

Name of the Module : Software Engineering
Module Code : CTE 303
Semester : V
Credit Value : 13
Module Leader : Tsheten Dorji
Module Tutor : Tsheten Dorji

General Objectives or aims of the module:

This module introduces students to analyse, design, develop, test, and implement software. It will develop students' ability to develop the software, and software maintenance. The module will help students to bring out the better products/systems.

Learning Outcomes:

At the end of this module, students are expected to be able to:

1. Differentiate between process, products and systems.
2. Use the software development process.
3. Analyze the software specifications and gather the software requirements.
4. Translate the requirements into different types of analysis diagrams and design.
5. Implement the different techniques and strategies of software testing.
6. Explain the quality management and review techniques.

Learning and Teaching Approach Used:

Lectures: 3 hours per week
Tutorial: 1 hour per week.
Self Study: 6 hour per week.

Assessment:

Continuous Assessment:

- 2 Assignments – 5 Marks
- Class Test/Quiz- 10 Marks [Date:]
- Midterm Exam-10 marks [Date:.....]
- Project Work – 25 Marks

Semester End Examination – 50 marks

SUBJECT MATTER:

THE PRODUCT AND PROCESS: Definition of terms, The evolution of software, Software characteristics, Software Applications, Software myths, A Layer technology, The software process.

PROCESS MODEL: Software process models, The Linear Sequential Model, The Prototyping Model, The RAD Model, Evolutionary Software Process Models: The Incremental Model, The Spiral Model, The Concurrent Development Model, Fourth generation Techniques.

SYSTEM ENGINEERING: System Engineering Hierarchy, Requirement Engineering: Requirement Elicitation, Requirements Analysis and Negotiation, Requirements Specification, Requirements Validation and Management, Functional Specifications.

ANALYSIS CONCEPTS AND MODELING: Requirement Analysis, Analysis Principles, Data Modeling: Data Objects, Attributes, and Relationships, Cardinality and Modality, Entity/Relationship Diagrams, Data flow Diagrams, Behavioral Modeling, The Data Dictionary, Other classical analysis methods.

DESIGN CONCEPTS: Design Principles and methods, Effective Modular Design: Functional Independence, Cohesion, and Coupling, Design Documentations, Architectural Design, Software Architecture, Mapping Design into Software Architecture: Transform flow, Transaction flow, User Interface Design, Interface design models, The User Interface Design process, Design Issues, Implementations Tools.

SOFTWARE TESTING TECHNIQUES: Software Testing Fundamentals, Test Case Design, Testing methods: White box testing, basic path testing, Control structure testing, Black box testing, Software Testing Strategies: Unit Testing, Integration Testing, Validation Testing, System Testing, Verification and Validation.

SOFTWARE QUALITY: Software Quality Assurance, Quality Concept-cost of quality, Software Quality Group, Roles and responsibilities of SQA group-Formal Technical reviews, Quality standards, Software Reliability.

References:

1. Roger S. Pressman , “Software Engineering – A Practitioner’s Approach”, McGRAW-HILL International Edition 2005, Sixth Edition.
2. Ian Sommerville, “ Software Engineering”, Pearson Education Asia, New Delhi, 2000.
3. Richard Fairley, “Software Engineering Concepts”, McGraw Hill International Edition, Singapore, 2000.
4. Ali Behforooz, Frederick J Hudson, “Software Engineering Fundamentals”, Oxford University Press, Indian Reprint, Noida, 2003.
5. Mall Rajib, “ Software Engineering”

11th December 2009