## Course Outline

|  |  |
| --- | --- |
| Title | Software Engineering |
| Code | CMP – 3901-3 |
| Credit Hours | 3  ***Theory/week:***  Weight: 3 Cr. Hrs.  Contact Hours: 3 Hrs.  Lectures: 2  Duration: 1.5 Hrs.  ***Lab/week:***  Weight 0 Cr. Hrs.  Contact Hours 0 Hrs.  Labs. 0  Duration 0 Hrs. |
| Prerequisite | CMP – 2402 – 4 Object-Oriented Programming |
| **Prerequisite Skills** | * Understanding of basic programming concepts. |
| Follow Up | CMP – 3902-3: Object Oriented Analysis & Design |
| **Category** | Major |
| Aims and Objectives | * Application of software engineering elements to the development of software in any computing application domain where professionalism, quality, schedule, and cost are important in producing a software system. * To convey the importance and need of software engineering * To discuss different software development models appropriate for the development and maintenance of software products * To introduce the basic project management concepts for the development of a high-quality product * To impart comprehensive knowledge regarding software development lifecycle * To demonstrate, with justification, an appropriate set of tools to support the development of a range of software projects |
| Learning Outcomes | * The students will recognize the importance and need of software engineering to cope with the modern trends in software industry * The students will have the knowledge of all phases of software life cycle including the artifacts that are produced * The students will be proficient enough to analyze, evaluate and apply a set of CASE tools * The term project will enable the students to implement the software engineering concepts in a disciplined way, to compete the local and international market |
| Syllabus | **The Scope of Software Engineering:** Definition, Motivation and Need, Software Characteristics and Applications. **The Software Process:** Definition, Introduction to Software Development Life Cycle, Software Process Models. **Project Management Concepts:** Essentials, 4Ps (People, Product, Process, Project); **Software Measurement concepts:** Project, Process, Product and Software Quality Metrics, Software Cost Estimation techniques; **Project Scheduling:** GANTT chart, Critical Path Method. **Requirements Engineering:** Definition, Requirements Elicitation, Requirement Traceability, Requirements Analysis and Specification. **Software Designing:** Elementary Concepts, Data Design, Software Architecture, Functional Independence, Interface Design, Component-level Design, Coding Standards. **Software Testing:** Fundamentals, Verification & Validation, Testing Plan, Testing Techniques, White Box Testing and Black Box Testing, Test Case Design, Unit Testing, Integration Testing, Validation Testing, System Testing, Debugging practices. |
| Text Book/s | A - Roger S. Pressman “Software Engineering- A practitioner’s approach”, 7th Ed. |
| Reference Material | 1. Ian Sommerville “Software Engineering”, 6th Ed. 2. An integrated approach to software engineering by Pankaj Jalote 3. Fundamentals of Software Engineering By Carlo Ghezzi 4. <http://www.mhhe.com> 5. <http://www.sei-cmu.edu> |
| Instructional Aids/Resources | * Class facilities to Instructor and students for Lectures per Room   + 1 Network enabled PC   + 1 Multimedia   + 1 Whiteboard   + 1 Rostrum   + 1 Board marker/Month i.e. 4 Week Lectures (eight sessions)   + Max 45 students sitting facility   + CASE tools must be available in the lecture room as well as in labs. * Evaluation and Support facilities   + Announced Tests i.e. pre mid and pre final Testes should not be conducted during class hours * Photocopy facility for different Handouts   + Subject description document that includes student version of course outline, prerequisite test and its solution should be distributed in first class.   + Handout: - Required handouts as mentioned in lecture framework   + Quizzes and Solution: - Quizzes and their standard solution must be provided to students in handout form after each quiz.   + Case Studies and Tutorials: - Description of case study and handout of each tutorial should be provided. |
| Assessment Criteria | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Sessional 15%** | **Project 10%** | **Mid 35%** | **Final 40%** | **Total 100%** | | Tests/Assignment and Presentations | Term Project 10 | Mid Paper 35 | Final Paper 40 | 100 | |
| Recommendations | Marks division for sessional or project may vary on the basis of complexity of project or available time for project execution and documentation. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Week | Lecture | Topic | Source (Book-Chapter No. Section No.) | **Recommendations for Learning Activities** |
