MOHAMED SATHAK A J COLLGE OF ENGINEERING, Chennai - 603103

Department of Information Technology

CS8494 SOFTWAREENGINEERING **Year/Sem:III/V**

PART A Questions

Unit 1- Software process and Agile development

1. **Differentiate : Verification Vs Validation. NOV / DEC 2017**

Verification	**Validation**
The set of activities that ensure that software correctly implements a specific function.	The set of activities that ensure that the software has been built is traceable to customer requirements
Verification represents the set of activities that are carried out to confirm that the software correctly implements the specific functionality	Validation represents the set of activities that ensure that the software that has been built is satisfying the customer requirements.

2. **What is Software Engineering? NOV/DEC 2013, NOV / DEC 2014, APRIL/MAY 2017,APRIL/MAY 2017**

Software engineering is the application of a systematic, disciplined, approach to the development, operation and maintenance of software as well as the study of approaches of the same.

3. **Define an evolutionary prototype.**

Evolutionary models are iterative. They are characterized in a manner that enables you to developincreasingly more complete versions of the software.

Software prototyping, refers to the activity of creating prototypes of software applications, i.e., incomplete versions of the software program being developed.

4. **List two deficiencies in waterfall model. Which process model do you suggest to overcome eachdeficiency? APRIL/MAY 2017**

- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- No working software is produced until late during the life cycle.

5. **What is the significance of the spiral model when compared with other models. NOV/DEC 2017**

- High amount of risk analysis.
- Good for large and mission-critical projects.
- Software is produced early in the software life cycle.

6. **What is software process?**

The process model can be defined as the abstract representation of process. The appropriate process model can be chosen based on abstract representation of process. These process models will follow some rules for correct usage.

7. **List any two agile process model.**

The following are the agile process model.

- Extreme Processing (XP)
- Agile Modeling
- Scrum

8. **What is software? List the characteristics. APRIL/MAY 2018**

Software is instructions (computer programs) that are intended to provide desired Features, function, and performance Characteristics:

Software is developed or engineered; Software doesn't "wear out" most software continues to be custom built.

9. **Write a note on the unique characters of software. NOV/DEC 2017**

• Functionality

• Reliability

• Usability

• Efficiency

10. If you have to develop a word processing software product, what process model will you choose? Justify your answer. NOV/DEC 2016

Incremental model: incremental paradigm might deliver basic file management, editing, and document production functions in the first increment; more sophisticated editing and document production capabilities in the second increment; Spelling and grammar checking in the third increment; and advanced page layout capability in the fourth increment

11. What are the advantages and disadvantages of iterative software development model NOV/DEC 2015

Advantages

In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product.

- Building and improving the product step by step.
- can get the reliable user feedback
- Less time is spent on documenting and more time is given for designing.

Disadvantages

- Each phase of an iteration is rigid with no overlaps
- Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle

12. Define agility and agile team. <u>April /May 2015</u>

- Agility-Effective (rapid and adaptive) response to change (team members, new technology, requirements)
- Effective communication in structure and attitudes among all team members, technological and business people, software engineers and managers。
- Drawing the customer into the team. Eliminate “us and them” attitude.Planning in an uncertain world has its limits and plan must be flexible.
- Organizing a team so that it is in control of the work performed
- The development guidelines stress delivery over analysis and design although these activates are not discouraged, and active and continuous

Unit-2 requirements Analysis and Specifications

1. **What is software? List the characteristics. APRIL/MAY 2018**

Software is instructions (computer programs) that are intended to provide desired Features, function, and performance

Characteristics:

Software is developed or engineered;

Software doesn't "wear out"

most software continues to be custom built.

2. **What are the linkages between data flow and E-R Diagram? (APRIL/MAY 2016)**

An ER diagram is the Entity Relationship Diagram, showing the relationship between different entities in a process.

A Data Flow diagram is a symbolic structure showing how the flow of data is used in different process stages.

3. **List the characteristics of a good SRS. (APRIL/MAY 2016)**

- Correct – The SRS should be made up to date when appropriate requirements are identified.
- Unambiguous – When the requirements are correctly understood then only it is possible towrite an unambiguous software.
- Complete – To make SRS complete,it shold be specified what a software designer wants tocreate software.
- Consistent – It should be consistent with reference to the functionalities identified.
- Specific – The requirements should be mentioned specifically.
- Traceable – What is the need for mentioned requirement? This should be correctly identified.

