Software Development Framework with Scrum

Trainer: Trinh Nguyen

Course Objectives

- At the end of the course, you will acquire sufficient knowledge of Scrum Framework and be able to apply the knowledge to project
- Knowledge to gain from the course include:
  - Scrum Framework: Roles, Backlog, Sprints, Events
  - Scrum terminologies

Course Audience and Prerequisite

- The course is for everyone who work in a project
- Pre-requisites:
  - None

Assessment Disciplines

- Class Participation: 100%
- Passed Score (mini test): 70%

Course Administration

- In order to complete the course you must
  - Sign in the Class Attendance List
  - Participate in the course and complete assignment
  - Provide your feedback in the End of Course Evaluation

Duration and Course Timetable

- Course Duration: 4 hours
  - Introduction: 2.5 hours
  - Discussion: 1.5 hours

Agenda

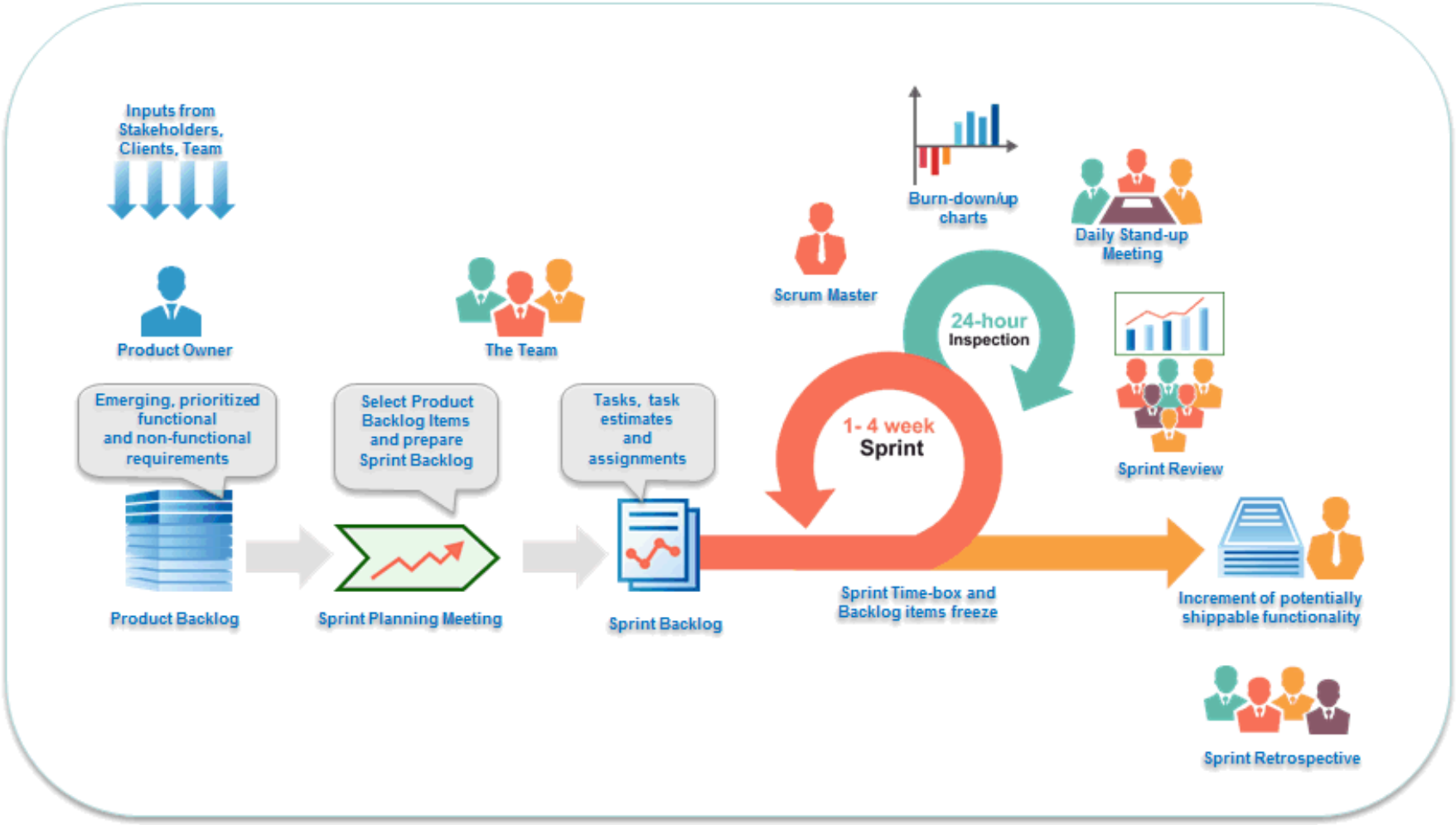
- Scrum Framework: Roles, Sprints, Events, Backlog
- Scrum Terminologies
- Workshop
- Q & A

