Overview

- The roadmap to building high quality software products is software process.
- Software processes are adapted to meet the needs of software engineers and managers as they undertake the development of a software product.
- A software process provides a framework for managing activities that can very easily get out of control.
- Different types of projects require different software processes.
- The software engineer's work products (programs, documentation, data) are produced as a consequence of the activities defined by the software process.
- The best indicators of how well a software process has worked are the quality, timeliness, and long-term viability of the resulting software product.

Software Engineering

- Software engineering is the establishment and sound engineering principles applied to obtain reliable and efficient software in an economical manner.
- Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software.
- Software engineering encompasses a process, management techniques, technical methods, and the use of tools.

Common Process Framework

- Communication (customer collaboration and requirement gathering)
- Planning (establishes engineering work plan, describes technical risks, lists resource requirements, work products produced, and defines work schedule)
- Modeling (creation of models to help developers and customers understand the requires and software design)
- Construction (code generation and testing)
- Deployment (software delivered for customer evaluation and feedback)

Software Engineering Umbrella Activities

- Software project tracking and control (allows team to assess progress and take corrective action to maintain schedule)
- Risk management (assess risks that may affect project outcomes or quality)
- Software quality assurance (activities required to maintain software quality)
- Formal technical reviews (assess engineering work products to uncover and remove errors before they propagate to next activity)
- Measurement (define and collect process, project, and product measures to assist the software team in delivering software meeting customer needs)
- Software configuration management (manage effects of change)
- Reusability management (defines criteria for work product reuse and establish mechanisms to achieve component reuse)
- Work product preparation and production (activities to create models, documents, logs, forms, lists, etc.)

Attributes for Comparing Process Models

- Overall flow and level of interdependencies among tasks
- Degree to which work tasks are defined within each framework activity
- Degree to which work products are identified and required
• Manner in which quality assurance activities are applied
• Manner in which project tracking and control activities are applied
• Overall degree of detail and rigor of process description
• Degree to which stakeholders are involved in the project
• Level of autonomy given to project team
• Degree to which team organization and roles are prescribed

Software Engineering Institute (SEI) Capability Maturity Model Integration (CMMI)

• Level 0: Incomplete (process is not performed or does not achieve all goals defined for
• Level 1: Performed (work tasks required to produce required work products are being conducted)
• Level 2: Managed (people doing work have access to adequate resources to get job done, stakeholders are actively involved, work tasks and products are monitored, reviewed, and evaluated for conformance to process description)
• Level 3: Defined (management and engineering processes documented, standardized, and integrated into organization-wide software process)
• Level 4: Quantitatively Managed (software process and products are quantitatively understood and controlled using detailed measures)
• Level 5: Optimizing (continuous process improvement is enabled by quantitative feedback from the process and testing innovative ideas)

Software Patterns

• Templates or methods for describing important characteristics of software processes
• Software teams can combine software patterns to construct processes that best meet the needs of specific projects
• Generic software pattern elements
  o Meaningful pattern name
  o Intent (objective of pattern)
  o Type
    ▪ Task pattern (defines engineering action or work task)
    ▪ Stage pattern (defines framework activity for the process)
    ▪ Phase pattern (defines sequence or flow of framework activities that occur within process)
  o Initial context (describes conditions that must be present prior to using pattern)
  o Solution (describes how to implement pattern correctly)
  o Resulting context (describes conditions that result when pattern has been implemented successfully)
  o Related patterns (links to patterns directly related to this one)
  o Known uses/examples (instances in which pattern is applicable)

Process Assessment

• SPICE (ISO/IE15504) standard defines a set of requirements for process assessment
  o Examines the processes used by organizations to determine whether they are effective in achieving their goals
  o Provides a reference model that examines the purpose and measurable objectives of the process (process dimension) and the set of process attributes that should be present (capability dimension)
  o SPICE model assessment is a structured evaluation of a process model (activities, tasks, work products, etc.)
• ISO 9001:2000 for Software defines requirements for a quality management system that will produce higher quality products and improve customer satisfaction
  o Stress the important for an organization to identify, implement, manage, and continually improve the effectiveness processes needed for a quality management system and to manage process interactions to achieve organization objectives
- Process effectiveness and efficiency are assessed through internal or external review of processes using maturity scales
- Results can be documented and monitored over time to reach improvement goals using “plan-do-check-act” cycle

**Personal Software Process (PSP)**

- **Framework activities**
  - Planning (size and resource estimates based on requirements)
  - High-level design (external specifications developed for components and component level design is created)
  - High-level design review (formal verification methods used to uncover design errors, metrics maintained for important tasks)
  - Development (component level design refined, code is generated, reviewed, compiled, and tested, metric maintained for important tasks and work results)
  - Postmortem (effectiveness of processes is determined using measures and metrics collected, results of analysis should provide guidance for modifying the process to improve its effectiveness)

- **Levels**
  - Level PSP 0 (personal measurement) - original process used
  - Level PSP 1 (personal planning) - original process is modified slightly by introducing a moderate number of PSP tasks to small project; resource, size, and defect estimates are made, results are measured and defect/effort measurements are logged and analyzed
  - Level PSP 2 (personal quality) - original process is modified significantly by introducing a moderate number of PSP tasks to small project; resource, size, and defect estimates are made, results are measured and defect/effort measurements are logged and analyzed
  - Level PSP 3 (cyclic process) - full scale PSP process implemented and applied to sequence of small projects

**Team Software Process**

- **Objectives**
  - Build self-directed teams that plan and track their work, establish goals, and own their processes and plans
  - Show managers how to coach and motivate their teams and maintain peak performance
  - Accelerate software process improvement by making CCM Level 5 behavior normal and expected
  - Provide improvement guidance to high-maturity organizations
  - Facilitate university teaching of industrial team skills

- **Scripts for Project Activities**
  - Project launch
  - Design
  - Implementation
  - Integration and system testing
  - Postmortem

**Process Technology Tools**

- Used to adapt process models to be used by software project team
- Allow organizations to build automated models of common process framework, task sets, and umbrella activities
- These automated models can be used to determine workflow and examine alternative process structures
- Tools can be used to allocate, monitor, and even control all software engineering tasks defined as part of the process model