4. **Classify the following as functional / non-functional requirements for a banking system.(NOV/DEC 2016)**

- Verifying bank balance - functional Requirements
- Withdrawing money from bank - functional Requirements

- Completion of transactions in less than one second – Non-functional Requirements
- Extending the system by providing more tellers for customers.- functional Requirements

5. What is a data dictionary? (NOV/DEC 2016) (NOV/DEC 2015)

The data dictionary can be defined as an organized collection of all the data elements of the system with precise and rigorous definitions so that user and system analyst will have a common understanding of inputs, outputs, components of stores and intermediate calculations.

6. What is the need for feasibility analysis? (APRIL/MAY 2015)

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions.

7. How are the requirements validated? (APRIL/MAY 2015)

While designing the user interface of software the requirement collection can be done by focusing on the profile of user who will interact with the system. Skill level, business understanding and general grasping to the new system are recorded. Users can be categorized into different categories and for each category of user requirements are elicitations.

8. **What do you mean by Functional and non-functional requirement? (APRIL/MAY 2014, APR/MAY 2019)**

Functional requirements:

These are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do.

Non-functional requirements:

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process and standards. Non-functional requirements often apply to the system as a whole. They do not usually just apply to individual system features or services.

9. What is Petri net? NOV/DEC 2019

Petri nets are a basic model of parallel and distributed systems. The basic idea is to describe state changes in a system with transitions.

Petri nets — Formal technique for describing concurrent interrelated activities.

10. Define feasibility study and list the types. (NOV/DEC 2015)

Feasibility is defined as the practical extent to which a project can be performed successfully. Type of feasibility:

- Technical feasibility,
- 2.Operational feasibility, and
- 3.Economic feasibility

11. What are the various types of traceability in software engineering? April/may 2018

- Source traceability – These are basically the links fromrequirement to stakeholders
- Requirements traceability – These are links betweendependant requirements.
- Design traceability – These are links from requirements to design.

12. What is the outcome of feasibility study?

The outcome of feasibility study is the results obtained from the following questions: x Which system contributes to organizational objectives? x Whether the system can be engineered? Is it within the budget? x Whether the system can be integrated with other existing system?

Unit -3 Softwre Design

1.Define a component. Give example. NOV/DEC 2019

❖ Component-level design occurs after the first iteration of the architectural design

❖ A component-level design can be represented using some intermediate representation (e.g.

graphical, tabular, or text-based) that can be translated into source code

❖ The design of data structures, interfaces, and algorithms should conform to well-established

guidelines to help us avoid the introduction of errors.

2 . List down the steps to be followed for User Interface design. (APRIL/MAY 2015)

1. Using information developed during interface analysis, define interface objects and actions (operations).

2. Define events (user actions) that will cause the state of the user interface to change. Model this behavior.

3. Depict each interface state as it will actually look to the end user.

4. Indicate how the user interprets the state of the system from information provided through the interface.

3.What are the golden rules for an interface design? (NOV/DEC 2015)

• Place the User in Control

• Reduce the User's Memory Load

• Make the Interface Consistent

4. What is the need for architectural mapping using data flow? (MAY/JUNE 2016, APRIL/MAY 2017)

It Provides a method to go from a DFD to program structure

- The type of information flow is established
- Flow boundaries are indicated
- The DFD is mapped into program structure
- Control hierarchy is defined
- Resultant structure is refined using design measures and heuristics
- The architectural description is refined and elaborated.

5. What architectural styles are preferred for the following systems? Why? (NOV/DEC 2016)

(a) Networking - Client server Architecture/Remote procedure call architectures

(b) Web based systems - N-Tier / 3-Tier Architecture

(c) Banking system. - Layered Architecture

6. Write a note on FURPS model of design quality. (NOV/DEC 2015) (NOV/DEC 2017)

FURPS is an acronym representing a model for classifying software quality attributes (functional and non-functional requirements):

- Functionality
- Usability
- Reliability

- Performance
- Supportability

7. If a module has logical cohesion, what kind of coupling is this module likely to have?

(MAY/JUNE 2016)

When a module that performs a tasks that are logically related with each other is called logically cohesive. For such module content can be suitable for coupling with other modules.