August 15, 2019

Scrum Framework

DXC.technology

Scrum Overview



Scrum is a project management framework - not a full-fledged methodology

Product Owner

- Define features of product
- Decide on release date and content
- Be responsible for profitability of product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results

Common Mistakes

- The Underpowered Product Owner
- The Overworked Product Owner
- The Partial Product Owner
- The Distant Product Owner
- The Product Owner Committee

Scrum Master

Scrum Master	
Responsible for success of project (on-time, on budget, high quality)	Facilitate meetings and collaboration across all roles
Manage Sprints and timelines	Keep information about the team's progress up-to-date and visible to all parties
Control budget burn and remaining work items	Ensure that team is fully functional and productive
Manage and improve project process, engineering activities, practices, and project artifacts	Lead team to transit to a self-managed team
Remove blockers, impediments	Provide Agile coaching and support to the Team
Guide project through the shoals of complexity	Stand firm and stick to the rules of project process
Foster Agility environment, communication, collaboration, coordination	Protect the team from interruptions and external interferences

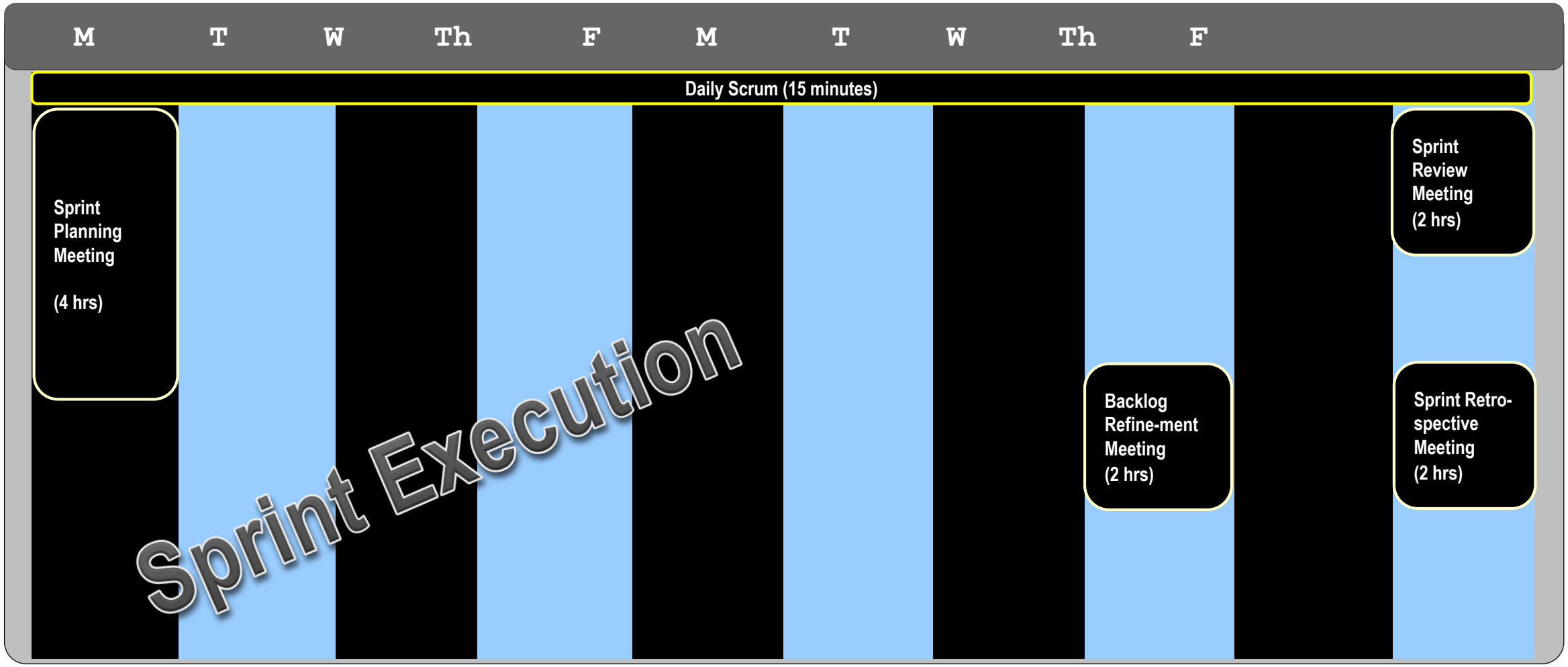
Team

- Cross-functional:
  - Business Analyst, Technical Architect, Designer, Developer/Programmer, DevOps Engineer, Tester, others.
- Self-organized, self-managed
- Collectively responsible for delivering committed items
- Commitments made to each other to get things done
- Intensively collaborative
- Continuous learning
- Typically 5-9 people
- Ideally, team members should be full-time and co-located

