3 & 4. Introduction to Scrum & Core Concepts in Scrum

IT 4406 – Agile Software Development

Level II - Semester 4

Intended Learning Outcomes

- Explain the significance of Scrum.
- Describe the usage of Scrum.
- Understand and apply the Scrum framework practices.
- Identify the Scrum roles and their activities.
- Describe the Scrum artifacts and rules.
- Define what a Sprint is.
- Describe the characteristics of a Sprint.
- Choose the most suitable time span for a Sprint.
- Explain the term called “Timeboxing”.

Intended Learning Outcomes

- Discuss how the requirements are handled in a Scrum project.
- Explain how and when to use User Stories
- Describe and practice how to handle non-functional requirements and knowledge acquisition in a scrum project.
- Explain the importance of the Product Backlog and Product Backlog characteristics.
- Describe how to determine which and how many backlogs should exist.
- Explain “Backlog Grooming”.
- Describe the concepts of estimation and velocity.

Intended Learning Outcomes

- Discuss the items that should estimate and when and how to estimate them.
- Describe how to estimate product backlog items.
- Explain velocity and how to use a velocity range.

List of subtopics

3.1 What Is Scrum?

3.2 History of Scrum

3.3 Advantages of Scrum

3.4 Genomica Results

3.5 Use of Scrum

List of subtopics

4.1 Scrum Framework

- 4.1.1 Introduction
- 4.1.2 Scrum Roles
  - 4.1.2.1 Product Owner
  - 4.1.2.2 Scrum Master
  - 4.1.2.3 Development Team
- 4.1.3 Scrum Activities and Artifacts
  - 4.1.3.1 Product Backlog
  - 4.1.3.2 Sprints
  - 4.1.3.3 Sprint Planning
  - 4.1.3.4 Sprint Execution
  - 4.1.3.5 Daily Scrum
  - 4.1.3.6 Done
  - 4.1.3.7 Sprint Review
  - 4.1.3.8 Sprint Retrospective

List of subtopics

4.2 Sprints

- 4.2.1 Introduction
- 4.2.2 Timeboxed
- 4.2.3 Short Duration
- 4.2.4 Consistent Duration
- 4.2.5 No Goal-Altering Changes
- 4.2.6 Definition of Done

4.3 Requirements and User Stories

- 4.3.1 Introduction
- 4.3.2 Using Conversations
- 4.3.3 Progressive Refinement
- 4.3.4 What Are User Stories?

List of subtopics

4.3.5 Level of Detail

4.3.6 INVEST in Good Stories

4.3.7 Non-functional Requirements

4.3.8 Knowledge-Acquisition Stories

4.3.9 Gathering Stories

4.4 Product Backlog

4.4.1 Introduction

4.4.2 Product Backlog Items

4.4.3 Good Product Backlog Characteristics

4.4.4 Grooming

4.4.5 Definition of Ready

4.4.6 Flow Management

4.4.7 Which and How Many Product Backlogs?

List of subtopics

4.5 Estimation and Velocity

- 4.5.1 Introduction
- 4.5.2 What and When We Estimate
- 4.5.3 PBI Estimation Concepts
- 4.5.4 PBI Estimation Units
- 4.5.5 Planning Poker
- 4.5.6 What Is Velocity?
- 4.5.7 Calculate a Velocity Range
- 4.5.8 Forecasting Velocity
- 4.5.9 Affecting Velocity
- 4.5.10 Misusing Velocity

Scrum - Introduction

- What is Scrum?
  - ➢ An Agile methodology
  - ➢ It is an iterative incremental process for software development
  - ➢ An approach that controls the chaos of changing requirements
  - ➢ A team-based approach to development
  - ➢ The whole team travels the distance as a unit (Rugby)

Scrum - Introduction

- What is Scrum?
  - ➢ The work itself is performed in short, time boxed iterations(usually range from a week to a month)
  - ➢ During each iteration, a self organizing, cross functional team does all of the work—such as designing, building, and testing

