|  |  |
| --- | --- |
| C:\Users\TEMP.WDC.013\Downloads\VIT logo.png  **Version 0120-7** | **Consolidated Academic Administration Plan for the Course**  ***CSC502 Software Engineering (Core) Semester V – Computer Engineering 2023-2024 – Odd Semester***  ***Faculty - Prof. Sachin Bojewar (Cluster Mentor) Prof. Sachin Bojewar*** |

**The academic resources available in VIT –**

|  |  |  |  |
| --- | --- | --- | --- |
| **VMIS (ERP)** | **V-Refer and V-Live** | **VIT Library** | **VAC & MOOC Courses** |
| Institute & Department Vision and Mission | Former IA question papers and solutions (prepared by faculty) | Former IA question papers solutions - hardcopy | Value Added Courses (VAC) are conducted throughout the semester & in the semester break - Enrol for the VACs |
| Program Educational Objectives (PEO) | MU end semester examination question papers and solutions (prepared by faculty) | MU end semester exam question paper & solutions - by faculty, hardcopy |
| Program Specific Outcome (PSO) | Class notes and Digital Content for the subject (scanned / typed by faculty) | All textbooks, reference books, e -books mentioned in the syllabus & AAP | Online courses from NPTEL, Coursera etc. are pursued throughout the semester - Register for the course & get certified |
| Program Outcome (PO) | Comprehensive question bank, EQ, GQ, PPT, Class Test papers | Technical journals and magazines for reference |
| Departmental Knowledge Map | Academic Administration Plan & Beyond Syllabus Activity report | VIT library is member of IIT Bombay Library | Watch former lectures captured in LMS at VIT |

|  |  |
| --- | --- |
| **1.a** | **Course Objectives (Write in detail – as per NBA guidelines)** |

|  |  |  |
| --- | --- | --- |
| Cognitive | What do you want students to know? | To gain knowledge of software engineering discipline. |
| Affective | What do you want students to think / care about? | To apply analysis, design, and testing principles to software project development. |
| Behavioural | What do you want students to be able to do? | To demonstrate and evaluate real world software projects. |

**Advice to Students:**

Attend every class!!! Missing even one class can have a substantial effect on your ability to understand the course. Be prepared to think and concentrate, in the class and outside. I will try to make the class very interactive. Participate in the class discussions. Ask questions when you don’t understand something. Keep up with the class readings. Start assignments and homework early. Meet me in office hour to discuss ideas, solutions or to check if what you understand is correct. The v-Refer Link for this course (**Creation of microsite (vit.edu.in or teams) https://tinyurl.com/y8cus3vz**

**Collaboration Policy:**

We encourage discussion between students regarding the course material. However, no discussion of any sort is allowed with anyone on the assignment and homework for the class. If you find solution to some problems in a book or on the internet, you may use their idea for the solution; provided you acknowledge the source (name and page in the book or the website, if the idea is found on the internet). Even though you are allowed to use ideas from another source, you must write the solution in your own words. If you are unsure whether or not certain kinds of collaboration is possible, please ask the teacher.

|  |  |
| --- | --- |
| **1.b** | **Course Outcome (CO) Statements and Module-Wise Mapping (follow NBA guideline)** |

|  |  |  |
| --- | --- | --- |
| CO No. | Statements | Related Module/s |
| CO1 | Students will be able identify requirements and assess the process models | 1,2 |
| CO2 | Students will be able to create schedule and track the project progress | 3 |
| CO3 | Students will be able to design software project | 4 |
| CO4 | Students will be able to test software | 5 |
| CO5 | Students will be able to identify risk, manage change and assure quality in software project. | 6 |

|  |  |
| --- | --- |
| **1.c** | **Mapping of COs with POs (mark S: Strong, M: Moderate, W: Weak, Dash ‘–’: not mapped)**  **(List of POs is available in V-refer)** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 |
| CO 1 | S | M | W | W | W | W | M | W | M | M | M | S |
| CO 2 | S | S | M | S | S | W | S | W | M | S | W | M |
| CO 3 | S | M | M | M | S | W | W | W | M | S | W | M |
| CO 4 | S | S | M | M | S | W | S | W | M | S | S | M |
| CO 5 | S | M | M | M | M | M | M | M | M | S | M | M |

|  |  |
| --- | --- |
| **1.d** | **Mapping of COs with PSOs (mark S: Strong, M: Moderate, W: Weak, Dash ‘–’:not mapped)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | PSO 1 | PSO 2 | PSO 3 | PSO 4 |
| CO 1 | S | W | S | - |
| CO 2 | S | M | M | - |
| CO 3 | S | S | W | - |
| CO 4 | S | S | M | - |
| CO 5 | S | W | W | - |

