



Government of Karnataka
DEPARTMENT OF COLLEGIATE AND TECHNICAL EDUCATION

Programme	Computer Science and Engineering	Semester	IV
Course Code	20CS44P	Type of Course	Programme Core
Course Name	Software Engineering principles and practices	Contact Hours	8 hours/week 104 hours/semester
Teaching Scheme	L:T:P :: 3:1:4	Credits	6
CIE Marks	60	SEE Marks	40

1. Rationale

Digital reality has become an integral part of human life with software tools being used to deal with virtually every part of life. A process is key to develop a quality software successfully. Principles and practices of software engineering blends engineering, computing, project management and software development. It's essential to understand the life cycle of software development and the process followed to develop a quality software. Design thinking methodology encourages identifying alternative strategies and solutions to solve a problem in best possible way.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Explain the typical software development life cycle (SDLC), list and differentiate the various SDLC models along with identifying where each model could be beneficial when applied.
CO-02	Demonstrate the application of design thinking as a process, explain how it helps in requirement engineering and mitigate risks.
CO-03	Study a given application requirement, create user stories, draw the appropriate UML diagram and validate to ensure user story/UML diagram meet with the given requirement.
CO-04	Document standard test procedures and test cases for a given requirement to ensure the software gives the desired results for which it is built.

3. Course Content

Week	CO	PO	Lecture (Knowledge Criteria)	Tutorial (Activity Criteria)	Practice (Performance Criteria)
			3 hours/week	1 hour/week	4 hours/week (2 hours/batch twice in a week)
1	1	1	Overview Software engineering; Need of software engineering; Software paradigms; Software product types: generic, customized; characteristics of good software; Challenges in software projects; Factors that influence software development; understanding success Software process; need of process, components of process, process	Refer Table 1	1. Discuss success and failure stories 2. Presentation of collected case studies 3. Enact the importance of ethical practices

			activities; Differentiate product, project and process; process assessment and improvement; Software engineering ethics.		
2	1	1, 5	SDLC and Process Models SDLC; Software process model; How to choose process model? Comparison between a defined process and an empirical process; Traditional process models: waterfall; Incremental; Agile process- manifesto; principles; practices; A paradigm shift from plan driven mentality.		1. Case study to understand the SDLC 2. Organize and play games to understand the agile process like, morning wake up game <ul style="list-style-type: none"> ▪ the marshmallow challenges ▪ White Elephant Sizing ▪ Easter Egg Challenge 3. Create JIRA (similar tool) account and learn interface <ul style="list-style-type: none"> •
3	1	1, 5	Agile frameworks; Ceremonies; Roles; Overview of XP – XP practices Scrum: Overview; framework; ceremonies and artifacts		1. Play and act agile ceremonies 2. Play different agile roles Eg. Product owner, business analyst
4	1,2	1	Risk Risk, characteristics, categories; why risk management is critical; risk management framework; Activities; Principles of risk management, Risk identification, Risk assessment – risk analysis; risk prioritization; Risk Mitigation; need and importance of risk mitigation; Risk Control – planning, resolution, monitoring; How to use tool to manage and mitigate risks in an organization.		1. case study to understand the importance of risk management and mitigation of risk 2. How to use tools to manage and mitigate risks [eg. Logicgate, AuditBoard etc]
5	2	1, 2	Design Thinking Introduction, 5 stages of design thinking Understand the process of design thinking using an example Case Study		1. Conduct warmup activities to Ignite Design Thinking 2. Organize and conduct design thinking exercises and games

6	1,3	1, 2	Requirement Engineering & Modeling Overview; what is requirement? Importance; Requirement types; Sources of requirements; Requirement engineering Process; Feasibility study; Typical Requirements Engineering Problems; Requirement modeling strategies; Overview of UML; types of diagrams; Note: Take a case study to understand requirement engineering and prepare use cases or user stories	1. Organize role play for requirement activities 2. Identify a problem and prepare requirement document or Epics and user stories 3. Configure JIRA for the managing the project to solve the identified problem 4. Draw UML diagram for given use case
7	1,3	1, 2, 4	User stories What are user stories? Why user story? Basic concepts; Characteristics; How to write/create user stories? Steps; 3C's in user stories; Life cycle of user story. User story map. Estimation: User story point: basics; components of story point estimation; Steps involved in estimation;	1. Create detailed user stories for the above identified problem 1. Organize and play planning poker to decide on user points.
8	1,2	1, 2, 3	Design Objectives; design Concepts; Levels of design; Architectural styles; Monolithic and Microservices; UI and UX: Overview of UI and UX, UI types, essential properties, elements of UI design; relationship between UI and UX; Importance of good UI/UX. Wireframes: overview, purpose, benefits;	1. Create sitemap and wireframe for above created user stories. (Tools such as sketch, Adobe XD, Figma, etc can be used)
9	1	1, 2, 3, 4	Development Overview of DevOps; working principle; Benefits; DevOps culture; DevOps practices: continuous integration, continuous delivery, version control, configuration management, Build process;	1. Create Git (similar tool) account and configure repository 2. Upload the artifacts created to Git Learn version control and configuration management with Git
10	1	1, 2, 3, 4	Code quality and code security: overview; importance; issues caused by poor code; tools to check code quality Containerization: Container, why container, containerization; working principle; benefits; Hello world example Note: Docker or similar tool can be used to explain the containers.	1. Install and configure Jenkins 2. Create a container image for Hello world project 3. Setup build for container image using Jenkins (Hello world application)

