What is Software Engineering?

Software engineering is a systematic and disciplined approach to developing software systems that meet user requirements. It involves the entire software development life cycle, from understanding the user's needs and defining the software requirements, to designing, coding, testing, and maintaining the software. The process includes techniques and tools to ensure software quality, reliability, and maintainability.

Example of Software Engineering in real life

Let’s say our college needs to build a website for campus dining then they will hire a software engineering team to build that website. The team will follow a systemic approach like below:

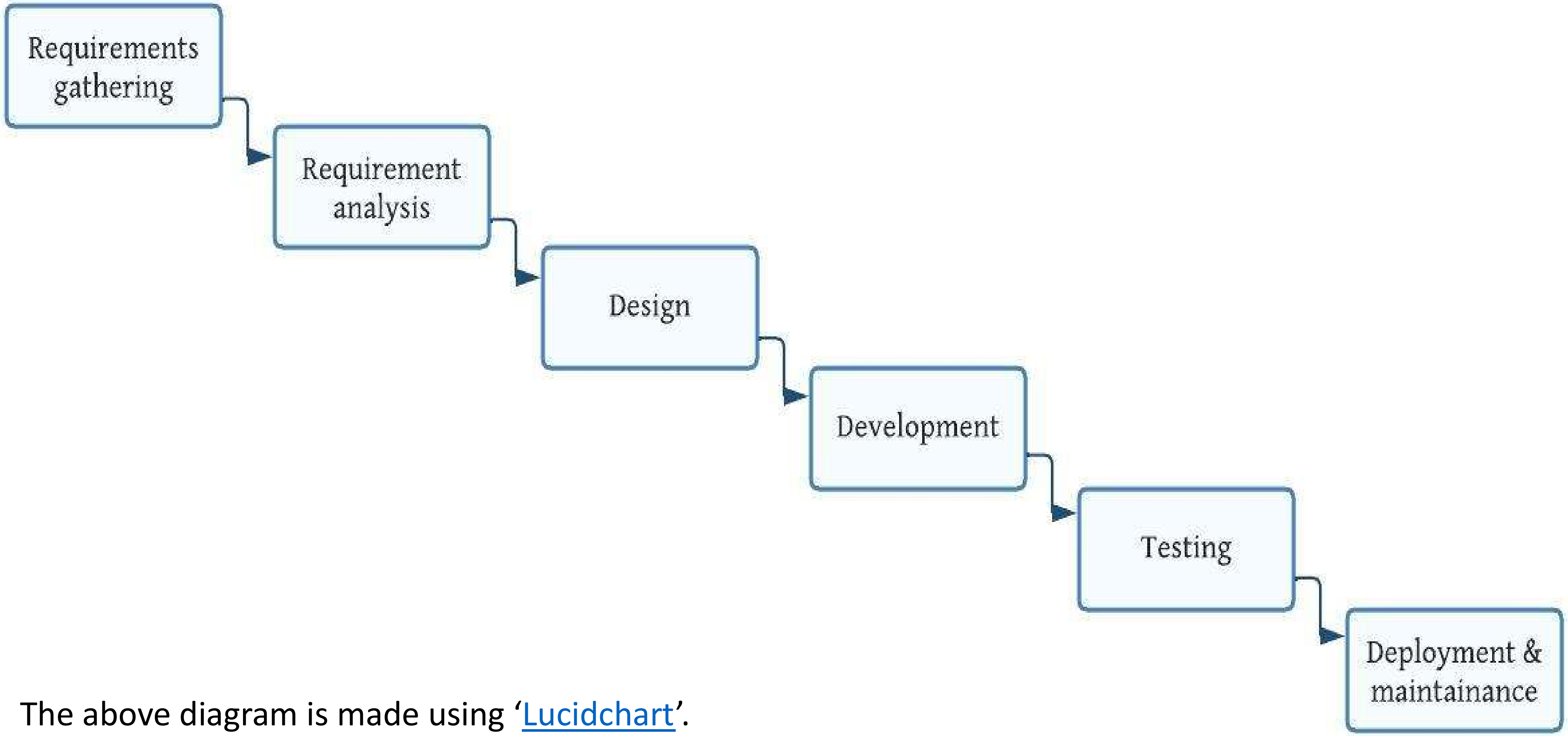
- Requirement gathering: the software team meets with our college dining department to understand their requirements and what features they are looking for in their website

- Design: based on the requirements they gathered they will design a model/blueprint of the website which includes the user interface and the tools and technologies they are going to use to build it

- Implementation: they will write the code and build the website based on the blueprint.