|  |  |
| --- | --- |
| **1.e** | **Teaching and Examination Scheme (As specified by the University) for the Course** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Categories | Mathematics | Basic Science & General Engg. | Humanities & Soft Skill | Core Engg./ Technology - Design & Analysis | Multidisciplinary |
| Tick suitable category | **--** | **--** | **--** | √ | **--** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Subject Code | Subject Name | **Teaching Scheme** | | | Credits Assigned | | | |
| Theory | Practical | Tutorial | Theory | TW/Practical | Tutorial | Total |
| CSC502  CSL501 | Software Engineering  Software Engineering Lab | 3 | 2 | - | 3 | 1 | - | 4 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Subject Code | Subject Name | **Examination Scheme** | | | | | | | |
| Theory Mark | | | End Sem. Exam Marks | TW | Practical | Oral | Total |
| ISA | MSE | ESE |
| CSC502 | Software Engineering | 20 | 30 | 50 |  |  |  |  | 100 |
| CSL501 | Software Engineering Lab | -- | -- | -- |  | 25 | -- | 25 | 50 |

|  |  |
| --- | --- |
| **1.f** | **Faculty-Wise Distribution of all Lecture-Practical-Tutorial Hours for the Course** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Divisions | Lecture (Hrs.) | Practical (Hrs.) | | | | Tutorial (Hrs.) | | | |
| Batch 1 | Batch 2 | Batch 3 | Batch 4 | Batch 1 | Batch 2 | Batch 3 | Batch 4 |
| **A** | SBJ 03 | SA | SBJ | SA | SA | SA | SA | SA | SA |
| **B** | SBJ 03 | SA | SA | SA | SA | SA | SA | SA | SA |

|  |  |
| --- | --- |
| **1.g** | **Office Hours (Faculty will be available in office in this duration for solving students’ query)** |

|  |  |  |  |
| --- | --- | --- | --- |
| Division | Day | Time (at least 1 Hr. / Division) | Venue (Office Room No.) |
| A | Friday | 3.30 to 4.30 pm | M-209 |
| B | Friday | 3.30 to 4.30 pm | M-209 |

|  |  |
| --- | --- |
| **2.a** | **Syllabus: Module Wise Teaching Hours and % Weightage in University Question Paper** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Module** |  | **Contents** | **Hrs** |
| **1** |  | **Introduction To Software Engineering and Process Models** | **7** |
|  | 1.1 | Software Engineering-process framework, the Capability Maturity Model (CMM), Advanced Trends in Software Engineering |  |
| 1.2 | Prescriptive Process Models: The Waterfall, Incremental Process Models, Evolutionary Process Models: RAD & Spiral |
| 1.3 | Agile process model: Extreme Programming (XP), Scrum, Kanban |
| **2** |  | **Software Requirements Analysis and Modeling** | **4** |
|  | 2.1 | Requirement Engineering, Requirement Modeling, Data flow diagram, Scenario based model |  |
| 2.2 | Software Requirement Specification document format (IEEE) |
| **3** |  | **Software Estimation Metrics** | **7** |
| 3.1 | Software Metrics, Software Project Estimation (LOC, FP, COCOMO II ) |  |
| 3.2 | Project Scheduling & Tracking |
| **4** |  | **Software Design** | **7** |
|  | 4.1 | Design Principles & Concepts |  |
| 4.2 | Effective Modular Design, Cohesion and Coupling, Architectural design |  |
| **5** |  | **Software Testing** | **7** |
|  | 5.1 | Unit testing, Integration testing, Validation testing, System testing |  |
| 5.2 | Testing Techniques, white-box testing: Basis path, Control structure testing black-box testing: Graph based, Equivalence, Boundary Value |
| 5.3 | Types of Software Maintenance, Re-Engineering, Reverse Engineering |
| **6** |  | **Software Configuration Management, Quality Assurance and Maintenance** | **7** |
|  | 6.1 | Risk Analysis & Management: Risk Mitigation, Monitoring and Management Plan (RMMM). |  |
| 6.2 | Quality Concepts and Software Quality assurance Metrics, Formal Technical Reviews, Software Reliability |
| 6.3 | The Software Configuration Management (SCM, version Control and  Change Control |
|  |  |  | **39** |

|  |  |
| --- | --- |
| **2.b** | **Prerequisite Courses** |

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Semester | Name of the Course | Topic/s |
| 1 | III | OOPM | Design |
| 2 | IV | DBMS | Design |

