

Academic year: 2020-21
Sem-In Examinations-I,
B. Tech. (CSE), 2019 Batch
II/IV, 2nd Semester
19CS2211: Software Engineering
Scheme of Evaluation

Time: 2 hours

Max. Marks: 50

(Assume any missing data suitably and design adequate hypothesis, if required)

CO1		Max.Marks 25
Answer ALL Questions		
1.	<p>Define Software & Software Engineering. 4.5 M</p> <p><i>instructions (computer programs) that when executed provide desired features, function, and performance</i> 2M</p> <p><i>[Software engineering is] the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.</i> 2.5M</p>	
	(Or)	
2.	<p>List out the characteristics of software in comparison with hardware characteristics. 4.5 M</p> <ul style="list-style-type: none"> • <i>Software is developed or engineered, it is not manufactured in the classical sense.</i> 1M • <i>Software doesn't "wear out."</i> 1M • <i>Although the industry is moving toward component-based construction, most software continues to be custom-built.</i> 1M <p>They need to explain and draw graphs 1.5M</p>	
3.	<p>Outline Software Application Domains. 8M</p> <ul style="list-style-type: none"> • System software • Application software • Engineering/scientific software • Embedded software • Product-line software • WebApps (Web applications) • AI software <p>Each domain carries one mark 7*1==7M and example carries one mark == 1M</p>	
	(Or)	
4.	<p>In what way can we relate Product to Process. 8M</p>	

	<p>Product: In the context of software engineering, Product includes any software manufactured based on the customer's request. This can be a problem solving software or computer based system. It can also be said that this is the result of a project. 3M</p> <p>Process: Process is a set of sequence steps that have to be followed to create a project. The main purpose of a process is to improve the quality of the project. The process serves as a template that can be used through the creation of its examples and is used to direct the project. 3M</p> <p>The main difference between a process and a product is that the process is a set of steps that guide the project to achieve the convenient product. while on the other hand, the product is the result of a project that is manufactured by a wide variety of people. 2M</p>
(1 X 12.5M=12.5M)	
5.	<p>(a) Summarize Management Software Myths and realities in software engineering. 6M</p> <ul style="list-style-type: none"> • We already have a book of standards and procedures for building software. It does provide my people with everything they need to know ..." 2M • "If my project is behind the schedule, I always can add more programmers to it and catch up ..." 2M (a.k.a. "The Mongolian Horde concept") <p>"If I decide to outsource the software project to a third party, I can just relax: Let them build it, and I will just pocket my profits 2M</p> <p>They has to write realities also</p> <p>(b) Explain how process framework activities helpful while developing the software. 6.5M</p> <ol style="list-style-type: none"> 1. Communication 1M 2. Planning 1M 3. Modeling 1M <ul style="list-style-type: none"> – Analysis of requirements – Design 4. Construction 2M <ul style="list-style-type: none"> – Code generation – Testing 5. Deployment 1.5M <ul style="list-style-type: none"> – Delivery – Feedback
(Or)	
6.	<p>(a) Examine how spiral model is useful in real life and write its phase 6M</p>

	<ul style="list-style-type: none"> • It is a realistic approach to the development of large-scale systems and software. • The software evolves as the process progresses, the developer and customer better understand and react at each evolutionary level. • It uses prototyping as a risk reduction mechanism. <p>It demands considerable risk assessment expertise and realise on this expertise for success 2M</p> <p>Diagram-2M</p> <p>Adv and disadvantages -2M</p> <p>(b) Provide three examples of software projects that would be amenable to the waterfall model and explain it.</p> <p><u>Waterfall Model</u></p> <ul style="list-style-type: none"> • Oldest software lifecycle model & best understood by upper management • Used when requirements are well understood and risk is low • Work flow is in a linear fashion (i.e., sequential) <p>Used often with well-defined adaptations or enhancements to current software2M</p> <p>Diagram-2M</p> <p>Examples-0.5M</p> <p>Adv and disadvantages -2M 6.5M</p>
CO2 Max.Marks 25	
Answer ALL Questions	
7	<p>Sketch the Lifecycle activities of Extreme Programming with a neat diagram.</p> <ul style="list-style-type: none"> • XP Planning <p>XP Design</p> <ul style="list-style-type: none"> – Follows the KISS(Keep it Simple, Stupid) principle – Encourage the use of CRC cards (Class Responsibility Collaboration) – For difficult design problems, suggests the creation of “spike solutions”—a design prototype(A Working Model) – Encourages “refactoring”—an iterative refinement of the internal program design <p>XP Coding</p> <ul style="list-style-type: none"> – Recommends the construction of a unit test for a store <i>before</i> coding commences – Encourages “pair programming” <p>XP Testing</p> <ul style="list-style-type: none"> – All unit tests are executed daily <p>“Acceptance tests” are defined by the customer and executed to assess customer visible functionality 2M</p>