Scrum - Introduction

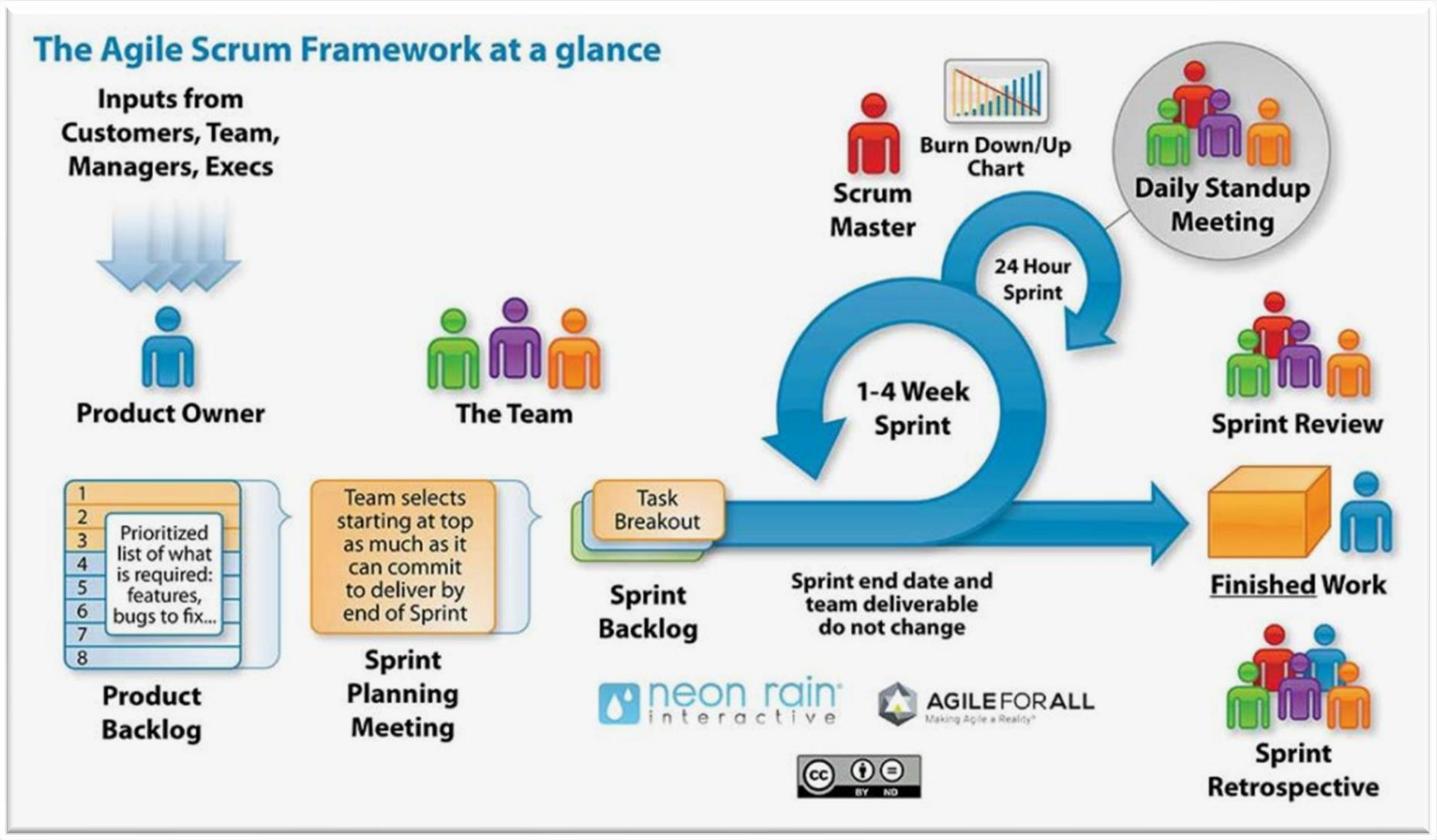
- Begins by creating a product backlog
- Work on the most important or highest priority items first.
- Time boxed iterations
- A self-organizing, cross-functional team
- At the end of the iteration, the team reviews the completed features with the stakeholders to get their feedback.
- Based on the feedback, the product owner and team can change what they plan to work on next and how the team plans to do the work.