|  |  |
| --- | --- |
| **2.c** | **Relevance to Future Courses** |

|  |  |  |
| --- | --- | --- |
| No. | Semester | Name of the Course |
| 1 | VII&VIII | Final Year Project |

|  |  |
| --- | --- |
| **2.d** | **Identify real life scenarios / examples which use the knowledge of the subject** |

|  |  |
| --- | --- |
| Real Life Scenario | Concept Used |
| Application development | PLC, SDLC |

|  |  |
| --- | --- |
| **3.** | **Past Results – Division-Wise** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Details | Target – Oct 2023 | Oct 2022 | Oct 2020 | Oct 2019 |
| Course Passing % – Average of 2 Divisions | 100 | 100 | -- | -- |
| Marks Obtained by Course Topper (marks 100) | 90+ | 80 | -- | -- |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Division A | | Division B | | Division C | |
| Year | Initials of Teacher | % Result | Initials of Teacher | % Result | Initials of Teacher | % Result |
| Dec 2022 | SBJ | 100% | SA | 100% |  |  |
| Dec 2021 | SBJ | 100% | SA | 100% | -- | -- |
| Dec 2020 | DN | 100% | PTT | 100% | -- | -- |

|  |  |
| --- | --- |
| **4** | **All the Learning Resources – Books and E-Resources** |

|  |  |
| --- | --- |
| **4.a** | **List of Textbooks (T – Symbol for Textbook) to be Referred by Students** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. No | Textbook Titles | Author/s | Publisher | Edition | Module Nos. |
| 1 | Software Engineering: A Practitioner‘s Approach", | Roger Pressman | McGraw-Hill Publications | - | 1,2,3,4,5,6 |
| 2 | Software Engineering‖, | Ian Sommerville, | Pearson Education | (9th edition) | 2,4 |
| 3 | "Software Engineering Fundamentals", | Ali Behfrooz and Fredeick J.Hudson | Oxford University Press | - | 1,2,3,4,5,6 |

|  |  |
| --- | --- |
| **4.b** | **List of Reference Books (R – Symbol for Reference Books) to be Referred by Students** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. No | Reference Book Titles | Author/s | Publisher | Edition | Module Nos. |
| 1 | Software Engineering – Concepts and Practices | Ugrasen Suman | Cengage Learning | - | 1,2,3,4,5,6 |
| 2 | "An integrated approach to Software Engineering" | Pankaj Jalote, | Springer/ Narosa | - | 1,2,3,4,5,6 |
| 3 | Software Engineering | Jibitesh Mishra and Ashok Mohanty, | Pearson | - | 1,2,3,4,5,6 |
| 4 | "Fundamentals of Software Engineering" | Rajib Mall | Prentice Hall India | - | 1,2,3,4,5,6 |

|  |  |
| --- | --- |
| **4.c** | **List of E - Books (E – Symbol for E-Books) to be Referred by Students** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. No | E- Book Titles | Author/s | Publisher | Edition | Module Nos. |
| 1 | Software Engineering: A Practitioner ‘s Approach" | Roger Pressman | McGraw-Hill Publications | - | 1,2,3,4,5,6 |

