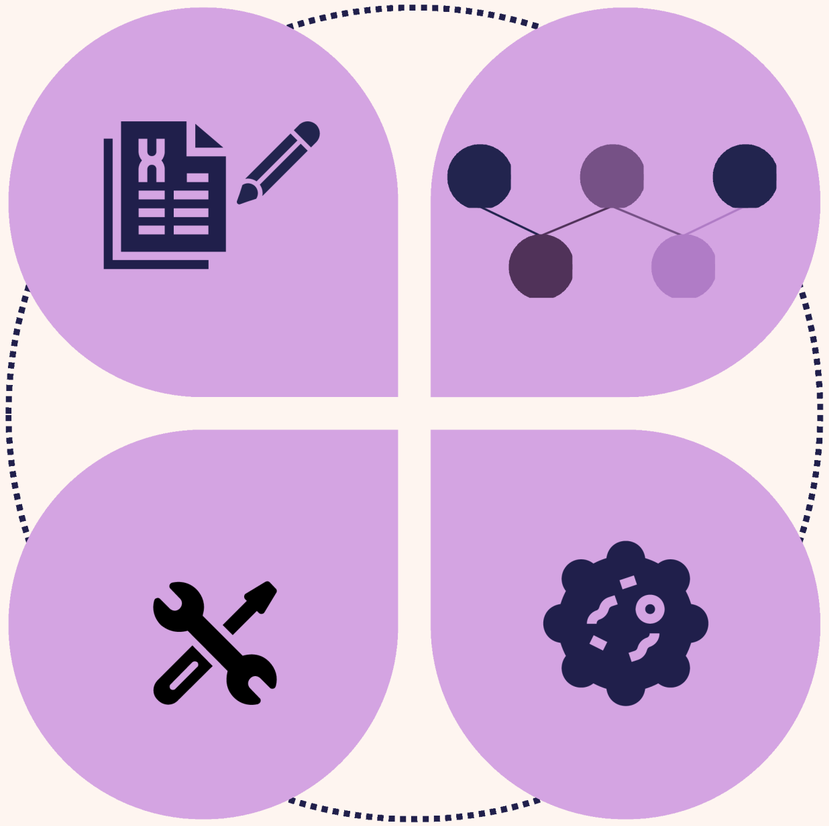
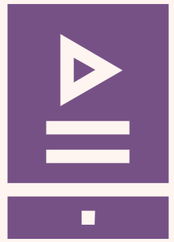
عن عُمَرَ بْنَ الخَطَّابِ قَالَ: سَمِعْتُ رَسُولَ اللَّهِ ﷺ يَقُولُ: "إِنَّمَا الأَعْمَالُ بِالنِّيَّاتِ، وَإِنَّمَا لِكُلِّ امْرِئٍ مَا نَوَى، فَمَنْ كَانَتْ هِجْرَتُهُ إِلَى دُنْيَا يُصِيبُهَا، أَوْ إِلَى امْرَأَةٍ يَنْكِحُهَا، فَهِجْرَتُهُ إِلَى مَا هَاجَرَ إِلَيْهِ"

Quiz+

Software Engineering



Majd Riyad

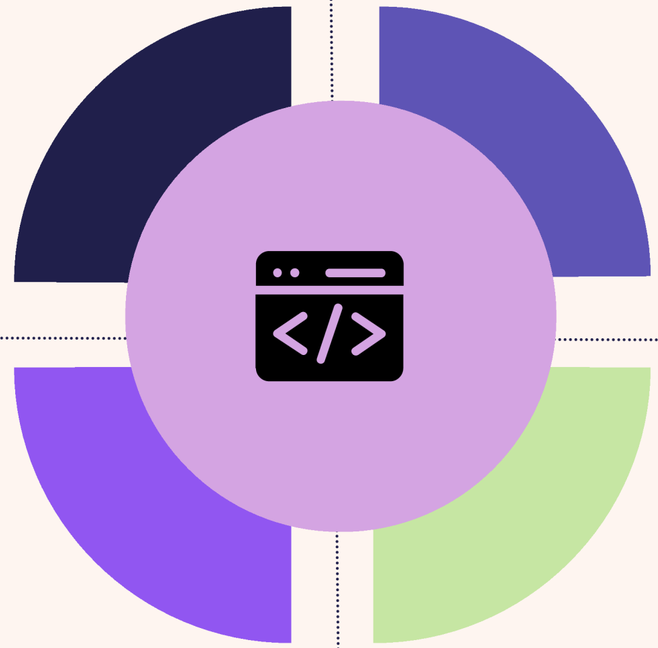


What is Software ?

