Pokhara University

Faculty of Science and Technology

Course No.: CMP 348	Full marks: 100
Course title: **Software Engineering**	Pass marks: 45
Nature of the course: Theory/Tutorial/Practical	Time per period: 1 hour
	Total periods: 45
Level: Undergraduate	Program: BE

1. Course Description

This course intends to provide students with an in-depth knowledge and practical skills pertaining to the field of Software Engineering. It intends to make students familiar with the principles and practices followed in the field of software development and gives a comprehensive understanding of various phases of Software Development activities including Requirements Engineering, Design, Testing and Maintenance. Apart from that the course has also been designed to provide students with a managerial perspective needed to oversee and plan large software development activities using contemporary technologies ensuring product and process quality.

2. General Objectives

The course is designed with the following objectives:

1. To make the students familiar with the principles and practices of Software Engineering required for successfully carrying out development activities
2. To provide students with a detailed knowledge of different phases of software development lifecycle like requirements engineering, architectural design, implementation and testing
3. To equip students with necessary skills and knowledge required for overseeing software development activities including project management, quality assurance, configuration management
4. To familiarize students with the recent trends in the field of software engineering

3. Methods of Instruction

The method of instruction includes lectures, tutorials and practical classes to cover the theoretical, tutorial and practical aspects. Students can be involved in group discussions and presentations in order to assimilate new ideas and current trends in the field of software engineering. Short quizzes can be held in the class to check the students' level of comprehension. Project work needs to be assigned to students to come up with a comprehensive documentation of a software product demonstrating their level of understanding of the contents studied in the course.

4. Contents in detail with specific objectives

Specific Objectives	Contents
The chapter intends to provide a brief introduction of the field of software engineering and software project management and ethics Students make use of the management aspect of software engineering required to undertake software related projects like estimation, measurement, risk management and ethics	**Chapter 1: Software Engineering and Project Management (7 Hrs)** 1.1 Nature and Characteristics of Software 1.2 Software versus System Engineering 1.3 Software Crisis and Myths 1.4 Four Ps of Software Project Management 1.5 Process and Project Metrics 1.6 Measurement of Software: Metrics, Measure, Indicator 1.7 Project Estimation, Empirical Estimation Models 1.8 Software Risks, Assumption, Issues, Dependency 1.8.1 Identification 1.8.2 Mitigation

	1.8.3 Monitoring 1.8.4 Management 1.9 Software Engineering Ethics and Professional Practice
In this chapter students learn about different types of software process models, their strength and weaknesses and their applicability Students also learn to make use of Agile development strategies for software development process	**Chapter 2: Software Process Models and Agility (5 Hr)** 2.1 Software Development Lifecycle (SDLC) 2.2 Waterfall, Incremental, Prototyping, Iterative, Spiral, Rapid Application Development, Aspect Oriented Software Engineering 2.3 Agile Software Development 2.3.1 Extreme Programming 2.3.2 Scrum 2.3.3 Agile Project Management and Scaling Agile Methods 2.4 Pros and Cons of Process Models and their Applicability
This chapter helps students learn how to gather, analyze, and specify software requirements Students make use of various techniques like structured and object oriented approaches for requirements modeling and management	**Chapter 3: Requirements Engineering and Principles (7 Hr)** 3.1 Types of Requirements 3.2 Requirements Modeling Principles 3.3 Domain Analysis and System Models 3.3.1 Context Models 3.3.2 Behavioural Models 3.3.3 Data Models

	3.4 Requirements Engineering Process 3.4.1 Feasibility Study 3.4.2 Requirements Elicitation, Analysis, Modeling, Specification 3.4.3 Requirements Validation 3.5 Object Oriented Analysis
This chapter intends to get students well acquainted with software design principles and patterns. Students use various architectural styles and design patterns including object oriented approach for designing software systems	**Chapter 4: Software Design, Architecture and Principles (7 Hr)** 4.1 Design Modeling Principles 4.2 Design Process 4.3 Architectural design 4.3.1 Layered 4.3.2 Repository 4.3.3 Client Server 4.3.4 Pipe and Filter 4.4 Interface, Component, Database Design 4.5 Design Patterns 4.6 Security by Design 4.7 Object Oriented Design 4.8 Embedded System Design

This chapter imparts knowledge regarding various testing techniques used for proper validation and verification of software. It also lets students become aware of the inevitable changes that occur and ways of maintaining software systems Students design test cases for proper validation and verification of the software product	**Chapter 5: Testing Techniques and Maintenance (7 Hr)** 5.1 Validation and Verification 5.2 Testing Phases 5.2.1 Development Testing 5.2.1.1 Unit Testing 5.2.1.2 Component Testing 5.2.1.3 System Testing 5.2.2 Release Testing 5.2.2.1 Requirements-based Testing 5.2.2.2 Scenario Testing 5.2.2.3 Performance Testing 5.2.3 User Testing 5.2.3.1 Alpha Testing 5.2.3.2 Beta Testing 5.2.3.3 Acceptance Testing 5.3 Test Case Development Strategies 5.3.1 Boundary Value Analysis 5.3.2 Equivalence Partitioning 5.3.3 Basis Path Testing 5.3.4 Control structure Testing 5.4 Software Changes and Evolution 5.5 Maintenance Process and Reengineering

