RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

Academic year 2024-2025 (Even Sem)

DEPARTMENTS OF ISE & CSE

Date: 29/04/2025	**CIE - 1**	**Max. Marks:60**
Semester: VI	**UG**	**Duration: 2 Hour**
Course Title: Software Engineering with Agile Technologies		**Course Code: IS364TA**

Sl. No.	Questions	M	BT	CO
	PART A(QUIZ)			
1.1	List the key attributes of professional software.	1	1	1
1.2	Name any two common Software Process Models.	1	1	2
1.3	Give one project characteristic that would make a plan-driven development process (a waterfall-like process) more appropriate than an agile development process.	1	2	3
1.4	Categorize the following relationship into generalization, aggregation or association. a) Person works for a company b) Car contains engine and a battery	1	3	3
1.5	Categorize the following as functional or non-functional requirements (a) "The system shall allow users to log in using a username and password." (b) "The system shall generate monthly reports." (c) "The system shall respond to user input within 2 seconds." (d) "Users shall be authenticated using multi-factor authentication."	2	3	1
1.6	Describe the four major areas of ethical responsibility for software engineers (as per ACM/IEEE Code)?	2	2	2
1.7	Create a class hierarchy to organize the following vehicle classes: Vehicle, Car, Truck, Motorcycle, ElectricSUV, SportsCar, Van, Bike.	2	4	3

PART B(TEST)				
1a.	Summarize the essential activities in a software process.	6	2	1
1b.	Differentiate between Plan-Driven and Agile processes with suitable examples.	4	1	1
2a.	Explain the main features of Functional and Non-functional requirements with respect to software process. What challenges are faced if functional and non-functional requirements are not stated properly, justify with a suitable example.	6	3	3
2b.	Describe the different ways of writing a system requirements specification.	4	2	3
3a.	Identify the actors, classes and attributes for the following scenario to register a patient in a hospital management system and represent the same using UML class notation. The administrator enters the patient's name, address, date of birth and emergency contact details into the system. If the patient has only public health insurance, the administrator enters the patient's medicare number, and the system verifies this with government health database. If the patient also has private health insurance, then the administrator enters also the patient's private health insurance details, and the system verifies these details with the private health insurance system. Once these details are verified as correct, the system saves the patient's details and confirms the registration.	6	4	4
3b.	Identify the actors and use-cases for the given scenario. Represent the relationship between the actors and the use-cases in a Usecase diagram with standard symbols and notations. Exams are conducted for UG students for computer science courses. Instructor prepares exam question paper and students' take-up the exams. The instructor invigilates the exam and evaluates the papers.	4	4	4
4a.	Describe the different phases of process requirements elicitation and analysis, with a block diagram.	8	2	3
4b.	Mention the specific UML diagram used to represent the dynamic behavior of the system and demonstrate how it responds to the events.	2	2	2
5.	Explain with a block diagram the Incremental development method of software process model. Also, discuss the benefits and problems associated with Incremental development model.	6+4	3	2

Date: 29/04/2025	**CIE – 1 SCHEME AND SOLUTION**	**Max. Marks:60**
Semester: VI	**UG**	**Duration: 2 Hour**
Course Title: Software Engineering with Agile Technologies		**Course Code: IS364TA**

Sl. No.	Questions	M
1.1	Key attributes of professional software: a. Maintainability: Easy to modify or extend. b. Dependability: Reliable, secure, and safe. c. Efficiency: Uses system resources optimally. Acceptability: Meets customer requirements and complies with standards.	0.5x2=1
1.2	Name any two common Software Process Models. Waterfall Model / Incremental Development / Integration and Configuration / Spiral Model / Agile (e.g., Scrum, XP)	0.5x2=1
1.3	One project characteristic that would make a **plan-driven development process** more appropriate than an **agile development process** is: **Stable and well-understood requirements.** In plan-driven (waterfall-like) processes, all requirements are typically gathered and finalized upfront. This approach works best when requirements are unlikely to change throughout the project, such as in projects with strict regulatory or contractual obligations (e.g. payroll system, aerospace, medical devices, or infrastructure systems).	1x1=1
1.4	Person works for Company - Association Car contains engine, battery - Aggregation	0.5x2=1
1.5	Categorize the following as functional or non-functional requirements a) "The system shall allow users to log in using a username and password." - Functional Requirement b) "The system shall generate monthly reports." - Functional Requirement c) "The system shall respond to user input within 2 seconds." - Non-functional Requirement "Users shall be authenticated using multi-factor authentication."-Non-functional Requirement	0.5x4=2