|  |  |
| --- | --- |
| **4.d** | **Reading latest / top rated research papers (at least 5 papers)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of Paper | Authors with Background | Published in | | Problem Statement |
| Date | Journal |
| E-GOVERNMENT MATURITY MODELS: A COMPARATIVESTUDY | Abdoullah Fath-Allah | 2022 | IJSEA | Many maturity models have been used to assess or rank e-government portals. In order to assess electronic services provided to the citizens, an appropriate e-government maturity model should be selected. This paper aims at comparing 25 e-government maturity models to find the similarities and differences between them and, to identify their weaknesses and strengths. Although the maturity models present large similarities between them, our findings show that the features included in those models differ from a maturity model to another. Furthermore, while some maturity models are covering some features and introducing new ones, it seems that others are just ignoring them. |
| PROGRAM SLICING TECHNIQUES AND ITS APPLICATIONS | N.Sasirekha  A.Edwin Robert |  | IJSEA | Program understanding is an important aspect in Software Maintenance and Reengineering. Understanding the program is related to execution behaviour and relationship of variable involved in the program. The task of finding all statements in a program that directly or indirectly influence the value for an occurrence of a variablegives the set of statements that can affect the value of a variable at some point in a program is called a program slice. Program slicing is a technique for extracting parts ofcomputer programs by tracing the programs’ control and data flow related to some data item. This technique is applicable in various areas such as debugging, program comprehension and understanding, program integration, cohesion measurement, re-engineering, maintenance, testing where it is useful to be able to focus on relevant parts of large programs. This paper focuses on the various slicing techniques (not limited to) like static slicing, quasi static slicing, dynamic slicing and conditional slicing. This paper also includes various methods in performing the slicing like forward slicing,s yntactic slicing and semantic slicing. The slicing of a program is carried out using Java which is a object oriented programming language. |
| DESIGNING CODE LEVEL REUSABLESOFTWARE COMPONENTS |  |  | IJSEA | The basic idea behind building Reusable software components is to design interchangeable parts from other industries to the software field of construction. A reuse library or component reuse repository organizes stores and manages reusable components. The biggest advantage of the building reusable software components is that it reduces the time and energy in developing any software. Frameworks provides a standard working system through which user ‘s focus is on developing desired modules instead of developing lower-level details. By using this facility, the software developers can spend more time in developing the requirement of software, rather than preparing the tools of application development. Framework is set of reusable software program that forms the basis for an application. Frameworks help the programmers to build the application quickly. At its best code reuse is accomplished through the sharing of common classes and/or collections of functions, frameworks and procedures. This paper describes how to build the code level reusable components and how to design code level components. Finally providing coding guidelines, standards and best practices used for creating reusable code level components and guidelines and best practices for making configurable and easy to use. |
| SOFTWARE METRICS VALIDATION METHODOLOGIES INSOFTWARE ENGINEERING |  |  | IJSEA | In the software measurement validations, assessing the validation of software metrics in software engineering is a very difficult task due to lack of theoretical methodology and empirical methodology [41, 44, 45]. During recent years, there have been a number of researchers addressing the issue of validating software metrics. At present, software metrics are validated theoretically using properties of measures. Further, software measurement plays an important role in understanding and controlling software development practices and products. The major requirement in software measurement is that the measures must represent accurately those attributes they purport to quantify and validation is critical to the success of software measurement. Normally, validation is a collection of analysis and testing activities across the full life cycle and complements the efforts of other quality engineering functions and validation is a critical task in any engineering project. Further, validation objective is to discover defects in a system and assess whether or not the system is useful and usable in operational situation. In the case of software engineering, validation is one of the software engineering disciplines that help build quality into software. The major objective of software validation process is to determine that the software performs its intended functions correctly and provides information about its quality and reliability. This paper discusses the validation methodology, techniques and different properties of measures that are used for software metrics validation. In most cases, theoretical and empirical validations are conducted for software metrics validations in software engineering |
| A NEW APPROACH TO REQUIREMENT ELICITATION BASED ON STAKEHOLDER RECOMMENDATION AND COLLABORATIVE FILTERING | Nilofar Mulla  Sheetal Girase |  | IJSEA | The customers' needs in a software project are identified in the process of Software requirements elicitation. For building a software system this process is considered as one of the most important parts. In this part it is decided precisely what will be built. A close interaction between developers and end-users of the system is needed by requirements’ gathering. Meetings can be costly, inconvenient, and infrequent if developers and end-users are in different organizations or different cities. The quality of the elicited requirements can greatly be impacted if there is a problem of communication. Requirement elicitation is a process difficult to scale to large software projects with many stakeholders which involves identifying and prioritizing requirements. A stakeholder is an individual or a group who can influence or be influenced by the success or failure of a project. Existing methods to identify and prioritize requirements do not scale well to large projects. Large projects tend to be beset by three problems: information overload, inadequate stakeholder input, and biased prioritization of requirements. |

|  |  |
| --- | --- |
| **4.e** | **Based on research paper an identify the current Problem statement** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Problem Statement | Used in | | | | | |
| Quiz | Assignment | Lab | Mini Project | Poster Presentation | Test |
| 1. Considering your mini project, identify all the stakeholder and their requirements using collaborating requirements gathering.  2 . Design of a digital platform |  |  |  | ✓ |  | ✓ |

|  |  |
| --- | --- |
| **4.f** | **Identify Companies / Industries which use the knowledge of the subject and thus may provide Internships and final Placements** |

|  |  |  |  |
| --- | --- | --- | --- |
| Name of the Company | To be / Contacted for | | |
| Student Internship | Student Final Placement | Faculty Internship |
| TCS | ✓ | ✓ | ✓ |
| Google | ✓ | ✓ |  |
| Cloud Counselage | ✓ | ✓ | ✓ |
| Carwale | ✓ | ✓ | ✓ |

