

Unit-I

Engineering Practices

Software Engineering Process

- Software methods provide the technical way of building software.
- Methods encompass a broad array of tasks that include communication, requirements analysis, design modeling, program construction, testing, and support.
- Software engineering tools provide automated or semi-automated support for the process and the methods.
- When tools are integrated so that information created by one tool can be used by another, a system for the support of software development, called computer-aided software engineering.

The Software Process

- What is a software process?
- What are the generic framework activities that are present in every software process?
- How are processes modeled and what are process patterns?
- What are the prescriptive process models and what are their strengths and weaknesses?
- Why is agility a watchword in modern software engineering work?
- What is agile software development and how does it differ from more traditional process models?

The essence of practice

- Understand the problem (communication and analysis)
 - Who has a stake in the solution to the problem? That is, who are the stakeholders?
 - What are the unknowns? What data, functions, and features are required to properly solve the problem?
 - Can the problem be compartmentalized? Is it possible to represent smaller problems that may be easier to understand?
 - Can the problem be represented graphically? Can an analysis model be created
- Plan the Solution (modeling and software design)
 - Have you seen similar problems before? Are there patterns that are recognizable in a potential solution? Is there existing software that implements the data, functions, and features that are required?
 - Has a similar problem been solved? If so, are elements of the solution reusable?

- Carryout the plan (code generation)
- Does the solution conform to the plan? Is source code traceable to the design model?
- Is each component part of the solution provably correct? Have the design and code been reviewed, or better, have correctness proofs been applied to the algorithm?
- Examine the result for accuracy (Testing and Validation)
- Is it possible to test each component part of the solution? Has a reasonable testing strategy been implemented?
- Does the solution produce results that conform to the data, functions, and features that are required? Has the software been validated against all stakeholder requirements?