

Table of Contents

Table of Contents	1
Learning Objectives	2
What is a Software Process?	2
What is Software Engineering	2
Definition: Software Process	2
Definition: Software Engineering	3
Definition: Software Engineering [IEEE]	3
Layered Technology	3
Process	3
Methods.....	3
Tools	3
Software Engineering Phases.....	4
Definition Phase.....	4
Development Phase.....	4
Support Phase.....	5
Umbrella Activities	5
Software Process Framework	5
Software Process Maturity	6
Process maturity level 2	7
Process maturity level 3	7
Process maturity level 4	7
Process maturity level 5	7
Reference for Further Reading.....	8

Learning Objectives

After completing this handout, you will be able to:

- Understand software process.
- Understand software engineering.
- Define software process.
- Define software engineering.
- Understand the software engineering layered technology.
- Understand software engineering phases.
- Understand software process maturity.

What is a Software Process?

- When you build a product or system, you go through a series of predictable steps (software process).
- A software process is a road map that helps you create a timely, high-quality result.
- Software engineers adopt the process to their needs and follow it.
- A software process provides stability, control, and organization.
- The software process you adopt depends on the software you're building.
- The work products are the programs, documents, and data produced as a consequence of the software engineering activities defined by the software process.
- A number of software process assessment mechanisms enable organizations to determine the "maturity" of a software process.
- A software process defines the approach that is taken as software is engineered.

What is Software Engineering

- Software engineering encompass technologies that populate the software process—technical methods and automated tools.
- Software engineering is the analysis, design, construction, verification, and management of technical (or social) entities. Regardless of the entity to be engineered.

Definition: Software Process

- A software process is a framework for the tasks that are required to build high-quality software.

Definition: Software Engineering

- Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

Definition: Software Engineering [IEEE]

- Software Engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).

Layered Technology

Process

- The foundation for software engineering is the process layer.
- Software engineering process is the glue that holds the technology layers together and enables rational and timely development of computer software.
- Process defines a framework for a set of key process areas (KPAs) [PAU93] that must be established for effective delivery of software engineering technology
- The key process areas form the basis for management control of software projects and establish the context in which technical methods are applied, work products (models, documents, data, reports, forms, etc.) are produced, milestones are established, quality is ensured, and change is properly managed.

Methods

- Software engineering methods provide the technical how-to's for building software.
- Methods encompass a broad array of tasks that include requirements analysis, design, program construction, testing, and support.
- Software engineering methods rely on a set of basic principles that govern each area of the technology and include modeling activities and other descriptive techniques.

Tools

- Software engineering tools provide automated or semi-automated support for the process and the methods.

- CASE combines software, hardware, and a software engineering database (a repository containing important information about analysis, design, program construction, and testing) to create a software engineering environment analogous to CAD/CAE (computer-aided design/engineering) for hardware.

Software Engineering Phases

- The work associated with software engineering can be categorized into three generic phases, regardless of application area, project size, or complexity.

Definition Phase

- The definition phase focuses on what.
 - What information is to be processed?
 - What function and performance are desired?
 - What system behavior can be expected?
 - What interfaces are to be established?
 - What design constraints exist?
 - What validation criteria are required to define a successful system?
- The key requirements of the system and the software are identified.
- Three major tasks occur
 - System or information engineering
 - Software project planning
 - Requirements analysis

Development Phase

- The development phase focuses on how.
 - How data are to be structured?
 - How function is to be implemented within software architecture?
 - How procedural details are to be implemented?
 - How interfaces are to be characterized?
 - How the design will be translated into a programming language?
 - How testing will be performed?
- Three specific technical tasks should always occur
 - Software design
 - Code generation
 - Software testing

Support Phase

- The support phase focuses on
 - change associated with error correction
 - adaptations required as the software's environment evolves
 - changes due to enhancements brought about by changing customer requirements
- Four types of change are encountered during the support phase:
 - Correction: maintenance changes the software to correct defects.
 - Adaptation: maintenance results in modification to the software to accommodate changes to its external environment.
 - Enhancement: maintenance extends the software beyond its original functional requirements.
 - Prevention: maintenance makes changes to computer programs so that they can be more easily corrected, adapted, and enhanced.

Umbrella Activities

- Umbrella activities are applied throughout the software process and include:
 - Software project tracking and control
 - Formal technical reviews
 - Software quality assurance
 - Software configuration management
 - Document preparation and production
 - Reusability management
 - Measurement
 - Risk management

Software Process Framework

- A common process framework is established by defining a small number of framework activities that are applicable to all software projects, regardless of their size or complexity.
- A number of task sets—each a collection of software engineering work tasks, project milestones, work products, and quality assurance points—enable the framework activities to be adapted to the characteristics of the software project and the requirements of the project team.
- Finally, umbrella activities—such as software quality assurance, software configuration management, and measurement²—overlay the process model.

Umbrella activities are independent of any one framework activity and occur throughout the process.

Software Process Maturity

- The Software Engineering Institute (SEI) has developed a comprehensive model predicated on a set of software engineering capabilities that should be present as organizations reach different levels of process maturity.
- The SEI uses an assessment that results in a five point grading scheme.
- The grading scheme determines compliance with a capability maturity model (CMM) [PAU93] that defines key activities required at different levels of process maturity.
 - Level 1: Initial. The software process is characterized as ad hoc and occasionally even chaotic. Few processes are defined, and success depends on individual effort.
 - Level 2: Repeatable. Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.
 - Level 3: Defined. The software process for both management and engineering activities is documented, standardized, and integrated into an organizationwide software process. All projects use a documented and approved version of the organization's process for developing and supporting software. This level includes all characteristics defined for level 2.
 - Level 4: Managed. Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled using detailed measures. This level includes all characteristics defined for level 3.
 - Level 5: Optimizing. Continuous process improvement is enabled by quantitative feedback from the process and from testing innovative ideas and technologies. This level includes all characteristics defined for level 4.
- The SEI has associated key process areas (KPAs) with each of the maturity levels.
 - Goals—the overall objectives that the KPA must achieve.
 - Commitments—requirements (imposed on the organization) that must be met to achieve the goals or provide proof of intent to comply with the goals.

- Abilities—those things that must be in place (organizationally and technically) to enable the organization to meet the commitments.
 - Activities—the specific tasks required to achieve the KPA function.
 - Methods for monitoring implementation—the manner in which the activities are monitored as they are put into place.
 - Methods for verifying implementation—the manner in which proper practice for the KPA can be verified.
- Eighteen KPAs (each described using these characteristics) are defined across the maturity model and mapped into different levels of process maturity.
 - Each of the KPAs is defined by a set of key practices that contribute to satisfying its goals.
 - The key practices are policies, procedures, and activities that must occur before a key process area has been fully instituted.

Process maturity level 2

- Software configuration management
- Software quality assurance
- Software subcontract management
- Software project tracking and oversight
- Software project planning
- Requirements management

Process maturity level 3

- Peer reviews
- Intergroup coordination
- Software product engineering
- Integrated software management
- Training program
- Organization process definition
- Organization process focus

Process maturity level 4

- Software quality management
- Quantitative process management

Process maturity level 5

- Process change management

- Technology change management
- Defect prevention

Reference for Further Reading

- Book: Software Engineering, A Practitioner's Approach, Fifth Edition, Roger S. Pressman, chapter 2.