King Saud University
College of Computer and Information Sciences
Department of Software Engineering

SWE 312 – SOFTWARE REQUIREMENTS ENGINEERING 3(3-0-1)

Required Course: 3 hours lecture and 1 hour lab per week

Course Instructor and/or Coordinator: Dr. Mohammad Anwar Hossain

Course Description (catalog): The course covers requirements engineering in depth including the followings: Requirements Engineering Process - Elicitation of requirements - Functional and non functional requirements - System services and constraints – Quality of Requirements - Requirements traceability matrix - Metrics for non-functional requirements - Use case description - Use case and context diagrams - Software Requirements Specification - IEEE Standard - Requirements for agile developments - Requirements for various systems: embedded systems, web-based systems, business systems, etc. – Requirements management. Ethical behavior of software analysts with respect to stakeholders when gathering the requirements will be also discussed. This course gives also a brief introduction to formal specifications using specification languages such as Z or B. Students participate in a group project on software requirements analysis and specification and requirements management case tools.

Prerequisite(s): SWE 211 – Introduction to Software Engineering **Co-requisite(s):** None

Textbook(s) and/or Other Supplementary Materials:

Primary:

1. Karl Wiegers and Joy Beatty: Software Requirements (3rd Edition) (Developer Best Practices). Microsoft Press, 2013.
2. Bernd Bruegge and Allen H. Dutoit. Object-Oriented Software Engineering Using UML, Patterns, and Java. 3rd Edition, Prentice Hall, 2010.

Supplementary:

1. Axel van Lamsweerde: Requirements Engineering, from System Goals to UML Models to Software Specifications. John Wiley, 2007.
2. Peter Zielczynski, Requirements Management Using IBM Rational RequisitePro. IBM Press, 2008.

Major Topics Covered:

Topic	# Weeks	# Contact hours*
Review of SDLC and Methodologies	1	4
Overview of Project planning and link with requirements management	1	4
Introduction to Software Requirements Engineering and Management; Ethical behavior of software engineers	1	4
Analyzing the problem	0.75	3
Understanding of user and stakeholder needs	1	4
Defining and specifying the system: A use case approach	2	8
Managing the scope; Change management, etc.	0.75	3
Refining the System: Advanced use case modeling; other specification models; Metrics for non-functional requirements	2	8
Building the right system: Compliance with requirements, Traceability, user expectations, conflict management, etc.	0.75	3
Requirements for agile developments; Typical/special systems:	0.75	3

Scientific/intensive-computing systems; etc.		
Overview about formal methods in specification	1.5	6
Concluding remarks, Review, and Evaluation	0.5	2
Total	**13**	**52***

*Contact hours include lectures and labs. Extra eight contact hours will be cancelled and/or used for exams.

Specific Outcomes of Instruction (Course Learning Outcomes):

1. Recognize professional and ethical responsibility of a software engineer, in particular when dealing with customers in order to gather the requirements. [**about 10%**]
2. Recognize the impact of development methodologies on requirements analysis. [**about 7%**]
3. Recognize basics of formal methods in requirements specification [**about 10%**]
4. Discover, elicit, and record user requirements for a software project [**about 30%**]
5. Specify requirements in use cases and other specification techniques [**about 28%**]
6. Recognize the critical aspects of requirements engineering when working in a team [**about 5%**]
7. Use a software case tool to effectively manage and specify requirements. [**about 10%**]

Student Outcomes (SO) Addressed by the Course:

#	Outcome Description	Contribution
	General Engineering Student Outcomes	
(a)	an ability to apply knowledge of mathematics, science, and engineering	
(b)	an ability to design and conduct experiments, as well as to analyze and interpret data	
(c)	an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability	
(d)	an ability to function on multidisciplinary teams	L
(e)	an ability to identify, formulate, and solve engineering problems	M
(f)	an understanding of professional and ethical responsibility	L
(g)	an ability to communicate effectively	L
(h)	the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context	
(i)	a recognition of the need for, and an ability to engage in life-long learning	
(j)	a knowledge of contemporary issues	
(k)	an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice	M
	Specific Software Engineering Student Outcomes	
(l)	the ability to analyze, design, verify, validate, implement, apply, and maintain software systems	H
(m)	the ability to appropriately apply discrete mathematics, probability and statistics, and relevant topics in computer science and supporting disciplines to complex software systems	
(n)	the ability to work in one or more significant application domains	
(o)	the ability to manage the development of software systems	

H=High, **M**= Medium, **L**=Low

ASSESSMENT

Assessment will be distributed as follows:

One midterm exam	**15 %**
Two assignments	**10 %**
Quizzes (2 Quizzes)	**10 %**
Project – demonstration at lab	**20 %**
Participation/interactivity	**5%**
Final Exam	**40 %**
Total	**100**

Grades will be distributed based on the total obtained by each student as follows:

Total	**Grade**
>= 95%	**A+**
< 95% and >= 90%	**A**
< 90% and >= 85%	**B+**
< 85% and >= 80%	**B**
< 80% and >= 75%	**C+**
< 75% and >= 70%	**C**
< 70% and >= 65%	**D+**
< 65% and >= 60%	**D**
< 60% and >= 55%	**F**
< 55% and >= 50%	**F**
< 50%	**F**

LATE SUBMISSIONS AND ABSENCES

- Attendance is noted 5 minutes after the start of the class. Any student attending after will be counted as absent.
- **Only a medical note is accepted as an excuse for an absence. The medical note must be given to the instructor within a week after the missed class**.
- If a student does not deliver an assignment/project at the beginning of the class at the day of submission deadline, the student loses 25% of the full mark. The assignment can still be accepted before 5 pm the next business day.
- No excuse is accepted for late assignments/project.
- **Every student must obtain 50% of the project + lab marks combined in order to pass this course**
- Plagiarism is not tolerated and will result immediately in failing the course (e.g. copying from the work of other students, cheating in the exam, and missing proper acknowledgement and referencing)