Daily Engineering Activities

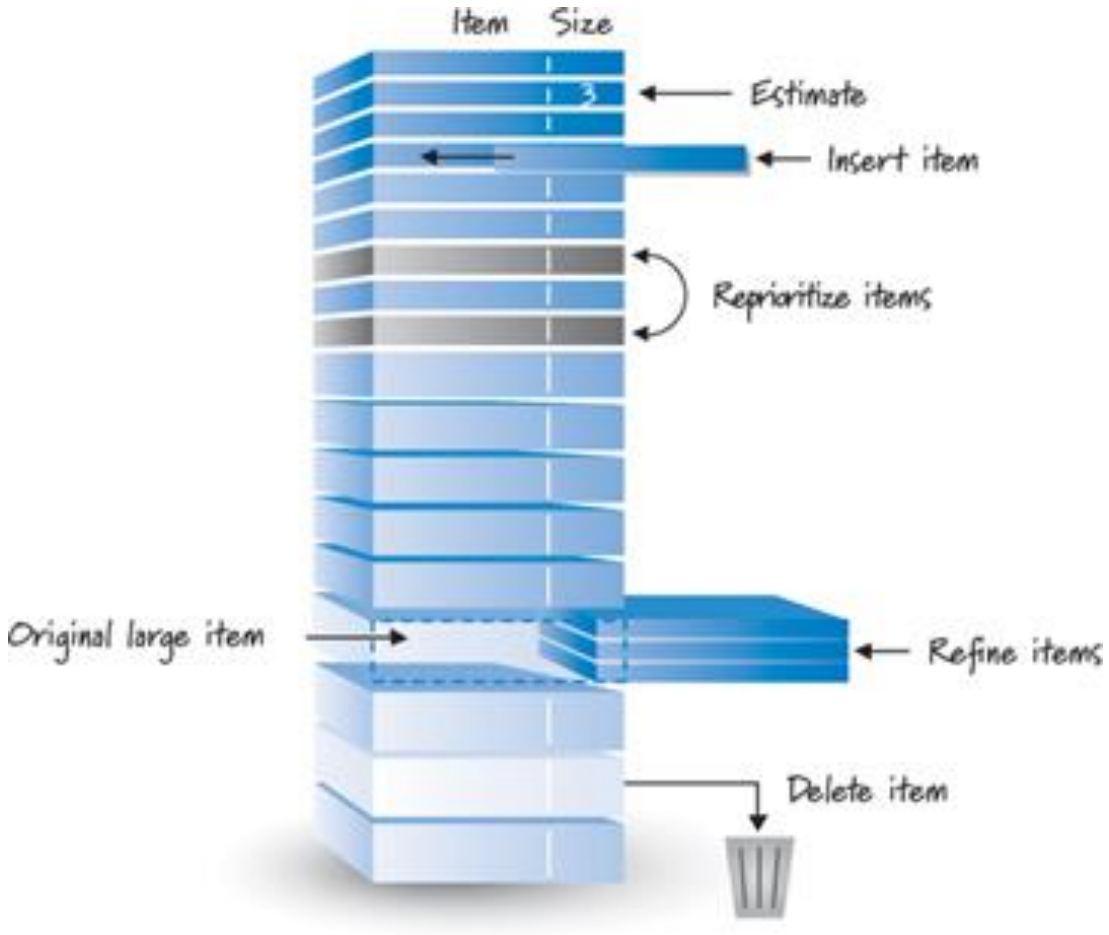
Engineering	Planning	Daily Engineering Activity				Review	Release
Analyze Requirement	Sprint Backlog Items (SBIs) to be delivered within the time box	Standup Meeting (morning)	• Create business req. spec. • Clarify business requirement • Create/Update User Stories, User Story Mappings	• Create/Update Screen flows, Mockup / wireframe • Create Data Mapping • Validate software quality		Live product demonstration	Release Candidates – Features
Define Solution Architecture		Other Meetings	• Define Solution Architecture • Resolve blockers, impediments • Review/Approve Code	• Guide implementation of practices • Tech. Support	Resolve Technical Debts System Integration Regression Test Performance Load Test		
Design			• Create/Update Design Documents • Review/Approve Code	• Code, Unit Test, Integrate code			
Code and Integrate		Update Work progress to JIRA (afternoon)	• Code, Unit Test, Integrate (ITS, Junit, Nunit) • Perform integration test • Check-in code once approved				
Build and Deploy			• Build, deploy and validate • Improve Build and Deploy jobs • Update Deployment Guides, Check List	• Monitor and control builds deployed			
Test			• Create and execute test automation with ITS • Leverage and Improve ITS Common Library				

