NANYANG POLYTECHNIC

IT2755
Software Engineering

Overview of Software Engineering & Development Methodologies

NANYANG THE INNOVATIVE POLYTECHNIC

Learning Outcomes

- Define software engineering & its best practices
- Describe Waterfall model within SDLC
- Describe AGILE methodology
- Describe SCRUM framework within Agile

What is Software Engineering?

Software Engineering is a discipline whose goal is the production of:

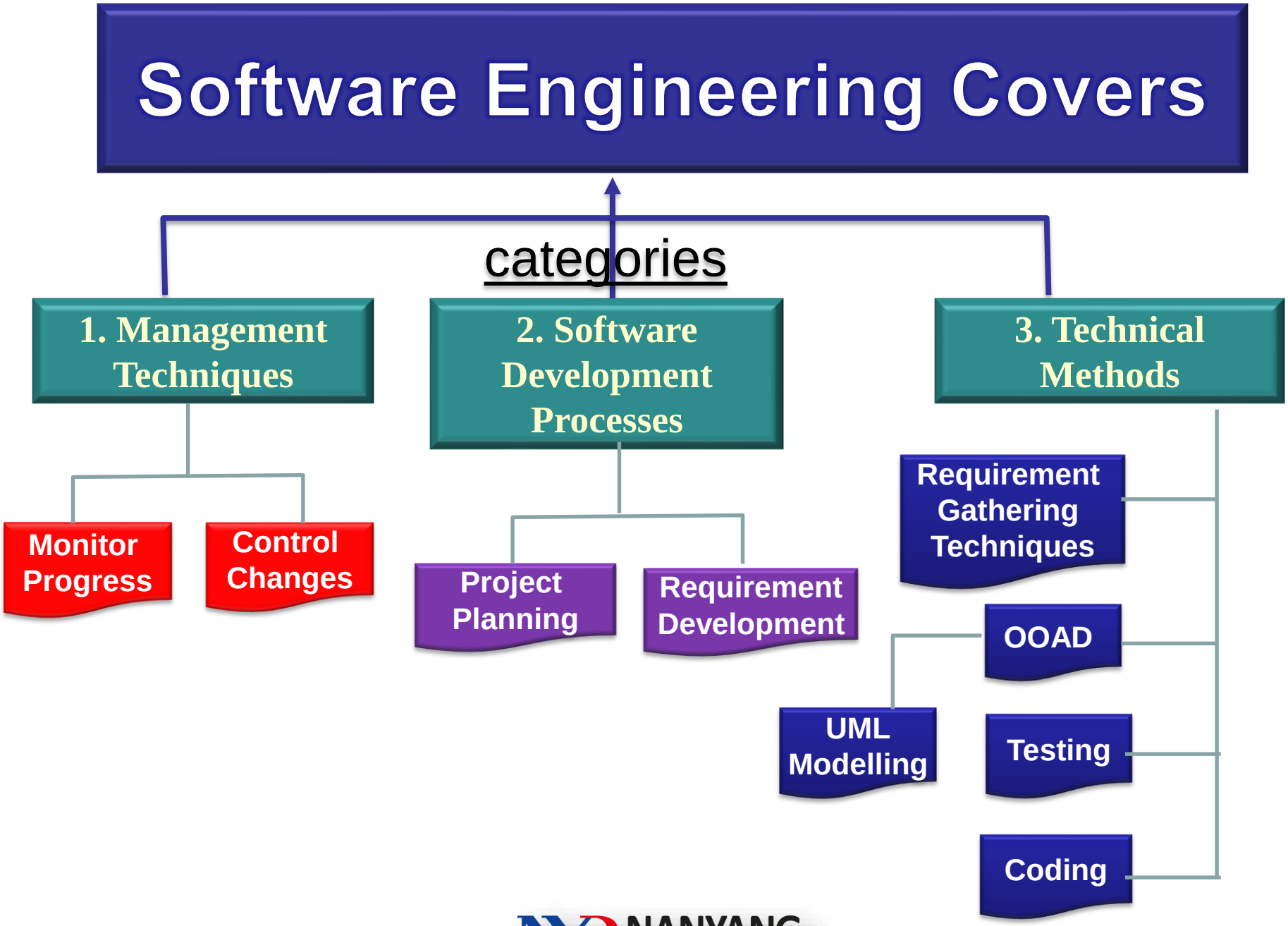
- **quality** software,
- delivered **on time** and
- **within budget**, that
- **satisfies the user's needs**

It includes:

- **processes**,
- **management techniques** and
- **technical methods**

to achieve the software engineering goals.