Scrum - Origin

- “The New New Product Development Game”
  - ➢ all-at-once product development
- A term borrowed from the sport of rugby
- 1993 : Jeff Sutherland and his team at Easel Corporation
- The first paper on Scrum at OOPSLA 1995

Scrum can be used to develop different types of products or to organize the flow of various types of work

Scrum - Structure



https://www.neonrain.com/agile-scrum-web-development/

Why Scrum

- Delighted customers
- Improved return on investment
- Reduced costs
- Fast results
- Confidence to succeed in a complex world
- More joy

Cynefin framework

- While Scrum is an excellent solution for many situations, it is not the proper solution in all circumstances
  - ➢ Complex Domain
  - ➢ Complicated Domain
  - ➢ Simple Domain
  - ➢ Chaotic Domain
  - ➢ Disorder

Scrum Rules

- Teams
  - ➢ 7 – 9 members (excluding PO & SM)
  - ➢ Cross-functional
- Time-boxes
- Feedback and learning through frequent delivery
- Predictability
- Collective and Adaptive planning to build the right product
- Productivity & Quality
  - ➢ Via Reduction of waste
  - ➢ Through team and time-boxed focus

SCRUM Roles - SCRUM Master

- Be a facilitator
- Remove obstacles faced by the team
- Assist the team in achieving the iteration goals
- Coach the team on SCRUM principles

SCRUM Roles - Product Owner

- Share the product vision/ goals with the team
- Identify the requirements
- Prioritize the requirements
- Be the “GO TO” person for requirement clarifications
- Provide feedback
- Negotiate with stakeholders

SCRUM Roles - Development Team

- Traditional software development teams have various individuals with different skills and job roles.
  - Architect
  - Programmer
  - Tester
  - Database Administrator
  - UI Designer
- The SCRUM development team is a diverse, cross-functional collection of these roles who are responsible for designing, building, and testing the desired product.

Portfolio Planning

- Determine which products to work on, in what order, and for how long.

Product Planning

- Product Vision
  - Provides a clear description of the areas in which the stakeholders get value
- High-Level Product Backlog
- Product Roadmap

The outputs of product-level planning became inputs to portfolio planning

Release Planning

- Making scope, date, and budget trade-offs for incremental deliveries.
- To get an idea of what you can deliver by a fixed date, create and estimate a sufficient number of product backlog items.
  - Draw a line through the product backlog.
  - All of the items above the line are planned for the release.
- Time dimension.

Plan just in time and just enough

Sprint Planning

- Determine the specific product backlog items that the SCRUM team will work on in the next sprint

Daily Planning

- The most detailed level of planning
- Team members get together, and each person takes turns stating what he/she got done since the last daily scrum and the plan for today

Daily Planning

"Today I am going to work on the stored procedure task, and I should have that done by lunch. Whoever is going to work on the business logic task, please keep in mind that the business logic task is on the critical path for getting our work done this sprint and you should be ready to work on it right after lunch."

Meeting Types

- Release Planning
- Iteration Planning
- Stand-Up Meeting
- Retrospective Meetings