Scrum Events or Meetings



Backlog Refinement Meeting

- Purpose
  - Ensure the team and Product Owner have the same understanding of the items that are going to be developed in the next two or three Sprints
- When
  - Before new development work starts
- Participants
  - Product Owner
  - Scrum Master
  - Team



Backlog Refinement Meeting

- Actions
  1. Product Owner presents any new items requested by the customers
  2. Team captures requirements to the product backlog
  3. Product Owner prioritizes product backlog items
  4. Team and Product Owner refine product backlog items from top-down
- Output
  - Refined, prioritized product backlog items
  - Definition of Done
  - Understanding of requirements

Sprint Planning Meeting

- Purpose
  - Determine what product backlog items (functionality) the team will deliver
- When
  - Conducted before Sprint
- Participants
  - Product Owner
  - Scrum Master
  - Team

Sprint Planning Meeting

- Actions
  1. Product Owner presents top priority items from product backlog to the team
  2. The team selects as many items as they can handle in the next sprint based on team capacity
  3. The team breaks items down into sprint tasks and estimate them based on performance of previous sprints
  4. Product Owner and the team work together to come up with a sprint goal
- Output
  - Tasks and estimated effort (sized in Story Points using Planning Poker technique)
  - Sprint goal

Planning Poker Technique

- Planning Poker technique is used to estimate the size of tasks and then user stories
- In order to play Planning Poker, the following is needed:
  - The list of features to be estimated
  - Decks of numbered cards
  - A typical deck has cards showing the Fibonacci sequence: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, ? (Don’t know), ∞ (Too large, need to break down into smaller items)

Planning Poker Technique

1. Product Owner explains user stories
2. The Scrum Master facilitates the process and keeps an eye on "anchoring" in discussions
3. Each player estimates relative size and complexity (not effort)
4. Size is estimated by comparing user stories
5. Each player throws card
6. Lowest and highest bidder discuss their reasoning
7. Replay until bids converge to one number and it is the size of the story
   - Maximum up to three rounds
   - If no result after three rounds, majority rules

Daily Scrum Meeting

- Each day during a Sprint, the team hold a Daily Scrum Meeting (or Daily Stand-up Meeting) with specific guidelines:
  - All members of the development team come prepared with updates for the meeting
  - The meeting starts **on time** even if some development team members are missing
  - Timebox the meeting in 15-minute
  - The meeting should happen at the same location and same time every day

Daily Scrum Meeting

- During the meeting, each team member answers three questions
  - What have you done since yesterday?
  - What are you planning to do today?
  - Any impediments/stumbling blocks?
    - *Any impediment/stumbling block identified in this meeting is documented by the Scrum Master and displayed on the scrum board*
    - *No detailed discussions shall happen in this meeting*

Sprint Review Meeting

- Purpose
  - Team demonstrates to customer work done within the sprint, which enables an inspection of project progress
  - Based on customer feedback, team adjust the work
- When
  - Conducted at the end of every sprint
- Participants
  - Product Owner/Stakeholders
  - Scrum Master
  - Team

Sprint Review Meeting

- Actions
  1. The team presents the product increments that they have built to Product Owner and Stakeholders (management, customers and users)
  2. Any items which are not completed during the Sprint should not be shown and are returned to the product backlog
  3. Stakeholders and the Product Owner decide on what's "done", what's "not done", and what's to do next
- Output
  - Acceptance or rejection of the product increments
  - Incomplete items are returned to product backlog

Sprint Retrospective Meeting

- Purpose
  - Determine what need to be changed to improve the team in the next Sprint
- When
  - Conducted at the end of every sprint and after a Sprint Review Meeting
- Participants
  - Product Owner
  - Scrum Master
  - Team