- Testing: the team will test the code to make sure it works as expected and to check whether it meets the user requirements or not

- Deployment: in this phase team will release the website to the public/users. They will maintain the website by fixing issues or adding features etc…



The above diagram is made using ‘Lucidchart’.

SDLC: Waterfall model

Waterfall model Approach

The waterfall model is also called a linear sequential model. Each phase is completed first before moving to the next phase. Phases distinct and separate from each other it makes easier to plan and control the project.

Requirements gathering and analysis: working with the Product Owner to understand a software system's requirements.

Design: In this phase, a team of software architects and designers works with the requirements to create a blueprint for the software system.

Development: writing code to implement the software system, following best practices for readability, maintainability, and performance.

Testing: writing and executing tests to verify that the software system meets the requirements and works as intended.

Deployment and maintenance: fixing bugs, adding new features, and updating the software system over time to meet changing requirements and maintain its health

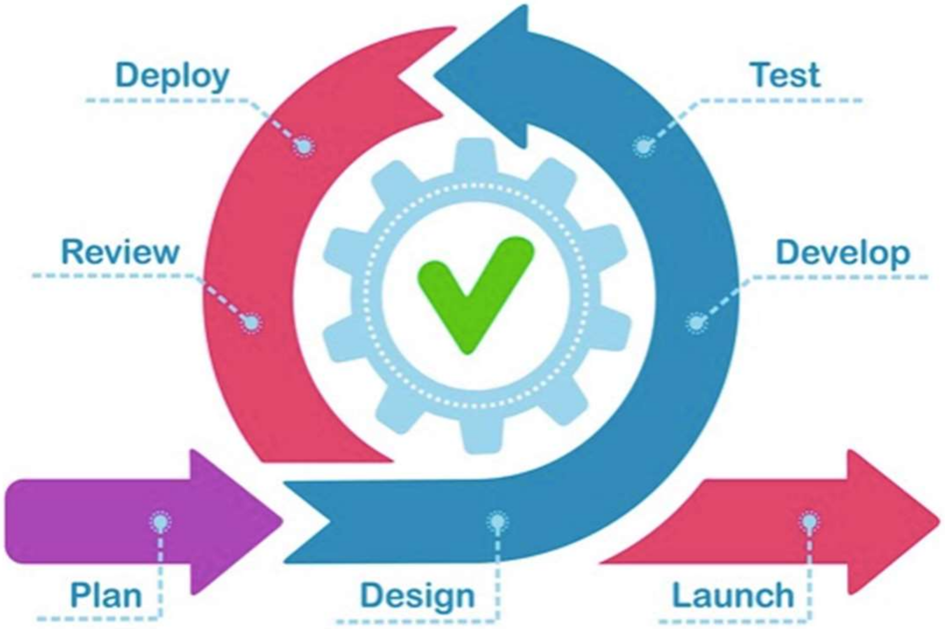
Advantages and disadvantages of Waterfall model

- Advantages:
- 1. Simple and easy to understand and use
- 2. Easy to manage
- 3. Phases are processed and completed at a time
- 4. Requirements are well understood

- Disadvantages:
- 1. No working software is produced until late during the life cycle
- 2. A high amount of work
- 3. Poor for large projects
- 4. Little opportunity for customer feedback
- 5. All requirements must be known upfront

Agile Methodolog y

Agile is a method for delivering software projects. It involves working in small, time-limited stages and regularly checking in with the customer to ensure their needs are being met. Agile is an iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver all at once. In iteration 1 we take 2 or 3 features, then we plan, design, develop, test, and review, in iteration 2 we add another 2 or 3 features on top of the iteration 1 feature so we plan, design, develop, test, and review. Each iteration is meant to be iterative and increment is from iteration 2 onwards we are adding/increasing the features to existing features.