Software Challenges

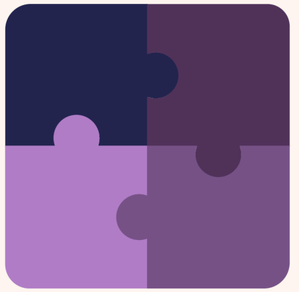
Complexity

Bugs

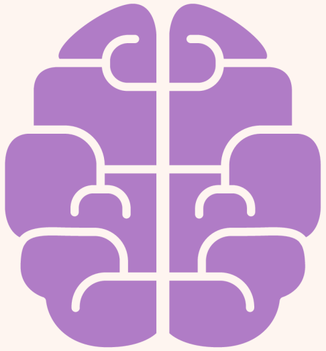


Maintenance

Testing



What is Engineering?



Software Engineering

is the process of designing, developing, testing, and maintaining software by applying engineering principles in a systematic and structured way to ensure that the software is reliable, efficient, and meets user needs.

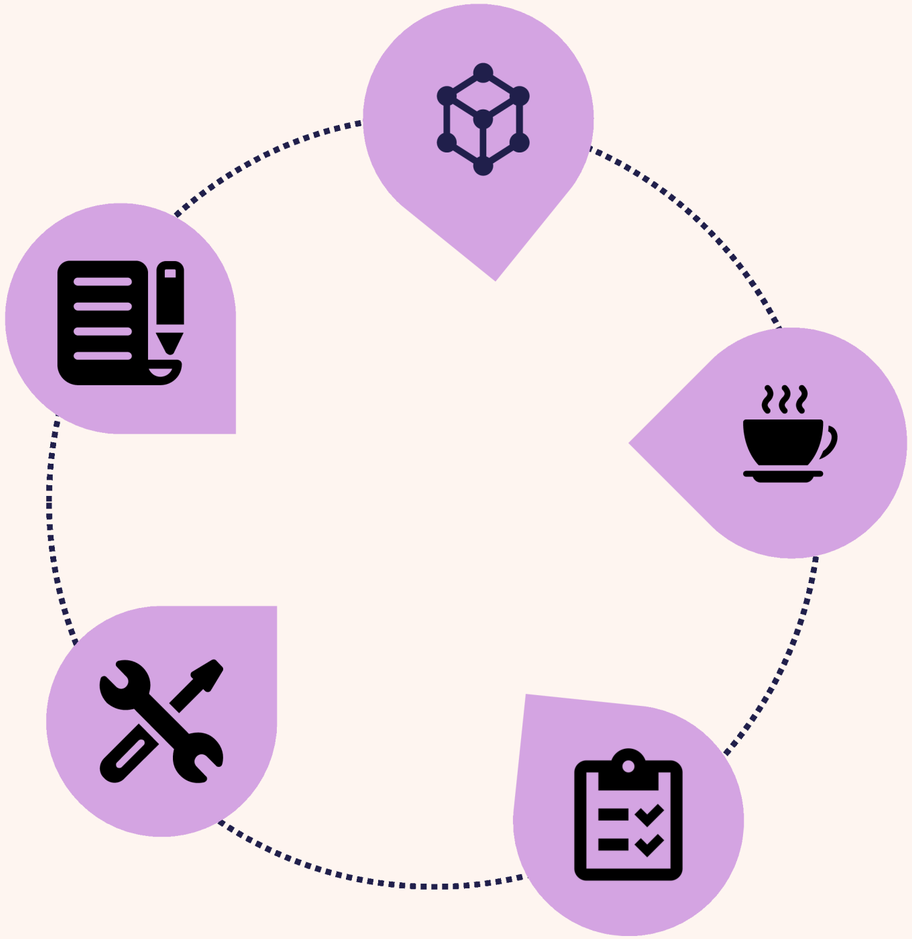
Software engineering is a

Modeling activity

Problem-solving activity

Rationale-driven

Knowledge-gaining activity



What do we mean by software engineering is a modeling activity?