Sprint Retrospective Meeting

- Actions
  1. All team members reflect on the past Sprint
  2. Make continuous improvements
  3. Two main questions are asked in the Sprint retrospective
     a. What went well during the Sprint?
     b. What could be improved in the next Sprint?
- Output
  – Lessons learnt and adapted

August 15, 2019

Scrum Terminology

DXC.technology

User Story

- A convenient format for expressing the desired business value for many types of product backlog items
- User Stories are crafted in a way that makes them understandable to both business people and technical people
- User Stories provide a great placeholder for a conversation
- Typically expressed in a format such as

Sample

As a customer representative I want to search for my customers by their first and last name to find those I like to get in touch with

User Story

- **I.N.V.E.S.T** Criteria – Method for determining if a user story is ready to be slotted into a sprint
  - **Independent** – The story can be worked on and completed independently from other stories
  - **Negotiable** – The implementation details are not included in the story (left to the team to decide how it will be technically implemented)
  - **Valuable** – The story has a high degree of business value relative to most others remaining on the product backlog
  - **Estimable** – The story can be estimated in relative Level of Effort
  - **Small** – The story is not too large to be completed in a sprint, or too small to be its own story
  - **Testable** – The story can be definitively tested to determine measurable success

Story Point

- A measure of the relative size of product backlog items that takes into account factors such as complexity and physical size
- Determined by using Planning Poker Technique

Definition of Done

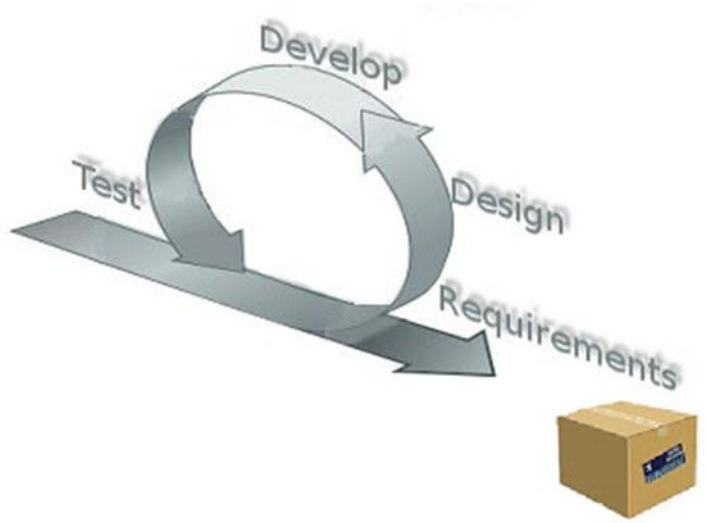
- A list of criteria which must be met before a product increment “often a user story” is considered “done”
- Failure to meet these criteria at the end of a sprint normally implies that the work should not be counted toward that sprint's velocity

Definition of Done

#	Project Artifacts	Done criteria	Owner
1	Business Requirement	Signed off	Business Analyst
2	User Stories, User Story Mapping, Business Flows	Meet INVEST	Business Analyst
3	Solution Architecture (SA)	Meet solution framework	Solution Architect
4	High level design (HLD), LLD, TSD	Coupled with solution framework	Solution Architect/TL, Dev Lead
5	Root Cause Analysis Impact Analysis Code and Unit Test (Unit Test document and/or Junit)	Adhere to HLD, LLD, TSD Pass Unit Test, Functional Test No SonarQube technical debs Release Candidate	Developer Dev Lead Solution Architect Business Analyst
6	Development environment	Meet CI, RC standards	DevOps Engineer
7	DevOps Scripts, Deployment Guides, Check List	80% automated 100% meet Check List	DevOps Engineer
8	ITS Scripts for Functional Test, Regression Test, System Test, Performance Load Test	100% Test Coverage Meet performance expectation	Tester Test Lead
9	Release Notes (Delivery Notes)	Meet document standard	Project Manager, Program Manager