|  |  |
| --- | --- |
| **4.g** | **Identify suitable relevant TOP Guest Speakers from Industry (CS50 Lecture by Mark Zuckerberg - 7 December 2005 - YouTube)** |

|  |  |  |
| --- | --- | --- |
| Name of the Identified Guest Speaker | Designation | Name of the Company |
| Mr. Nitin Komawar | Founder | Grok Learning Pvt. Ltd |
| Mr.Aniket Mahala | Global Head, Devopp | Oracle |
| Dr.Swapnail Dambe | Director UI/UX | HCL |

|  |  |
| --- | --- |
| **4.h** | **Identify relevant technical competitions to participate [Competitions -Paper Presentations, Projects, Hackathons, IVs etc..]** |

|  |  |  |
| --- | --- | --- |
| Name of the Relevant Technical Competition Identified to participate | Organized by | Date of the Event |
| SIH/Cavach | AICTE | 25-26 August 2023 |
| ICPC | ICPC Global | 6-11 November 2023 |

|  |  |
| --- | --- |
| **4.i** | **Identify faculty in TOP schools / Universities who are teaching same / similar subject and develop rapport e.g. Exchange Lecture Material (Assignments / Tests / Project etc..), Joint Paper Publication** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| University | Name of the Course | Name of Faculty | Type of Collaboration | | |
| Exchange of Lecture Material | Joint Publication/ Research | Other |
| IIT Kanpur | Software Engineering | Mr.Pankaj Jalote |  |  | Book reference |
| IIT Bombay | Software Architecture | Mr.Rishikesh Joshi | ✓ |  |  |

|  |  |
| --- | --- |
| **4.j** | **Web Links and Names of Magazines, Journals, E-journals – [VIT is member of IIT Bombay Library]** |

Refer online journals subscribed in VIT library. You can also access IIT Bombay online library for journals from IITB campus.

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Web-Links and Names of Journals and E-Journals Recommended to Students for this Course | Web-Links and Names of Magazines Recommended to Students for this Course | Module Nos. |
| 1 | IJSEA https://www.researchgate.net/publication | SD Times | 2,3 |
| 2 | Advances in software Engineering  https://www.sciencedirect.com/journal/advances-in-engineering-software | Developer Tech | 5 |
| 3 | Information and Software Technology  https://www.sciencedirect.com/journal/information-and-software-technology | Today Software Magazine | 4,6 |

|  |  |
| --- | --- |
| **4.k** | **Module Best Available in - Tick ONE best resource [from *4.a* to *4.d* in this AAP] & give details** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Module No. | Category ( Please Tick Mark ) - √ | | | | | | Available In VIT Library? | | Details of the Resource  (i.e. Name, Chapter no.etc.) |
| Book | | | Maga-zine | Journals | |
| Text | Reference | E-Book | Regular | E-Journal | Y | N |
| 1 | ✓ |  |  |  |  |  |  |  |  |
| 2 | ✓ |  |  |  |  |  |  |  |  |
| 3 | ✓ |  |  |  |  |  |  |  |  |
| 4 | ✓ |  |  |  |  |  |  |  |  |
| 5 | ✓ |  |  |  |  |  |  |  |  |
| 6 | ✓ |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **4.l** | **Referred to any top-rated university in that subject for content** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| University | Name of the Course | Name of Faculty | Date of Delivery of the Course | Remarks |
| Oxford | Software Engineering | Jim Davies |  | Not available for external students |
| Cambridge | Software Engineering | Ross Anderson |  | Not available for external students |
| ETH Zurich | Software Engineering | Peter Muller |  | Not available for external students |

|  |  |
| --- | --- |
| **4.m** | **Faculty received any certification related to their subject. List of Certifications Identified / Done** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Course | Certifying Agency | Certification | | Remarks |
| Done on | Proposed to be on |
| Software Engineering | NPTEL |  | Dec 2023 | Registered |

|  |  |
| --- | --- |
| **4.n** | **Completed subject wise/cluster wise training with cluster mentor.**  **List of relevant Refresher Course Identified / Done** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Course | Certifying Agency  (As suggested by DAB/Cluster Mentor/Industry/University other than MU) | Certification | | Remarks |
| Done on | Proposed to be on |
| Pedagogy | Introducing ICT systems |  | December 2022 |  |
|  |  |  |  |
| PBL | Swayam |  | December 2022 |  |
|  |  |  |  |
| Sub. Content Training |  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| **4.o** | **Best Practices Identified and adopted** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Item | Best Practices Identified | | |
| University at Albany | CMU | IITB |
| 1 | Microsite |  |  |  |
| 2 | Video Lectures |  |  |  |
| 3 | Assignments |  | √ | √ |
| 4 | Mini Project |  |  |  |
| 5 | Assessment Metric |  | √ |  |
| 6 | Quizzes |  |  |  |
| 7 | Labs/ Practical (PBL) | √ |  |  |
| 8 | Tests |  |  |  |
| 9 | Etc |  |  |  |
| 10 | Peer Assessment etc. |  | √ |  |

