Scrum

CEN 4010 Intro to Software Engineering

Professor Alex Roque

What is scrum? Scrum 101

- Scrum is an agile approach for developing innovative products and services.
- With an agile approach, you begin by creating a product backlog(prioritized list of the features)
- product backlog leads you always work on the most important or highest priority items first.
- Work itself is performed in short, timeboxed iterations or sprints
- Work that is not completed in an iteration gets reprioritized for upcoming iterations.

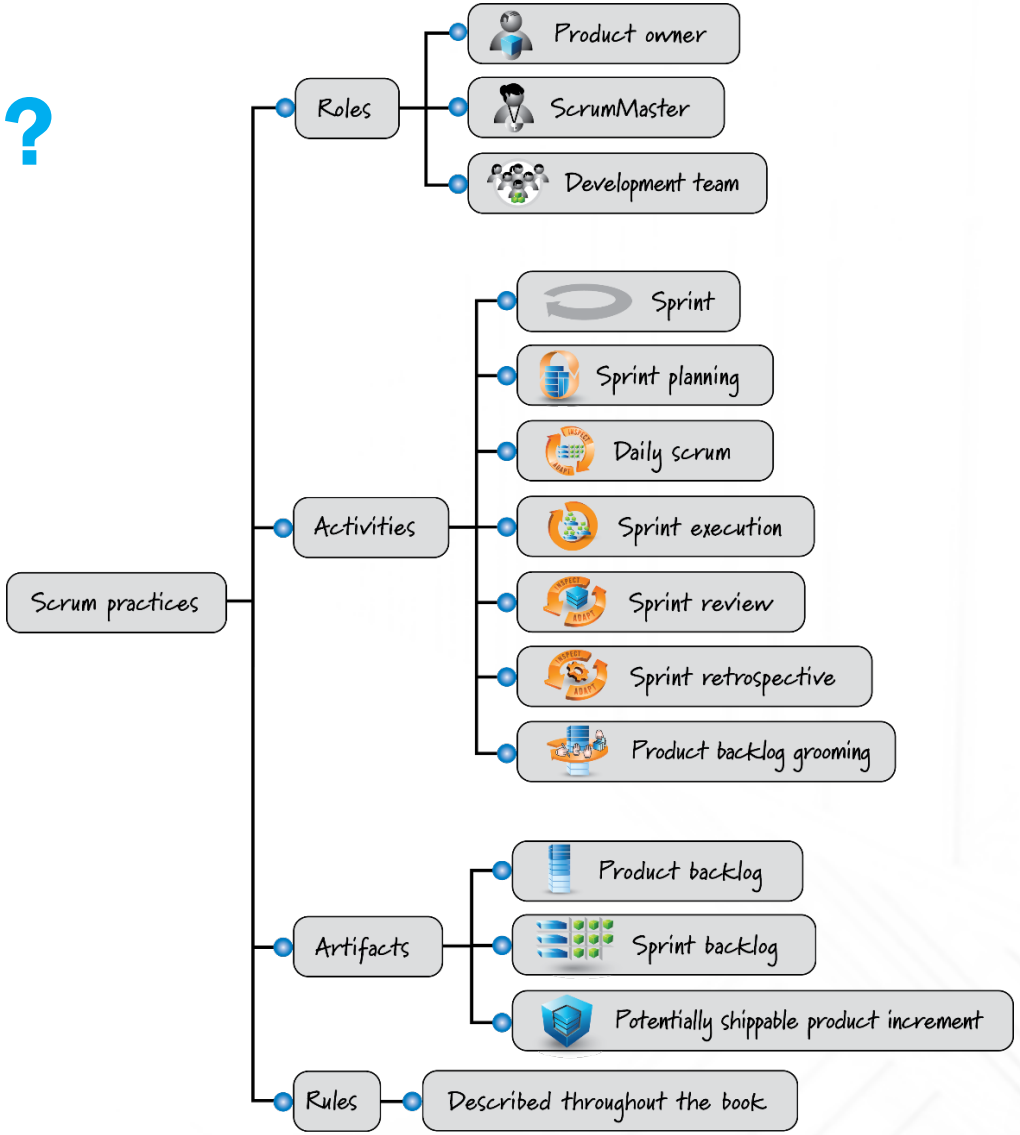
What is Scrum?

