

SE - Assignment - 1

Name: Kashmira suktankar

Roll no - 9642

Q1) significance of recognizing software requirements in the software engineering

- As technology advances so does the user requirement and the environment in which the software is working thus every organisation is ranked based on the software engineering principles by the organisation

- Implementing and managing large size of software program requires a specific method modularize the tasks so that size of software can't lean quality
- Software engineering provides methodology to plan out complex software systems
- Without any standard method or management, it is difficult to point out faults in the product & improve
- Extending a software's features becomes easier due to functionality because the cost to create a new software much greater than the cost to add few functions.

2) Describe main characteristics of different process models used in software development.

Waterfall model	V-model	Incremental model	Iterative model
-----------------	---------	-------------------	-----------------

Waterfall Model

- characteristics → Sequential & linear approach [Each phase must be completed before moving on to the next]
- clear + structured.
- suitable for projects with • lesser complexity
- Well defined requirements
 - minimal changes • stable space
- Drawbacks → limited flexibility.
- minimal changes thus less adaptability.
 - potential for late stage discovery of errors

Validation & Verification model (V-model)

- Characteristics
- Parallel development + testing approach. each development phase is followed by a corresponding testing phase.
 - Strong emphasis laid on verification and validation of developments in model. Verification at this stage, clear documentation reduces risk by early stage error identification
- Drawbacks → limited adaptability to changing environment
- potential scope for miscommunication between development and testing phases.

Incremental model

- model built in increments each delivering a certain functionality.
 - early arrival of functional modules reduces time to market thus allowing for better integration testing
- Drawbacks - requires careful planning to define increments possible integration challenges.

Iterative model

- similar to agile but with more structured and defined phases. Each iteration may include a subset of the software functionality.
 - allows for iterations, refined features and defined phases. Each iteration may include a subset of the software functionality.
- Drawbacks - requires clean planning coordination between iteration. Potential for scope creep

How does the capability maturity model (CMM) contribute to improving software development process?

CMM models application in software development has sometimes been problematic. Applying multiple models that are not integrated is their within and across an organisation could be costly in training appraisers and improvement activities.

→ The capability maturity model integration (CMMI) project was formed to sort out the problem of using multiple models for software development processes capability. Thus the CMMI model continues to be a general theoretical process capability model used in the public domain.

→ CMMI framework consists of a collection of computer programs based on knowledge, engineering, software engineering, integrated product and process development and providing sourcing

CMMI framework has 3 groups as

- 1) CMMI for development (CMMI-DEV)
- 2) CMMI for service (CMMI-SVC)
- 3) CMMI for acquisition (CMMI-AQ)

<p>Perspective process model</p> <ul style="list-style-type: none"> → Developed to bring order & structure to software development process → It can accommodate changing requirements → More popular → Waterfall model and incremental models are a few examples of perspective process model. 	<p>Evolutionary process model</p> <ul style="list-style-type: none"> → Stages consist of growing increments of an operational software product with evolution → Improvement is required in the product → Less popular → e.g. Spiral and prototyping model as well as RAD model
<p>Q5. Provide examples of situations where using a specific process model would be more suitable</p> <ol style="list-style-type: none"> ① <u>Incremental model</u> - When a project can be divided into smaller functional increments, allowing certain modules to be developed and delivered independently while ensuring integration and testing along the way. ② <u>RAD model</u> - When there is a need to quickly produce a working prototype to gather user feedback and make requirements before proceeding with full development. ③ <u>Waterfall model</u> - When requirements are stable and changes are minimal, making it possible to plan and execute the project in a linear sequence of phases. ④ <u>Agile model (Scrum)</u> - When flexibility and adaptability are crucial and project can be divided into smaller increments with frequent interactions allowing for continuous feedback and changes. 	

Q6 Compare and contrast the waterfall model and agile methodologies in terms of project planning and progress tracking

- Waterfall model is the first approach used in software development process
- It is also called as a classical life cycle model or linear sequential model.
- In waterfall model any phase of development process begins only if previous phases is completed
- Agile software development describes an approach to software development where the requirements are unclear and the solutions evolve through the collaborative effort of self organizing and cross functional teams and their ^{custom} customization
- It advocates adaptive planning, evolutionary development, delivery of developments and continual improvement and it encourages rapid and flexible responses to change
- the term agile was popularized in this context by the manifesto for agile software development

Q7 Apply process metrics to evaluate the efficiency and effectiveness of waterfall, agile (both scrum and kanban) methodologies, considering factors such as development speed adaptability speed adaptability to change and customer satisfaction.

WATERFALL

- * Development speed - linear and sequential methodology where each phase must be completed before moving on to the next leading to longer development cycles
Metrics - Time taken for each phase [requirements, design, development, testing]
- * Adaptability to change - less adaptable to changes due to rigid structure
Metrics - Number of change requests, impact analysis time & delays caused by requests.
- * Customer satisfaction - might have limited customer involvement until the end of project
Metrics - Customer feedback at the end of project post-deployment support requirements.

AGILE

- * Development speed - Agile methodologies emphasize iterative development for quicker delivery of working features
Metrics - Number of user stories completed per sprint or cycle time, velocity.
- * Adaptability to change - highly adaptable to changing requirements due to regular iterations and flexibility.
Metrics - No. of changes incorporated per sprint, cycle time, taken to respond to change requests.
- * Customer satisfaction - agile methodologies involve continuous customer feedbacks and collaboration, leading to improved satisfaction
Metrics - Regular customer feedback scores, frequency of customer involvement.

Q8. Justify the relevancy of the following comparison for software development.

Features	Waterfall model	Incremental Model	Prototyping model	Spiral model
requirement specification	well understood	not well understood	not well understood	well understood
understanding requirements	well understood	not well understood	not well understood	well understood
Availability of reusable components	No	Yes	Yes	Yes
Risk analysis	only initially	no risk analysis	no risk analysis	Yes
User involvement	only at beginning	intermediate	high	high
implementation time	long	less	less	depends on project
flexibility	Rigid	less	high	flexible
expertise required	High	high	medium	high
Cost central	Yes	No	No	Yes
Resource control	Yes	Yes	No	Yes