

Rapid Application Development

Lecture 1

Binuri Raigamkorale
binuriyr@dcs.ruh.ac.lk

Today's Outline

- Introduction to software development life cycles.
- Software development methodologies
- Agile
 - Introduction to scrum.
 - Concept of sprints and scrum terminology
 - Scrum roles
- Waterfall
- RAD

Learning Objectives

- Explain how to apply Agile methodology to a development process
- Understand the principles behind the Agile.
- Explain the significance of Scrum & the usage of Scrum
- Understand and apply the Scrum framework practices
- Identify the Scrum roles and their activities
- Describe the Scrum terminology Sprinting, backlog, stories, standup s.
- Understanding the purpose of using RAD in the industry.

Introduction to software development life cycle.

- The life cycle defines a methodology for **improving the quality** of software and the overall development process.
- SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhancements.

Why development process is important ?

For any company..

- Customer satisfaction is critical
- Project should successfully meet the deadlines.
- Different teams, different location, distributed developments

Introduction to software development life cycles cont..

7 phases of SDLC

System Planning

feasibility study and practicality of the system

System Analysis

defines system from users' view
study old system and identify new requirements

System Design

design new/alternative system
design system from technical view

System Development

new software is acquired, developed and tested.

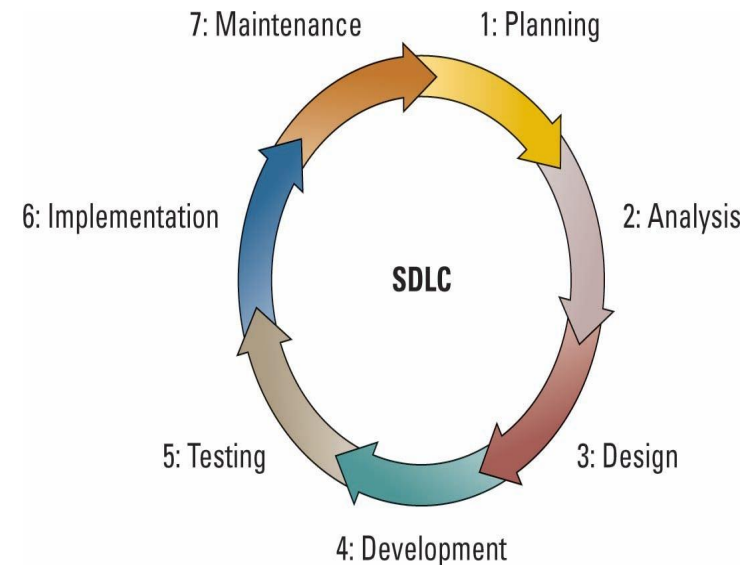
System Testing

System Implementation

