# Software engineering II MODULE

### THE SOFTWARE PROCESS FRAME WORK

- The process of framework defines a small set of activities that are applicable to all types of projects.
- The software process framework is a collection of task sets.
- Task sets consist of a collection of small work tasks, project milestones, work productivity and software quality assurance points.



#### Fig.- A software process framework

A number of task sets –each a collection of software engineering work tasks, project milestones, work products and quality assurance points



- ► Typical umbrella activities are:
- 1. Software project tracking and control
- In this activity, the developing team accesses project plan and compares it with the predefined schedule.
- If these project plans do not match with the predefined schedule, then the required actions are taken to maintain the schedule.
- > 2. Risk management
- Risk is an event that may or may not occur.
- ▶ If the event occurs, then it causes some unwanted outcome. Hence, proper risk management is required.



#### ► 3. Software Quality Assurance (SQA)

SQA is the planned and systematic pattern of activities which are required to give a guarantee of software quality.

**For example,** during the software development meetings are conducted at every stage of development to find out the defects and suggest improvements to produce good quality software.

- ▶ 4. Formal Technical Reviews (FTR)FTR is a meeting conducted by the technical staff.
- ▶ The motive of the meeting is to detect quality problems and suggest improvements.
- ► The technical person focuses on the quality



- 5. Measurement
- Measurement consists of the effort required to measure the software.
- ▶ The software cannot be measured directly. It is measured by direct and indirect measures.
- Direct measures like cost, lines of code, size of software etc.
- Indirect measures such as quality of software which is measured by some other factor. Hence, it is an indirect measure of software.
- 6. Software Configuration Management (SCM) It manages the effect of change throughout the software process.
- **7. Reusability management**It defines the criteria for reuse the product.
- ▶ 8. Work product preparation and production It consists of the activities that are needed to create the documents, forms, lists, logs and user manuals for developing a software.
- The quality of software is good when the components of the software are developed for certain application and are useful for developing other applications. Downloaded from Ktunotes.in

# Capability Maturity Model

### What is CMM?

- CMM: Capability Maturity Model
- Developed by the Software Engineering Institute of the Carnegie Mellon University
- Framework that describes the key elements of an effective software process.
- Describes an evolutionary improvement path for software organizations from an ad hoc, immature process to a mature, disciplined one.
- Provides guidance on how to gain control of processes for developing and maintaining software and how to evolve toward a culture of software engineering and management excellence

### What are the CMM Levels? (The five levels of software process maturity)



### Level 1: Initial

- Initial : The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort.
  - At this level, frequently have difficulty making commitments that the staff can meet with an orderly process
  - Products developed are often over budget and schedule
  - > Wide variations in cost, schedule, functionality and quality targets
  - Capability is a characteristic of the individuals, not of the organization

### Level 2: Repeatable

- Basic process 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.
  - Realistic project commitments based on results observed on previous projects
  - Software project standards are defined and faithfully followed
  - Processes may differ between projects
  - Process is disciplined
  - > earlier successes can be repeated

### Level 3: Defined

- The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization.
- All projects use an approved, tailored version of the organization's standard software process for developing an maintaining software.

### 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.
  - Narrowing the variation in process performance to fall within acceptable quantitative bounds
  - > When known limits are exceeded, corrective action can be taken
  - > Quantifiable and predictable

▶ predict trends in process and product quality

### Level 5: Optimizing

- > Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.
- Goal is to prevent the occurrence of defects
  Causal analysis
  - Causal analysis
- > Data on process effectiveness used for cost benefit analysis of new technologies and proposed process changes

### ISO 9000

- The ISO 9000 family of <u>quality management systems</u> standards is designed to help organizations ensure that they meet the needs of customers.
- KTUNOTES.IN The seven quality management principles are:
  - •QMP 1 Customer focus
  - •QMP 2 Leadership
  - •QMP 3 Engagement of people
  - •QMP 4 Process approach
  - •QMP 5 Improvement
  - •QMP 6 Evidence-based decision making
  - •QMP 7 Relationship management

ISO 9000 principles of quality management

The ISO 9000:2015 and ISO 9001:2015 standards are based on seven quality management principles that senior management can apply for organizational improvement:

#### >Principle 1 – Customer focus

- Organizations depend on their customers and therefore should understand current and future customer needs, should meet customer requirements and strive to exceed customer expectations.
- > Understand the needs of existing and future customers
- > Align organizational objectives with customer needs and expectations
- > Meet customer requirements
- > Measure customer satisfaction
- Manage customer relationships
- Aim to exceed customer expectations