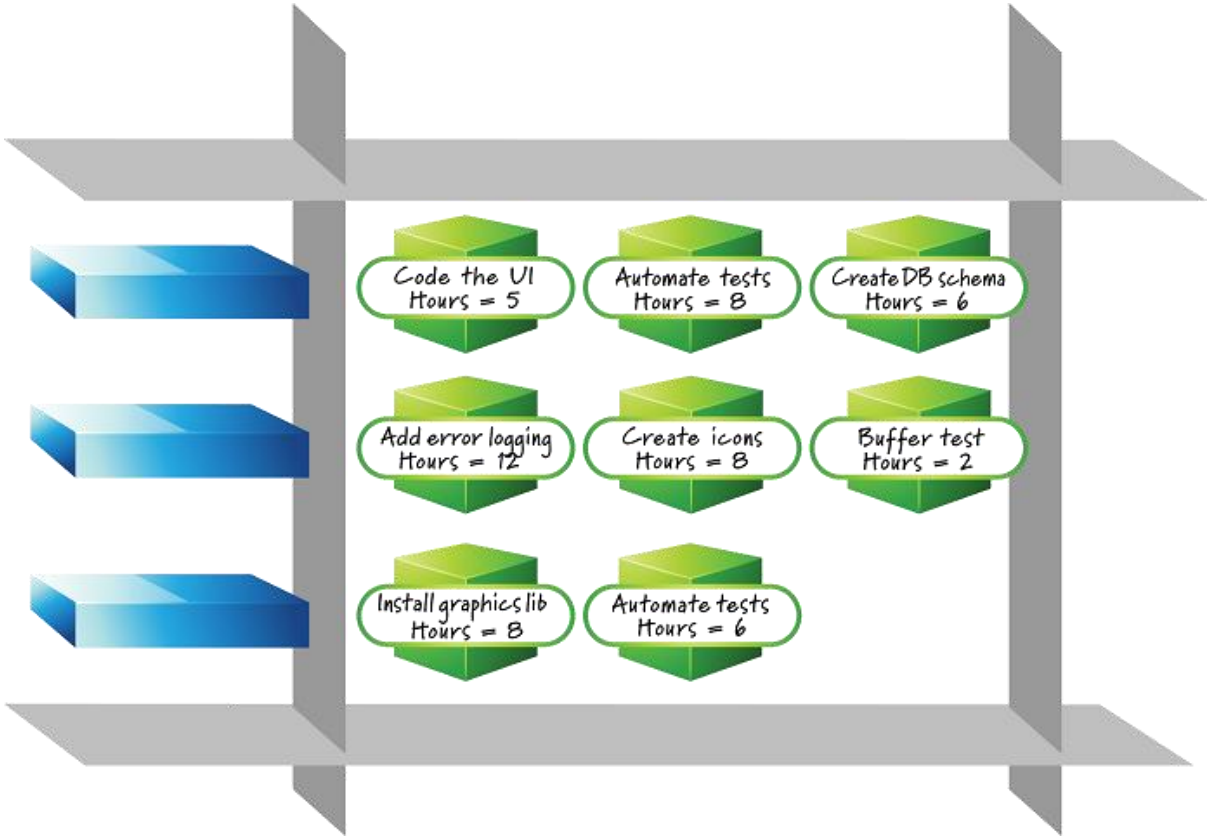
Sprint

- The basic unit of development in Scrum
- Time-boxed: fixed period of time
- Sprint cycle varies from 1 to 4 weeks
- A potentially shippable software is delivered at the end of each sprint
- For each sprint, the should be a sprint goal



Sprint Backlog

- Contains stories with the highest priority from the product backlog
- Amount of stories depends on capacity (velocity)
- For each story, tasks are added and estimated by hours
- The estimated work remaining is updated daily
- Any team member can add, delete, or change tasks within the sprint backlog