System installation and training

System maintenance

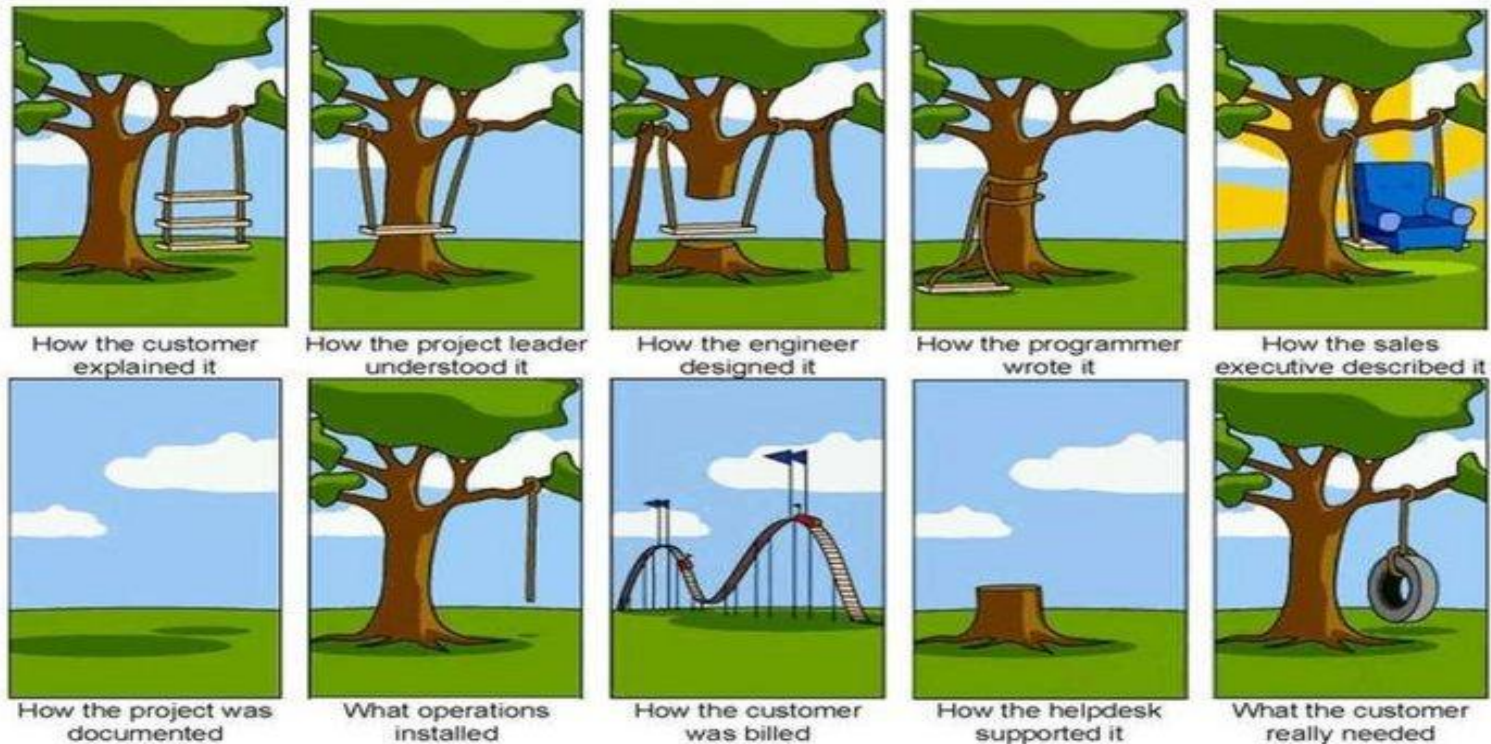
periodic evaluation and updating



Introduction to software development life cycle cont..

An average of 70% software development projects fail!! So Why SDLC goes wrong?

- Ambiguity problem of requirements gathering.



- No meetings in the timelines
- Costs overrun

Software development methodologies

Custom software development companies use many software development methodologies for their day-to-day operations, there are pros and cons associated with each.

The basic purpose of these methodologies is to provide smooth software development according to the project requirements.

Ex:

- Agile

- Waterfall

- Iterative

- Rapid

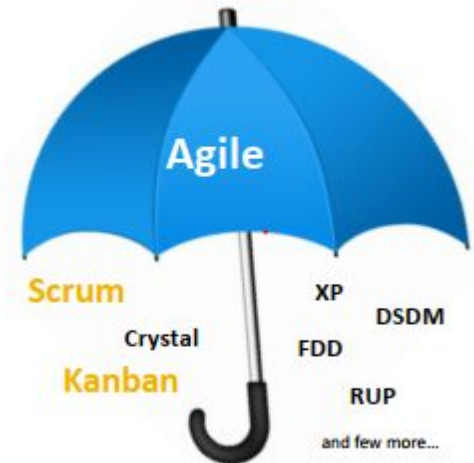
What is Agile?

- Individuals and interactions over process and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan.



What is Agile? Cont..

- Agile approach, teams develop in short sprints or iterations, each of iteration includes a well defined duration and list of deliverables, but in no particular order.
- During sprints, teams work towards the goal of delivering working software deliverables. Focusing on team strengths and efficiency, along with internal feedback from various departments and clients.
- Client satisfaction is the highest priority, which teams achieve by continuously delivering working, tested, prioritized features.
- In general, **agility** is defined as
"the ability to both create and respond to
Change in order to profit in a turbulent
environment".
- Famous types of agile methodologies.
Scrum, Kanban, XP, etc.



Why Agile?