### >Principle 2 – Leadership

- > Leaders establish unity of purpose and direction of the organization.
- They should create and maintain the internal environment in which people can become fully involved in achieving the organization's objectives.
- > Establish a vision and direction for the organization
- > Set challenging goals
- Model organizational values
- Establish trust
- Equip and empower employees
- > Recognize employee contributions

#### >Principle 3 – Engagement of people

- People at all levels are the essence of an organization and their full involvement enables their abilities to be used for the organization's benefit.
- > Ensure that people's abilities are used and valued
- > Make people accountable
- > Enable participation in continual improvement
- Evaluate individual performance
- Enable learning and knowledge sharing
- > Enable open discussion of problems and constraints

### >Principle 4 – Process approach

- > A desired result is achieved more efficiently when activities and related resources are managed as a process.
- Manage activities as processes
- Identify linkages between activities
- Prioritize improvement opportunities
- Deploy resources effectively

#### >Principle 5 – Improvement

- > Improvement of the organization's overall performance should be a permanent objective of the organization.
- > Improve organizational performance and capabilities
- > Align improvement activities
- Empower people to make improvements
- > Measure improvement consistently
- Celebrate improvements

#### >Principle 6 – Evidence-based decision making

- > Effective decisions are based on the analysis of data and information.
- > Ensure the accessibility of accurate and reliable data
- > Use appropriate methods to analyze data
- Make decisions based on analysis
- Balance data analysis with practical experience

#### >Principle 7 – Relationship management

An organization and its external providers (suppliers, contractors, service providers) are <u>interdependent</u> and a mutually beneficial relationship enhances the ability of both to create value.

- > Identify and select suppliers to manage costs, optimize resources, and create value
- > Establish relationships considering both the short and long term
- Share expertise, resources, information, and plans with partners
- Collaborate on improvement and development activities
- Recognize supplier successes

### Requirement Engineering

#### Requirements describe -What to build not How

- Developers do not know what to build
- Customers do not know what to expect
- This phase produces one large document written in natural language contains a description of what the system will do without describing how it will do it
- Crucial process

## 4 STEPS IN REQUIREMENT ENGG

#### 1. Requirement elicitation

- Also called gathering of requirement
- Identified with the help f customer

#### 2. Requirement analysis

Analyzed in order to identify inconsistencies, defects, omissions etc.

#### **3. Requirement documentation**

- ► This is the end product of requirement elicitation and analysis
- Primary output of requirement engineering is requirement specification
- ▶ If it describes only the software requirement, its is software requirement specification
- ▶ If it describes booth hardware and software requirements, it is called system requirement specification
- This document is called software requirement specification(SRS)

#### 4. Requirement review

Review process is carried out to improve the quality of SRS

### Types of requirements





- **Known requirement-** Something stakeholder believes to be implemented
- **Unknown requirement-** Forgotten by the customer
- Undreamt requirements- stakeholder may not be able to think of new requirements due to limited domain knowledge
- A known, unknown or undreamt requirement may be functional or non functional

### Functional requirements

- Functional requirements describe what the software has to do.
- ► They are often called product features.
- Ex in case of mobile phone calling and messaging

### Non functional requirements

- Non Functional requirements are mostly quality requirements.
- ► That stipulate how well the software does, what it has to do.
- ex: battery back up, security
- ► For users-Availability, Reliability, Usability ,Flexibility
- ► For developers Maintainability, Portability, Testability

### Requirement elicitation

- Most difficult
- Most critical
- Most error prone
- Most communication intensive
- Selection of any method
- ▶ 1. It is the only method that we know
- > 2. It is our favorite method for all situations
- ▶ 3. We understand intuitively that the method is effective in
- ▶ the present circumstances.

### 1. Interviews

- After receiving a problem form the customer, first step is to arrange a meeting with customer
- Specialised developers called requirement engineer interact with customer
- Both parties have a common goal
- **• Objective : To understand customer expectation from software**
- ► Types of Interview
- open ended Success -no pre defined agenda
- **structured-** have pre defined agenda
- Selection of stakeholder
- ▶ 1. Entry level personnel
- ► 2. Middle level stakeholder
- ► 3. Managers
- 4. Users of the software (Most important)-



#### Types of questions normally asked

- Any problems with existing system
- Any Calculation errors
- Possible reasons for malfunctioning
- Possible benefits
- Satisfied with current policies
- Any requirement of data from other system
- Any specific problems
- Any additional functionality
- Most important goal of the proposed development

### 2. Brainstorming Sessions

- It is a group technique
- It promotes------ $\rightarrow$





- Provides platform to share ideas
- Participants are encouraged to share what comes to their mind whether relevant or not
- group consists of actual users, middle level managers, total stakeholder
- Written in simple English
- Every idea will be documented, such that everyone can see it
- After session, detailed report will be made and facilitator(ne who controls the session) will review it.