The content coupling is a kind of coupling when one module makes use of data or control information maintained in other module.

8. What is Inheritance? NOV/DEC 2019

Inheritance in software design model is interfacing module will be done from top (base class) to bottom (derived classes)

9. What UI design patterns are used for the following? (NOV/DEC 2016) (APRIL/MAY 2017)

(a) Page layout – cards, Grid

(b) Tables - Alternating row color, Table filter, sort by column

(c) Navigation through menus and web pages - vertical dropdown menu, horizontal dropdown menu, accordion menu

(d) Shopping cart- product page, pricing table, coupon, shopping cart

10. List four design principles of a good design? APR/MAY- 11APRIL/MAY 2018

- Process should not suffer from tunnel vision.
- It should be traceable to the analysis model
- It should not reinvent the wheel
- It should exhibit uniformity & integration.

11.What are the elements of design model?

- Data design
- Architectural design
- Interface design

- Component-level design

12.What is Coupling?What are the various types of coupling APRIL/MAY-15

Coupling is the measure of interconnection among modules in a program structure. It depends on the interface complexity between modules.

- Data coupling – The data coupling is possible by parameter passing or data interaction.
- Control coupling – The modules share related control data in control coupling.
- Common coupling – The common data or a global data is shared among modules.
- Content coupling-Content couplingoccurs when one module makes use of data or control information maintained in another module.

Unit-4 Testing and Maintenance

1. **What is the need for regression testing? (APR/MAY 2015)**

Regression testing may be conducted to ensure that new errors have not been introduced. Regression Testing is required when there is a

- Change in requirements and code is modified according to the requirement
- New feature is added to the software
- Defect fixing
- Performance issue fix

2. **Define Reverse Engineering. APR/MAY 2019**

Reverse engineering is a process of de-constructing a system in order to extend the functionalities or in order to understand the working of the system.

3. **What is test case? NOV/DEC 2019**

A TEST CASE is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

4. **List the levels of testing. APR/MAY 2019**

The developed software should be tested in the following order

- Unit Testing
- Integration Testing

- System testing
- Acceptance Testing.

5. What are the testing principles the software engineer must apply while performing the software testing? (MAY 18)

- All tests must be traceable to customer requirements
- Tests should be planned long before testing begins
- Testing should begin in small and progress towards testing in large.
- Exhaustive testing is not possible.
- Testing should be done independent third party.

6. List two customers related and technology related risks. (APR/MAY 2017)

Customer related risk

• Have you worked with the customer in the past?

• Does the customer have a solid idea of what is required?

• Technology related risk

• Is the technology to be built new to your organization?

•Do the customer's requirements demand the creation of new algorithms or input or output technology?

7. How can refactoring be made more effective? (APR/MAY 2016)

There are two general categories of benefits to the activity of refactoring.

1. **Maintainability**. It is easier to fix bugs because the source code is easy to read and the intent of its author is easy to grasp. This might be achieved by reducing large monolithic routines into a set of individually concise, well-named, single-purpose methods.

2. **Extensibility**. It is easier to extend the capabilities of the application if it uses recognizable design patterns, and it provides some flexibility where none before may have existed.

8. Outline the needs of system testing. NOV/DEC 2019

A classic system-testing problem is "finger pointing." This occurs when an error is uncovered, and the developers of different system elements blame each other for the problem.

9. Why does software fail after it has passed from acceptance testing? (APR/MAY 2016)

During acceptance testing, the random input is used for testing. This may lead to the situation that some input values that may cause failure go unhandled. The practical problem with acceptance testing is that it is time consuming. Hence in order to keep testing cost low, there is restricted number of test cases.

10. What is smoke testing ? APRIL /MAY 2017

Smoke Testing, also known as "Build Verification Testing", is a type of software testing that comprises of a non-exhaustive set of tests that aim at ensuring that the most important functions work. The results of this testing is used to decide if a build is stable enough to proceed with further testing.