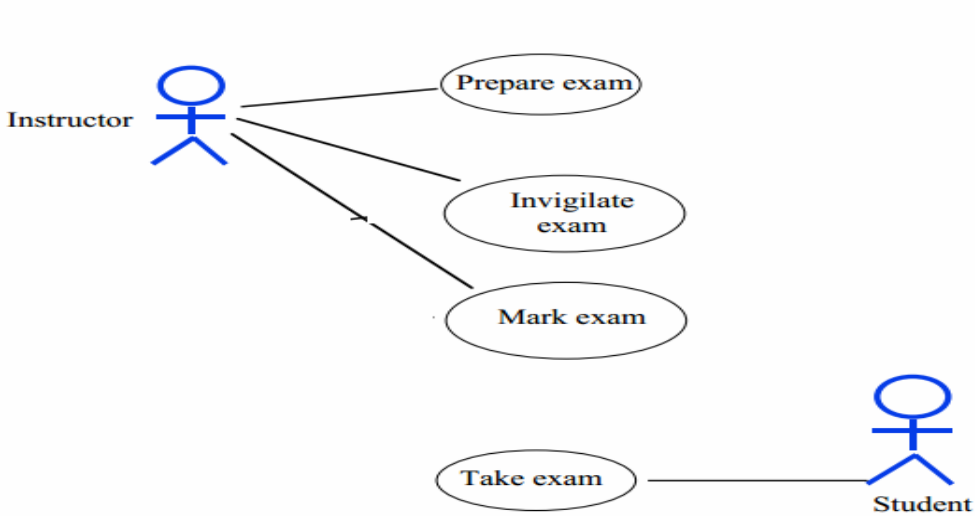
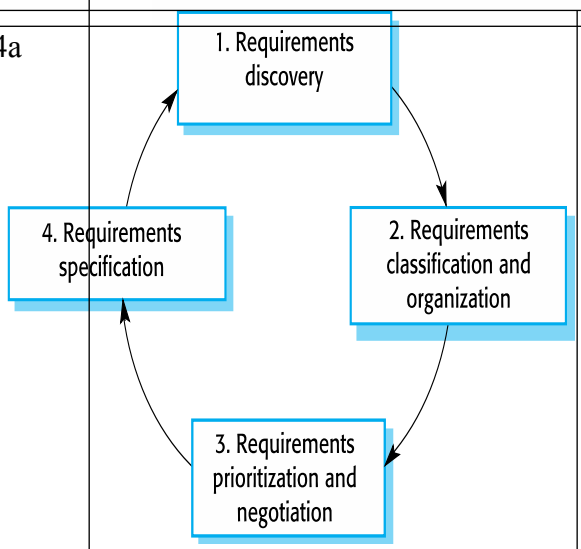
1.6	**ACM/IEEE Software Engineering Code of Ethics**: 1. **Public** – Software engineers shall act consistently with the public interest. 2. **Client and Employer** – Software engineers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest. 3. **Product** – Software engineers shall ensure that their products and related modifications meet the highest professional standards possible. **Judgment** – Software engineers shall maintain integrity and independence in their professional judgment.	0.5x4= 2
1.7	A class hierarchical tree representing generalization and specification.	2