### Facilitated application development(FAST)

- Similar to brainstorming sessions.
- Team oriented approach
- forms a team of customers and developers



### Guidelines for FAST

- ▶ 1. Arrange a meeting at a neutral site.
- ▶ 2. Establish rules for participation.
- ▶ 3. Informal agenda to encourage free flow of ideas.
- ▶ 4. Appoint a facilitator.
- ▶ 5. Prepare definition mechanism board, worksheets, wall stickier.
- ▶ 6. Participants should not criticize or debate.



### FAST session Preparations

Each attendee is asked to make a list of objects that are:

- ▶ 1. Part of environment that surrounds the system.
- ▶ 2. Produced by the system.
- ▶ 3. Used by the system.



### Activities of FAST session

- ► 1. Every participant presents his/her list
- ▶ 2. Combine list for each topic, Eliminate redundancies
- ▶ 3. Discussions on combined list
- 4. Consensus list are finalized by facilitator
- 5. Forms sub teams, each works to develop for mini specifications for one or more entries in the list
- ▶ 6. Presentations of mini-specifications to all FAST attendees.
- ▶ 7. Each attendee prepares a list of validation criteria for system
- ▶ 8. A sub team may asked to write complete draft specifications

### Quality Function Deployment(QFD)

- Incorporates voice of the customer and the voice is then translated to technical requirements
- ▶ What is important for customer?
- Prime concern of QFD is customer satisfaction
- ► Three types of requirements are identified
- -- Normal requirements-If present customer get satisfied
- Expected requirements-these are requirements that are not explicitly stated by customer but if not present they will be dissatisfied
- -- Exciting requirements- if present they are highly satisfied



- Ex: In the case of a result analysis s/w
- Normal: entry of mark ,calculation of result, merit list report etc.
- **Expected requirements**: protection from unauthorized access.
- Exciting: if an unauthorized access is noticed shut down all systems and mail is sent to system admin about the access

### Steps of QFD

- 1. Identify stakeholders(customers, users, developers)
- ▶ 2. List out requirements from user
- 3. Value indicating the degree of importance assigned to each requirement
  - ► 5 Points : V. Important
  - ► 4 Points : Important
  - ▶ 3 Points : Not Important but nice to have
  - ▶ 2 Points : Not important
  - ▶ 1 Points : Unrealistic, required further



- ► Finally Requirement Engineer may categorize like:
- ► (i) It is possible to achieve
- ► (ii) It should be deferred & Why
- ▶ (iii) It is impossible and should be dropped from consideration
- ► First Category requirements will be implemented as per priority assigned with every requirement.
- Requirements of importance to each requirement

## Software prototyping

- Technique of constructing a partial implementation of system so that customers, users or developers can learn more about the problem.
- ► The process of prototyping involves the following steps
- Identify basic <u>requirements</u> Determine basic requirements including the input and output information desired. Details, such as security, can typically be ignored.
- Develop initial prototype
- Review The customers, including end-users, examine the prototype and provide feedback on potential additions or changes.
- Revise and enhance the prototype Using the feedback both the specifications and the prototype can be improved.

### **Benefits of s/w prototyping**

- The software designer and implementer can get valuable feedback from the users early in the project.
- Missing user requirement can be detected
- Misunderstanding between user and developers may be identified
- A working model is available quickly
- The client and the contractor can compare if the software made matches the <u>software</u> <u>specification</u>, according to which the software program is built.
- ► It also allows the software engineer some insight into the accuracy of initial project estimates and whether the deadlines and <u>milestones</u> proposed can be successfully met



### Throwaway prototyping

- Also called close-ended prototyping.
- Throwaway refers to the creation of a prototype model that will eventually be discarded rather than becoming part of the final delivered software.
- After preliminary requirements gathering is accomplished, a simple working model of the system is constructed to visually show the users what their requirements may look like when they are implemented into a finished system.
- ► This throwaway prototype will not be included in the final system



### Evolutionary prototyping

- Evolutionary prototyping is quite different from <u>throwaway prototyping</u>.
- The main goal when using evolutionary prototyping is to build a very robust prototype in a structured manner and constantly refine it.
- Evolutionary prototype, when built, forms the heart of the new system, and the improvements and further requirements will then be built.
- It forms the part of the final system