11. Why does software fail after it has passed from acceptance testing?APR/MAY 2016

Each acceptance test represents some expected result from the system. Customers are responsible for verifying the correctness of the acceptance tests and reviewing test scores to decide which failed tests are of highest priority. Acceptance tests are also used as regression tests prior to a production release. A user story is not considered complete until it has passed its acceptance tests. This means that new acceptance tests must be created for each iteration or the development team will report zero progress.

12. Distinguish between alpha and beta testing. MAY/JUNE 2016

Alpha and beta testing are the types of acceptance testing.

Alpha test: The alpha testing is attesting in which the version of complete software is tested by the customer under the supervision of developer. This testing is performed at developer's site.

Beta test: The beta testing is a testing in which the version of the software is tested by the customer without the developer being present. This testing is performed at customer's site.

13. Write about drivers and stubs. NOV/DEC 2017

Drivers and stub software need to be developed to testincompatible software. The "driver" is a program that accepts the test data and prints the relevant results.

The "stub" is a subprogram that uses the module interfaces and performs the minimal data manipulation if required.

Unit –5 Project management

1. Highlight the activities in Project Planning. (APR/MAY 2015)

- Software scope

- Resources
- Project estimation
- Decomposition

2. What are the different types of productivity estimation measures? (APR/MAY 2017)

- Function Point and Function Point Analysis
- COCOMO
- Cyclomatic Complexity

3. List out the principles of project scheduling. (NOV/DEC 2017)

- Compartmentalization
- Interdependency
- Time allocation
- Effort validation
- Defined responsibilities
- Deined outcomes
- Defined milestones

4. Write a note on Risk Information Sheet (RIS). (NOV/DEC 2017)

The Risk Information Sheet documents a Risk that may during the life-time of a specific Software Project. Risk Information Sheets can be used in to supplement or in the place of a formal Risk Mitigation, Monitoring and Management (RMMM) Plan

5. Compare project risk and Business risk. APR/MAY 2019

Project risk - that the building costs may be higher than expected because of an increase in materials or labor costs.

Business risk - even if the stadium is constructed on time and within budget that it will not make money for the business.

6. What is budgeted cost of work scheduled? NOV/DEC 2019

The budgeted cost of work scheduled (BCWS) is determined for each work task represented in the schedule. During estimation, the work (in person-hours or person-days) of each

software engineering task is planned. BCWS= sum of BCWSi Hence, BCWSi is the effort planned for work task i

7. Write any two differences between "known risk" and predictable risk". NOV/DEC 2019

Known risk: It can be uncovered after careful evaluation project plan, business and technical environment in which the project is being developed, other reliable information resources. E.g. unrealistic delivery date, lack of software poor development environment. Predictable risks are those risks that can be identified in advance based on pas project experience. For example: Experienced and skilled staff leaving the organization in between

8. List two advantages of COCOMO model. APR/MAY 2019

- COCOMO Model is used to estimate the project cost
- COCOMO is easy to interpret, predictable and accurate.

9. List two customers related and technology related risks. (APR/MAY 2017)

Customer related risk:

Have you worked with the customer in the past?

Does the customer have a solid idea of what is required?

Technology related risk:

Is the technology to be built new to your organization?

Do the customer's requirements demand the creation of new algorithms or input or output technology?

10. How is productivity and cost related to function points? (NOV/DEC 2016)

Inconsistent productivity rates between projects may be an indication that a standard process is not being followed. Productivity is defined as the ratio of inputs/outputs. For software, productivity is defined as the amount of effort required to deliver a given set of functionality. The true cost of software is the sum of all costs for the life of the project including all expected enhancement and maintenance costs. The more invested up front should reduce per unit cost for future enhancement and maintenance activities. The unit cost can be hours/FP or $/FP

11. What are the types of metrics? MAY/JUNE 2016

Direct metrics – It refers to immediately measurable attributes.

Example- Lines of code,execution speed.

Indirect metrics – It refers to the aspects that are not immediately quantifiable or measurable.

Example – functionality of a program.

12. What is risk management? NOV/DEC2016

Risk management is the identification, assessment, and prioritization of risks followed by coordinated and economical application of resources to minimize, monitor, and control the probability and/or impact of unfortunate eventsor to maximize the realization of opportunities. Risk management's objective is to assure uncertainty does not deflect the endeavor from the business goals.