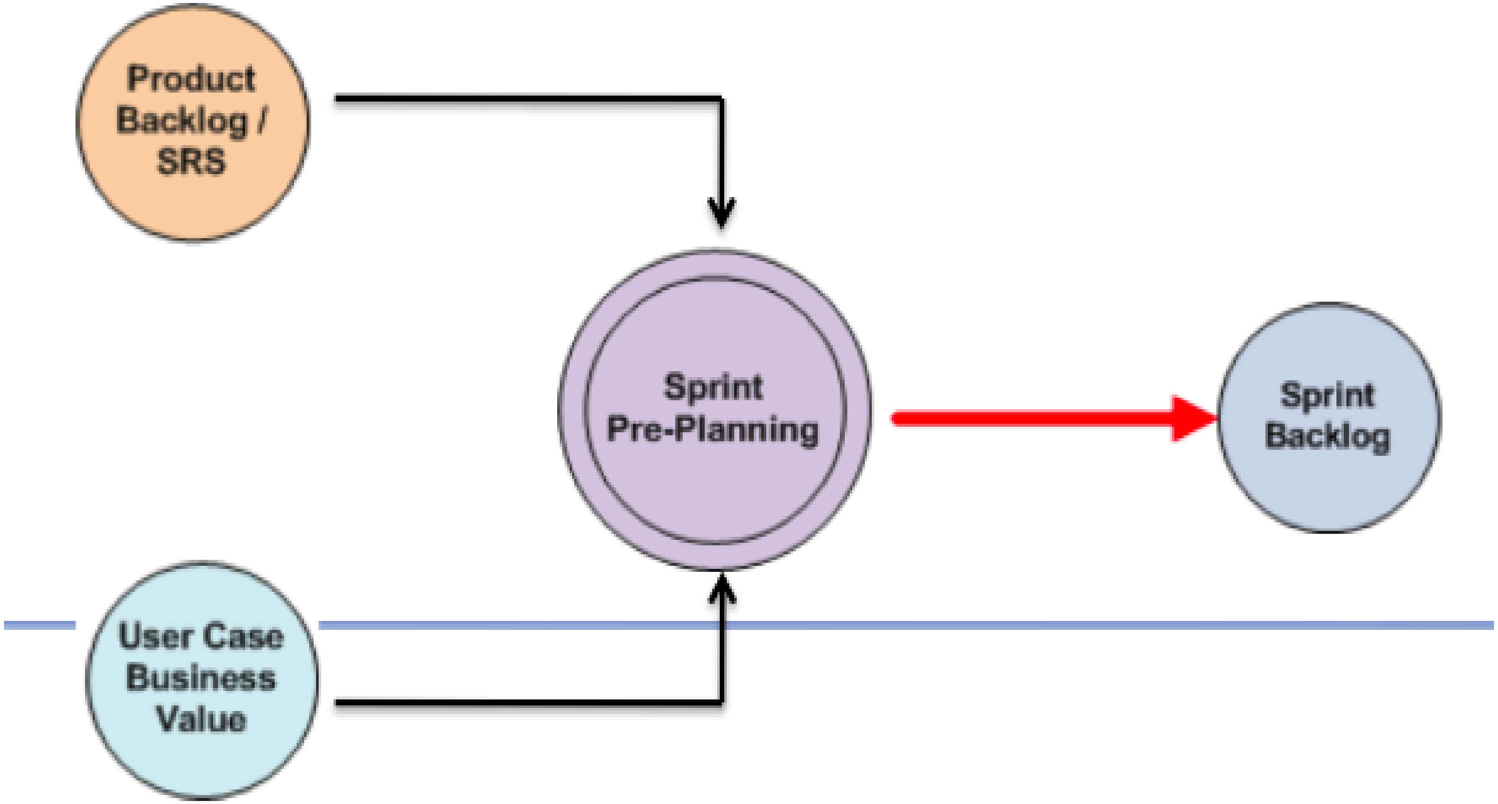
You can have other meetings!!!

SCRUM - Phases

- Pre-game
  - Planning - define system. Product Backlog
  - Architecture - high level design of system
- Sprints
  - Iterative cycles(sprints) - new or enhanced functionality
- Post-game
  - Requirements satisfied - no new items or issues

Pre-game

- This stage focuses on understanding the project in terms of business and technical requirements and setting the project governance along with the initial project management activities.



© e-Learning Centre, UCSC

Pre-game Cont...

- Key Activities
  - Conduct workshops to understand/clarify requirements
  - Document, review and refine requirements. This will generate the product backlog.
  - Estimate the effort required for the phase (High-level, ball part estimate)
  - Requirement planning activities are performed for the entire phase.
- Key Deliverables
  - Software Requirements Specification (updated with use cases)
  - High-level Architecture & Design Document
  - Deployment Model
  - Project Schedule (high-level milestones, etc)
  - Quality Assurance Test Strategy / Plan

Release Planning - Step 1

- Break the Requirements into
  - ➢ EPIC
    - ❑ Epics are large user stories, typically ones which are too big to implement in a single iteration and therefore they need to be disaggregated into smaller user stories at some point.
  - ➢ User Stories
    - ❑ Very slim and high-level requirements artifacts.