Software engineers need to understand two things:

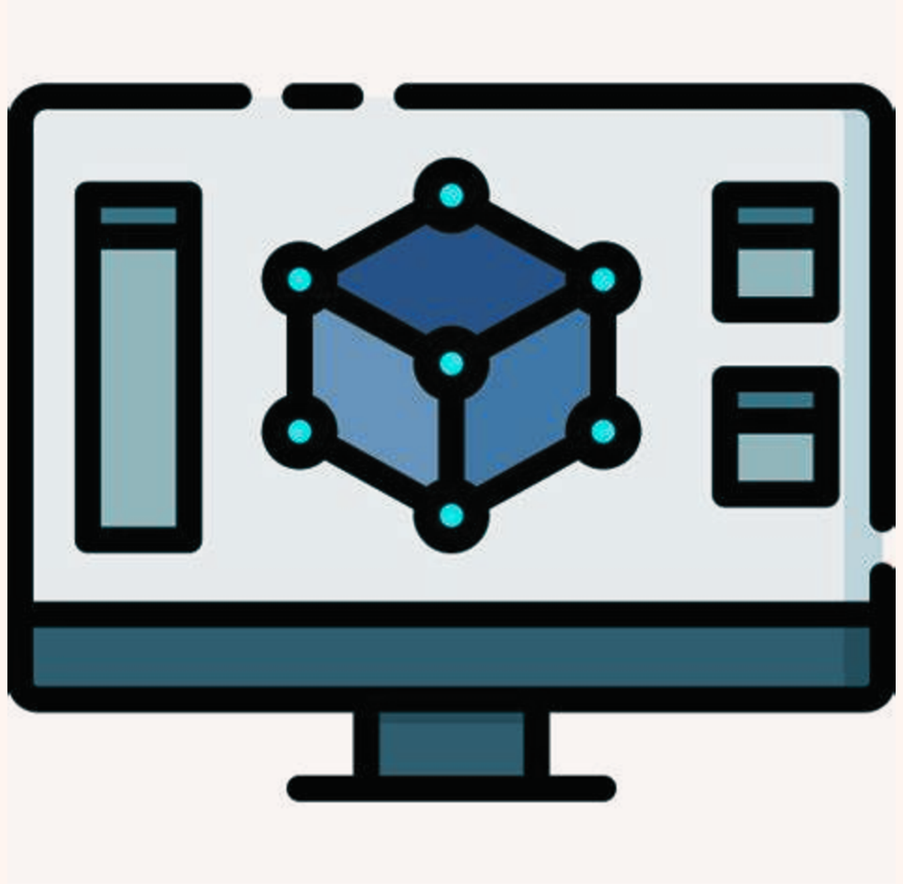
- understand the environment in which the system has to operate.
- understand the systems they could build.

So what helps software engineers understand that?