Software Engineering Covers



Software Development Life Cycle (SDLC)

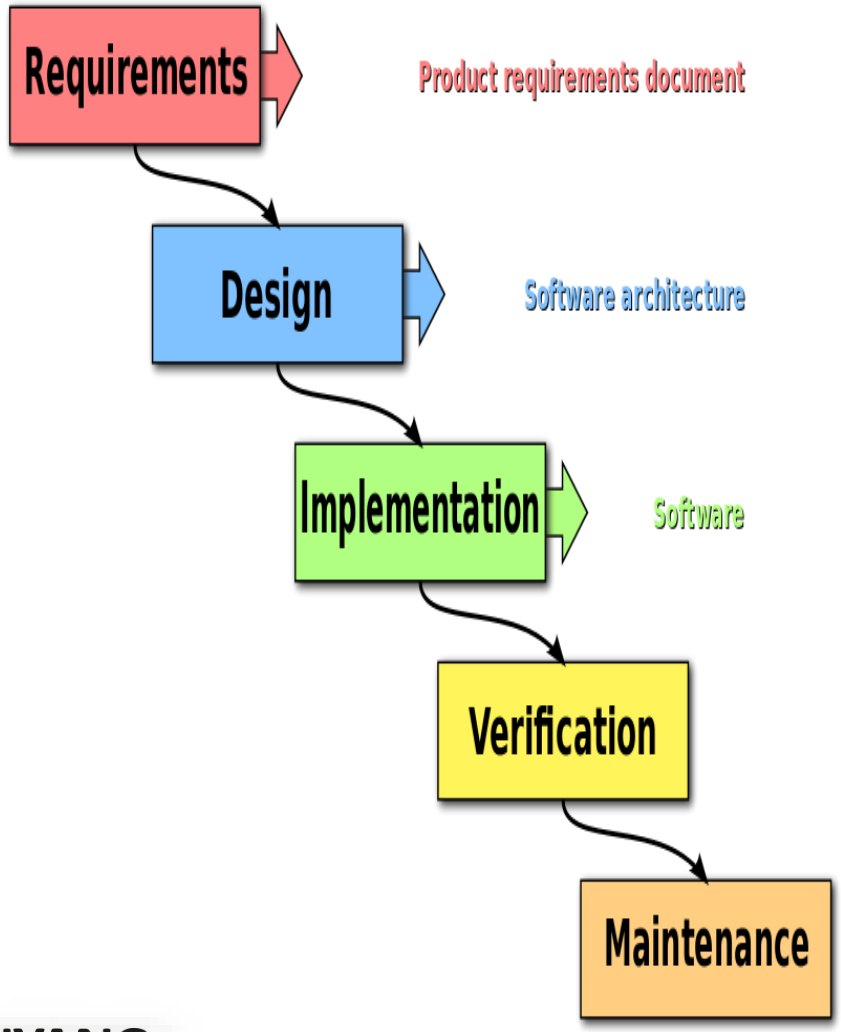
- **SDLC**: process of building, deploying, using, and maintaining a system
- **Five phases** of project development:
  1. **P**lanning,
  2. **A**nalysis,
  3. **D**esign,
  4. **I**mplementation,
  5. **S**upport

SDLC Phases and their Objectives

SDLC PHASE	OBJECTIVE
Project Planning	To identify the scope of the new system, ensure that the project is feasible, and develop a schedule, resource plan, and budget for the remainder of the project
Analysis	To understand and document in detail the business needs and the processing requirements of the new system
Design	To design the solution system based on the requirements defined and decisions made during analysis
Implementation	To build, test, and install a reliable information system with trained users ready to benefit as expected from use of the system
Support	To keep the system running productively initially and during the many years of the system's lifetime

Waterfall Methodology

- **Waterfall** - A highly structured design process that relies heavily on up-front planning and a set of **sequential** and **prescribed steps** like a waterfall.