- Each iteration usually range from a week to a calendar month in length
- During each iteration, a self-organizing, cross-functional team does all of the work required to produce completed, working features that could be put into production. (such as designing, building, and testing)
- at the start of each iteration, the team plans which high-priority subset of the product backlog to create in the upcoming iteration.
- At the end of the iteration, the team reviews the completed features with the stakeholders to get their feedback.

What is scrum?

- Based on the feedback, the product owner and team can alter both what they plan to work on next and how the team plans to do the work.
- At the end of each iteration(or Sprint), the team should have a potentially shippable product (or increment of the product).
- If releasing after each iteration isn't appropriate, a set of features from multiple iterations can be released together.
- As each iteration ends, the whole process is begun anew with the planning of the next iteration.

What is Scrum?



Where did Scrum come from?

- Scrum's rich history can be traced back to a 1986 Harvard Business Review article, "The New New Product Development Game" (Takeuchi and Nonaka 1986).
- In 1993, Jeff Sutherland and his team at Easel Corporation created the Scrum process for use on a software development effort by combining concepts from the 1986 article with concepts from object-oriented development, empirical process control, iterative and incremental development, software process and productivity research, and complex adaptive systems.
- Though Scrum is most commonly used to develop software products, the core values and principles of Scrum can and are being used to develop different types of products or to organize the flow of various types of work.

The Agile Manifesto

- Created in 2001 by independent software practitioners:
  - Kent Beck, Mike Beedle, Arie van Bennekum, Alistai Cockburn, Ward Cunningham, Martin Fowler, Robert C. Martin, Steve Mellow,
  - Dave Thomas, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Ken Schwaber, Jeff Sutherland
- 12 Principles,
- 4 core Values

The Agile Manifesto

- We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
  - 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.

Scrum Benefits

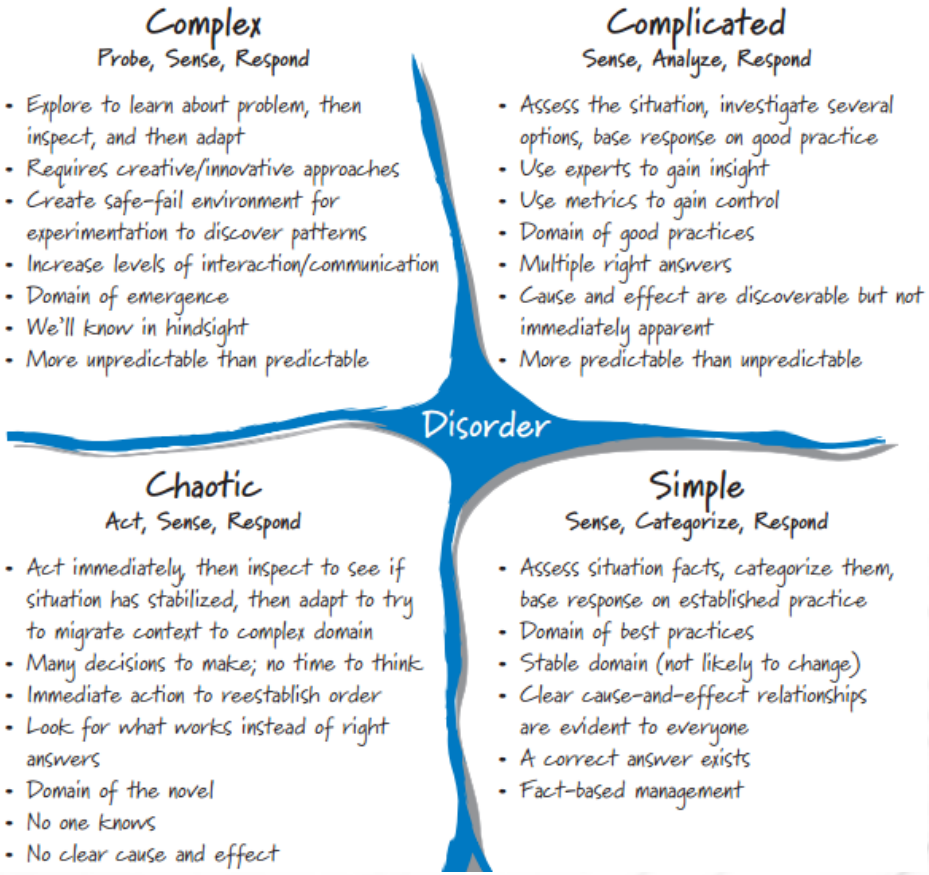
- Delighted customers
- Improved return on investment
- Reduced costs
- Fast results
- Scrum can best be applied to complex work

Is Scrum always the best framework?