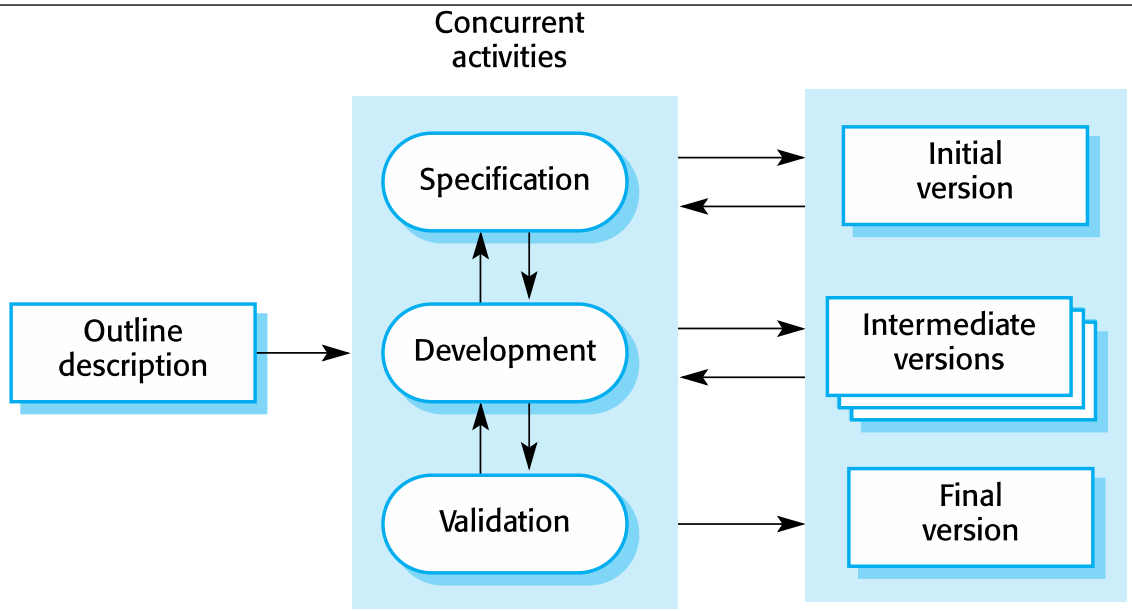
Sl.No.	PART B - SCHEME & SOLUTIONS	M
1a	The typical **steps (or activities)** in a **software process** are: 1. **Specification** – Define the system requirements. 2. **Design** – Plan the architecture and components of the system. 3. **Implementation (or Coding)** – Write the actual code for the system. 4. **Validation (or Testing)** – Ensure the system meets requirements and works correctly. 5. **Deployment** – Release the system to users. 6. **Maintenance (or Evolution)** – Fix bugs, improve performance, or adapt the system to new requirements.	6x1=6
1b	Differentiate between Plan-Driven and Agile processes with suitable examples. ✧ Plan-driven processes are processes where all of the process activities are planned in advance and progress is measured against this plan. ✧ In agile processes, planning is incremental and it is easier to change the process to reflect changing customer requirements. ✧ In practice, most practical processes include elements of both plan-driven and agile approaches. ✧ There are no right or wrong software processes.	2+2=4
2a	**Functional Requirements** These define **what** the system should do. They describe the **specific behaviors, functions, or features** of the system. **Examples:** • The system shall allow users to log in using a username and password. • The software shall generate monthly sales reports. • The application shall send an email confirmation after a purchase. **Non-Functional Requirements** These define **how** the system performs its functions. They are often related to **quality attributes** like performance, security, usability, etc. **Examples:** • The system shall respond to user actions within 2 seconds. • The software shall be available 99.9% of the • time. • The application shall encrypt user data using AES-256. **Functional = Features and behavior**, **Non-functional = Quality and constraints** **Poorly Stated Functional Requirement:** "The system should allow users to manage their accounts." **Issue:** Too vague — what does "manage" mean? It's unclear what specific actions are included. **Better Version:** "The system shall allow users to update their profile information, change passwords, and delete their accounts."	2+2+1+1=6

<table>
<tr><td></td><td>Poorly Stated Non-Functional Requirement:
"The system should be fast."
Issue:
"Fast" is subjective — there’s no measurable benchmark.
Better Version:
"The system shall load the dashboard page within 2 seconds under normal network conditions."</td><td></td></tr>
<tr><td>2b</td><td><table>
<tr><th>Notation</th><th>Description</th></tr>
<tr><td>Natural language</td><td>The requirements are written using numbered sentences in natural language. Each sentence should express one requirement.</td></tr>
<tr><td>Structured natural language</td><td>The requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement.</td></tr>
<tr><td>Design description languages</td><td>This approach uses a language like a programming language, but with more abstract features to specify the requirements by defining an operational model of the system. This approach is now rarely used although it can be useful for interface specifications.</td></tr>
<tr><td>Graphical notations</td><td>Graphical models, supplemented by text annotations, are used to define the functional requirements for the system; UML use case and sequence diagrams are commonly used.</td></tr>
<tr><td>Mathematical specifications</td><td>These notations are based on mathematical concepts such as finite-state machines or sets. Although these unambiguous specifications can reduce the ambiguity in a requirements document, most customers don’t understand a formal specification. They cannot check that it represents what they want and are reluctant to accept it as a system contract</td></tr>
</table></td><td>4x1=4</td></tr>
<tr><td>3a</td><td>Actors: Administrator, Government Health Database, Private Health Insurance System