Waterfall Methodology

- **Pros:** Offers a simple approach to software development with stable requirements.
- **Cons:** Not suitable to develop solution with complex and changing requirements
- Customer need to wait for a long time.
- **When to use:** For projects with very stable requirements

Agile methodolgy

- **Agile** aims to cope with changes fast through **iterative** and **incremental development.**

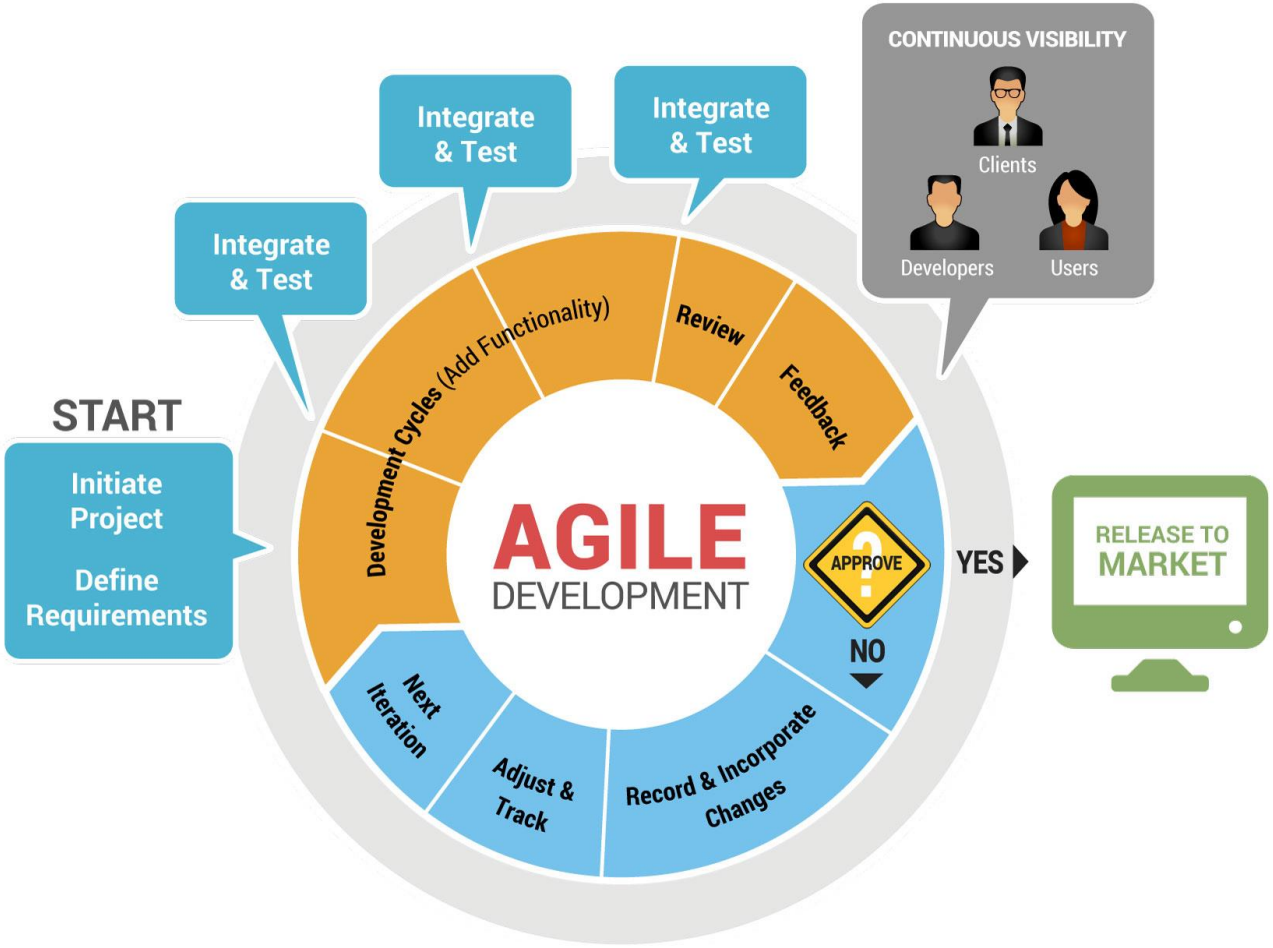
Ie. Repeative rounds of development steps to produce system that gradually grow with improved features

- **Agile Software Development** produces solutions through evolution and collaboration between self-organizing, cross-functional teams by utilizing the appropriate practices. Eg SCRUM.