- Scrum is an excellent solution for many situations, it is not the proper solution in all circumstances.
- The Cynefin framework (Snowden and Boone 2007) is a sense-making framework that helps us understand the situation in which we have to operate and decide on a situation-appropriate approach.
- It defines and compares the characteristics of five different domains: simple, complicated, chaotic, complex, and a fifth domain, disorder, which occurs when you don't know which other domain you are in .

Cynefin Framework

- Cynefin framework is used to discuss situations in which Scrum is and is not a good fit.



Cynefin Framework

Complex Domain:

- When dealing with complex problems, things are more unpredictable than they are predictable.
- This is the domain of emergence.
- explore to learn about the problem, then inspect and adapt based on our learning.
- complex domains requires creative and innovative approaches. Routine solutions simply don’t apply.
- need to create a safe-fail environment for experimentation so that important information can be discovered. In this environment high levels of interaction and communication are essential.

Cynefin Framework

Complicated Domain:

- Complicated problems are the domain of good practices dominated by experts.
- There might be multiple right answers, but expert diagnosis is required to figure them out.
- Although Scrum can certainly work with these problems, it might not be the best solution.

Cynefin Framework

Simple Domain:

- Often the right answer is obvious and undisputed.
- There are known solutions.
- Scrum can be used for simple problems, but it may not be the most efficient tool for this type of problem.

Cynefin Framework

Chaotic Domain:

- Chaotic problems require a rapid response.
- You are in a crisis and need to act immediately to prevent further harm and reestablish at least some order.

Cynefin Framework

Disorder:

- You are in the disorder domain when you don’t know which of the other domains you are in.
- When you are in the disorder domain, the way out is to break down the situation into constituent parts and assign each to one of the other four domains.
- You are not trying to apply Scrum in the disorder domain; you are trying to get out of this domain.

Interrupt-drive work

- Scrum is not well suited to highly interrupt-driven work.
- In interrupt-driven environments you would be better off considering an alternative agile approach called **Kanban**.
- Kanban is not a stand-alone process solution, but instead an approach that is overlaid on an existing process.

How software was previously delivered: Waterfall or Plan Drive Development

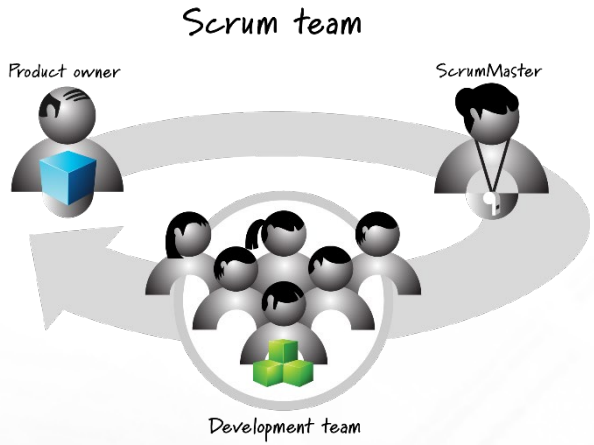
- Step 1: Analysis ← Customer Interaction
  - Step 2: Design
    - Step 3: Coding
      - Step 4: Testing
        - Step 5: Production ← Customer Interaction with product (Too long!!)
- Disadvantages:
  - No immediate feedback
  - Costly to implement any changes late in the process
  - No way to inspect and adapt or pivot
  - Customers were usually kept away from the project until the very end