Classes: Patient, Administrator, Address, EmergencyContact, PublicHealthInsurance, PrivateHealthInsurance, Registration

Attributes: Patient Name, Patient ID….</td><td>2+2+ 2=6</td></tr>
</table>

3b	Actors: Instructor, Student Usecases: Prepare exam, Invigilate exam, Mark exam, Take exam 	1+1+2
4a	 ✧ Requirements discovery ▪ Interacting with stakeholders to discover their requirements. Domain requirements are also discovered at this stage. ✧ Requirements classification and organisation ▪ Groups related requirements and organises them into coherent clusters. ✧ Prioritisation and negotiation ▪ Prioritising requirements and resolving requirements conflicts. ✧ Requirements specification ▪ Requirements are documented and input into the next round of the spiral.	4+4=8
4b	State Chart Diagram / State Diagram	2

5	 Specification, development and validation are interleaved. May be plan-driven or agile. **Incremental development benefits** ✧ The cost of accommodating changing customer requirements is reduced. ▪ The amount of analysis and documentation that has to be redone is much less than is required with the waterfall model. ✧ It is easier to get customer feedback on the development work that has been done. ▪ Customers can comment on demonstrations of the software and see how much has been implemented. ✧ More rapid delivery and deployment of useful software to the customer is possible. ▪ Customers are able to use and gain value from the software earlier than is possible with a waterfall process. **Incremental development problems** ✧ The process is not visible. ▪ Managers need regular deliverables to measure progress. If systems are developed quickly, it is not cost-effective to produce documents that reflect every version of the system. ✧ System structure tends to degrade as new increments are added. ▪ Unless time and money is spent on refactoring to improve the software, regular change tends to corrupt its structure. Incorporating further software changes becomes increasingly difficult and costly.	6+2+2

RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

Academic year 2024-2025 (Even Sem)

SCRUTINY & EVALUATION OF CIE QUESTION PAPER

Date: 29/04/2025	**CIE - 1**	**Max. Marks:60**
Semester: VI	**UG**	**Duration: 2 Hour**
Course Title: Software Engineering with Agile Technologies		**Course Code: IS364TA**

Declaration by the Course handling faculties:

As a teaching faculty of the course _ **Software Engineering with Agile Technologies** _ with code: _**: IS364TA** _, We hereby confirm that the question paper with Scheme and Solutions is thoroughly reviewed and we ensure that it adheres to the following criteria:

Sl. No	Criteria	Yes/No
1	The question paper adequately covers the prescribed syllabus contents.	**Yes**
2	The question paper is in line with the recommended pattern, taking into consideration the structure and format suitable for the Continuous Internal Evaluation.	**Yes**
3	The question paper is designed to align with the Revised Bloom's Taxonomy, encompassing various levels of cognitive skills such as remembering, understanding, applying, analysing, evaluating, and creating.	**Yes**
4	The question paper is aligned with the defined course outcomes, ensuring that it effectively assesses the knowledge and skills acquired during the course.	**Yes**
5	Course handling faculty (as applicable) who are responsible for preparing the question paper, Scheme and Solutions have unanimously agreed to utilize this Question paper for conducting the Continuous Internal Evaluation.	**Yes**
6	The Question paper, Scheme and complete Solutions have been submitted to the Test coordinators within the designated time-frame to ensure the smooth conduction of Continuous Internal Evaluations	**Yes**

Course handling Faculties:

Name: **Signature**

1.

2.

3.

4.

5.

6.

Academic year 2024-2025 (Even Sem)

To be filled by the Scrutinizer:

Sl. No	Rubrics	Points	
		Max	Awarded
1	Timely submission of the question paper along with the scheme & solution	10	
2	Heterogeneous nature of QP with respect to BTs and Cos	10	
3	Format with proper entry of all particulars including test, course name, code, date, max marks, BT CO table, efficient use of paper (proper spacing, figures)	10	
4	No handwritten data or diagrams, and uniform fonts throughout	10	
5	Scheme & complete Solutions in the format	10	
	Total points	**50**	
Any other comments by the scrutinizer :			

Note: Course coordinators to obtain scrutinizer's acceptance by incorporating all suggestions from scrutiny into the final versions of QP, Scheme, and Solutions.

All corrections suggested by the scrutinizer are incorporated and both the copies are re-submitted	**Signature of Course coordinator**
Accepted/Rejected	**Signature of Scrutinizer (Name:)**

Signature of HOD