|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| D:\UAAR\UIIT\courseOutlineCommittee\CourseContents_Final_V02\New folder\logo4.png | **PMAS Arid Agriculture University Rawalpindi**  **University Institute of Information Technology** | | | |
| CS-453 Software Engineering | | | | | |
| **Credit Hours:** | | **3(3-0)** | **Prerequisites:** | **None** | |
| **Teacher:** | |  |  |  | |
| **Course Description:** | | | | | |
| Introduction to Software Process Models; Programming in the Large vs. Individual Programming; Evaluation of Software Process Models; Requirements Analysis and Design Modeling Tools; Testing Tools; Programming Environments that Automate Parts of Program Construction Processes; Tool Integration Concepts and Mechanisms; Functional Requirements; Properties of Requirements; Software Requirements Elicitation; Describing System Data; Non-Functional Requirements; Requirements Specifications; System Design Principles; Design Paradigms; Structural and Behavioral Models of Software Designs; Design Patterns; Relationships between Requirements and Designs; Software Architecture; Refactoring Designs using Design Patterns; The Use of Components in Design; Coding Practices; Coding Standards; Integration Strategies; Verification and Validation; Inspections; Reviews; Audits; Testing Types; Testing Fundamentals; Debugging; Defect Tracking; Limitations of Testing. | | | | | |
| **Course Objective:** | | | | | |
| 1. To understand basic SW engineering methods and practices, and their appropriate application. 2. To understand software engineering layered technology and Process frame work. 3. To understand the role of project management including planning, scheduling, risk management. 4. To understand data models, object models, context models and behavioral models. 5. To understand different software architectural styles. 6. To understand software testing approaches such as unit testing and integration testing. 7. To understand debugging and defect tracking. | | | | | |
| **Teaching Methodology:** | | | | | |
| Lectures, Assignments, labs, Projects, Presentations, etc. Major component of the course should be covered using conventional lectures. | | | | | |
| **Courses Assessment:** | | | | | |
| Exams, Assignments, Quizzes, Project, Presentations. Course will be assessed using a combination of written examinations and project(s). | | | | | |
| **Reference Materials:** | | | | | |
| * Software Engineering: A Practitioner’s Approach, Pressman, R.S. & Maxim B., 8th Edition (2015), McGraw-Hill. * Software Engineering, Sommerville, I., 10th Edition (2016), Pearson. | | | | | |

|  |  |  |
| --- | --- | --- |
| **Week/Lecture #** | | **Theory** |
| Week 1 | Lect-I | Introduction to software engineering and software development life cycle (SDLC), Why we need a SDLC? How different process models have evolved from general SDLC. Waterfall & incremental model is discussed in detail along with the similarities and the differences between them. In addition explaining in which conditions they can be used. |
| Lect-II | Rational unified process model, RAD, Spiral and Evolutionary are discussed in detail along with the similarities and the differences between them. |
|  | Lect-III | In addition, explaining in which conditions they can be used. |
| Week 2 | Lect-I | Agile process model along with various agile development techniques will be discussed |
| Lect-II | Introduction to requirement engineering. Discussion on different types of requirements system, user and domain requirements. |
|  | Lect-III | What are functional requirements and Non-functional requirements? |
| Week 3 | Lect-I | How to write the functional requirements using use case or user stories concepts? Discussion on non-functional requirements and its different types |
| Lect-II | Requirement engineering process in detail will be discussed along with requirement specification |
|  | Lect-III | Different methods /tools used for requirement gathering will be discussed. |
| Week 4 | Lect-I | Stakeholder identification, their possible issues will be discussed. |
| Lect-II | Concepts of requirement validation and requirement change and management will be discussed. |
|  | Lect-III | Discuss the current SRS document of Final Year Project. |
| Week 5 | Lect-I | Introduction to different representation of requirements will be discussed. ERD are described here. |
| Lect-II | Concepts of system design and modeling will be discussed, relationship between requirements and design, Design modeling tools. |
|  | Lect-III | Structural Models |
| Week 6 | Lect-I | Behavioral Models & diagram with description & example. |
| Lect-II | Context models |
|  | Lect-III | (Data flow diagram with description and diagrams). |
| Week 7 | Lect-I | What is Software Architecture, The Software Architecture, Role of the Software Architecture, Assessment and Evaluation, |
| Lect-II | Dynamic Software Architectures, Architecture Design Process. |
|  | Lect-III | Assessment of the Quality Attributes |
| Week 8 | Lect-I | Architectural Styles, Semantics of Components. |
| Lect-II | Model-View-Controller. |
|  | Lect-III | Software architectural designs and patterns |
| **Mid Term Exam** | | |
| Week 10 | Lect-I | Different software architectures |
| Lect-II | Software Project Management, Product and the Project Life Cycles, |
|  | Lect-III | Project Management Life Cycle |
| Week 11 | Lect-I | Knowledge Areas for Project Management, Software Project Planning, The Project Plan, |
| Lect-II | Scope Definition, Project Analysis |
|  | Lect-III | Work Break Down Structure |
| Week 12 | Lect-I | Activity Planning, Six Methods for Estimating Activity Duration. |
| Lect-II | Concepts of verification and validation along with various techniques required for software testing. |
|  | Lect-III | How to document the V&V process? |
| Week 13 | Lect-I | Static testing and dynamic testing will introduce. Inspection review, and audit processes will be discussed in detail |
| Lect-II | Dynamic testing will introduce. Testing types and where are they required. |
|  | Lect-III | Development testing (unit and integration testing ) |
| Week 14 | Lect-I | Release testing , user testing , manual or automatic testing , making of test cases, Testing tools |
| Lect-II | Software Evolution and its process. |
|  | Lect-III | Different laws for software evaluation will be discussed |
| Week 15 | Lect-I | Characteristics of maintainable software |
| Lect-II | Software reuse, Software reliability. |
|  | Lect-III | System reliability and failure behavior. |
| Week 16 | Lect-I | Project Presentation |
| Lect-II | Project Presentation |
|  | Lect-III | Project Presentation |
| **Final Term Exam** | | |

|  |
| --- |
| **Course Learning Outcomes (CLOs)** |
| At the end of course the students will be able to: |
| * Explain the concept of a software life cycle along with its phases including the deliverables that are produced. |
| * Discover both functional and non-functional requirements for a medium sized software system. |
| * Produce appropriate design models for the structure and behavior of a medium sized software system. |
| * Describe various code integration strategies. |
| * Prepare a set of tests for a medium-sized code segment. |