Scrum: Deliver on Cadence

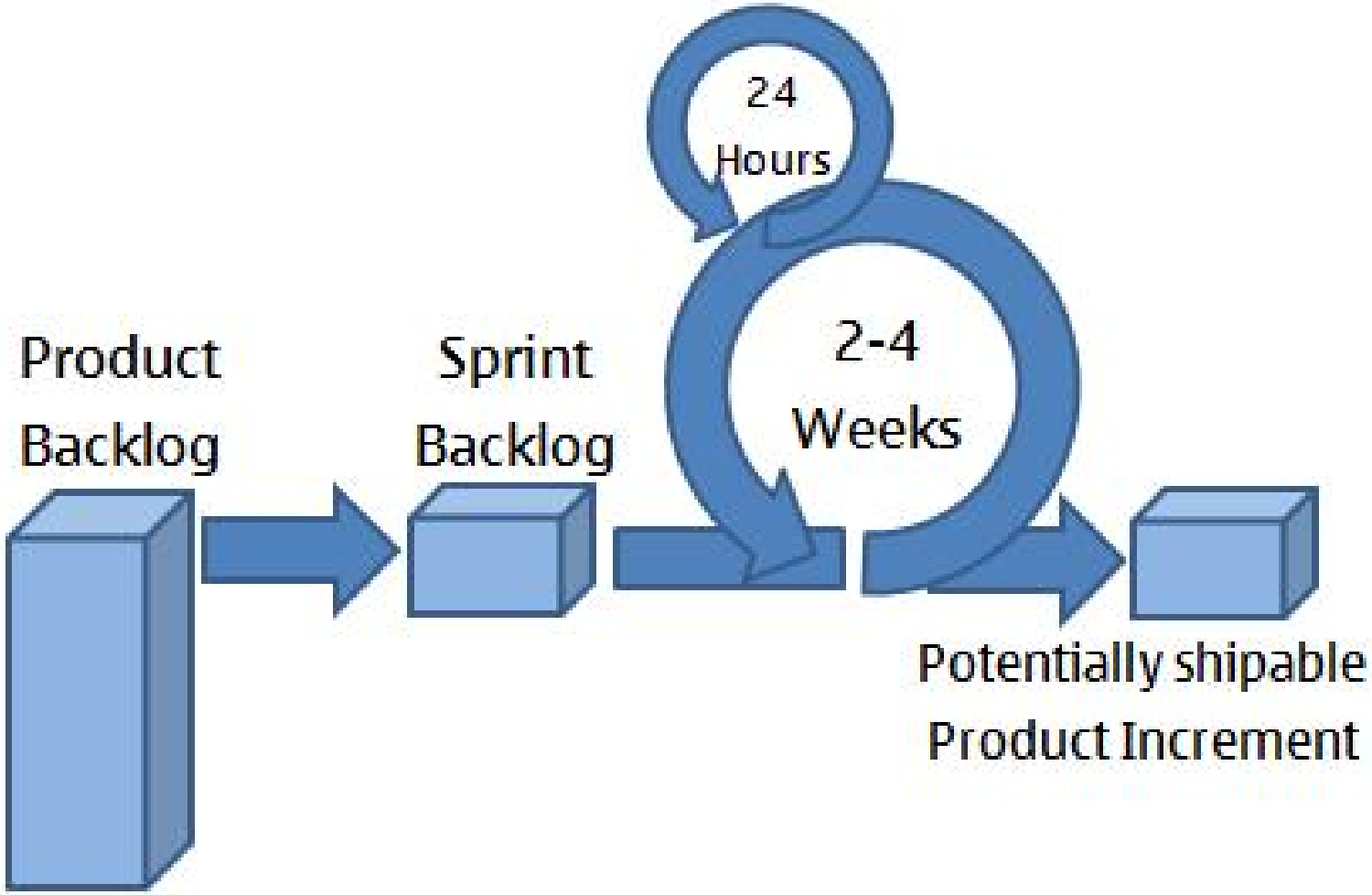
- Because software is delivered every sprint, the disadvantages of waterfall are addressed.
  - Stakeholders provide immediate feedback
  - Changes can be implemented after every iteration instead of waiting until the end
  - Sprint review ceremony allows for inspection of working software
  - We value working software that is delivered after every sprint and can be shipped
  - Encourages the team to solicit feedback from customers often

Roles in Scrum

- Product Owner: The role responsible for delivering value for the product and creating the product backlog of “user stories” . The product owner must groom and prioritize the backlog so that it is ready for the team to pull in work into the sprints.
- Scrum Master: The role responsible for advocating on behalf of the team and removing any impediments from the team.
- The Team: The members of a scrum team are responsible for implementing the user stories created by the product owner into deliverable functionality. The team is self-organized and product-focused.