Agile methodology

- Agile development methodology provides opportunities to assess & control the progress of a project throughout the development lifecycle

Agile Development



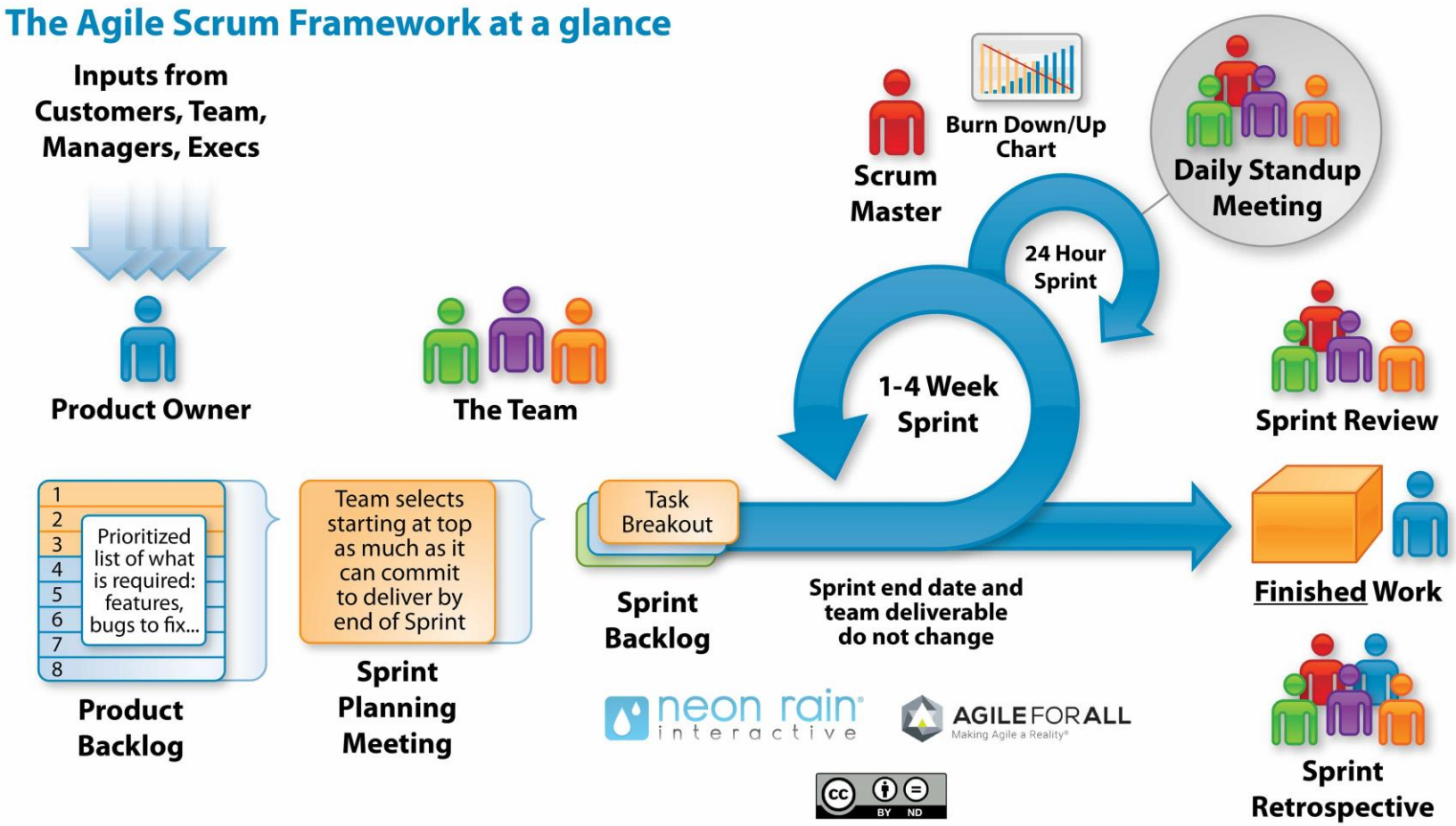
Agile Methodology

- **Pros:** Working product is delivered frequently (weeks rather than months)
- Changes can effectively be taken care of during rounds of development cycles
- **Cons:** Lack of emphasis on necessary designing and documentation
- **When to use:** project with unstable user requirements + customer's involvement is available during development cycles.