Writing user stories

- Write in the voice of the customer
- Use the following template

As a <role>,

I can <activity>

so that <business value>

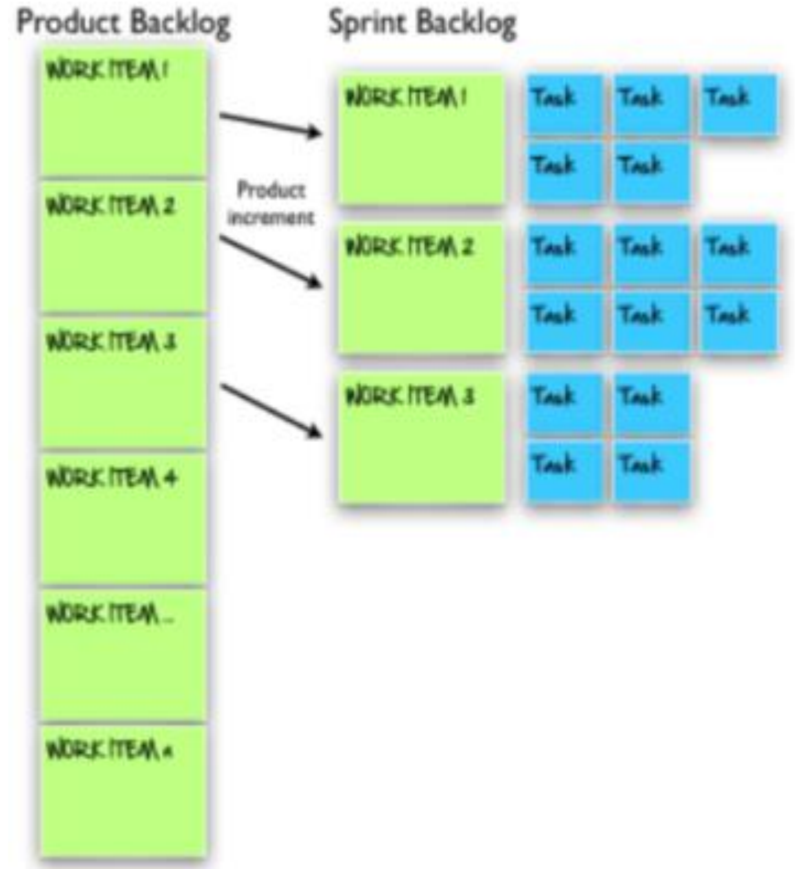
Writing user stories cont...

- Students can purchase monthly parking passes online.
- Parking passes can be paid via credit cards.
- Parking passes can be paid via PayPal ™.
- Professors can input student marks.
- Students can obtain their current seminar schedule.
- Students can order official transcripts.
- Students can only enroll in seminars for which they have prerequisites.
- Transcripts will be available online via a standard browser.

Release Planning - Step 2

- Let’s Size !!!
  - ➢ Estimates are too difficult early on so let’s first have a sense of the complexity.
- Planning Poker
  - ➢ Stick to 0, 1, 2, 3, 5, 8, 13