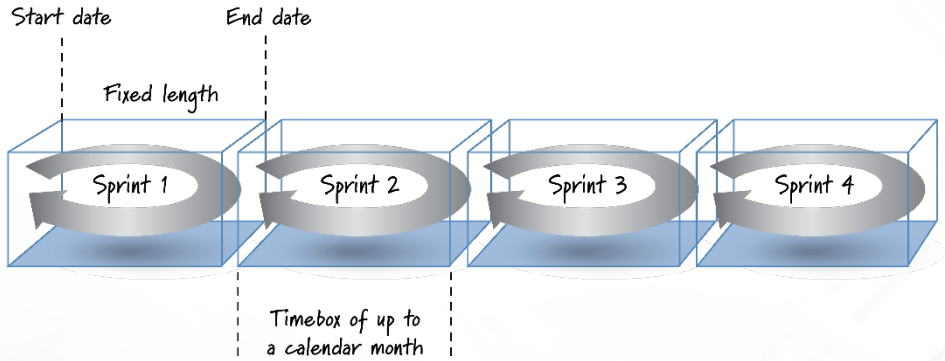


Scrum Framework



Sprints

- Work is performed in **iterations/cycles** up to one calendar month in duration
  - Timeboxed
  - Fixed start & end dates
  - Generally, same duration for each Sprint
- Work completed should create something of **tangible value** to the PO
- Guidelines suggest **no scope, goal, or personnel-altering** during a Sprint



Scrum Ceremonies

- Sprint Planning:
  - Description: Discussion where product Owner and the team commit to a sprint goal(s) and agrees pull certain stories from the product backlog into the sprint backlog.
  - Time: It should be timeboxed to 1hr per week of Sprint work. A 2 week sprint would need 2 hours of planning.
- Sprint Standup (Daily Scrum):
  - Description: A daily meeting meant to keep the team on track. Each member answers:
    - What have you done since the last Daily standup?
    - What are you going to do until the next Daily Standup?
    - What impediments are standing in your way?
  - Time: Short, usually 15 minutes max. If there are topics that need to be discussed in detail, the team members involve should take it outside of the stand up.

Scrum Ceremonies

- Sprint Review:
  - Description: Meeting where all the completed stories are demoed to the product owner and other stakeholders. Feedback is collected from stakeholders on the completed product increment.
  - Time: It should be timeboxed to 1hr per week of Sprint work.
- Sprint Retrospective:
  - Description: Performed at the end of the sprint. Ceremony where the team gets together to discuss:
    - What has been going well in the last sprint and how do we continue to do it?
    - What hasn't been working well in the sprint and how improve?
  - Time: It should 2 hours per week of Sprint.

Scrum Artifacts

- Product Backlog: This is a prioritized list of user stories or features that have business value for the product. The product owner is constantly grooming this backlog to deliver functionality that is valuable to the product users.
- Sprint Backlog: This is the estimated list of user stories that has been pulled in to the team and the team has agreed to complete in the sprint. They usually line up with the Sprint Goal.
- Shippable Product Increment: Completed functionality that has been completed, tested and demoed for the given Sprint and can be deployed to production.
- User Story: A description of a the functionality from the point of view of a person (or user) of the system, and what value it delivers.
- Sprint: A timeboxed iteration where the development of user stories gets done

Definition of “Done”

- In Scrum, the definition of done is a tested, potentially shippable product increment.
  - Functionality has been completed based on the user stories completed and the agreed to Sprint goals.
  - Quality has been verified with a high degree of confidence.
- It’s potentially shippable because this is a business decision.
  - Can we ship? We must consider:
    - Has the customer been trained?
    - Does it integrate with out features that have not been developed?
    - Change management communication taking place?