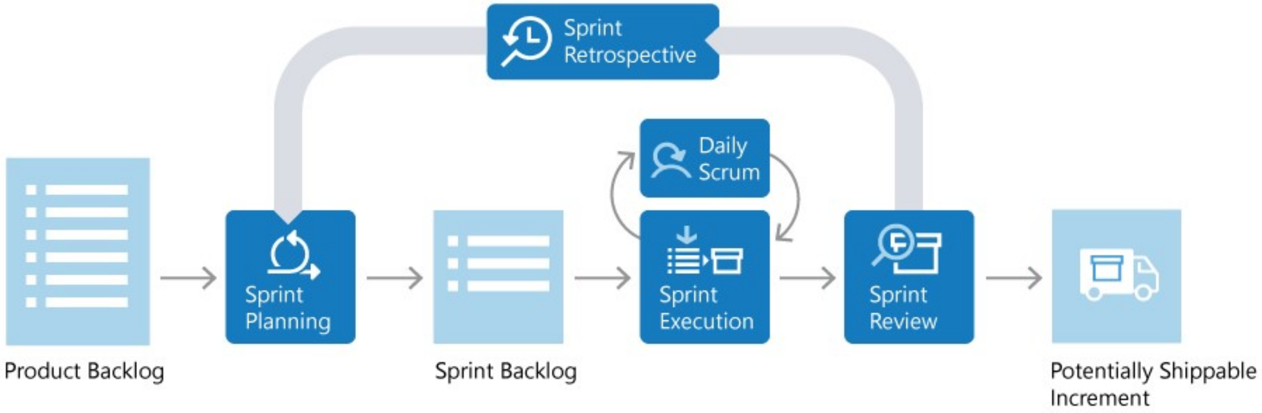
Four main tenets of Agile

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

The manifesto doesn't imply that the items on the right side of these statements aren't important or needed. Rather, items on the left are simply more valued.

Scrum Agile approach

Scrum is an Agile framework that provides a structure for teams to collaborate and iteratively develop, deliver, and improve products. It consists of defined roles such as Product Owner, Scrum Master, and Development Team, and has key practices such as Sprint, Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective. The goal of Scrum is to help teams deliver valuable, high-quality products in an efficient and flexible manner.



Source of image ‘learn.microsoft.com’

Scrum Agile Approach

In a nutshell, Scrum requires a Scrum Master to foster an environment where:

1. A Product Owner orders the work for a complex problem into a Product Backlog.
2. The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
3. The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.

