Course Name: Software Engineering -2
Course Code: CS352

I. Basic Course Information

Major or minor element of program: Both Major and Minor
Department offering the course: Computer Science Department

Academic level: 300 Level
Semester in which course is offered: Second (spring) semester
Course pre-requisite(s): Software Engineering- 1 CS251

Credit Hours: 3
Contact Hours Through:

Lecture	Tutorial*	Practical*	Total
2.5	0.0	1.5	4.0

* 1.5 hours for **either** Tutorial or Practical

Approval date of course specification: January 2015

II. Overall Aims of Course

- Study how to conduct software system: design, V&V, management, and maintenance.
- Understand four dimensions of system dependability of system dependability: availability, reliability, safety, security.
- Learn about different types of software maintenance.
- Be aware of the processes involved in software evolution, including the process of software re-engineering.
- Understand the differences between agile development methods and the traditional software development methods.
- Understand four dimensions of system dependability of system dependability: availability, reliability, safety, security.
- Learn about different types of software maintenance.
- Be aware of the processes involved in software evolution, including the process of software re-engineering.
- Understand the differences between agile development methods and the traditional software development methods.

III. Program ILOs covered by course

Program Intended Learning Outcomes (By Code)			
Knowledge & Understanding	Intellectual Skills	Professional Skills	General Skills
K2,K3,K9,K18	I4,I5,I6,I18,I19	P6,P9,P10,P14	G2,G4,G8

IV. Intended Learning Outcomes of Course (ILOs)

a. Knowledge and Understanding

On completing the course, students should be able to:

K.1 Define the basic steps of software system development and management.
K.2 Describe the essential elements of software engineering.
K.3 Recognize the Unified Modeling Language basic elements.
K.4 Recognize the basics of modelling and design of computer-based systems

b. Intellectual/Cognitive Skills

On completing the course, students should be able to:

I.1 Make a feasibility study for a specific system.
I.2 Differentiate between different software models.
I.3 Choose the appropriate testing approach for a software application.
I.4 The ability to choose an appropriate design pattern for a certain system design.
I.5 Perform system re-engineering.

c. Practical/Professional Skills

On completing the course, students should be able to:

P.1 Design, test, and evolve a software system.
P.2 Maintain software systems.
P.3 Develop a comprehensive project plan for a specific development effort.
P.4 Apply a wide variety of testing techniques in an effective and efficient manner.
P.5 Conduct reviews and inspections.

d. General and Transferable Skills

On completing the course, students should be able to:

G.1 Demonstrate ability in time management, organization skills, communication skills, report writing skills, and presentation skills for a variety of audiences
G.2 Demonstrate ability to work as a team member
G.3 Demonstrate an appreciation and ability to continue professional development and ensure life-long self-learning

V. Course Matrix Contents

	Main Topics / Chapters	Duration (Weeks)	Course ILOs Covered by Topic (By ILO Code)			
			K & U	I.S.	P.S.	G.S.
1-	Software Quality management	2	K1	I1,I2	P5	
2-	Software Design and implementation	3	K2,k3,K4	I4,I5	P1,P4	G1,G2
3-	Software Testing	2		I3	P1,P4	
4-	Software Critical System	3	K4	I3	P1,P3	
5-	Software Evolution	2			P1,P2,P3	G2,G3
	Net Teaching Weeks	**13**				

VI. Course Weekly Detailed Topics / hours / ILOs

Week No.	Sub-Topics	Total Hours	Contact Hours	
			Theoretical Hours	Practical Hours*
1	Code Inspections and Reviews	2.5	2.5	
2	Verification and Validation techniques	4	2.5	1.5
3	Design patterns	4	2.5	1.5
4	Design patterns	4	2.5	1.5
5	Creating UML diagrams which model aspects of the domain and software architecture	4	2.5	1.5
6	Testing Techniques	4	2.5	1.5
7	**Midterm Exam**			
8	Detecting testing coverage	4	2.5	1.5
9	Critical Systems	4	2.5	1.5
10	Testing critical systems	4	2.5	1.5
11	Testing critical systems	4	2.5	1.5
12	Rapid Software development	4	2.5	1.5
13	Software Evolution	4	2.5	1.5
14	Software Evolution	4	2.5	
15	**Final Exam**			
Total Teaching Hours		**51**	**33**	**18**

* **No Practical/Tutorial during the first week of the semester**

VII. Teaching and Learning Methods

Teaching/Learning Method	Selected Method	Course ILOs Covered by Method (By ILO Code)			
		K & U	Intellectual Skills	Professional Skills	General Skills
Lectures & Seminars	Y	K1,K2,K4	I1,I2,I5	P2	G2
Tutorials					
Computer lab Sessions					
Practical lab Work	Y	K3	I3,I4	P1,P3,P4	
Reading Materials					
Web-site Searches					
Research & Reporting					
Problem Solving / Problem-based Learning					
Projects					
Independent Work	Y		I5		G3
Group Work	Y		I4,I5	P5	G1
Case Studies					
Presentations					
Simulation Analysis					
Others (Specify):					

VIII. Assessment Methods, Schedule and Grade Distribution

Assessment Method	Selected Method	Course ILOs Covered by Method (By ILO Code)				Assessment Weight / Percentage	Week No.
		K & U	I.S.	P.S.	G.S.		
Midterm Exam	Y	All	All	P1,P2,P3	All	20%	7
Final Exam	Y	All	All	P1,P2,P3	All	60%	15
Quizzes							
Course Work	Y	All	All	P1,P2,P3,P4	All	5%	Every week
Report Writing							
Case Study Analysis	Y				All		
Oral Presentations							
Practical	Y		I4	P1,P3,P5,P4	All	5%	11
Group Project							
Individual Project	Y	All	All	P1,P2,P3	All	10%	12
Others (Specify):							

IX. List of References

Essential Text Books	• Software Engineering, Ian Somerville, 7th edition, 2004
Course notes	• None
Recommended books	• Software Engineering, Roger Pressman, 6th edition, 2005
Periodicals, Web sites, etc …	• None

X. Facilities required for teaching and learning

- None

Course coordinator: Dr. Mohamed El Ramly

Head of Department: Prof. Abeer El Korany

Date: January 2015