PART B Questions

Unit 1- Software process and Agile development

1. Compare the following life cycle models based on their distinguishing factors, strengths and weaknesses — Waterfall Model, RAD Model, Spiral Model and Formal Methods Model. (Present in the form of table only — use diagrams wherever necessary) **NOV/DEC 2013, NOV/DEC 2018**

2. Discuss the prototyping model. What is the effect of designing a prototype on the overall cost of the software project? **MAY/JUNE 2016**

3. Describe the type of situations where iterative enhancement model might lead to difficulties

   MAY/JUNE 2016

4. Elucidate the key features of the software process models with suitable examples. **MAY/JUNE 2016**

5. List the principles of agile software development. **NOV/DEC 2016, NOV/DEC 2019**

6. What is a process model? Describe the process model that you would choose to manufacture a car.Explain giving suitable reasons. **MAY/JUNE 2017, NOV/DEC 2019 (Spiral Model)**

Unit-2 requirements Analysis and Specifications

1. What is feasibility study? How it helps in requirement engineering process? **NOV / DEC 2017**

2. What is requirement engineering? Explain in detail the various processes in requirements engineering. **APRIL / MAY 2017, NOV/DEC 2019**

3. Write a note on what are the difficulties in elicitation, requirement elicitation. **APRIL / MAY 2017,APRIL/MAY 2017**

4. Explain the organization of SRS and highlight the importance of each subsection. **MAY /JUNE2016, APRIL/MAY 2017**

5. Differentiate between user and system requirements. **MAY /JUNE 2016**

6. Describe the requirements change management process in detail. **MAY /JUNE 2016**

7. What is requirements elicitation? Briefly describe the various activities performed in requirementselicitation phase with an example of a watch system that facilitates to set time and **alarm. NOV / DEC 2016**

Unit -3 Softwre Design

1. Explain the various coupling and cohesion methods used in Software design. **(APR/MAY 2015and NOV/DEC 2015, APR/MAY 2017)**

2. Discuss about User Interface Design of a software with an example and neat sketch **.(NOV/DEC 2015 and NOV/DEC 2017)**

3. Write short notes on the following

(i) Design heuristics

(ii) User-interface design

(iii) Component level design

(iv) Data/Class design **(APR/MAY 2016)**

4. Discuss the differences between Object Oriented and Function Oriented Design. **(APR/MAY2016)**

5. What is structured design? Illustrate the structured design process from DFD to structured chartwith a case study. **(NOV/DEC 2016)**

6. Describe the golden rules for interface design**. (NOV/DEC 2016)**

7. Explain component level design with suitable examples. **(NOV/DEC 2016)**

8. What is software architecture? Describe in detail different types of software architectures with illustrations. **(APR/MAY 2017,NOV/DEC 2019) – Architectural styles**

9. Discuss about the design concepts in a software development process. **(NOV/DEC 2017)**

10. Outline the steps in designing class based components with an example**. NOV/DEC 2019**

Unit-4 Testing and Maintenance

1. Elaborate path testing and regression testing with an example**. NOV/DEC2019**

2. Explain how business process Reengineering helps to achieve a defined businessoutcome**. NOV/DEC2019**

3. Explain various levels of software testing with suitable example .**NOV/DEC2019**

4. List the process in software reengineering process model and explain in detail.**APR/MAY 2019**

5. With suitable example, explain boundary value analysis. **APR/MAY 2019**

Unit -5 Project management

1. State the need for Risk Management and explain the activities under Risk Management.
(APRIL/MAY 2015) (NOV/DEC 2015) (APRIL/MAY 2017)

2. Write short notes on the following**(APRIL/MAY 2015)**

(i) Project Scheduling
(ii) Project Timeline chart and Task network

3. Discuss about COCOMO II model for software estimation.
(NOV/DEC2015)(APRIL/MAY 2017)(NOV/DEC 2019)

4. Write short notes on the following :**(APRIL/MAY 2016)**

(i) Make/Buy decision
(ii) COCOMO II

5. Discuss in detail about project scheduling techniques.
(APRIL/MAY 2021)

6. List the features of LOC and FP based estimation models. Compare the two models and list the advantage of over one other. **APR/ MAY 2019**

7. Define risk. List types of risk and explain phases in risk management**. APR/ MAY 2019, NOV/DEC 2019**