|  |  |
| --- | --- |
| **4.p** | **Web Links for Online Notes/YouTube/VIT Digital Content/VIT Lecture Capture/NPTEL Videos** |

Students can view lectures by VIT professors, captured through LMS ‘Lecture Capture’ in VIT campus for previous years.

|  |  |  |
| --- | --- | --- |
| No. | Websites / Links | Module Nos. |
| 1 | <http://freevideolectures.com/Course/2318/Software-Engineering> | 1,2,3,4,5 |
| 2 | <https://www.youtube.com/watch?v=7ykI5NxyO5E> | 5 |
| 3 | <https://www.youtube.com/watch?v=izAq05SBvMI> | 4 |

|  |  |
| --- | --- |
| **4.q** | **Recommended MOOC Courses like Coursera / NPTEL / MIT-OCW / edX/VAC etc.** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | MOOC Course Link | Course conducted by – Person / University / Institute / Industry | Course Duration | Certificate (Y / N) |
| 1 | Course Name-: Software Engineering Basics  <https://nptel.ac.in/courses/106/105/106105182/> | NPTEL | 8 Weeks | Y |

|  |  |
| --- | --- |
| **5** | **Consolidated Course Lesson Plan** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | From (date/month/year) | From (date/month/year) | Total Number of Weeks |
| Semester Duration | 11/07/2022 | 22/10/2022 | 15 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Week | Lecture no. | Module No. | Lecture Topics / IA 1 and IA 2 / BSA planned to be covered | Actual date of Completion | Cos | Recommended  Prior Viewing / Reading | |
| Lecture No. (on LMS) | Chapter No. / Page Nos./ Books/ Web Site |
| 1 | 1 | 1 | Software Engineering-process framework, the Capability Maturity Model (CMM) |  | CO1 |  | T1/ch-1/3 |
| 1 | Advanced Trends in Software Engineering |  | CO1 |  | T1/ch-2/31/ |
| 2 | 1 | 1 | **Prescriptive Process Models:** The Waterfall Model, V-model, Incremental Process Models, Evolutionary Process Models, Concurrent Models |  | CO1 |  | T1/ch-1/38 |
| 1 | Agile process, Agility Principles, Extreme Programming (XP), Scrum, Kanban model |  | CO1 |  | T1/ch-3/65 |
| 3 | 2 | 2 | Requirement Engineering, Requirement Modelling, Data flow diagram, Scenario based model |  | CO1 |  | T1/ch-5/128 |
| 2 | Software Requirement Specification document format (IEEE) |  | CO1 |  | T1/Appendix1 |
| 4 | 3 | 3 | Software Metrics |  | CO2 |  | T1/Appendix1 |
| 3 | Software Project Estimation LOC, FP, |  | CO2 |  | T1/Appendix1 |
| 5 | 3 | 3 | Software Project Estimation (COCOMO II)  ESE 1 |  | CO2 |  | T1/ch-24/645 |
|  | 3 | Project Scheduling & Tracking |  | CO2 |  | T1/ch-26/692 |
| 6 | 4 | 4 | Design Principles & Concepts |  | CO3 |  | T1/ch-26/708 |
| 4 | Design Principles & Concepts |  | CO3 |  | T1/ch-27/721 |
| 7 |  |  | **Presentations** |  |  |  |  |
| 8 | 7 | 4 | Effective Modular Design, |  | CO3 |  | T1/ch-8/215 |
| 4 | Cohesion and Coupling, Architectural design |  | CO3 |  | T1/ch-9/242 |
| 9 | 8 | 5 | Unit testing, Integration testing, |  | CO4 |  | T1/ch-10/276 |
| 5 | Validation testing, System testing |  | CO4 |  | T1/ch-10/276 |
| 10 | 9 | 5 | Testing Techniques, white-box testing: Basis path, Control structure testing |  | CO4 |  | T1/ch-11/312 |
| 5 | black-box testing: Graph based, Equivalence, Boundary Value |  | CO4 |  | T1/ch-28/744 |
| 11 | 10 | 5 | Types of Software Maintenance, Re-Engineering, Reverse Engineering |  | CO4 |  | T1/ch-22/584 |
| 5 | Types of Software Maintenance, Re-Engineering, Reverse Engineering |  | CO4 |  | T1/ch-14&16/398&432 |
| 12 | 11 | 6 | Risk Analysis & Management: Risk Mitigation, Monitoring and Management Plan (RMMM). |  | CO5 |  | T1/ch-16/432 |
| 6 | Risk Analysis & Management: Risk Mitigation, Monitoring and Management Plan (RMMM). |  | CO5 |  | T1/ch-17/449 |
| 13 | 12 | 6 | Quality Concepts and Software Quality assurance Metrics, |  | CO5 |  | T1/ch-17/456 |
| 6 | Formal Technical Reviews, Software Reliability |  | CO5 |  | T1/ch-18/481 |
| 14 | 13 | 6 | The Software Configuration Management (SCM), Version Control and  Change Control  ESE 2 |  | CO5 |  | T1/ch-18/495 |
|  | **Revision**  ESE 3 |  |  |  | T1/ch-29/761 |
| 15 | 14 |  | **Seminar Presentation** |  |  |  |  |
|  | **Seminar Presentation** |  |  |  |  |