	Diagram-2.5M 4.5 M
	(Or)
8	<p>Illustrate the process of incorporating quality in generating requirements, for any social networking website? 4.5 M</p> <p>Requirement Engineering is the process of defining, documenting and maintaining the requirements. It is a process of gathering and defining service provided by the system. Requirements Engineering Process consists of the following main activities:</p> <ul style="list-style-type: none"> • Requirements elicitation 1M • Requirements specification 1M • Requirements verification and validation 1M • Requirements management 1M <p>Examples 0.5M</p>
9	<p>Identify specific agile methods that follow the core principles of Agility. 8M</p> <p>Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</p> <ol style="list-style-type: none"> 2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. 3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. 4. Business people and developers must work together daily throughout the project. 2M 5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. 6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. 2M 7. Working software is the primary measure of progress. 8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. 9. Continuous attention to technical excellence and good design enhances agility. 2M 10. Simplicity – the art of maximizing the amount of work not done – is essential. 11. The best architectures, requirements, and designs emerge from self-organizing teams. 12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. 2M
	(Or)
10	Outline the validating activities that take place at the beginning of each software

	<p>process iteration.</p> <p style="text-align: right;">8M</p> <ol style="list-style-type: none"> 1. Is each requirement consistent with the overall objectives for the system/product? 2. Have all requirements been specified at the proper level of abstraction? That is, do some requirements provide a level of technical detail that is inappropriate at this stage? 3. Is the requirement really necessary or does it represent an add-on feature that may not be essential to the objective of the system? 4. Is each requirement bounded and unambiguous? 5. Does each requirement have attribution? That is, is a source (generally, a specific individual) noted for each requirement? 6. Do any requirements conflict with other requirements? 7. Is each requirement achievable in the technical environment that will house the system or product? 8. Is each requirement testable, once implemented? 9. Does the requirements model properly reflect the information, function, and behavior of the system to be built? 10. Has the requirements model been “partitioned” in a way that exposes progressively more detailed information about the system? 11. Have requirements patterns been used to simplify the requirements model? Have all patterns been properly validated? Are all patterns consistent with customer requirements?
(1 X 12.5M=12.5M)	
11.	<p>(a) Discuss the various steps in negotiation requirements. 6M</p> <p>□ A new technology is being used Resources with needed skill set are not available</p> <p>♣ There are some high risk features and goals.</p> <p>♣ The WINWIN Spiral Model This is an adaptation of the spiral model which emphasis is explicitly placed on the involvement of the client in a negotiation process at the genesis of the product development. Ideally, the developer would simply ask the customer what is required and the customer would provide sufficient detail to proceed. Unfortunately this rarely happens and significant negotiations between both parties are required to balance functionality, performance, etc... derives its name from the objective of these negotiations, i.e. “win- win”. The client gets the product that satisfies the majority of his/her needs, and the developer wins by working to realistic and achievable budgets and deadlines. 2M</p> <p>Identification of the system stakeholders. That is the people on the organization that have</p> <p>1. Identification of the system or subsystem’s key stakeholders. 2M</p>