Product backlog

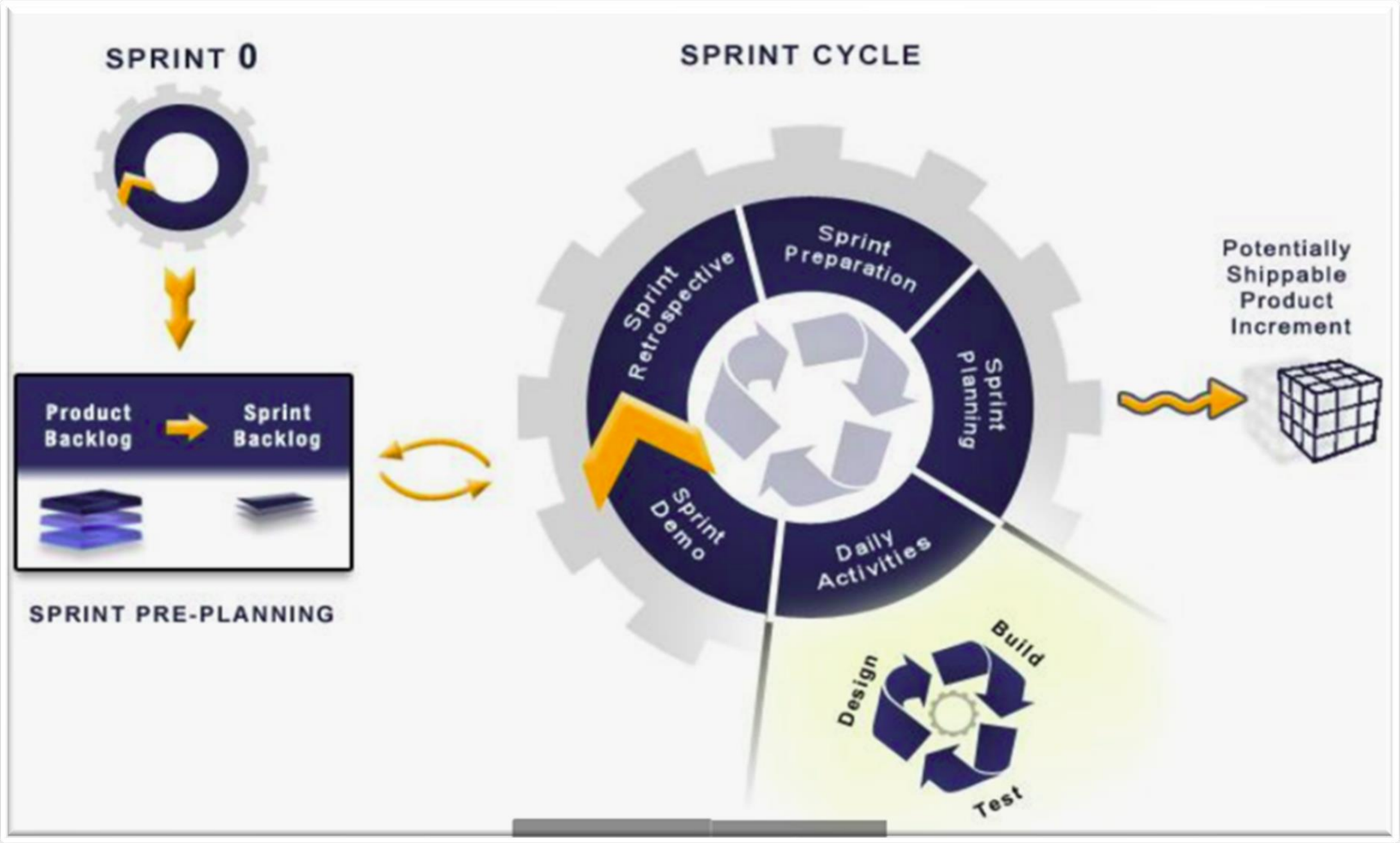


- Always do the most valuable work first.
- The product owner is responsible for determining and managing the sequence of this work.
- This prioritized (or ordered) list is known as the product backlog.
- The development teams break down each targeted feature in the product backlog into a set of tasks.
- This forms a second backlog called the sprint backlog.

Sprints

- Each sprint involves:
  - ➢ Sprint Planning
  - ➢ Sprint Preparation
  - ➢ Daily Activities
  - ➢ Sprint Demo
  - ➢ Sprint Retrospective

Sprint Cycle



During the Sprint

- Once team has committed, no changes to Sprint scope.
- No changes to deliverables.
- Details will emerge during Sprint, but no new work or substantially changes work.
- Customer can terminate the Sprint if necessary.
- No changes to Sprint Duration.
- Sprints end on panned date whether team has completed its commitment or not.

❖ Note: - Customer can make any changes to the remaining Product Backlog before the start of the next Sprint.