|  |  |
| --- | --- |
| **6** | **Rubric for Grading and Marking of Term Work (inform students at the beginning of semester)** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lecture + Practical (% Attendance) & Marks | Assign-ments | Tutorial | Lab / Practical Performance | Lab Journal Assessment | Class Tests (Other than IA) | Other (1)  specify | Other (2)  specify | Total |
| 75% | -- | -- | 25 | -- |  |  |  | 25 |

**ISA 20 Marks**

11 assignments 2 marks each, best of 10 will be considered.

**MSE 30 Marks**

Following activities will be considered for MSE.

1. Online course of minimum 5 hours on ChatGPT **5 Marks** (Completion certificate to be submitted)

2. Online course of minimum 5 hours on Scrum Master **5 Marks** (Completion certificate to be submitted)

3. Think of a use case to create a digital platform. Elaborate on its need. Submit report **5 marks.**

4. Field study (Group activity of 5) - Report on problem identification and its solution **15 marks**

Batch 1: Visit Department store/Mall/Supermarket on weekends, identify pain areas and suggest solutions.

Batch 2: Visit Toll on weekdays/weekends during peak hours, identify pain areas and suggest solutions.

Batch 3: Identify issues on the campus and suggest solutions.

Batch 4: Suggest solution for traffic congestion.

**\*Single report copy per group**

**ESE (**Best of 2 will be considered.) 5**0 marks.**

Test 1Module 1,2,3 20 Marks

Test 2 Module 4,5,6 20 Marks

Test 3 Module 1,2,3,4,5,6 20 Marks

Presentation on assigned topic 10 Marks.

|  |  |
| --- | --- |
| **7** | **Assignments / Tutorials Details** |

**Analysis of Assignment / Tutorial Questions and Related Resources**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Assignment / Tutorial No. | Week No. | Type\* (√) | | | Module No. | Based on # | | | Question Type (√) | |
| R | PQ | OBT | Text Book | Reference  Book | Other Learning Resource | MU EQ | Thought Provoking |
| 1 | 12-15 |  |  |  | 1,2,3,4,5,6 | ✓ | ✓ | ✓ |  | ✓ |
| 2 | 10-15 |  |  | ✓ | 1,2,3,4,5,6 | ✓ | ✓ | ✓ |  | ✓ |
| 3 | 9 |  | ✓ |  |  |  |  |  |  | ✓ |
| 4 | 5 |  |  | ✓ | 1,2,3 | ✓ |  |  |  | ✓ |

\* Tick (√) the Type of the Assignment: Regular (R); Pop Quiz (PQ) ; Open Book Test for TE/BE/ME (OBT)

# Write number for textbook, reference book, other learning resource from this AAP – *from* *Points 4.a to 4.d*

|  |  |
| --- | --- |
| **8** | **Internal Assessment / Other Class Test / Open Book Test (OBT)/Take Home Test (THT) Details** |