To adapt

- Changes in Requirements
- Changes in Design, Implementation
- Changes in Technology
- Changes in Team
- Changes in users/client contact

Practices including

- version control.
- setting coding standard.
- giving weekly demos to your stakeholders.



Agile principles

- **Deliver Value Faster** – Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- **Welcome Change** – Welcome changes to requirements, even late in projects. Agile processes harness that change for the customer's competitive advantage. Deliver Working
- **Work Together Daily** – Business people and developers must work together daily throughout the project.
- **Build Projects Around Motivated Individuals** – Give them the environment and support they need and trust them to get the job done.
- **Face-to-Face Conversations** – The most efficient and effective method of conveying information to and within a development team is with face-to-face conversation.
- **Working Software is Key** – Working software is the primary measure of progress. Working software should be delivered after a couple of weeks to a couple of months, with a preference to the shorter timescale.

Agile principles cont..

- **Sustainable Development** – Agile processes promote sustainable development. The sponsors, project team members (developers), and users should be able to maintain a constant pace indefinitely.
- **Attention to Technical Excellence** – Continuous attention to technical excellence and good design practices enhances agility. (robustness of the development for customers new requirements)
- **Simplicity** – The art of maximizing the amount of work not done is essential.
- **Self-Organizing Teams** – The best architectures, requirements, and designs emerge from self organizing teams.
- **Reflect and Adjust** – At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Introduction to Scrum