2. Determination of the stakeholders' "win conditions."

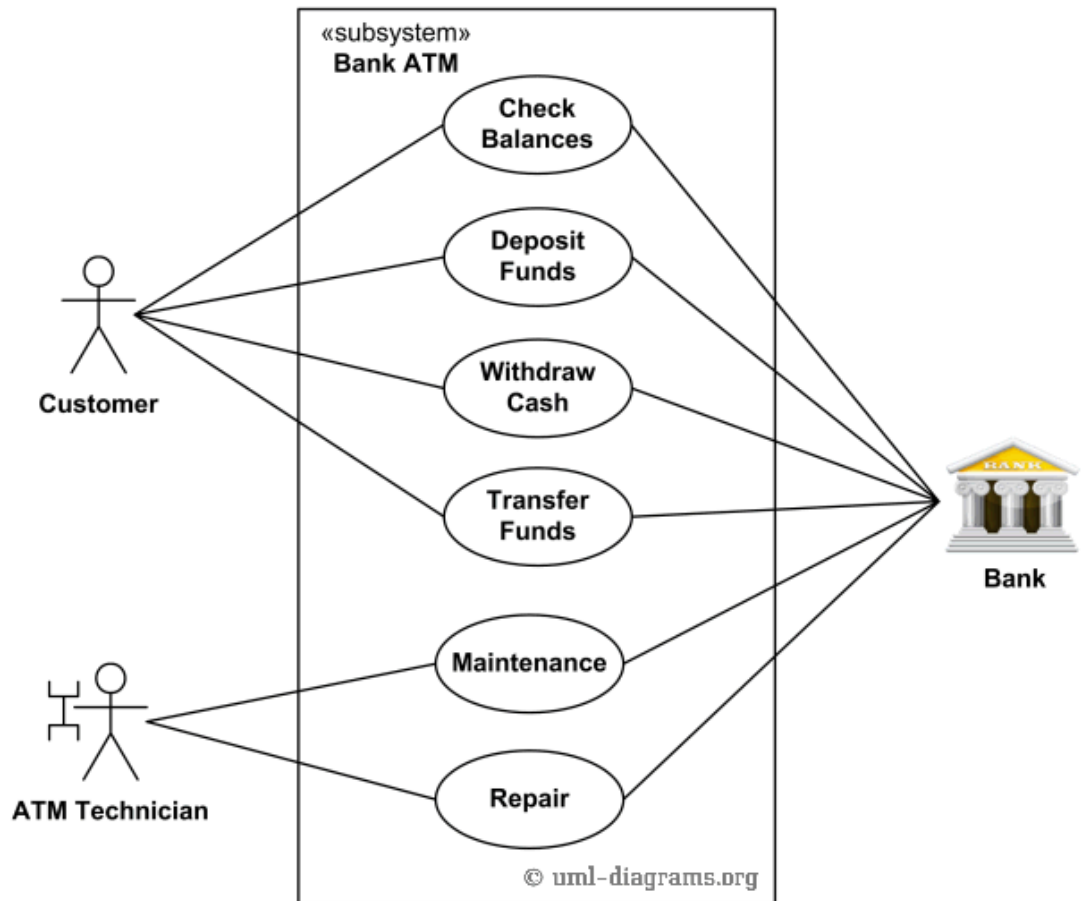
2M

3. Negotiation of the stakeholders' win conditions to reconcile them into a set of win-win conditions for all concerned (including the software team).

(b) Identify the Role of the use cases and develop a use case diagram for Online Bank ATM case study. 6.5M

Students can draw the diagram based on there perspective , but it should have proper meaning

One of the possible answer



(Or)

12. (a) List out the advantages and disadvantages of Extreme programming in the agile development process 6M

Advantages of Extreme Programming

The main advantage of Extreme Programming is that this methodology allows software development companies to save costs and time required for project realization. Time savings are available because of the fact that XP focuses on the timely delivery of final products. Extreme Programming teams save lots of money because they don't use too much documentation. They usually solve problems through discussions inside of the team.

Simplicity is one more advantage of Extreme Programming projects. The developers who prefer to use this methodology create extremely simple code that can be improved at any moment.

The whole process in XP is visible and accountable. Developers commit what they will accomplish and show progress.

Constant feedback is also the strong side. It is necessary to listen and make any changes needed in time.

XP assists to create software faster thanks to the regular testing at the development stage.

Extreme Programming contributes increasing employee satisfaction and retention.

However, as I've mentioned at the beginning of this article, there is no ideal method. That's why it's a high time to list some XP drawbacks. **3M**

Extreme Programming disadvantages

Some specialists say that Extreme Programming is focused on the code rather than on design. That may be a problem because good design is extremely important for software applications. It helps sell them in the software market.

Additionally, in XP projects the defect documentation is not always good. Lack of defect documentation may lead to the occurrence of similar bugs in the future.

One more disadvantage of XP is that this methodology does not measure code quality assurance. It may cause defects in the initial code.

XP is not the best option if programmers are separated geographically. **3M**

(b) Outline the steps required to establish the groundwork for an understanding of software requirements. **6.5M**

Requirement Engineering

The process to gather the software requirements from client, analyze and document them is known as requirement engineering.

The goal of requirement engineering is to develop and maintain sophisticated and descriptive 'System Requirements Specification' document.

Requirement Engineering Process

It is a four step process, which includes –

- Feasibility Study **2M**
- Requirement Gathering **2M**
- Software Requirement Specification **2.5M**

	<ul style="list-style-type: none">• Software Requirement Validation
--	---