Pokhara University
Faculty of Science and Technology

Course No.: xxx xxx
Course title: **Software Engineering Fundamentals**
Nature of the course: Theory & Practical
Year: Second
Level: Bachelor

Full marks: 100
Pass marks: 45
Time per period: 1 hour
Total periods: 45
Program: BE

1. Course Description

Software Engineering Fundamentals is designed to equip students with essential knowledge and practical skills required for a successful career in software development. Starting with an introduction to the field of software engineering, students will journey through the various phases of software development, including requirements gathering, design, development, testing, and maintenance. The course provides a comprehensive understanding of key concepts such as software architecture, project management, and software quality assurance. They will also delve into software testing methodologies and gain hands-on experience in creating test cases.The aim of this course is to provide solid foundation in software engineering principles, practices, and tools, making them well-prepared for further studies in the field of software development.

2. General Objectives

- To provide students with a solid foundation in the principles and practices that are essential for successful software development projects.
- To provide the concepts and techniques required to direct and control the development of medium to large-scale software, including project management, quality assurance, software process improvement and software metrics.
- To broaden student's understanding of possible software development paradigms (e.g., structured analysis and design, object-oriented approaches).

3. Methods of Instruction

3.1.General instructional Techniques: Lecture, discussion, readings.
3.2.Specific instructional Techniques: Lab works, case study

4. Contents in Detail.

Specific Objectives	**Contents**

- Students will be familiarized with the different software process models - Students will be familiarized with agile development model and software management concept	**Unit 1: Introduction to software engineering (5 hrs)** 1.1 Overview of software engineering and its application domain 1.2 Software process models (waterfall, iterative, spiral, rapid application development,reuse oriented) 1.3 Introduction to agile development (extreme programming,scrum) 1.4 Software Crisis and Myths 1.5 Four P’s of software
- Students will be familiarized with software measurement ,cost estimation, software risk and projection of risk	**Unit 2: Software metrics, Software Project Planning and Risk (5 hrs)** 2.1 basic terminologies of software metrics(Measures, Metrics, and Indicators) 2.2 software metrics guidelines 2.3 product,process and project metrics 2.4 Objectives, Scope, Resources, Project Estimation, Decomposition Techniques 2.5 Empirical Estimation Models, Risk Management Strategies 2.6 Software Risks, risk Identification, Risk Projection
- Students will be able understand how to gather requirements and model them using different diagramming notation.	**Unit 3: Analysis Concepts and Principles(7 hrs)** 3.1 Requirement elicitation(Initiating the Process Facilitated Application Specification Techniques Quality Function Deployment) 3.2 Use Cases and User Stories 3.3 Reqirement specification 3.4 Analysis principles 3.5Software prototyping. Specification and Specification Review 3.6 Analysis Modeling: Elements of Analysis Model 3.7 Data Modeling Functional Modeling and Information Flow 3.8 Behavioral Modeling, Structured Analysis- Data

	Dictionary
- Students will be familiarized with the design concept, principles, software architecture styles and design pattern.	**Unit 4: Design Concepts and Principles(7 hours)** 4.1 Design Process, Principles and Concepts 4.2 Architectural and Component Level Design 4.3 Software Architecture, Data Design, Architectural Styles 4.4Mapping Requirements into a Software Architecture 4.5Structured Programming, Comparison of Design Notation 4.6 Introduction to design pattern
- Students will be familarized with different types of testing and able to design test case.	**Unit 5: Software Testing Techniques and Maintenance (7 hrs)** 5.1 Testing Fundamentals 5.2 Test Case Design 5.3 White Box Testing. Basis Path Testing 5.4 Control Structure Testing 5.5 Black - Box Testing. Unit Testing 5.6 Integration Testing 5.7 Validation Testing, System Testing 5.8 software maintenance and its types
- Students will be familarized with quality standards, software reviews and CMM model.	**Unit 6: Software quality assurance(4 hrs)** 6.1 Concepts, Software Quality Assurance 6.2 Software Reviews, Formal Technical Reviews 6.3 Formal approaches to SQA 6.4 Statistical Quality Assurance. Software Reliability 6.5 ISO 9000 Quality Standards, SQA Plan 6.6 Capability Maturity Model (CMM)
- Students will be able to understand SCM activities like version control and change control..	**Unit 7: Software Configuration Management (4 hrs)** 7.1 Software Configuration Management, SCM Process 7.2 Identification of objects in the Software Configuration, Version Control

	7.3 Change Control, Configuration Audit 7.4 Status Reporting, SCM Standards
- Student will be familarized with basic of object oriented development	**Unit 8: Introduction to object oriented development (6 hrs)** 8.1 Object - Oriented paradigm 8.2 Object - Oriented Concepts 8.3 Identifying the Elements of an Object Model 8.4 Domain Analysis 8.5 Components of the OO Analysis Model 8.6 The OOA Process, Design for Object-Oriented Systems 8.7 The System Design Process 8.8 The Object Design Process

5. List of Experiments:

Laboratory work of 45 hours per group of maximum 24 students should cover all the concepts of software engineering fundamentals studied in the lectures. Students should submit a final case study report that uses all the activities involved in software development life cycle.The marks for the practical work will be based on the case study work. . Students should be able to draw Use case Diagram, ER Diagram, data flow diagram, sequence diagram, activity diagram,

Tools: draw.io or Dia or Visio

6. List of Tutorials:

The various tutorial activities that suits this course should cover all the content of this course to give students a space to engage more actively with the course content in the presence of instructor. Students should submit tutorials as assignments or class works to the instructor for evaluation. The following tutorial activities of 15 hrs should be conducted to cover all the content of this course:

A. Discussion-based Tutorials: (2 hrs)

1. Software myth from customer and management perspective (Class discussion)
2. Process models
3. Agile development.

B. Problem solving-based Tutorials: (10 hrs)

4. Numerical examples based on function point and loc based metric.

5. Numerical examples based on COCOMO model.
6. Numerical examples for calculating risk exposure.
7. Creating ER diagram for common system(library,online booking,online shopping system)
8. Creating Data flow diagram for common system.
9. Creating use case model for common system.
10. Designing test case for different scenario of system.

C. Review and Question/Answer-based Tutorials: (3 hrs)

- Students ask questions within from the course content and assignments and review key course content in preparation for tests or exams.

7. Evaluation system and Students' Responsibilities

Internal Evaluation

The internal evaluation of a student may consist of assignments, attendance, test-exams, term-exams, lab reports and projects etc. The tabular presentation of the internal evaluation is as follows:

External Evaluation	**Marks**	**Internal Evaluation**	**Weight**	**Marks**
Semester-End examination	50	Assignments	10%	50
		Attendance	10%	
		Practical	40%	
		Term exam	40%	
	50	Internal Final	100%	50
Full Marks 50+50= 100				

Student Responsibilities:

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 score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear the Semester-End Examination. Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during the period. If a student fails to attend a formal exam, test, etc. there won't be any provision for re-exam.

8. Prescribed Books and References

Textbooks:

1. Pressman. R.S., Software Engineering a practitioners Approach. 7th Edition. McGraw Hill. 2010. ISBN: 978-0-07-337597-7

References

2. Mall. R., Foundations of Software Engineering. PHI. 2000. ISBNM 81-203-1445
3. Somnterville. I., Software Engineering, 5th Edition. Addison - Wesley. 1995. ISBN: 0-201-43579-9