



Course description:

Topics covered include: introduction to software engineering, software processes, software processes models, software requirements: functional and non-functional requirements, system models, and architectural design. Also contains a lab to applying modeling by using UML.

Aims of the course:

The course introduces

1. The problems and challenges that caused the birth of Software Engineering as a discipline, as well as to its scope and body of knowledge.
2. Software characteristics and problems
3. Software development processes and its types
4. Domain and Software modeling
5. Software requirements analysis
6. Software design
7. Software construction
8. General information about software testing and evaluation, software development tools and environments, and ethics and professionalism.

Intended Learning Outcomes: (ILOs)

A. Knowledge and Understanding

A1. Concepts and Theories:

List the basic of software engineering terminologies

List the software process.

List the concept of ethics.

List the concept of software product

List the concept of software quality

A2. Contemporary Trends, Problems and Research:

- Solving problems depending on methods, theories, and tools
- Effective way of requirements gathering
- Produce high quality software
- Case studies

A3. Professional Responsibility:

- A brief introduction to ethical issues that affect software engineering and professional responsibility



B. Subject-specific skills

B1. Problem solving skills:

Learn how to use the methods and techniques to build software with high qualities

B2. Modeling and Design:

Learn how to build the software using models and architectural design

B3. Application of Methods and Tools:

Learn how to use technique, methods and tools.

C. Critical-Thinking Skills

C1. Analytic skills: Assess

Learn how to analyze the problem

C2. Strategic Thinking:

Understanding the techniques used in software engineering

C3. Creative thinking and innovation:

An idea to make the creativity and modern solution

D. General and Transferable Skills (other skills relevant to employability and personal development)

D1. Communication:

How to be more communicate with each other

D2. Teamwork and Leadership:

Discuss and work in a group in order to study several cases, each of which has issues affecting the software engineering in common.

Course structures:

Week	Credit Hours	ILOs	Topics	Teaching Procedure	Assessment methods
1 st – 2 nd	6	A1,A3	Introduction <ul style="list-style-type: none">Professional software developmentSoftware engineering ethicsCase studies	Presentation methods and techniques, Sources of information and Instructional Aids	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation according to class Participation
3 rd – 5 th	9	A1	Software processes <ul style="list-style-type: none">Software process modelsProcess activitiesCoping with changeThe Rational Unified Process	Presentation methods and techniques, Sources of information and Instructional Aids	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation according to class Participation
6 th – 7 th	6	A1,A2	Software requirements <ul style="list-style-type: none">User and system requirementsFunctional and non-functional	Presentation methods and techniques, Sources of	Diagnostic tests to identify the students level and areas of weakness



			requirements <ul style="list-style-type: none"> • Domain requirements • The software requirements document • Requirements specification • Requirements engineering processes • Requirements elicitation and analysis • Requirements validation • Requirements management 	information and Instructional Aids	Formal (stage) evaluation according to class Participation First Exam
8 th – 9 th	6	B1,B2,B3	System models <ul style="list-style-type: none"> • Context models • Interaction models • Structural models • Behavioral models • Model-driven engineering 	Presentation methods and techniques, Sources of information and Instructional Aids	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation according to class Participation
10 th – 11 th	6	B1,B2,B3	Lab Practical modeling <ul style="list-style-type: none"> • Use case diagrams • Sequence diagrams • Class diagrams • Activity diagrams 	Lab Practical using case tools	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation according to class Participation
12 th – 13 th	6	B2,B3	Architectural design <ul style="list-style-type: none"> • Architectural design decisions • Architectural views • Architectural patterns • Application architectures 	Presentation methods and techniques, Sources of information and Instructional Aids	Diagnostic tests to identify the students level and areas of weakness Formal (stage) evaluation according to class Participation
14 th	6	C1,C2,D	Project management <ul style="list-style-type: none"> • Software management distinctions • Management activities Project plan • Project scheduling • Risk management 	Presentation methods and techniques, Sources of information and Instructional Aids	Second Exam
15 th					Final Exam

References:

A. Main Textbook:



Summerville, I. *Software Engineering*, Addison Wesley Longman Publishing Co., Inc.(5),
March 2011.

B. Supplementary Textbook(s):

Software Engineering A PRACTITIONER'S APPROACH FIFTH EDITION Roger S. Pressman, Ph.D.

Assessment Methods:

Methods	Grade	Date
First Exam	20%	
Second Exam	20%	
Assignments (Reports /Quizzes/ Seminar / Tutorials)	10%	
Final Examination	50%	