Sprint Goal

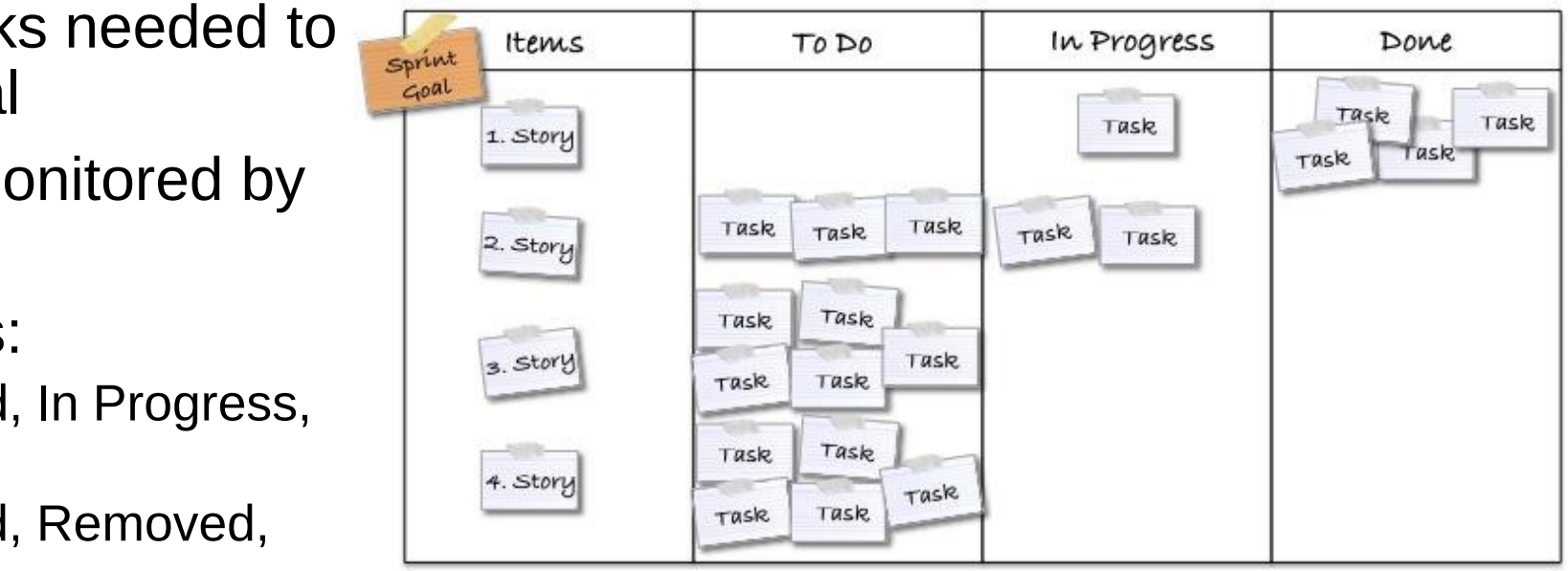
- A short expression of the purpose of a Sprint, often a business problem that is addressed
- Functionality might be adjusted during the Sprint in order to achieve the Sprint Goal

Sample

- In this sprint, we will allow users to log-in to the site, retrieve a forgotten password, and manage their own profile
- In this sprint, we will implement basic shopping cart functionality including add, remove, and update features
- In this sprint, we will integrate VISA payment gateway into our billing module

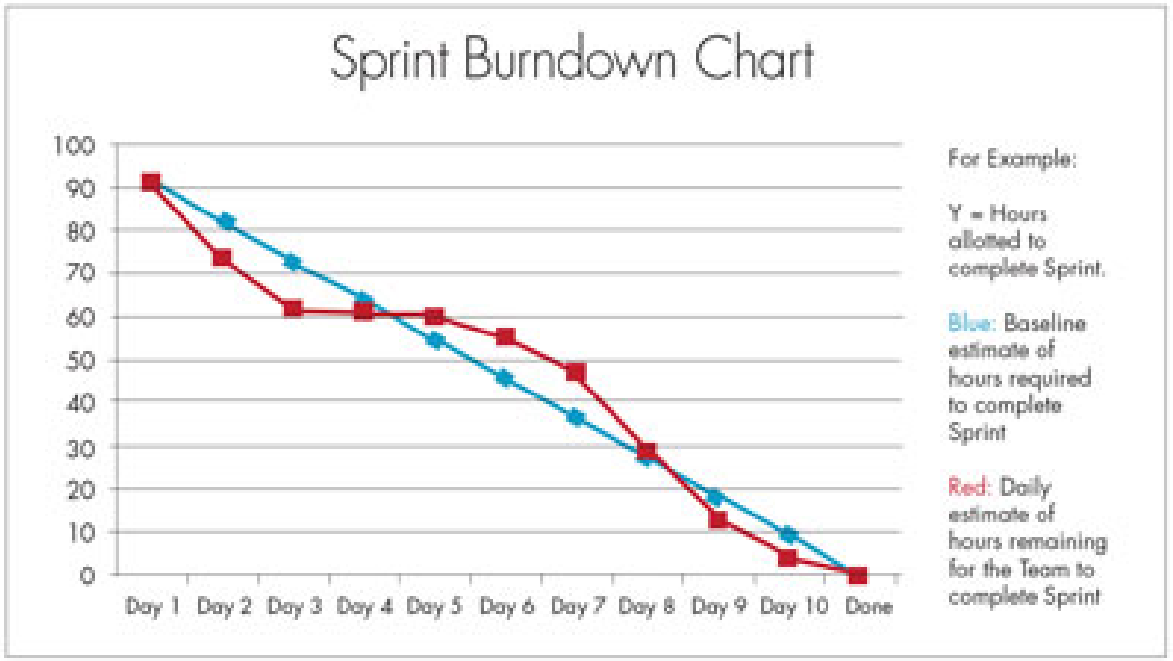
Scrum Board

- Contains all stories/tasks needed to complete the sprint goal
- Maintained by Team, monitored by Scrum Master
- At a minimum, contains:
  - Categories for Not Started, In Progress, Done
  - Other categories: On Hold, Removed, Ready to Test, Verify…
  - Copy of Sprint Burndown
- Serves as a manual “dashboard” for Team status and progress