**Refer 6**

|  |  |
| --- | --- |
| **9.a** | **Practical Activities – Regular Experiments** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Practical  No. | Module  No. | Title of the Regular Experiments | Concepts to be highlighted | LO Map | Audit / Quality Rate  (0 to 4) |
| 1 | 1 | Identify the role of the software in today’s world across a few significant domains related today life. | Nature of Software | LO1 | 4 |
| 2 | 1 | Prepare detailed statement of problem for the selected / allotted mini project and identify suitable traditional process models(min 2) for the same with justification. | Process model identification | LO1 | 4 |
| 3 | 1 | Prepare detailed statement of problem for the selected / allotted mini project and identify suitable agile development models (min 2) for the same with justification. | Process model identification | LO1 | 4 |
| 4 | 2 | Develop Software Requirement specification (SRS) document in IEEE format for the project. | Requirement identification | LO1 | 4 |
| 5 | 4 | Draw DFD (minimum 2 levels) for given mini projects | Software Design | LO2 | 4 |
| 6 | 3 | Perform Software Cost estimation for selected Mini Projects |  | LO1 | 4 |
| 7 | 3 | Use project management tool to prepare schedule for the project | Project scheduling and tracking | LO3 | 4 |
| 8 | 5 | Develop test cases for the project using white box testing. | Software Testing | LO2 | 4 |
| 9 | 5 | Develop test cases for the project using black box testing. | Software Testing | LO2 | 4 |
| 10 | 6 | Prepare RMMM plan for the project. | Risk management | LO1 | 4 |
| 11 | 6 | Change specification and make different versions using any SCM Tool. | Configuration management | LO3 | 4 |

|  |  |
| --- | --- |
| **9.b** | **Practical Activities – Newly Added Experiments** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Practical No. | Module No. | Title of the **Newly Added Experiments** | Concepts to be highlighted | CO Map |
| 1 | 1 | Study of different Agile models | Agile development | 1 |
| 2 | 3 | Estimation | Project management | 2 |

|  |  |
| --- | --- |
| **9.c** | **Practical Activities – PBL Experiments** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Practical No. | Module No. | Title of the **PBL Experiments** | Concepts to be highlighted | CO Map |
| 1 | 3 | Create Gantt chart and network diagram for your chosen mini project | Scheduling | 2 |
| 2 | 2 | Analysis model for mini project | Communication | 2,5 |
|  |  |  |  |  |

|  |  |
| --- | --- |
| **10** | **Beyond Syllabus Activities for Gap Mitigation** |

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Type of the Activity | Activities | Details – no of attendees, guest, feedback, mark sheet, report |
| 1 | **Experiential learning/Interaction with Outside World** | 1- Guest Lectures by Industry Expert | Aniket Mahala, Global head DevOps, Oracle  Dr.Swapnil Dambe, Directoe UI/UX, HCL |
| 2- Workshops | NA |
| 3- Mini Project | Laboratory work |
| 4- Industrial Visit | Paloalto, BKC, Mumbai  Grok Learning, Mumbai |
| 5- Any other activity | NA |
| 2 | **Collaborative & Group Activity** | 1- Poster Presentation | NA |
| 2- Minute Papers | NA |
| 3- Students Seminars | Scheduled |
| 4- Students Debates | NA |
| 5- Panel Discussion / Mock GD | NA |
| 6- Mock Interview | NA |
| 7- Any other activity | NA |
| 3 | **Co-Curricular Activity** | 1- Informative videos (NPTEL/Youtube /TEDx/ MIT OW/edX) | Will be screened during lecture |
| 2- Lecture Capture Usage | NA |
| 3-Any other activity | NA |
| 4 | **Tests & Assessments** | 1- Class Tests/ Weekly Tests | NA |
| 2- Pop Quiz | NA |
| 3- Mobile App Based Quiz |  |
| 4- Open Book | ESE 3 |
| 5- Take Home Test |  |
| 6-Any other activity |  |

|  |  |
| --- | --- |
| **11.1** | **One-on-One Academic Mentoring Meetings done** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name of Mentee | Date of One-On-One Meeting | | |
| Beginning of Sem. | After Mid Term Results | Before End Sem. |
|  |  |  |  |  |

|  |  |
| --- | --- |
| **11.2** | **Identify Financial Concerns and refer appropriately** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name of Mentee |  | | |
| Individual Goals Identified | Any Financial Concern which needs to be referred to | Any Emotional Concern to be referred to |
|  |  |  |  |  |

**\* Do not delete any activity. Give details for planned events. Write ‘NA’ for activity Not Planned.**

Consolidated Academic Administration PlanPrepared by (mention all theory teaching faculty names with signature)

Please write below your name and sign with date of the external cluster mentor meeting

|  |  |  |
| --- | --- | --- |
| Faculty 1 | Faculty 2 | Faculty 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| External Industry Mentor | External Academic Mentor | VIT Cluster Mentor | Program HOD |