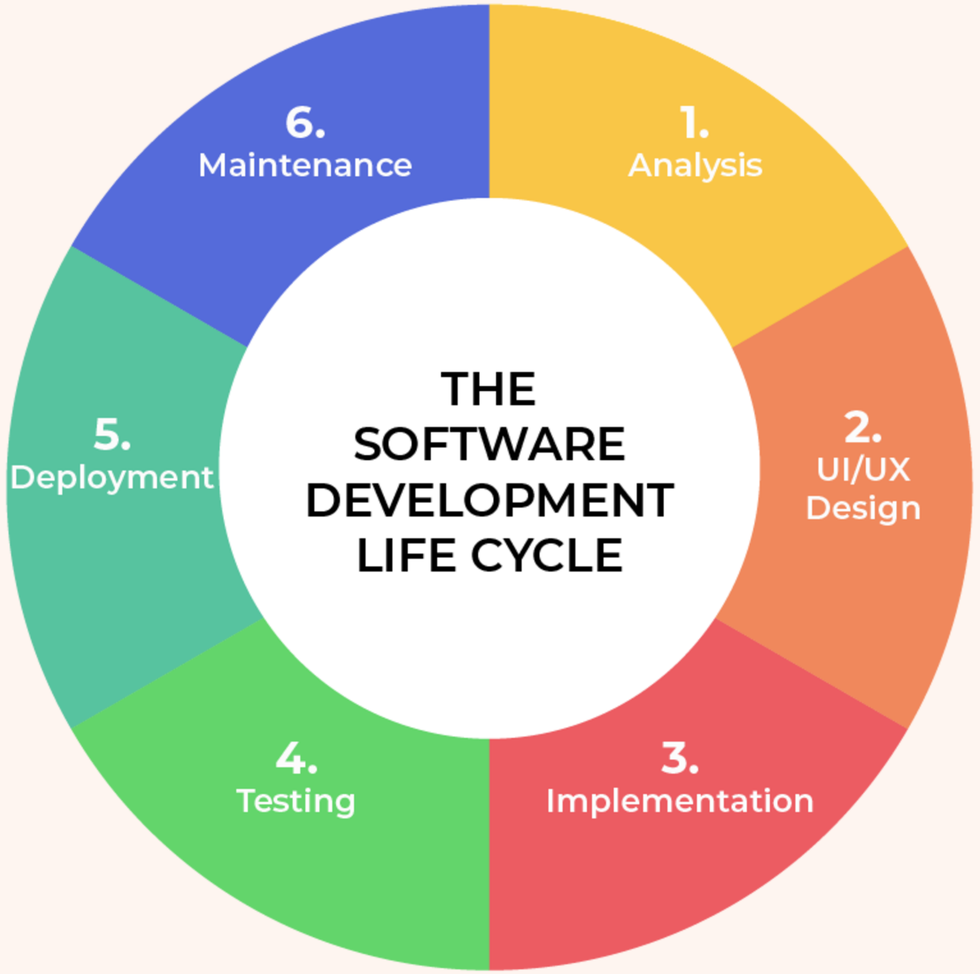
Modeling

- A model is an abstract representation of a system that enables us to answer questions about the system.

- Models also allow us to visualize and understand systems that either no longer exist or that are only claimed to exist.



Software Development Life Cycle (SDLC)



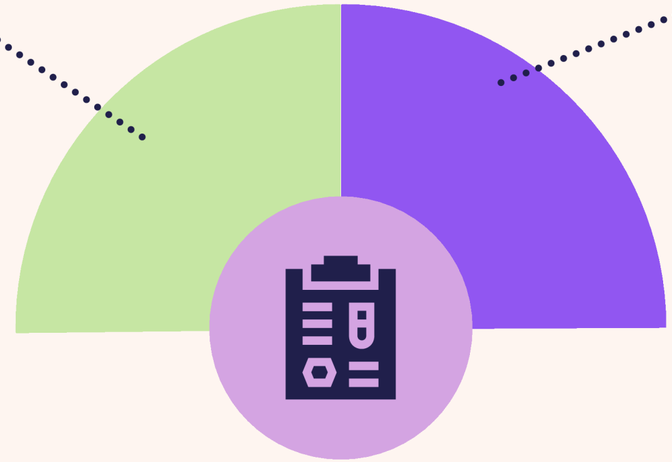
1. Requirements engineering

- Requirements engineering aims at defining the requirements of the system under construction (understand what the software should do by gathering detailed information).

- Requirements engineering includes two main activities:

Requirements gathering

which results in the specification of the system that the client understands.



Analysis

which results in an analysis model that the developers can clearly interpret

Requirements

FUNCTIONAL

- ***Functional requirements*** *describe the interactions between the system and its environment independent of its implementation.*

NONFUNCTIONAL

- ***Nonfunctional requirements*** *describe the quality attributes, performance, and constraints of the system.not directly related to the functional behavior of the system.*
- *categories of nonfunctional requirements: Performance,Reliability and Usability ..etc*

Requirements example

Example: An Online Banking System

FUNCTIONAL

User Authentication:

- *The system shall allow users to log in using a username and password.*
- *The system shall provide a "Forgot Password" feature to retrieve lost passwords.*

NONFUNCTIONAL

Performance:

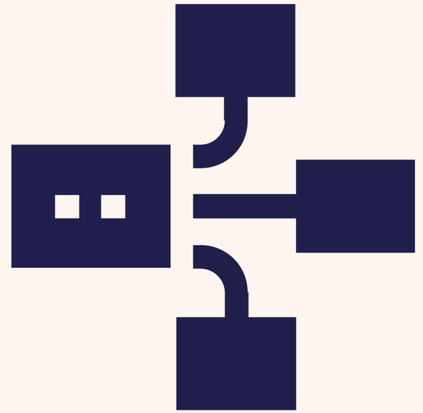
- *The system shall process transactions within 2 seconds.*
- *The system shall support up to 10,000 concurrent users without performance losses.*

2. System design

- During system design, developers define the design goals of the project and decompose the system into smaller subsystems that can be realized by individual teams.

