Home Assignment -1 2300090045

Adaptive Software Engineering

1) Define Software Engineering!

Software Engineering is the systematic application of engineering principles to the development, operation, and maintanance of software. It involves a structured and disciplined approach to design, develop test and manage software systems efficiently and reliably.

2) Explain the Software Myths.

Software myths are common misconceptions about software and the software development process.

Management Myths:

Myth1: If we have a problem, we can just add more people to fix it.

- Reality: Adding people to a late project makes it later.

Myth 2: Once we write the software, the job is done.

- Reality: Maintanance and updates often take more effort than initial development.

Customer Myths:

- Myth 1: Requirements can change easily because software is flexible.
  - Reality: Changing requirements later in the development is costly and complex.
- Myth 2: A general statement of need is enough for developers
  - Reality: Developers need detailed, precise requirements

Developer Myths:

- Myth 1: Once the code is written and works, our job is over.
  - Reality: Testing, debugging, documentation, and support are equally important.
- Myth 2: Software tools and techniques guarantee success
  - Reality: Tools support but do not replace sound software engineering practices.

) Explain the Spiral Model and Unified Process

a) Spiral Model:

The Spiral Model is a risk-driven software development process that combines iterative development with systematic aspects of the waterfall model

Phases:-

1. Planning: Define objectives, constraints, and alternatives.
2. Risk Analysis: Identify and resolve risks.
3. Engineering: Develop and verify the next product version.
4. Evaluation: Evaluating with the customer and plan the next iteration.

b) Unified Process (UP):

The Unified Process is an iterative and incremental software development process framework, commonly associated with Rational Unified Process (RUP)

Phases:-

1. Inception: Define project goals and scope.
2. Elaboration: Analyze the problem domain and define architecture.
3. Construction: Build the software system in increments
4. Transition: Deliver the system to users and deploy it.