Terminology

Sprints

Stories

Stand up meetings

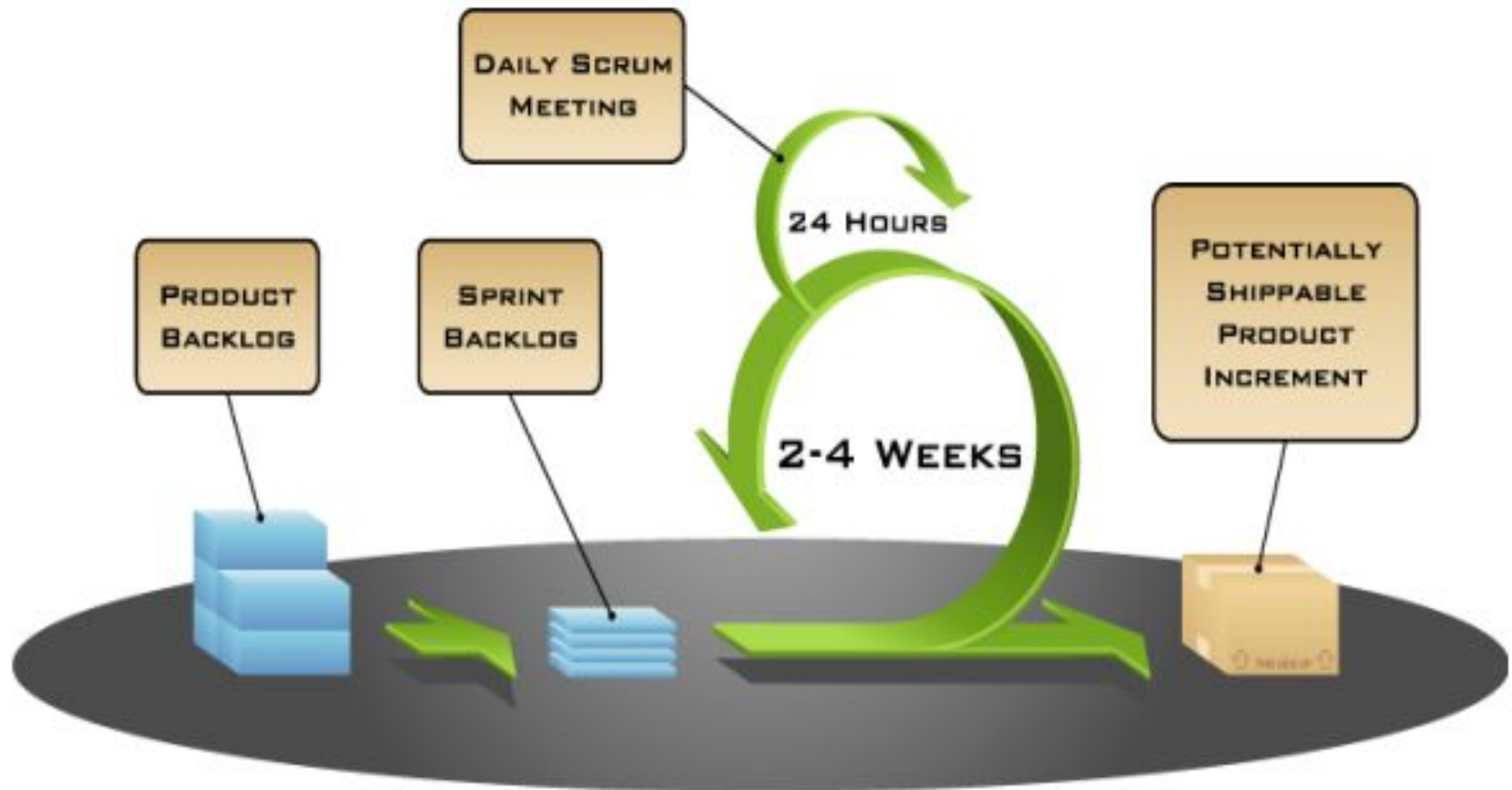
Agile board

Backlog

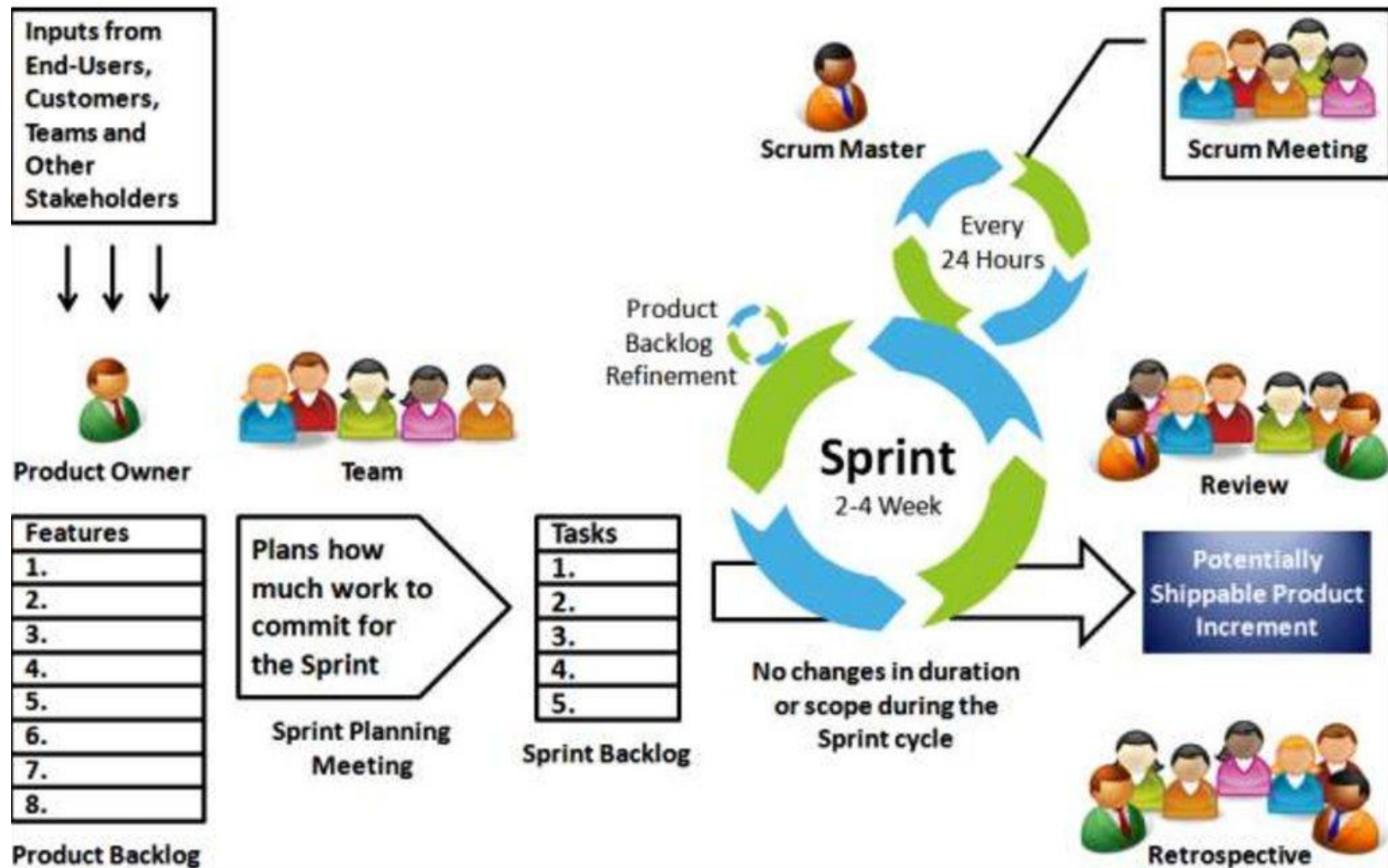
Scrum Roles

For more information : <https://www.atlassian.com/agile/tutorials/sprints>

Scrum in Nutshell



Scrum in nutshell



Introduction to Scrum cont.

Sprinting

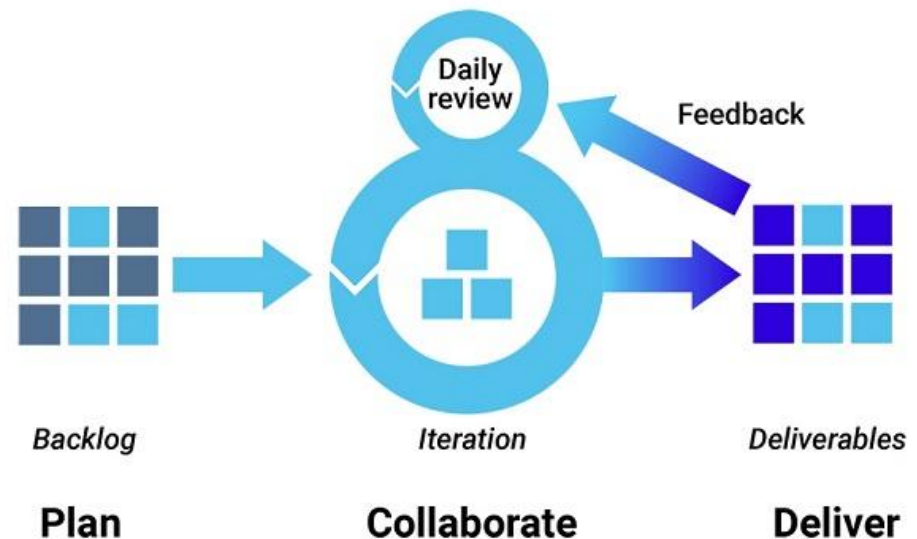
Sprints are a short iteration, usually between one to three weeks to complete, where teams work on tasks determined in the sprint planning meetings.

As you move forward, the idea is to continuously repeat these sprints until your product is feature ready.

Once the sprint is over, you review the product to see what is and isn't working, make adjustments, and begin another sprint to improve the product or service.

Each Sprint involves:

- Sprint planning
- Sprint preparation
- Daily Activities
- Sprint Demo
- Sprint Retrospective



Introduction to Scrum cont.

Stories

Story is a high-level definition of a work request.

It contains just enough information so the team can produce a reasonable estimate of the effort required to accomplish the request.

This short, simple description is written from the user's perspective and focuses on outlining what your client wants (their goals) and why.

Backlog

As project requests are added through your intake system, they become outstanding stories in the backlog.

During Agile planning sessions, your team will estimate story points to each task.

During sprint planning, stories in the backlog are moved into the sprint to be completed during the iteration.

Managing your backlog is a vital role for project managers in an Agile environment.

Introduction to Scrum cont.

Standup Meetings

Daily stand up (under 10 minutes), also known as “daily Scrum meetings,” are a great way to ensure everyone is on track and informed.

These daily interactions are known as “stand up” because the participants are required to stay standing, helping to keep the meetings short and to the point.

Introduction to Scrum cont.