User Story

- A User Story is a convenient format for expressing a desired business value for an item in the backlog.
- The product owner writes the user story once they understand the value they are trying to deliver in the product.
- Typical format is the format:

  <User Story Title>

  As a <user role> I want to <goal> so that <benefit>

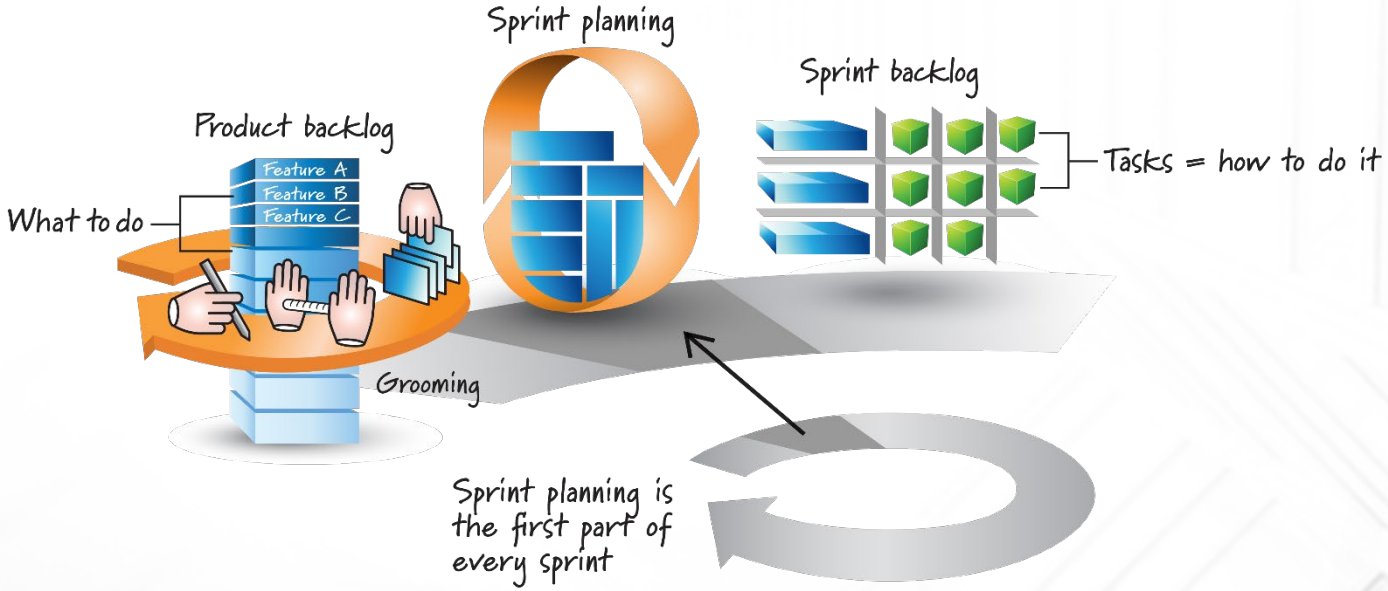
- It must also have an acceptance criteria, which details what the user story must successfully do to marked as completed.

User Story

- Example:
- As a book browser or purchaser I can create a user profile So that I do not have to enter my information each time I add books to my shopping cart or purchase books
- Possible Acceptance Criteria:
  - Must be able to click on New Profile Button to open New Profile Screen:
  - New Profile Screen must contain:
    - First Name label and Textbox
    - Last Name label and Textbox
    - Email label and Textbox
    - Address Line 1 label and Textbox
    - Address Line 2 label and Textbox
    - Save Button
  - Must be able to click Save Button and the profile information will be persisted in the database.
  - New Profile screen must adhere to UI / UX standards and client side validation.

Sprint Planning

- Product backlog usually represents many weeks or months of work
- Sprint is one calendar month or less
- Sprint planning is performed before each sprint by the PO, Dev team, and ScrumMaster to determine the next sprint's **goal(s)**