- The result of system design is a model that includes a subsystem decomposition and a clear description of each of these strategies (Design documents, system architecture diagrams, database schema, and interface specifications.).

- System design is decomposed into several activities :
  a. Identify design goals
  b. Design the initial subsystem decomposition
  c. Refine the subsystem decomposition to address the design goals.
- System design is not algorithmic :
  - Conflict in design goals.
  - They also cannot anticipate all design issues that they will face because they do not yet have a clear picture of the solution domain.

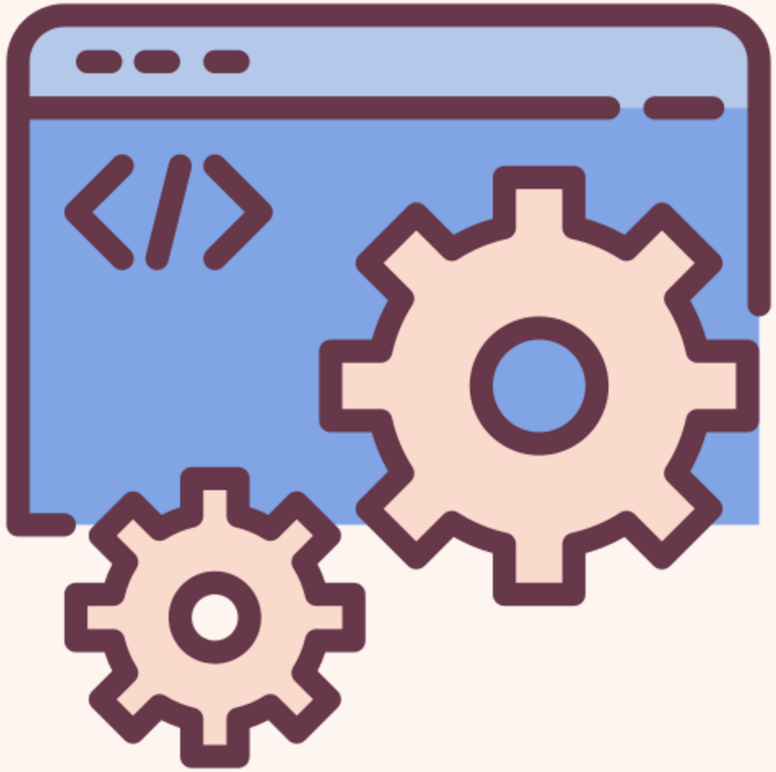


3. Implementation

- **Mapping Models to Code**

- The actual code is written based on the design documents. The goal is to build a working software product that meets the requirements.

