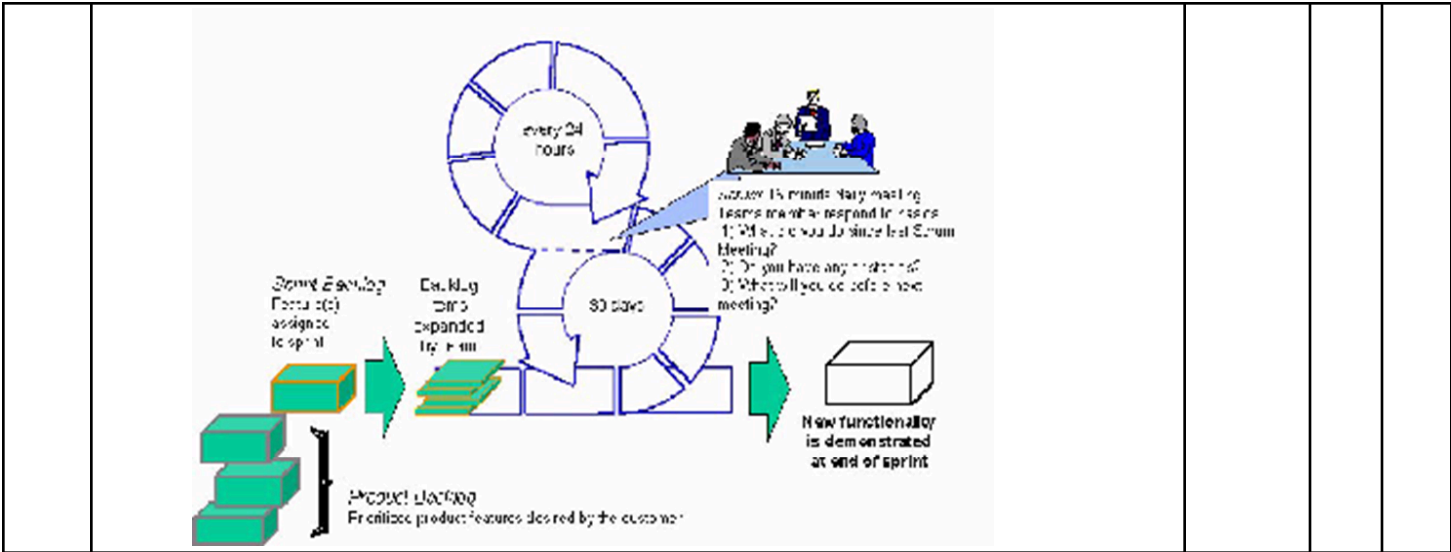


Internal Assessment Test 2 January 2024								
Sub:	Software Engineering and Project Management			Sub Code:	21CS61	Branch:	AIML	
Date :	8/07/24	Duration:90 m	Max Marks: 50	Sem :	V1		OBE	
<u>Answer any FIVE FULL Questions</u>						Marks	CO	R B T
1	What is an agile process? Explain agility principle -Is driven by customer descriptions of what is required (scenarios) -Recognizes that plans are short-lived -Develops software iteratively with a heavy emphasis on construction activities -Delivers multiple ‘software increments’ -Adapts as changes occur <u>agility principle</u> 1.Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. 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. 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. 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.				10		CO3 L3	

	<p>9. Continuous attention to technical excellence and good design enhances agility.</p> <p>10. Simplicity – the art of maximizing the amount of work not done – is essential.</p> <p>11. The best architectures, requirements, and designs emerge from self-organizing teams.</p> <p>12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.</p>			
2.	<p>Explain the extreme programming with a neat diagram.</p> <ul style="list-style-type: none"> The most widely used agile process, originally proposed by Kent Beck <p>XP Planning</p> <ul style="list-style-type: none"> Begins with the creation of “user stories” Agile team assesses each story and assigns a cost Stories are grouped to for a deliverable increment A commitment is made on delivery date After the first increment “project velocity” is used to help define subsequent delivery dates for other increments <p>XP Design</p> <ul style="list-style-type: none"> Follows the KIS principle Encourage the use of CRC cards (see Chapter 8) For difficult design problems, suggests the creation of “spike solutions”—a design prototype 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 “Acceptance tests” are defined by the customer and executed to assess customer visible functionality 	10	CO3L3	

3	<p>Explain scrum with a neat diagram..</p> <ul style="list-style-type: none"> ● Originally proposed by Schwaber and Beedle ● Scrum—distinguishing features ● Development work is partitioned into “packets” ● Testing and documentation are on-going as the product is constructed ● Work occurs in “sprints” and is derived from a “backlog” of existing requirements ● Meetings are very short and sometimes conducted without chairs ● “demos” are delivered to the customer with the time- box allocated 	10	CO3	L3



4

Explain agile modeling principles.

- Principle #1. The primary goal of the software team is to build software, not create models.
- Principle #2. Travel light—don't create more models than you need.
- Principle #3. Strive to produce the simplest model that will describe the problem or the software.
- Principle #4. Build models in a way that makes them amenable to change.
- Principle #5. Be able to state an explicit purpose for each model that is created.
- Principle #6. Adapt the models you develop to the system at hand.
- Principle #7. Try to build useful models, but forget about building perfect models.
- Principle #8. Don't become dogmatic about the syntax of the model. If it communicates content successfully, representation is secondary.
- Principle #9. If your instincts tell you a model isn't right even though it seems okay on paper, you probably have reason to be concerned.
- Principle #10. Get feedback as soon as you can.

10

CO3 L3

5

Explain the activities covered by software project management.

"A software project is not only concerned with the actual writing of software. In fact, where a software application is bought in "of the shelf", there may be no software writing as such, but this is still fundamentally a software project because so many of the other activities associated with software will still be present.

10

CO4 L3

	<pre> graph TD A[Feasibility study] --> B[Plan] B --> C[Project execution] A --- A1[Is it worth doing?] B --- B1[How do we do it?] C --- C1[Do it!] </pre>			
6	<p>Differentiate between jobs and projects. How are software projects distinct from other types of projects? Justify your answer.</p> <p>‘Jobs’ – repetition of very well-defined and well understood tasks with very little uncertainty</p> <p>‘Exploration’ – e.g. finding a cure for cancer: the outcome is very uncertain</p> <p>‘Projects’ – in the middle!</p> <ul style="list-style-type: none"> •<u>Invisibility</u>: With software, progress is not immediately visible •<u>Complexity</u>: Per dollar, software products contain more complexity than other engineered artefacts •<u>Conformity</u>: Software developers have to conform to the requirements of human clients •<u>Flexibility</u>: Software systems are particularly subject to change 	10	CO4 L3	