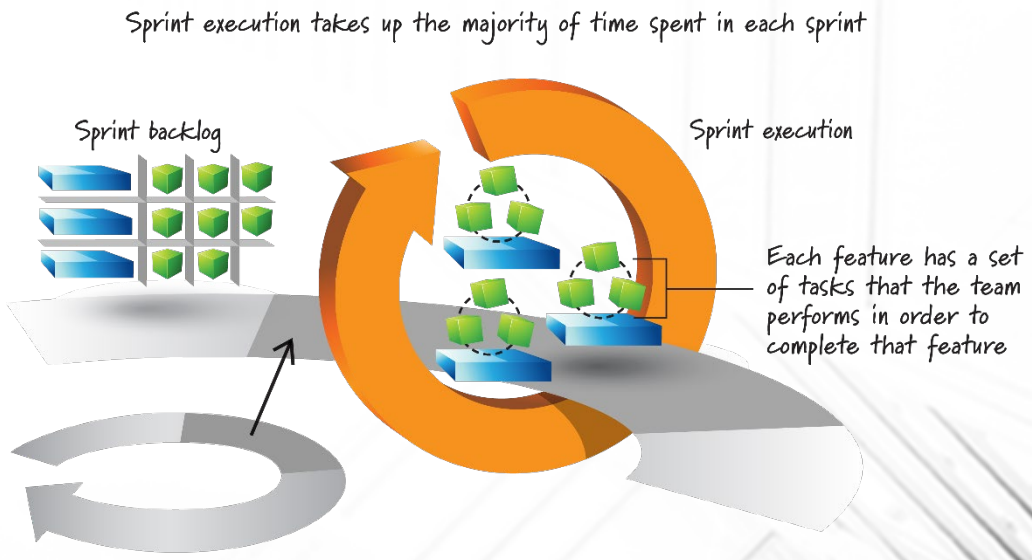


Sprint Planning

- The Dev team then selects the high-priority items from the product backlog that it believes it can realistically accomplish working at a **sustainable pace** (during the sprint)
- Many teams then create a **Sprint backlog** breaking down the features into a set of tasks
- Time (less than a day)

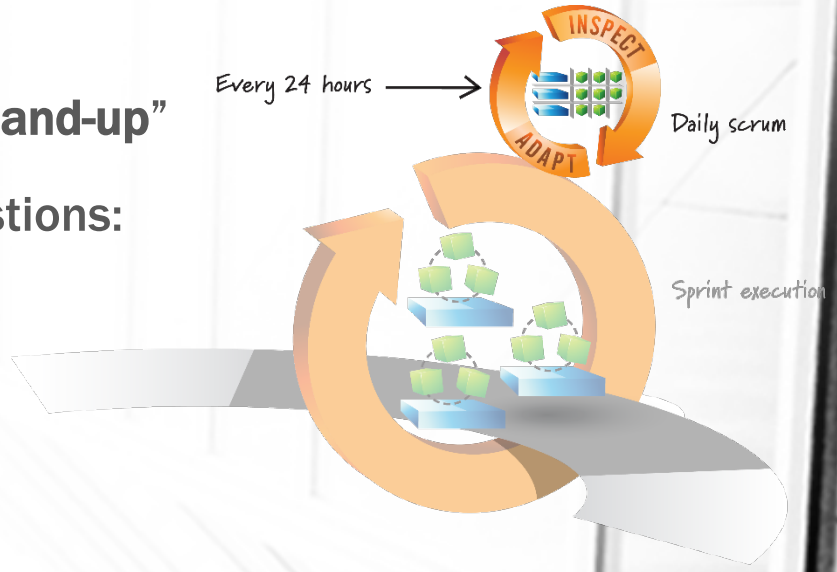
Sprint Execution

- ScrumMaster **coaches, guides** the Dev team
- Dev team responsible for **ordering the tasks**, **defining their own task-level work,** and **self-organizing** to best achieve the sprint goal
- Team performs necessary tasks to **complete** the sprint backlog
- Completion is referred to as **Done**



