DOMS | Page No.
Date / /

Ishita Yadav, 9649, TE Comps B.

SE Assignment 1.

1.) i.) As the technology changes, the user requirements and environment on which software is working also changes. So every organization is ranked based on the software engineering principles by that organization.

ii.) Implementing & managing large size of software programmer requires a specific method modularize the tasks so that size of software can't harm the software quality.

iii.) Software engineering provides methodology for implementing complex software systems with high quality.

iv.) Without any standard method or management, it is difficult to address defects in the product and correct them as early as possible. Software engineering provides this functionality.

v.) Extending the previous software to add new functionality requires more cost in terms of time to develop and efforts takes by people, as compare to the process of developing new software to provide that functionality.

vi.) Software engineering provides a way in which software system can be able to scale as needed in future.

2.] a) Waterfall model: Sequential and linear approach; Each phase must be completed before moving to the next one.

1) Clear and structured, suitable for projects with well-defined requirements, minimal changes and stable scope.

2) Limited flexibility for changes, difficult to adapt to evolving requirements, potential for late-stage or cross discovery.

b) V-model (validation and verification model): Parallel development and testing approach. Each development phase is followed by a corresponding testing phase.

DOMS | Page No.
Date / /

1.) Strong emphasis on validation and verification, clear documentation, reduces risk by identifying issues early.
2.) Limited adaptibility to changing requirements, potential for miscommunication between development and testing phases.

-) Incremental model: Similar to iterative modules, reduced time to market allows for better integration testing.
2.) Require 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 softwares functionality.
1.) Allows for iterations, refined features and early feedback, suitable for projects with evolving requirements.
2.) Requires clear planning and coordination between iterations, potential for scope creep.

3.] 1.) The CMM models application in software development has sometimes been problematic. Applying multiple models that are not integrated within and across an organization could be costly in training appraisals, and improvement activities.
2.) The capability maturity model integration (CMMI) project was formed to sort out the problem of using multiple models for software development processes, thus the CMMI model has superseded the CMM models though the CMM model continues to be a general theoritical process capability model used in the public domain.
3.) CMMI framework has three groups as:
i.) CMMI for development (CMMI-DEV)
ii.) CMMI for service (CMMI-SVC)
iii.) CMMI for aquisation (CMMI-ACQ)

DOMS | Page No.
Date / /

4.)

Perspective process model.	Evolutionary process model
i.) Developed to bring order and structure to the software development process.	i.) Stages consists of growing increments of an operational software product, with evolut^{n}.
ii.) It can accommodate changing requirement.	ii.) Improvement is required in the product.
iii.) It is more popular.	iii.) It is less popular.
iv.) Eg.: Waterfall model & incremental models.	iv.) eg.: spiral & prototyping model and RAD model.

5.] i.) Incremental model: 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.) RAD model: When there is a need to quickly produce a working prototype to gather user feedback and make refinements before proceeding with full development.

3.) Waterfall model: When requirements are stable and changes are minimal, making it possible to plan and execute the project in a linear sequence of phases.

4.) Agile method (scrum): When flexibility is crucial & project can be divided into smaller increments with frequent ~~int~~ iterations, allowing for continuing feedback & changes.

6.] i.) Waterfall model is the first approach used in software dev process.

2.) It is also called as classical life cycle model or linear sequential model.

3.) It advocated adaptive planning evolutionary development, early delivery and continual improvement & it encourages rapid & flexible responses to change.

4.) The terms agile was popularized, in this context, by the manifesto for agile software development.

7.] i) Waterfall

- Development speed:

i) Waterfall is a linear & sequential methodology where each phase must be completed before moving on the next. This can lead to longer development cycles.

- Adaptibility to change:

i) Waterfall is less adaptable to changes in requirement due to its rigid structure.

ii) Metrics: number of change requests, impact analysis time & delays caused by change requests.

- Customer satisfact^n:

i) Waterfall may have limited customer involvement until the end, which could affect satisfaction.

ii) Agile (scrum & kanban):

- Development speed:

i) Agile methodologies emphasize incremental development, allowing for quicker delivery of working features.

ii) Metrics: Number of user stories completed per sprint or cycle time, velocity.

- Adaptibility to change:

i) Agile methodologies are highly adaptable to changing requirements due to regular iterations & flexibility.

- Customer satisfaction:

i) Agile methodologies involve continous customer feedback and collaboration, leading to improved satisfaction.

ii) Metrics: Regular customer feedback scores, freq. of customer involvement.

DOMS | Page No.
Date / /

8.]

	Features	Waterfall model	Incremental model	Prototyping model	Spiral model
1.	Requirement specificat"	Well understood	Not well understood	Not well understood	Well understood
2.	Understanding requirements	Well understood	Not well understood	Not well understood	Well understood
3.	Availability of ~~the~~ reusable components	No	Yes	Yes	Yes
4.	Risk analysis	Only at the begianing	No risk analysis	No risk analysis	Yes
5.	User involvement	Only at the beginning	Intermediate	High	High
6.	Implementation time	long	less	less	depends on projects
7.	Flexibility	Rigid	less	high	flexible
8.	Expertise required	high	high	medium	high
9.	Cost control	Yes	no	no	yes
10.	Resource control	Yes	Yes	No	No