML, CSS, and JavaScript to look good and be responsive to user interaction

More web recommendations

- I recommend that groups creating a website use React for frontend work
 - It's an industry standard
 - There are millions of tutorials out there
 - It's not hard to make a good looking webpage
 - I can be more helpful if everyone is using a similar platform
- For those groups, I'll recommend Node.js for the backend
 - It's also an industry standard with lots of tutorials
 - Express is useful middleware
- If your site is filled with static content, you don't necessarily need a backend
- If you are dynamically scraping someone else's content, your backend will be integrated with that scraping tool

Upcoming

Reminders

- **Sprint 1 ends next Friday!**