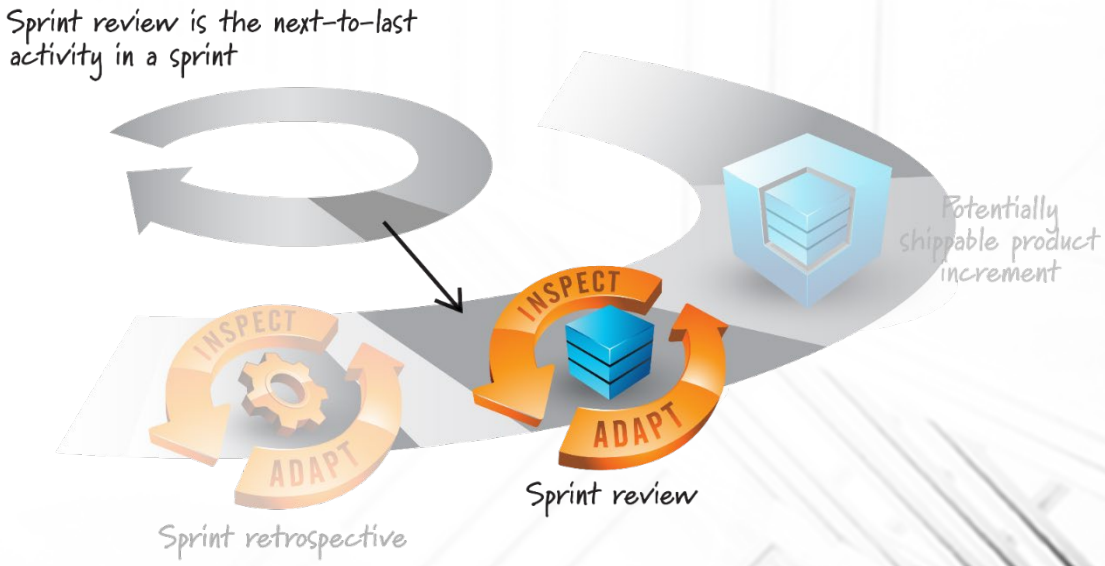
Sprint Standup or Daily SCRUM

- Each day, ideally @ the same time, the Dev team members hold a timeboxed (15 minutes or less) **daily scrum.**
- This is an **inspect-sync-&-adapt** session rather than a **problem solving** or traditional **status report** session for the ScrumMaster
- Common practice is to stand-up for this session, hence "**daily stand-up**"
- ScrumMaster **facilitates** as each team member answers 3 questions:
  1. **What did I accomplish** since the last stand-up?
  2. **What do I plan to work on** by the next stand-up?
  3. **What obstacles** are preventing my progress?
- The daily stand-up is for the **benefit** of the team
- Team members are making **commitments** to each other



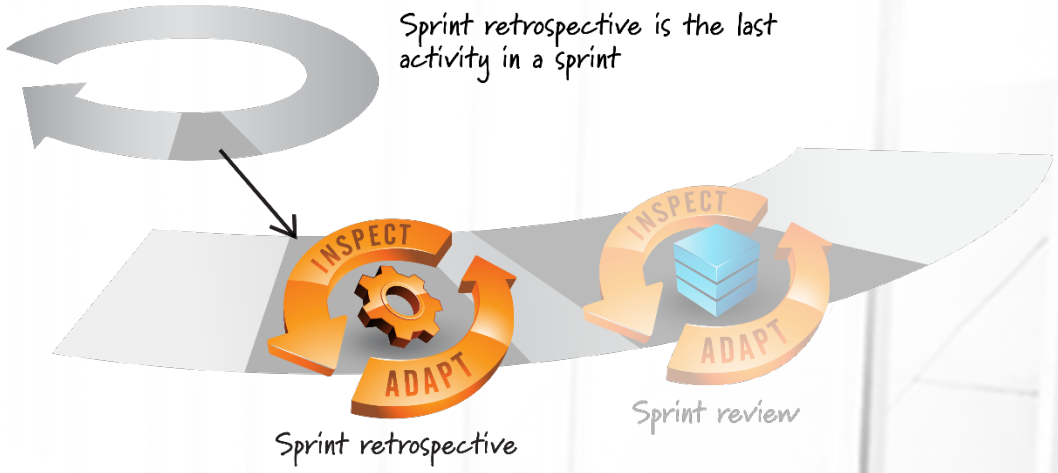
Sprint Review

- **Goal:** Inspect and adapt the **product** that is being built
- **Conversation** amongst Scrum team, stakeholders, sponsors, etc.
- Conversation focus is on reviewing the just-completed features in the context of the overall development effort
- Information flow is **bi-directional**



Sprint Retrospective

- **Goal: Continuous Process Improvement** by inspecting and adapting the **process** that is being used
- PO, ScrumMaster, & Dev Team discuss:
  - What's working with Scrum
  - What's not
- Improvements identified will be **implemented or undertaken** by the Scrum team in the next sprint



Scrum Project

- 5 students per project:
  - 1 Product Owner (Builds the product backlog
  - 1 Scrum master (removes impediments /develops/ test)
  - 3 team members (developers/ testers)
- 2 Week Sprints/Iterations
- Ceremonies must be completed every sprint
- Working software is the output of a Sprint/ Potentially shippable