

Name - Gilen Pereira

Rollno - 9770

Class - FE Comp 1B

SE Assignment 1

- 1) What is the significance of recognizing software requirements in the software engineering process?
 - As the technology changes the user requirement and environment on which software is working also change so every organization is ranked based on the software engineering principle used by that organization
- Implementing and managing large size of software programs requires a specific method modularize the tasks so that size of software not harm the software quality
- Software engineering provides methodology for implementing complex software systems with high quality
- Without any standard method for management it is difficult to address defects in product and correct them as early as possible Software engineering provides this functionality of developing new software to provide that functionality.

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

- waterfall model - sequential and linear approach each phase must be completed before moving to next one
 - clear and structured, suitable for projects with well defined requirement, minimal changes and stable scope
 - Limited flexibility for changes, difficult to adapt to changing requirement, potential for late stage error discovery

V-model (validation and verification model)
parallel development and testing approach
Each development phase is followed by corresponding testing phase

- strong emphasis on validation and verification, clear documentation, reduces risk by identifying issues only
- Limited adaptability to changing requirements, potential for miscommunication between development and testing phases

• Incremental model
similar to iterative models but the software is built in increments, each delivering specific functionality

Iterative model

similar to agile but with more structured and defined phase. Each iteration may include a subset of software functionality.

- Allows for iteration refined features and early feedback, suitable for projects with evolving requirements

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

→ the CMM models application in software development has sometimes been problematic. Applying multiple model that are not integrated within and across an organization would lead in public domain

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

- CMMI framework has three groups as

- 1) CMMI for development (CMMI-DFM)
- 2) CMMI for service (CMMI-SVC)
- 3) CMMI for acquisition (CMMI-ACQ)

n) Explain the difference between prescriptive process models and evolutionary process model

prescriptive process model

- Developed to bring order and structure to the software development process

- It can accommodate changing requirement

- It is more popular

- waterfall model and incremental models are few

examples of prescriptive process model

Evolutionary process model

stages consist of grouping increments of an operational software product with evolution

improvement is required in the product

It is less popular

Eg: spiral and prototyping model as well as RAD model

5) provide examples of situations where using a specific process model would be more suitable

→ Incremental model

when project can be divided into smaller increments, allowing certain modules to be developed and delivered independently while ensuring integration and testing along the way

→ RAP model - when there is a need to quickly produce working prototype to gather user feedback and make requirement before progressing with full development

→ Waterfall model - when requirements are stable and changes are minimal, making it possible to plan and execute the project in linear sequence of phases

→ Agile model (Scrum)

when flexibility and adaptability are crucial and the project can be divided into smaller increments with frequent iterations, allowing for continuous feedback and changes

f) compare and contrast the waterfall model and agile methodologies in term of project planning and progress tracking

→ waterfall model is the first approach used in software development process

· It is also called as classical life cycle model

· In waterfall model any phase is completed

· Agile software development describes an approach to software development under which requirements and solution evolve through the collaborative effort of self-organizing teams

· It advocates adaptive, planning, evolutionary development and flexible response to change

· term agile was popularized in context by manifesto for agile software development

g) Apply project metrics to evaluate the efficiency and effectiveness of waterfall, agile (both scrum & kanban) to change and customer satisfaction

→ waterfall

development speed

· waterfall is a linear and sequential methodology where each phase must be completed before moving on the next. This can lead to longer development cycles

· metrics - Time taken for each phase

customer satisfaction

- waterfall may have limited customer involvement until the end, which could affect satisfaction metrics

customer feedback at the end of the project post-deployment support requirements

2. Agile (scrum & kanban)

development speed:

Agile methodologies emphasize incremental development allowing for quicker delivery of working features

- metrics: number of user stories completed per sprint or cycle time, velocity

Adaptability to change

- Agile methodologies are highly adaptable to changing requirements due to regular iteration and flexibility
- metrics: number of changes incorporated per sprint / cycle, time taken to respond to change requests

customer satisfaction

- Agile methodologies involves continuous customer feedback and iteration, leading to improved satisfaction

8) Justify the relevancy of the following comparison for software development models

Features	waterfall model	Incremental model	Prototyping model	spiral model
Requirement specification	well understand	not well understood	not well understood	well understood
understanding requirement	well understood	not well understood	not well understood	well understood
Availability of reusable components	no	yes	yes	yes
Risk analysis	only at the begining	no risk	no risk	yes
User involvement	only at intermediate	analysis	analysis	high
Implementation	long	less	less	depends on project
Flexibility	rigid	less	high	flexible
Expenditure required	high	high	medium	high
cost control	yes	no	no	yes
Resource control	yes	yes	no	yes