

# **Software Engineering (CSE 415)**

## **Software Process: A Generic View**

# Software Process : Overview

- **What?** A software process is a series of predictable steps that leads to a timely, high-quality product.
- **Who?** Managers, software engineers, and customers.
- **Why?** Provides stability, control, and organization to an otherwise chaotic activity.
- **Steps?** A handful of activities are common to all software processes, details vary.
- **Work product?** Programs, documents, and data.
- **Correct process?** Assessment, quality deliverable.

# 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.
- Software engineering is an engineering discipline that is concerned with all aspects of software production.

# Software Engineering

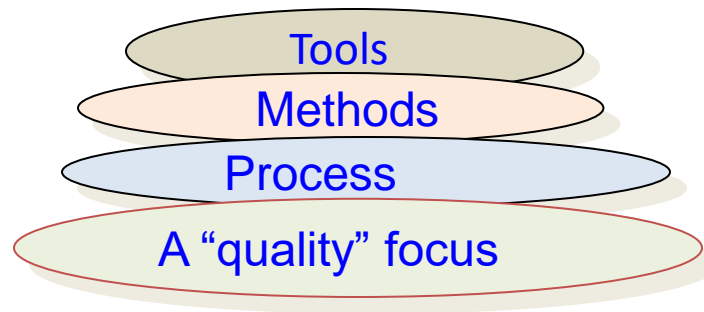
- Software Engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).
  - IEEE Standard 610.12-1990

# Software Engineering

- A strategy for producing high quality software.
- Software Engineering is a layered technology.
- Software engineering encompasses a process, methods, and tools.
- Quality focus is the bedrock that supports software engineering.

# Software Engineering

- Software Engineering –A Layered Technology



# Process

- A software process is the set of activities and associated results that are involved in developing software products.
- Foundation for software engineering.
- A software process defines a framework that must be established for effective delivery of software engineering technology.
- A software process is a framework for the tasks that are required to build high-quality software.
- Software process forms the basis for management control of software projects, establishes the context in which technical methods are applied, work products are produced, milestones are established, quality is ensured, and change is properly managed.

# Methods

- A software engineering method is a structured approach to software development whose aim is to facilitate the production of high-quality software in a cost-effective way.
- Software engineering methods provide the technical “how to’s” for building software.
- Methods are organized ways of producing software.
- Methods encompass a broad array of tasks –
  - Communication
  - Requirements analysis
  - Design
  - Program construction
  - Testing
  - Support



# Tools

- Provide support for the process & methods
- Automation/semi-automation
- Computer-aided software engineering (CASE)

# A Process Framework

- A *process framework* establishes the foundation for a complete software process by identifying a small number of *framework activities* that are applicable to all software projects, regardless of size/complexity.
- In addition, the process framework encompasses a set of *umbrella activities* that are applicable across the entire software process.

# Generic Process Framework Activities

- Five generic process framework activities are:
  - 1) Communication
  - 2) Planning
  - 3) Modeling
  - 4) Construction
  - 5) Deployment

# Generic Process Framework Activities

## ■ 1. Communication

- Involves heavy communication and collaboration with the customer and other stakeholders
- Encompasses requirements gathering and other related activities

# Generic Process Framework Activities

## ■ 2. Planning

- Establishes a plan for the software engineering work that follows
- Describes technical tasks, risks, resources, work product, and work schedule

# Generic Process Framework Activities

## ■ 3. Modeling:

- Encompasses the creation of models that allow the developer and customer to better understand software requirements and the design that will achieve those requirements
- Analysis of Requirements
- Design

# Generic Process Framework Activities

- **4. Construction:**
  - Code generation
  - Testing

# Generic Process Framework Activities

- **5. Deployment:**

- Software is delivered to customer who evaluates the delivered product and provides feedback.



# Umbrella Activities

- The generic process framework activities are complemented by a set of *umbrella activities* that persist across the entire software process.
- Each of these umbrella activities is defined by a set of tasks that are adapted to the project type and degree of rigor with which software engineering is to be applied.
- Umbrella activities occur throughout the software process and focus primarily on project management, tracking and control.

# What are the Umbrella Activities ?

- Umbrella Activities –
  - Software project management
  - Formal technical reviews
  - Software quality assurance
  - Software configuration management
  - Work product preparation and production
  - Reusability management
  - Measurement
  - Risk management

# The Process Model: Adaptability

- The framework activities will always be applied on every project ... BUT different projects demand different task sets.
- The tasks (and degree of rigor) for each activity will vary based on:
  - the type of project
  - characteristics of the project
  - common sense judgment
  - concurrence of the project team
- Software process adaptation is essential for project success.
- Software team chooses the task set based on problem and project characteristics.