Scrum Roles

1. Scrum Master:

A Scrum Master is the person responsible for making sure a Scrum team is operating as effectively as possible with Scrum values. He is responsible for:

- Facilitate daily Scrum meetings (also called “daily standups”)
- Lead sprint planning meetings
- Conduct “retrospective” reviews to see what went well and what can be improved for the following sprint
- Keep a pulse on team members, through individual meetings or other means of communication
- Manage obstacles that arise for the team by communicating with stakeholders outside of the team

Introduction to Scrum cont.

Scrum Roles

2. Product owner

A product owner makes sure the Scrum team is aligned with the goals of the overall product that the team is contributing to.

They understand the **business needs of the product**, Because they have to understand how the Scrum team fits into bigger picture goals, product owners usually stay in touch with product managers and other stakeholders outside the team.

Responsible for:

- Manage the product backlog by ordering work by priority
- Set the product vision for the team
- Communicate with external stakeholders and translate their needs to the team
- Make sure the team is focused on hitting product needs through communication and evaluating progress

Introduction to Scrum cont.

Scrum Roles

3. Development team

A development team is composed of professionals who are working for completing the tasks in a Scrum sprint.

Development team members can be SE engineers, tech. managers, QA Engineers, tech leads, architects, UI designers, data analysts, etc. to reach sprint goals, collaborate to map out goals and plans for achieving them.

The responsibilities of a development team will also depend on the end goals of the Scrum team. They are Responsible for:

- Help in sprint planning and goal setting
- Lend expertise to program, design, or improve products
- Use data to find best practices for development
- Test products and prototypes, plus other forms of quality assurance

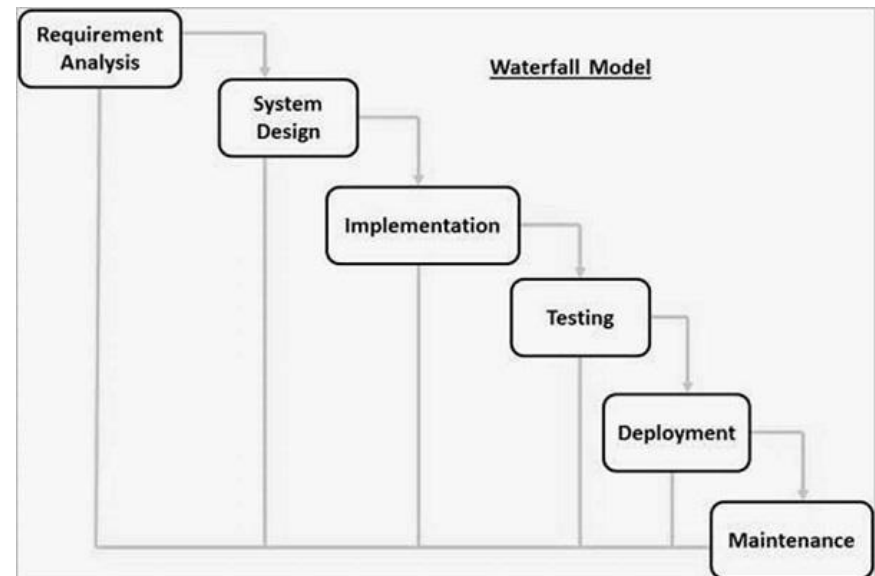
Scrum roles – as a summary

3 Key Roles	Key Value	Some Key Deliverables
Product Owner:	Accountable for maximizing the value of the product	<ul style="list-style-type: none"> ✓ Clearly expressing the Product Backlog items ✓ Ordering the items by Priority Order ✓ Optimizing the value of the work ✓ Ensuring that the PBIs are visible, transparent and clear to all ✓ Ensuring the Development Team understands the items listed
Scrum Master:	Servant Leader who is responsible for promoting and supporting SCRUM as defined in the SCRUM Guide.	<ul style="list-style-type: none"> ✓ Ensuring the goals, scope, and product are understood by all ✓ Removing impediments from the project ✓ Product release planning ✓ Facilitating SCRUM Events ✓ Planning SCRUM Implementations
Development Team: <ul style="list-style-type: none"> ▪ Developers ▪ Technical Analyst ▪ Business Analyst ▪ Solution Architect ▪ IT Operations 	Self-Organizing Team that is responsible to get the work completed.	<ul style="list-style-type: none"> ✓ Develop solution and build out solution ✓ Update system of record (ADO) project status ✓ Documentation (Help Guides, FAQs, Test Scripts, Presentations) ✓ Documentation Created for Architecture Review ✓ Documentation for Evergreen Support identified, created and logged ✓ Fully testing functionality for every increment and release

Introduction to Waterfall

The Waterfall model is the earliest SDLC approach that was used for software development.