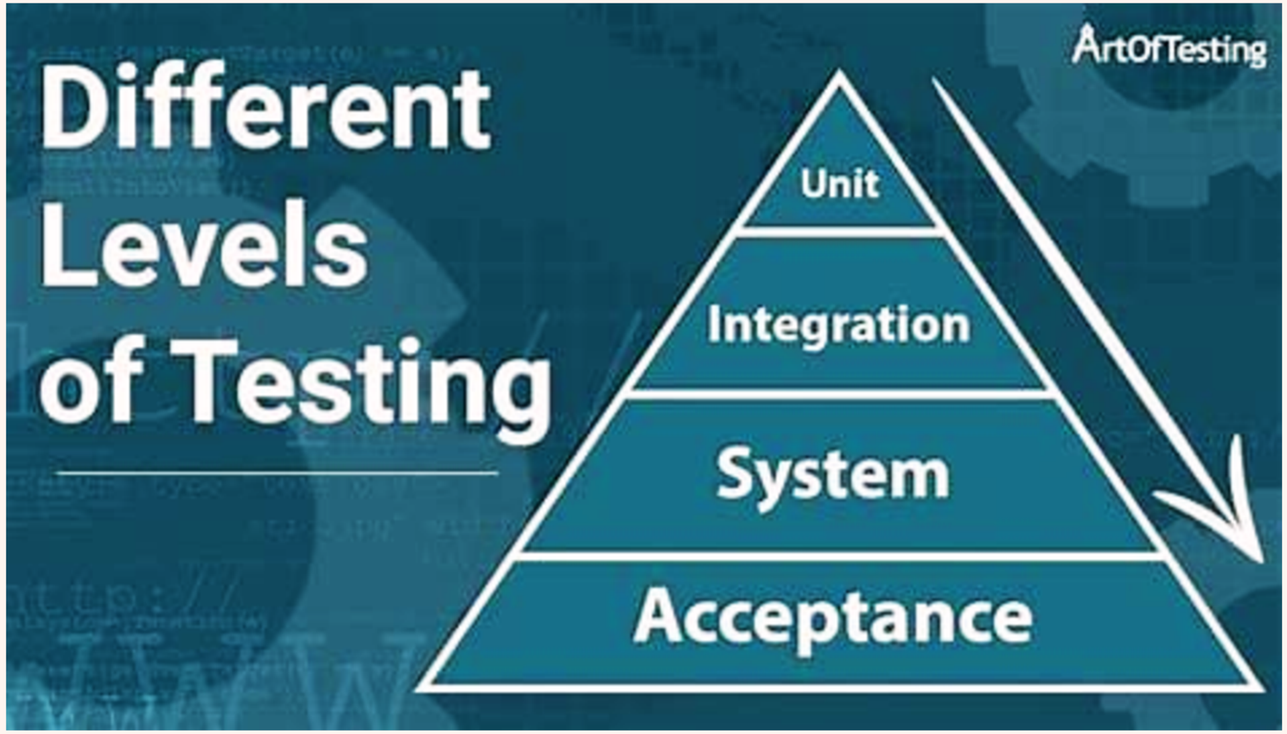
- Output is Executable software and source code files.



4. Testing

- The testing phase ensures that the software is free from defects and meets the specified requirements. It involves various levels of testing to validate and verify the software.

- Testing level :
  a. Unit Testing
  b. Integration Testing
  c. System Testing
  d. Acceptance Testing



Deployment && Maintenance

- The deployment phase involves delivering the software to the end-users and making it operational in the production environment.

- After deployment, the software enters the maintenance phase, where it is updated, enhanced, and supported to ensure it continues to meet user needs and functions correctly.

Software Engineering Methodologies

Waterfall vs Agile

HSC ITAS

Waterfall

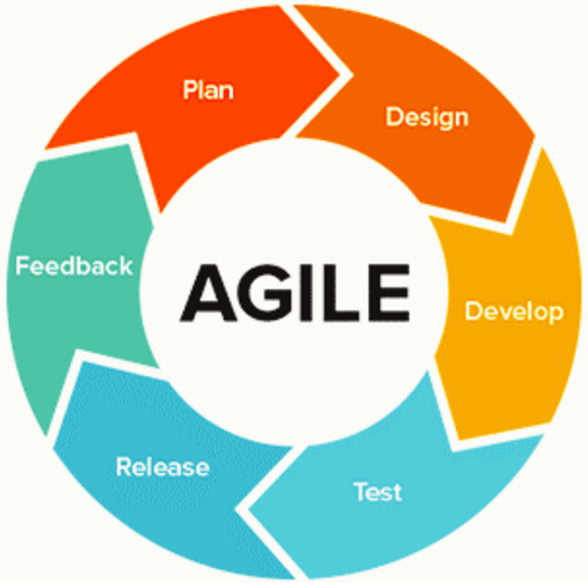
- Linear progression
- Defined long-term goal
- Completed product
- Almost without client feedback
- Fixed deadline
- Fixed budget, Higher costs
- Delivery in a later (testing) phase

VS

Agile

- Flexible progression
- Possible to alter the course of the project
- Minimum viable product
- Frequent communication with clients
- Short-term deadlines
- Flexible budget, Lower costs
- Delivery in an early phase

AGILE VS WATERFALL



سبحانك اللهم وبحمدك أشهد أن لا
إله إلا أنت أستغفرك وأتوب إليك