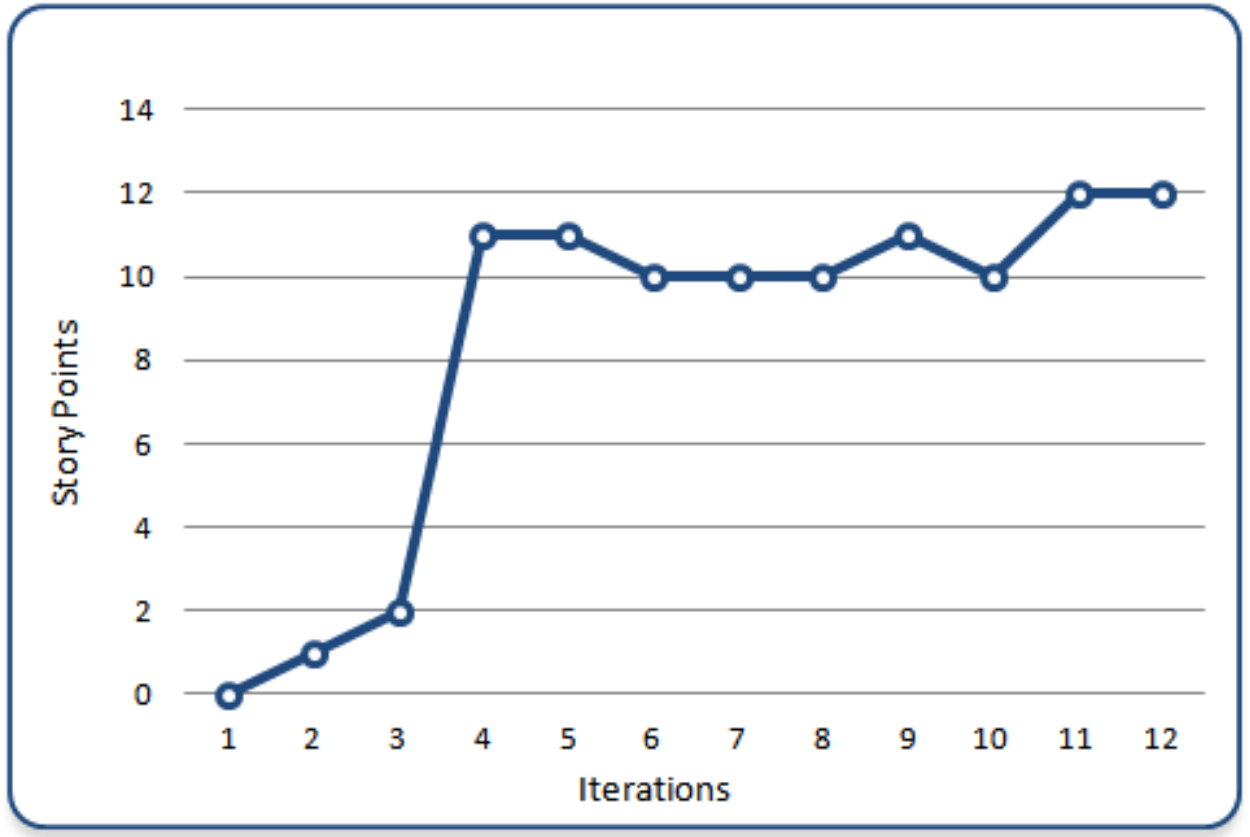
Burndown Chart

- Visualizes completed vs. remaining works for each sprint, release, or the whole product
- Can be shown in story points unit or ideal days
- Shows actual work remaining, plus slope of ideal time remaining (estimated)
- Data gathered daily by team reporting status
- Maintained by Scrum Master



Capacity (Velocity)

- An indicator of how much work the team can do in a sprint



Question & Answer

Thank You!

Revision History

Date	Version	Description	Updated by	Reviewed and Approved By
Aug, 2017	2.0	Re-theme with DXC template		Quang Tran, Long Truong
July 2019	3.0	Update content	Khanh Lam, Trinh Nguyen Mai Nguyen	Khanh Lam