This chapter focuses on quality management of software and the existing standards for compliance and assessing capability In this chapter, students make use of ways of ensuring quality in process and product by adhering to the standards which is essential in producing good software	**Chapter 6: Software Quality Assurance Process (4 Hr)** 6.1 Software Quality Concepts, Software Reliability 6.2 Software Quality Management and Planning 6.3 Software Standards and their Compliance 6.3.1 ISO, CMMI 6.4 Capability Assessment and Process Improvement 6.5 Process and Product Standards 6.6 Reviews and Inspections
This chapter focuses on the need and process of proper management of software configuration items produced as deliverables during the software development process Students use the techniques and tools that exist for keeping track of the software configuration items	**Chapter 7: Software Configuration Management (4 Hr)** 7.1 Software Configuration Items 7.2 Configuration Management Activities 7.2.1 Change Management 7.2.2 Version Management 7.2.3 System Building 7.2.4 System Release

This chapter deals with the recent trends and advancements in the field of software engineering This chapter enables students to make use of contemporary techniques for reusing software and software components and use advanced technologies for software development	**Chapter 8: Advanced Software Engineering Concepts (4 Hr)** 8.1 Software Reuse 8.1.1 Application Frameworks 8.1.2 Software Product Lines 8.1.3 COTS Product Reuse 8.2 Cloud Based Software Engineering 8.3 Artificial Intelligence in Software Engineering

5. List of Tutorials

The following tutorial activities of 15 hours per group of maximum 24 students should be conducted to cover all the required contents of this course.

S.N.	Tutorials
1	Discussions and exercises on Process models and their suitability
2	Exercises on computing metrics for a software system (Function Point, Lines of Code (LOC), COCOMO model) and using them to derive measures and indicators
3	Exercises on computing risk exposure factor and prioritizing risks
4	Exercises on developing test cases using testing techniques like Basis Path Testing, Boundary Value Analysis, Equivalence Partitioning and Control Structure Testing
5	Exercises on performing Domain investigation/analysis and specifying requirements
6	Exercises on drawing Data Flow diagram (Context and lower level diagrams)
7	Exercises on drawing Entity Relationship diagram

8	Exercises on drawing Use case, Sequence, Activity diagram
9	Exercises on drawing Architectural, Component and Class diagram

6. Practical Work

The practical work should consist of exercises focusing on all of the aspects covered in the course pertaining to software development. Students are required to make use of a CASE tool to come up with the documentation of the deliverables of each and every phase of the Software Development Life Cycle including analysis, design, implementation, testing.

A project work should be given to students in which they are required to demonstrate the assimilated knowledge in the course by coming up with detailed artifacts.

S.N.	Practical works
1	Introduction of a CASE tool for Software Engineering Activities (Rational Rose, Star UML, draw.io, Dia, Visio etc.)
2	Identify a domain that needs automation through the use of software and perform Domain analysis/investigation of that domain
3	Perform Requirements analysis of a system and develop a requirements catalog, translate them into task or user story format and enter them in a project management tool (Trello, Jira etc.)
4	Use a CASE tool to develop Data Flow Diagram of the functional requirements captured
5	Use a CASE tool to develop Entity Relationship (ER) diagram
6	Use a CASE tool to develop visual models of the requirements captured using Use case, Sequence and Activity diagram
7	Design the overall architecture of a software system
8	Develop a testing strategy and test cases for testing a software system
9	Use automated tools for properly managing software configuration items and their versions

7. Evaluation system and students' responsibilities

Internal Evaluation

In addition to the formal end-semester exam(s), the internal (formative) evaluation of a student may consist of quizzes, assignments, lab reports, projects, class participation and presentation etc. The tabular presentation of the internal evaluation is as follows. The components may differ according to the nature of the subjects.

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		30	Semester-End examination	50
Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Practical		20		
Attendance & Class Participation	20%			
Lab Report/Project Report	30%			
Practical Exam/Project Work	30%			
Viva	20%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Student requirements:

Each student must secure at least 45% marks in internal evaluation with 80% attendance in the class in order to appear in the semester-end examination. Failing to get such a score will be equated with NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End- Semester examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. Failure of a student to attend a formal exam, quiz, test, etc. won't qualify him/her for re-exam. ***Students are required to complete all the requirements defined for the completion of the course***

8. Prescribed Books and References

Text Books:

Ian Sommerville, Software Engineering, 10th Edition, Addison - Wesley, ISBN-13: 978-0-13-703515-1

Roger S. Pressman, Bruce R. Maxim, Software Engineering a Practitioner's Approach. 8th Edition, McGraw Hill, ISBN-13: 978-0-07-802212-8

References:

Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Addison Wesley, ISBN: 0-201-57168-4

Frederick P. Brooks, The Mythical Man Month: Essays on Software Engineering, ISBN-10. 9780201835953

Derek Partridge, Artificial Intelligence and Software Engineering, ISBN 9780893916060

Muthu Ramacharan, Zaigham Mahmood, Software Engineering in the Era of Cloud Computing, 2021, ISBN: 978-3-030-33623-3

Pankaj Jalote, Software Engineering- A Precise Approach, Latest Edition

Rajib Mall, Fundamental of Software Engineering, Latest Edition