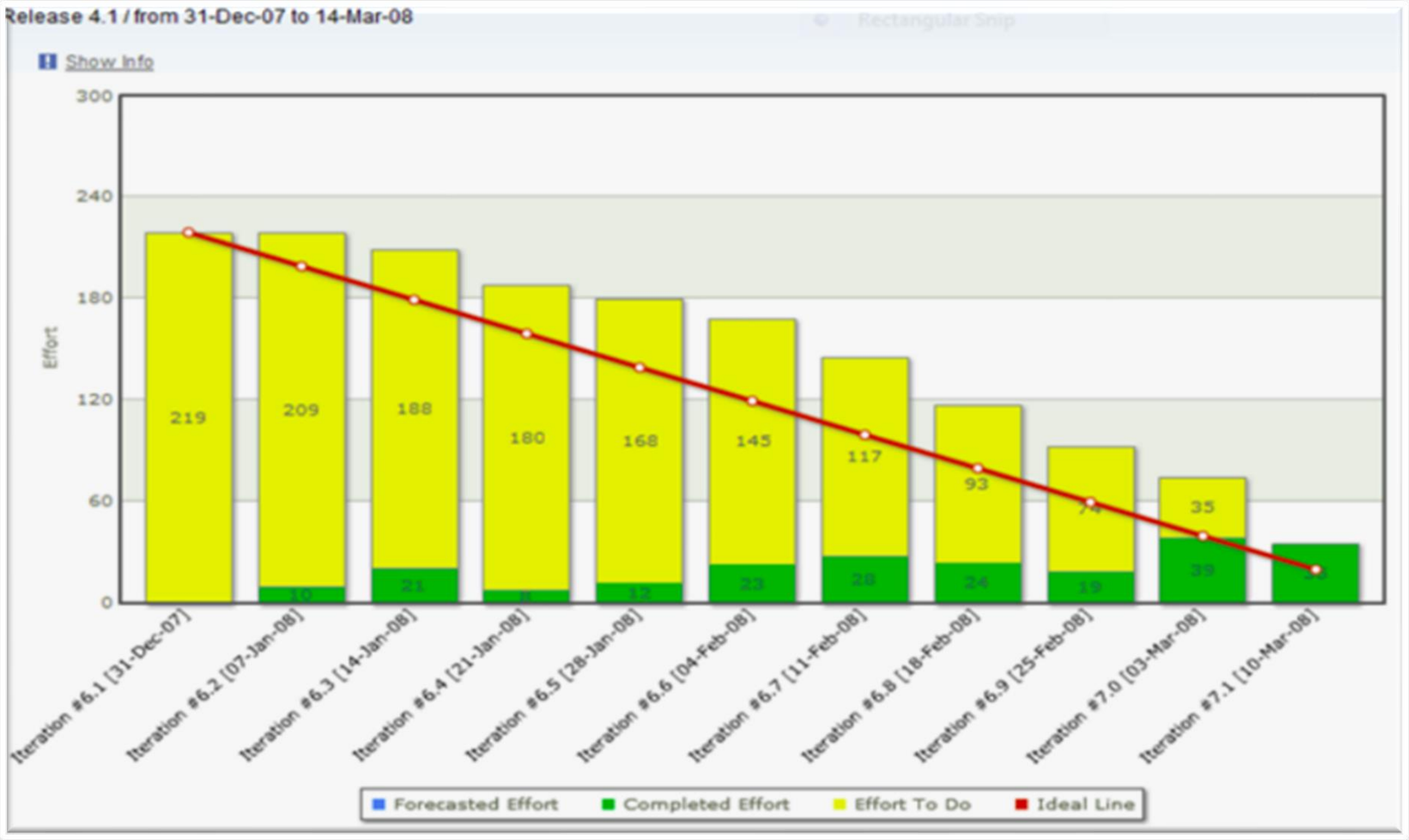
Post-game

- Sprint Demo
  - A walkthrough of all the functionalities developed in that particular sprint in order to collect client feedback.
- Sprint Retrospective
  - A meeting where all team members (including the client) reflect about the past sprint; especially on ‘what went right?’, ‘what went wrong?’, and ‘what could be improved in the next Sprint?’.
- Deliverables
  - Working and tested software (as per the sprint backlog)
  - Sprint Release Notes
  - Test Report

How do we monitor progress?