| 1 | 1 | **Introduction:**  The Scope of Software Engineering: Motivation and need for software engineering, Definition of Software Engineering, Introduction to software engineering vocabulary | A-2.1 |  |
| 2 | **The Software:**  Software characteristics, software application, Software Myths | A-1.2, 1.3, 1.5 |  |
| 2 | 3 | **A Generic View of Process:**  Software Engineering: A layered technology, A process framework, CMMI | A-2.1, 2.2,2.3 |  |
| 4 | **Process Models:**  Introduction, The Waterfall Model, Incremental Process Models-The Incremental Model | A-3.1, 3.2, 3.3-3.3.1 | Project Announced |
| 3 | 5 | **Process Models:**  The RAD Model, Evolutionary process models, Prototyping | A-3.3.2, 3.4, 3.4.1 | Case Study  Assignment 1 (Process Models) |
| 6 | **Process Models:**  The Spiral Model, A Final Comment on Evolutionary Processes, The Formal Methods Model, Introduction to Agile Process Models | A-3.4.2, 3.4.4, 3.5.2, 4.2 | Test 1 (The software process) |
| 4 | 7 | **Project Management:**  Project Management Concepts, Project Management Lifecycle | Handouts | Project Proposal Announced |
| 8 | **Project Management:**  The Management Spectrum: The people, The product, The process, The project | A-21.1, 21.2, 21.3, 21.4, 21.5 |  |
| 5 | 9 | **Estimation:**  Software Metrics: Metric, Measure, and Indicator, Metric for software quality, Decomposition Techniques, Software Measurements, Size oriented metrics, Function oriented metric | A-15.2.1, 22.3, 23.6, 22.2, 22.2.1, 22.2.2 | Project Phase 1: Project Planning |
| 10 | **Metrics for Process and Project:**  Function oriented Metrics, An example of FP based Estimation | A-22.2.2, 15.3.1, 23.6.4 |  |
| 6 | 11 | **COCOMO Model:**  The COCOMO Model | Ref. 2 |  |
| 12 | **Project Scheduling:**  GANTT chart, Critical Path Method | Handouts | Assignment 2 (Scheduling and Costing) |
| 7 | 13 | **LAB Session**: Microsoft Project | Handouts | LAB session  Case Study |
| 14 | **Requirements Engineering:**  Requirement gathering and fact finding techniques | A-7.1, 7.2, 7.3, 7.4  Kendall & Kendall,  Ref. 1-6.2.1 | Project Phase 2: Software Requirement Engineering  Test 2 (Scheduling, project management, and project metrics) |
| 8 | 15 | **Requirement Elicitation:**  FAST, QFD, Use – cases  Rational Rose 2002 overview | A-7.4, 7.5 |  |
| 16 | Pre Mid Review |  |  |
|  |  |  |  |  |
| 9 | 17 | **Building the Analysis Model:**  Requirements Analysis, Data modeling concepts | A-8.1, 8.3, Handouts (A-5th ed.) |  |
| 18 | **Flow-oriented Modeling:**  Creating a Data Flow Model | A-8.6.1, Handouts (A-5th ed.) | Case Study |
| 10 | 19 | **Behavioral Modeling:**  The Control Specification, The Process Specification | A-8.8, 8.6.3, 8.6.4, B-12, Handouts (A-5th ed.) |  |
| 20 | **LAB Session:** VISIO and ERWIN | Handouts | Case Study |
| 11 | 21 | **Software Design:** Design within the Context of Software Engineering, Design Process and Design Quality, Design concepts | A-9.1, 9.2, 9.3, Handouts (A-5th ed.) | Test 3 (Analysis Modeling)  Project Phase 3: Software Design |
| 22 | **Software Design:** Effective Modular Design | A-9.4, 11.2.3,11.23.4, Handouts (A-5th ed.) |  |
| 12 | 23 | **Software Design:**  Software Architecture, Architectural Styles, Mapping Requirements into software architecture | A-10.1, 10.3,10.6, Handouts (A-5th ed.) |  |
| 24 | **Software Design:**  Transform Mapping, Transactional Mapping | A-10.6, Handouts (A-5th ed.) | Case Study |
| 13 | 25 | **Lab Session:** VB- User Interface Design | A-12.1, 12.2, 12.4,  Handouts | Lab Session |
| 26 | **Component-level Design:**  Component-level design  **Testing Tactics:**  Software Testing Fundamentals, Black Box Testing | A-11.5  A-14.1, 14.2, 14.6 | Project Phase 4: Software Implementation  Project Phase 5: Software Testing |
| 14 | 27 | **Testing Tactics:**  White Box Testing, White Box testing with example | A-14.3, 14.4 |  |
| 28 | **Testing Strategies:**  A strategic approach to software testing, Strategic Issues, Testing Strategies for Conventional Software: Unit Testing, Integration testing | A-13.1, 13.2, 13.3 | Test 4 (Testing) |
| 15 | 29 | **Testing Strategies:**  Integration Testing, Validation Testing, System Testing, The art of debugging | A-13.3, 13.5, 13.6, 13.7 |  |
| 30 | Presentations on Final Project and viva |  | Group Presentations |
| 16 | 31 | Presentations on Final Project and viva |  | Group Presentations |
| 32 | Post Mid Review |  | Interactive session |