# Process Assessment

- The existence of a software process is no guarantee that software will be delivered on time, that it will meet the customer's needs, or that it will exhibit the technical characteristics that will lead to long-term quality characteristics.
- Assessment attempts to understand the current state of the software process with the intent of improving it.
- **Formal techniques** for assessing software process –
  - CMMI
  - SPICE
  - ISO 9001:2000 for Software

# The CMMI

- The **Software Engineering Institute (SEI)** has developed **Capability Maturity Model Integration (CMMI)**. CMMI is a process improvement approach whose goal is to help organizations improve their performance.
- CMMI is a model for judging the maturity of software processes of an organization & for identifying the key practices that are required to increase the maturity of these processes.
- CMMI is a comprehensive process meta-model that describes the specific goals, practices, and capabilities that should be present in a mature software process.

# The CMMI

- The CMMI defines each process area in terms of “specific goals” and the “specific practices” required to achieve these goals.
- *Specific goals* establish the characteristics that must exist if the activities implied by a process area are to be effective.
- *Specific practices* refine a goal into a set of process-related activities.

# The Maturity Levels of CMMI

- Maturity levels consist of a predefined set of process areas. The maturity levels are measured by the achievement of the **specific** and **generic goals** that apply to each predefined set of process areas.
- There are five maturity levels of CMMI:
  - Level 1 : Initial
  - Level 2 : Managed
  - Level 3 : Defined
  - Level 4 : Quantitatively Managed
  - Level 5 : Optimized

# Main focuses of different levels

- Initial ==> Ad-hoc/chaotic
- Managed ==> Basic Project Management
- Defined ==> Process Standardization
- Quantitatively Managed ==> Quantitative management
- Optimized ==> Continuous process improvement



# Level 1 - Initial

- No KPAs defined at this level [KPA- Key Process Area]
- Ad-hoc/Chaotic approach to software design
- The starting point for use of a new process
- The organization usually does not provide a stable environment. Organizations often produce products and services that work; however, they frequently exceed the budget and schedule of their projects.
- Organizations are characterized by a tendency to over commit, abandon processes in the time of crisis, and not be able to repeat their past successes.

## Level 2 - Managed

- 7 KPAs are defined at this level-
  - 1) Configuration Management(CM)
  - 2) Measurement & Analysis(MA)
  - 3) Project Monitoring & Control(PMC)
  - 4) Project Planning(PP)
  - 5) Process & Product Quality Assurance
  - 6) Requirements Managements(REQM)
  - 7) Supplier Agreement Management(SAM)

## Level 2 - Managed

- At maturity level-2, an organization has achieved all the **specific** and **generic goals** of the maturity level-2 process areas. In other words, the projects of the organization have ensured that requirements are managed and that processes are planned, performed, measured, and controlled.
- The process discipline reflected by maturity level-2 helps to ensure that existing practices are retained during times of stress. When these practices are in place, projects are performed and managed according to their documented plans.

# Level 3 - Defined

- Focuses on Process Standardization
- The process is defined as a standard business process.
- KPAs of Level-2 + 14 KPAs (e.g. Verification, validation, Risk management )
- At maturity level-3, processes are well characterized and understood, and are described in standards, procedures, tools, and methods.
- A critical distinction between maturity level-2 and maturity level 3 is the scope of standards, process descriptions, and procedures. Processes are typically described in more detail and more rigorously than at maturity level-2.

# Level 4 - Quantitatively Managed

- Focuses on Quantitative management
- All level-3 KPAs + 2 KPAs
  - Organizational Process Performance
  - Quantitative Project Management
- At maturity level 4, Sub-processes are selected that significantly contribute to overall process performance. These selected sub-processes are controlled using statistical and other quantitative techniques.
- Quality and process performance measures are incorporated into the organization's measurement repository to support fact-based decision making in the future.
- A critical distinction between maturity level 3 and maturity level 4 is the predictability of process performance.

# Level 5 - Optimized

- Continuous process improvement to meet changing customer needs
- Improve efficiency of the process area
- All Level 4 KPAs + 2 KPAs
  - Organizational Innovation and Deployment
  - Causal Analysis and Resolution
- Focuses on continually improving process performance through both incremental and innovative technological improvements.
- Optimizing processes that are agile and innovative depends on the participation of an empowered workforce aligned with the business values and objectives of the organization.