Course Name: Software Engineering

Course ID: IT314

Course Structure: 3 – 0 – 3 – 4.5

Course Objectives:
The following are the objectives of the course:

- The goal of this course is to provide the student with a solid understanding of the foundations of software engineering.
- The course provides a solid foundation for students who want to develop their career in the broad field of computing and specifically in the areas of Information and Communication technology.
- The course aims at teaching the students how to apply key engineering principles and mathematical models to application development projects.
- The course emphasizes the complete lifecycle of the software development process and the students learn how to design, develop, test, and deploy software using rigorous software engineering practices.
- The course will aid the students in developing the skills and abilities of applying the fundamental concepts of computing in industrial, business and other problems, in order to produce software solutions.
- The course also introduces the students to the role of software tools in the process of software development.

This course has been designed to provide the students with the opportunity to apply the software engineering principles learned in this course to a project, that is also a part of this course. Students will work on a significant software development project that may include any or all activities associated with creating a software solution to a client/customer problem. They would be taken through all the software engineering activities that are typically experienced from the initiation to the completion of a software development project. Special emphasis has been placed on defining the client/customer problem and determining requirements by either working with real clients on real world problems. Teams are encouraged to work autonomously following good software engineering practices, with guidance in the form of lectures and tutorials, from the course instructor and teaching assistants. In addition, issue based assistance is provided as and when required or as and when the same is sought by the team members.

Course Contents:
1. Introduction to Software Engineering
2. Software life cycle models
3. Software Project phases
   A] Pre-development phase
      i) Feasibility & Proposal
   B] Development phase
      i) Requirements analysis & specification
      ii) Design
      iii) Coding & Unit Testing
      iv) Testing
   C] Post-development phase
      i) Maintenance
4. Software Project Management
   i) Software Project Estimation
   ii) Software Project Scheduling
   iii) Risk Management
   iv) Configuration Management
   v) Software Reliability and Quality Assurance
5. Computer Aided Software Engineering (CASE)
6. Agile Methodologies

Expected Learning Outcomes:
Upon completion of this course the student should be able to:

1. Enumerate and define the phases in the software development process.
2. Describe the activities performed in each of the phases and how each phase relates to the others.
3. Develop a coherent set of software requirements for a particular application.
4. Convert a set of requirements into a set of specifications that can be validated.
5. Apply any of several design methodologies to the design of a software work product.
6. Develop and implement a test plan that will adequately exercise a software work product with the purpose of discovering defects.
7. Enumerate and define the steps in the post-implementation phases.
8. Describe the activities associated with corrective, adaptive, and perfective maintenance.
9. Describe the activities associated with the configuration management process and relate its importance during software development and maintenance.
10. Perform an impact analysis for a change request as it applies to a software work product.
11. Perform all software engineering tasks associated with developing a software system or product requiring a team of software engineers.
12. At the end of the project, assess a software development effort to determine the appropriate principles and practices that will maximize the probabilities for success.

**Additional Expectations – the student should be able to:**

1. Analyze a software development project and determine the most appropriate software engineering principles and practices for the given situation.
2. Evaluate the effectiveness of a given set of software engineering practices and make recommendations for changes that can improve the software development project.
3. Analyze a software development project to determine missing or inappropriate software engineering practices.
4. Assess the quality of software engineering processes, practices, products, and artifacts associated with a software engineering development effort.
5. Demonstrate interpersonal and team skills that support maximizing their team’s effectiveness.

**Special Expectations – the student should be able to:**

1. Work collaboratively and cooperatively with others as a team that produces the required software engineering work products.
2. Create and deliver a quality presentation (individually and as part of a team presentation) related to selected aspects of software engineering processes, practices and work products associated with a software engineering project.

**Books & Reference Material**


**Journals**

2. IEEE Software
3. Journal of Systems and Software

**Attendance Policy:**

There is no weightage attached to a student’s attendance in the lectures of the course i.e. the attendance of the student in the lectures has no bearing on the marks (and hence the grade) that the student earns in the course.