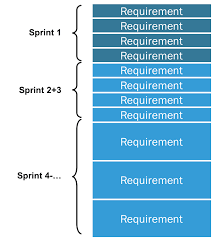
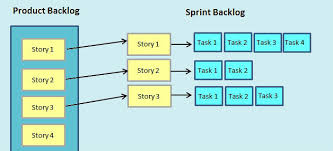
User Stories and Scrum

* Scrum is an implementation of Agile
* Scrum roles
  + Product owner
    - Person responsible for communicating with the end client
    - They ultimately control the project
    - Should tell you what needs to be worked on and priorities
  + Scrum Master
    - Allocate team members efficiently
    - Help create tasks and assign them
    - Keep people motivated, be a mentor
  + Team Member (scrumlings)
    - People who were on the Scrum Team
  + Stakeholder
    - Usually the people paying you for your labor!!!!
    - They are the people who need it to be successful





* Scrum Ceremonies
  + Backlog Grooming
    - Process at the start of a sprint
    - Done by product owner and scrum master (other senior developers)
    - They will look at the pending user stories and create tasks to be added to the current sprint backlog
      * To-do in asana
  + Planning
    - Scrum-master will allocate tasks/people/resources most efficiently to address the sprint backlog
  + Daily Standups
    - Team get together where every team member will give a status update
    - Request help, offer help
    - Talk about any blockers to their work
  + Sprint Retrospective
    - Will review what was accomplished and what was not
    - Review the lessons learned
    - Use the information to plan your next sprint
    - Show/prepare your working demo of the application so far
    - Update any documentation as necessary

User Stories

* Functional User stories
  + These are ways in which users will use and interact with your software
* Technical User stories
  + Stories developers should fulfill to accomplish better accomplish the function tasks
    - Scalability
    - Bug fixing
    - Good design practices

Structure of a User Story

As a <User> I want to <do something> so that <I can accomplish something>

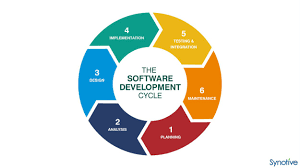
Acceptance Criteria

Acceptance criteria are the steps that a user should follow in order to complete the user story. Acceptance Criteria serves as a test to see if the user story is indeed fulfilled. They tend be specific to user GUI interaction. Clicking a specific button, navigating to a certain page, seeing a certain pop-up

Given (some precondition) The Employee is on the login page

When (some user action) The Employee clicks login

Then (some result) The Employee is redirected to their home page



1. Analysis of existing system (look at metrics)
2. Planning and designing
3. Implementation
4. Testing
5. Maintenance
6. Hmmm it appears that orders for specialty deals have been dropping significantly
7. Plan to promote the specialty deals by making them more visible on the front page
8. Code the feature into the front page
9. Test that those specialty deals do actually appear
10. Deploy/maintain the application (see if users are actually buying the specialty deals)
    1. User want tracking of their pizza orders. It a common feedback we hear
    2. We need to add a new table in the database to track the orders as we as create a TrackingOrderService
    3. We need to implement our code
    4. Test that the Tracking feature works
    5. Monitor our deployment see if we hear feedback from our users about the new feature