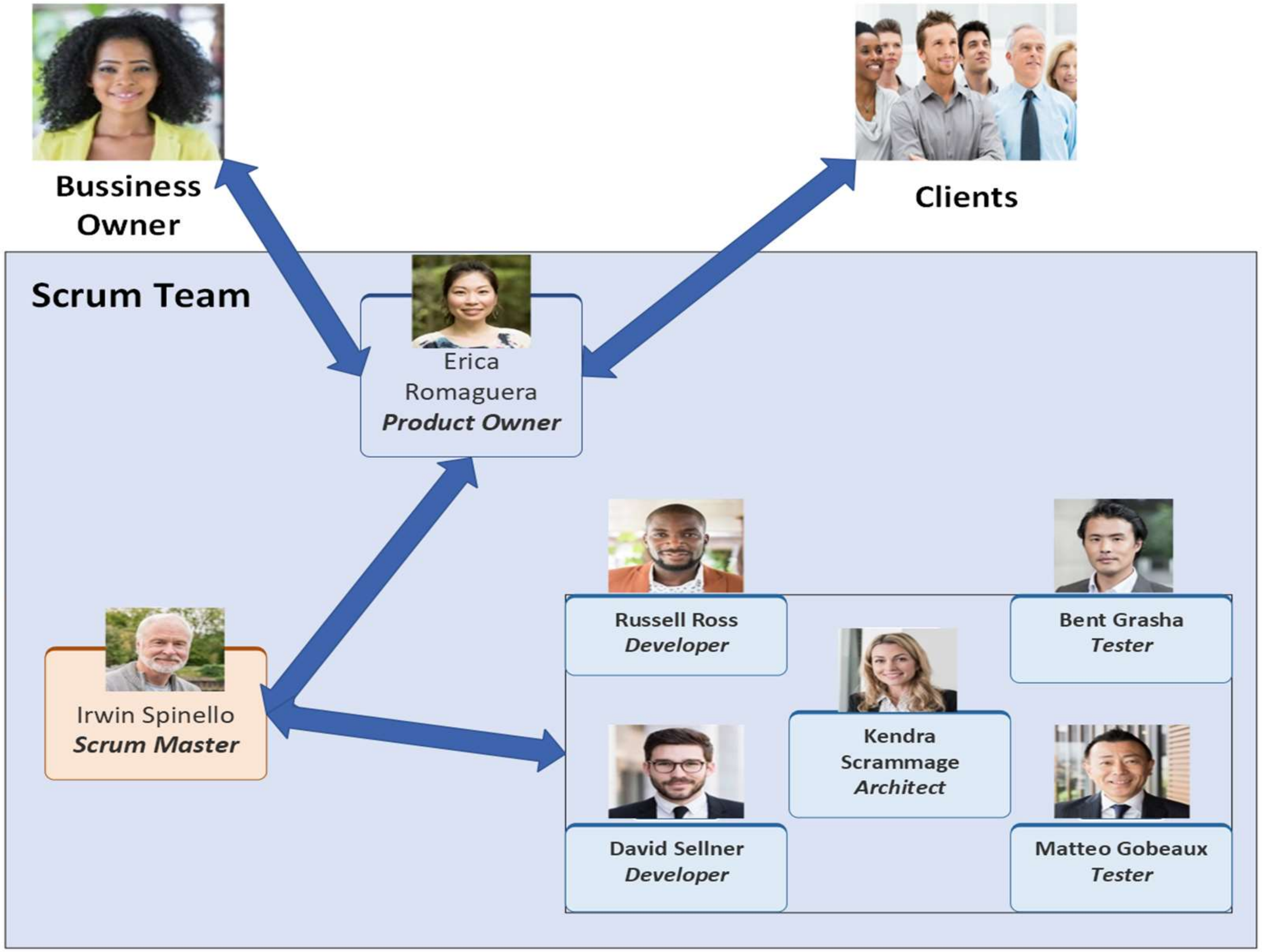
👍

1. It helps to maintain transparency.
2. Large and complex projects can be separated into partially manageable parts.
3. Developments are coded and tested during sprint review.
4. Works well for fast-moving development projects.

👎

1. It is very challenging to adopt scrum in large teams
2. It requires experienced personnel
3. Quality is difficult to achieve until the team undergoes a tough testing process.

Relation between Scrum team and PO



Bussiness Owner

Clients

Scrum Team

Erica Romaguera
Product Owner

Irwin Spinello
Scrum Master

Russell Ross
Developer

Bent Grasha
Tester

Kendra Scrammage
Architect

David Sellner
Developer

Matteo Gobeaux
Tester

The image was made using Visio. And the Images and data are collected from lucid chart

Product and Sprint Backlogs

Product Backlog:

The product backlog can be thought of as a task list for a team working on a product, such as a website. The list includes all the tasks that need to be completed to improve and complete the product. The person responsible for managing the list is called the "product owner." They ensure that the list is always up to date and that the most crucial tasks are at the top.

Sprint Backlog:

The sprint backlog is a list of tasks taken from the product backlog that the development team has agreed to complete during a specific sprint. At the beginning of each sprint, the team selects items from the product backlog that they feel they can complete within the sprint's timeline. These items are then added to the sprint backlog. This list serves as a detailed plan for the sprint and gives the team a clear understanding of what they need to accomplish and by when. It also helps to guide daily meetings and keep the team focused on their goals.

Scrum Meetings

Sprint Planning:

Sprint planning is about two things what and how?
What: it describes what tasks are going to do in the upcoming sprint.
How: it describes how they are going to achieve the tasks that are about to be done in the next sprint.
During the sprint planning meeting, the product owner chooses which PBIs fit the sprint goals. Those decisions are made in collaboration with the development team so they know exactly what work is for the upcoming sprint.
Generally, each sprint is around 3-4 weeks

Daily Scrum:

Also known as the daily stand-up, this is a short, daily meeting where each team member gives an update on what they did yesterday, what they plan to do today, and if there are any obstacles in their way.

Sprint Review:

This meeting is held at the end of each sprint to review what was accomplished during the sprint and to demonstrate the product to stakeholders.

Scrum Meetings

Sprint Retrospective:

Scrum master facilitates the retrospective meeting.
Follow 5 steps to an effective retrospective

- o Set the stage
- o Gather data (Capture what went well, what could have been gone better, Ideas)
- o Generate Insights
- o Decide what to do
- o Close the retrospective

Product Backlog Refinement:

The goal of the Product Backlog Refinement meeting is to keep the Product Backlog healthy and prepare for the upcoming 2 sprints at least. PBIs at the top of the Product Backlog is reviewed to ensure they are ready for the next sprint. Priorities, estimates, and stories are revisited for any changes to be ‘ready’ for the next sprints.
Takes place at least once per sprint

THANK YOU