- The waterfall Model illustrates the software development process in a **linear sequential flow**. This means that any phase in the development process begins only if the previous phase is complete.
- Applicable for:
 - Requirements are very well documented, clear and fixed.
 - Product definition is stable.
 - Technology is understood and is not dynamic.
 - There are no ambiguous requirements.
 - Ample resources with required expertise are available to support the product.
 - The project is short.



Scrum vs Waterfall

Scrum takes its cue from RUGBY SCRUM where a team is aggressively trying to advance the ball and working together as one unit.

Scrum says we need to have a cross functional team and they have to be focused on advancing the common goal.



Waterfall method is the relay race approach where every individual is doing their part and then passing the baton to the next person.

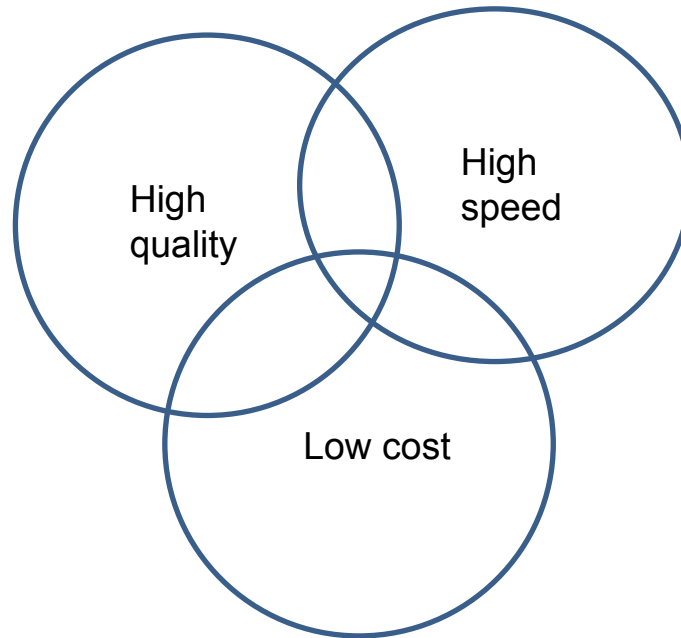


Introduction to RAD

RAD is a concept that products can be developed faster and of higher quality through:

- Gathering requirements using workshops or focus groups
- Prototyping and early, reiterative user testing of designs
- The re-use of software components
- A rigidly paced schedule that refers design improvements to the next product version
- Less formality in reviews and other team communication

Aim of RAD



More details will be covered in next lecture

Exercise

1. Merits, drawbacks and uses of SDLC methodologies.
1. Discuss the what types of software developments can be applicable for each SDLC methodologies.

software development methodologies - comparison

Parameter	Waterfall	Prototyping	Incremental	Spiral	RAD
Development type	Sequential	Iterative	Sequential + Iterative	Sequential + Iterative	Iterative
Most Appropriate for	Mainframe Applications	Small to Medium Applications	Web Applications	Real-time Applications	Small Applications
Least Appropriate for	Web Applications	Mainframe Applications	Interactive Applications	Low Risk priority Applications	Real-time Applications
Strengths	Software development measurable and Conserve resources	Encourages innovation and quick to implement	Customers involved at all stages and quick to implement	Promotes risk control and can include other methodologies	Can be modified rapidly and close to implementation at every stage
Weaknesses	Inflexible and excessive documentation	False customer expectations and may lead to poor design	Difficult implementation issues delayed	No firm deadlines and may end up waterfall model like	No firm deadlines and Difficult implementation issues delayed

References

- Scrum Guides. "[The Scrum Guide](https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf),
<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>." Accessed August 17, 2021.
- <https://scrumguides.org/scrum-guide.html>
- <https://www.orangescrum.com/tutorial/agile-scrum-management-an-overview/understanding-the-scrum-team-and-scrum-roles>