11	1,4	1,4	Testing Principles of testing; Need of testing; stages; Testing process and activities; classification; Testing strategies; Levels of software testing; Software testing types; (Integration testing, functional testing, end-to-end testing need to be explained in detail)		1. Prepare Test plan for the user stories using JIRA 2. Prepare RTM for the user stories created using JIRA. Create test cases for the user stories created.
12	1,4	1	Software Measurement and Metrics Measurement; need of Measurement; types; Metrics: characteristics; classification; Agile metrics; Application monitoring.		1. Use JIRA or similar tool to capture agile metrics 2. Use SonarQube to capture code quality metrics
13	1,4	1	Quality Control and Assurance Concept of software quality, Compliance, Quality Standards, quality control, quality assurance; Difference between QC & QA. Need for auditing. Auditing fundamentals: auditing, elements of auditing; audit types; auditing methods, benefits of auditing. Quality and Process improvement tools and techniques- pareto chart, PDCA cycle, Six sigma and Lean process		1. Organize Roleplay to understand the roles and responsibilities of QA and QC team. 2. Audit the artifacts produced in previous sessions
Total in hours			39	13	52

***PO = Program outcome as listed and defined in year 1 curriculum**

Table 1: Suggestive activities for tutorials (the list is only shared as an example and not inclusive of all possible activities for that course. Student and faculty are encouraged to choose activities that are relevant to the topic and the availability of such resources at their institution)

Sl No	Activity
1	Study the traffic signal and the importance of rules and process.
2	Visit various consulting company web portals and collect case studies.
3	Document the roles and responsibilities of different agile ceremonies
4	Identify cost of risk; Identify commonly used risk management tools.
5	Identify a problem and explain how design thinking can be applied to solve it. Design a shopping cart to achieve ease of use, applying design thinking.
6	Prepare RPM requirement traceability matrix for shopping cart List the criteria to select the requirement management tools. Identify different requirement management tools and list their features. Identify frequently used UML diagrams and also identify tools used to draw them.
7	Explore agile estimation techniques and prepare a report.
8	Study boiler plate and present necessary characteristics of boiler plate for a large and small project
9	Identify different DevOps Tools and list their features Study and report OWASP coding guidelines Learn and report Twelve Factor App methodology Identify different version control and configuration management tools and report their market share

10	Compare and contrast containerization and virtualization and identify importance of these in software development Identify container providers
11	Study and prepare report on testing tools. Compare manual and automation testing
12	Study and prepare report on widely used software metrics.
13	Identify different quality tools and report their features and usage

4. CIE and SEE Assessment Methodologies

Sl. No	Assessment	Test Week	Duration In minutes	Max marks	Conversion
1.	CIE-1 Written Test	5	80	30	Average of three tests 30
2.	CIE-2 Written Test	9	80	30	
3	CIE-3 Written Test	13	80	30	
4.	CIE-4 Skill Test-Practice	6	180	100	Average of two skill tests reduced to 20
5	CIE-5 Skill Test-Practice	12	180	100	
6	CIE-6 Portfolio continuous evaluation of Activity through Rubrics	1-13		10	10
Total CIE Marks					60
Semester End Examination (Practice)			180	100	40
Total Marks					100

5. Format for CIE written Test

Course Name	Software Engineering principles and practices	Test	I/II/III	Sem	III/IV
Course Code	20CS44P	Duration	80 Min	Marks	30
Note: Answer any one full question from each section. Each full question carries 10 marks.					
Section	Assessment Questions	Cognitive Levels	Course Outcome	Marks	
I	1				
	2				
II	3				
	4				
III	5				
	6				
Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, Cognitive level and course outcomes.					

6. Rubrics for Assessment of Activity (Qualitative Assessment)

Sl. No.	Dimension	Beginner	Intermediate	Good	Advanced	Expert	Students Score
		2	4	6	8	10	
1		Descriptor	Descriptor	Descriptor	Descriptor	Descriptor	8
2		Descriptor	Descriptor	Descriptor	Descriptor	Descriptor	6
3		Descriptor	Descriptor	Descriptor	Descriptor	Descriptor	2
4		Descriptor	Descriptor	Descriptor	Descriptor	Descriptor	2

	Average Marks= (8+6+2+2)/4=4.5	5
--	--------------------------------	---

Note: Dimension and Descriptor shall be defined by the respective course coordinator as per the activities

7. Reference:

Sl. No.	Description
1	Agile Software Development, principles, patterns and practices by Robert Martin
2	Art of agile development by James Shore & Shane Warden
3	Extreme programming explained: embrace change
4	Software-Engineering-9th-Edition-by-Ian-Sommerville
5	RPL-7th_ed_software_engineering_a_practitioners_approach_by_roger_s_pressman_
6	Becoming Agile..in an imperfect world by Greg Smith, Ahmed Sidky
7	scaledagileframework.com
8	Continuous Delivery Principles Atlassian
9	www.agilealliance.org/
10	www.udemy.com
11	www.tutorialride.com
12	www.interaction-design.org/
13	www.digite.com

8. SEE Scheme of Evaluation

SL. No.	Particulars/Dimension	Marks
1	Capturing the requirements of the client, documenting, reviewing and acceptance by the client of the documented requirement for given the problem statement.	30
2	Identify and document at least two associated risks for the above case.	20
3	Create user stories for the above requirements and draw a UML diagram using any of the modeling technique.	30
4	Document test suite for the above requirement. Viva-Voce, In the event of student unable to document a test suite, then student should be allowed to explain the test procedure.	20
Total Marks		100

9. Equipment/software list with Specification for a batch of 20 students

Sl. No.	Particulars	Specification	Quantity
1.	Computers		20
2.	Git, Jira, SonarCube, Lucidchart or any other UML design tool		