SCRUM

- SCRUM is one of the process framework within the Agile development that has been used to manage complex product development.
- SCRUM uses fixed-length iterations, called Sprints, which are typically 2 - 4 weeks long for the cross functional team to produce shippable product that the customer wants.

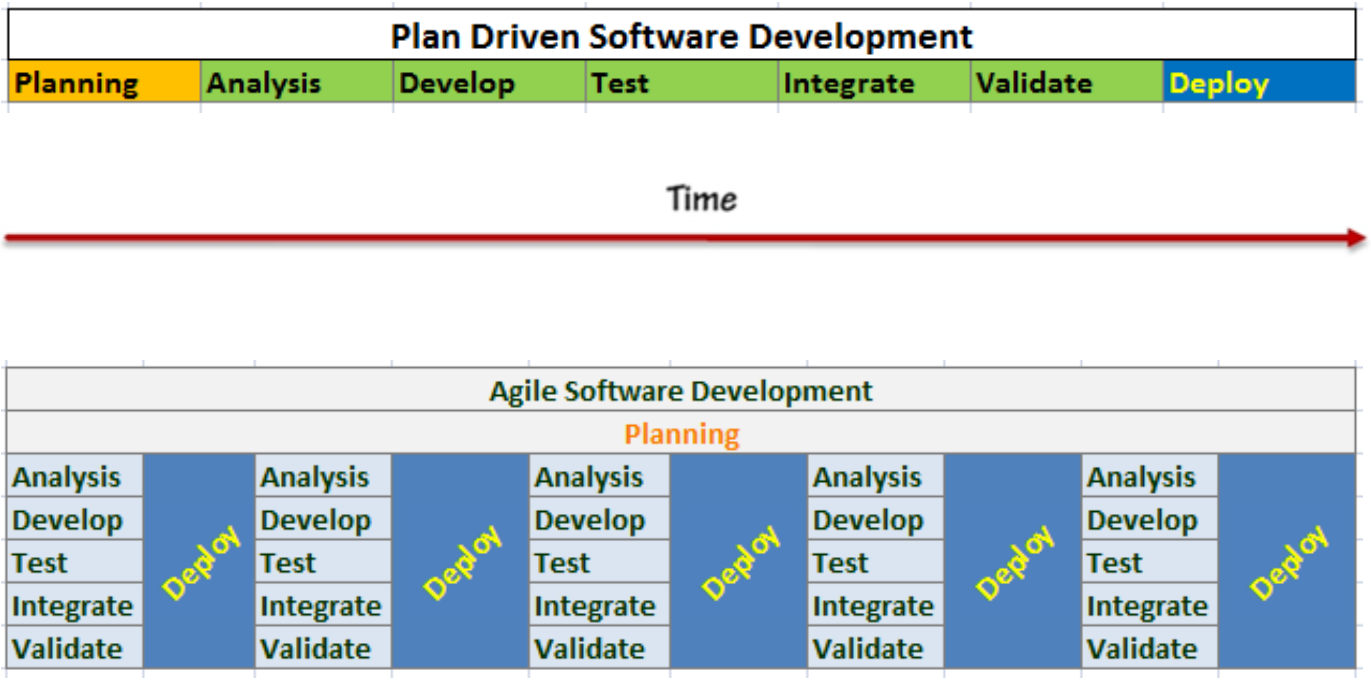
Scrum Framework



Agile Methodology - Scrum

- **2 Artifacts**
  - Product backlog
  - Sprint backlog
- **3 Roles**
  - Product Owner (set direction of the product)
  - Scrum Master (facilitator)
  - Team (developers, testers)
- **4 Events**
  - The sprint
  - Sprint Planning
  - Daily Scrum
  - Sprint Review

Planned (SDLC) vs. Agile



Summary

- Software engineering is concerned about development of quality software.
- **SDLC**- an approach consists of a set of planned activities required to complete development project.
- **Waterfall** - A highly structured development process that relies heavily on up-front planning and a set of **sequential** and **prescribed steps.**
- **Agile** - an approach that based on **iterative** and **incremental development**, where system solutions evolve through collaboration between self-organizing, cross-functional teams. **Scrum** is one of the Agile process frameworks.