- Burn-down charts
  - ➢ Iteration
  - ➢ Release
- VELOCITY is a key metric
  - ➢ Number of total story points / One iteration

Burn-down charts



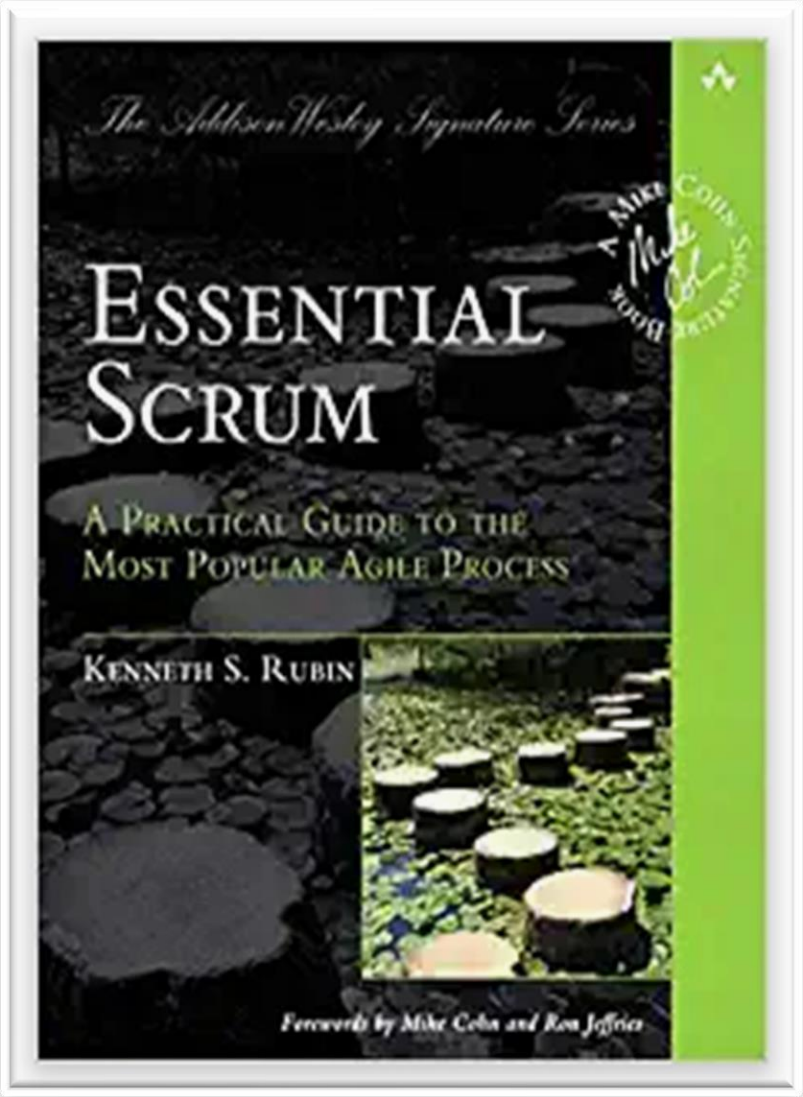
Summary

- Agile stresses four key issues.

1. Importance of self-organizing teams that have control over the work they perform.
2. Communicate and collaborate between team members and between practitioners and their customers.
3. A recognition that change represents on opportunity.
4. Emphasis of rapid delivery of software that satisfies the customers.

Reference

- Ref 1: [Addison-Wesley Signature Series (Cohn)] Kenneth S. Rubin - Essential Scrum_ A Practical Guide to the Most Popular Agile Process (2012, Addison-Wesley Professional)


The Addison Wesley Signature Series
A Mike Cohn Signature Book
Essential Scrum
A Practical Guide to the Most Popular Agile Process
Kenneth S. Rubin
Forewords by Mike Cohn and Ron Jeffries

Reference

- The Art of Agile Development by James Shore and Shane Warden.

- Agile and Iterative Development: A Manager’s Guide by Craig Larman, Agile Software Development Series, Alistair Cockburn and Jim Highsmith, Series Editors.

- http://agilemanifesto.org

- https://msdn.microsoft.com/en-us/library/hh533841.aspx