Arab Academy for Science and Technology and Maritime Transport
Software Engineering Curriculum
Course Syllabus

Course Code: SE392	**Course Title:** Software Requirements and Specifications	**Classification: E**	**Coordinator's Name:** Prof. Dr. Mohamed Shaheen	**Credit Hours:** 3
Pre-requisites: SE291 (Introduction to Software Engineering)	**Co-requisites:** None	**Schedule:** Lecture: 2 hours Tutorial-Lab: 2 hours		

Office Hours: (Room 308)
Wednesday 12:30 p.m. -2:30 p.m.

Course Description:
This course helps to improve the way any organization elicits, analyzes documents, validates, and manages its software requirements. It provides students with a tool kit of "good practices" to improve the requirements processes. Topics covered include Data gathering techniques, The software requirements document, User requirements definition, System requirements specification, Functional and non-functional requirements, Requirements engineering processes, Process modeling, Critical systems specification, Architectural Design, Object Oriented Design using UML (Unified modeling language UML: Class diagram, Use-Case diagram, Sequence diagram, and State diagram), Software Verification and Validation, Software Testing and Maintenance, CASE tools for software requirements and specifications.
.

Textbook:

Karl E Wiegers and Joy Beatty. 2013. Software Requirements 3. Microsoft Press, Redmond, WA, USA.

References:
J.A. Hoffer, J.F. George, *Modern Systems Analysis & Design*, Prentice Hall, 2002.

Roger Pressman, *Software Engineering: A Practitioner Approach*, McGraw-Hill 2004.

Course Objective/Course Learning Outcome: Upon completion of this course, students should be able to:	**Contribution to Program Student Outcomes:**
1. Design and conduct interviews, questionnaires, observations and documents investigation,	(CS-SO-3) Communicate effectively in a variety of professional contexts. (IS-SO-3) Communicate effectively in a variety of professional contexts.

<table>
<tr><td>2. Develop a software requirement document,
3. Understand the concepts of user requirements and system requirements,
4. Understand the differences between functional and non-functional requirements,
5. Develop a system process model
6. Develop a system data model,
7. Model a system object model using the unified modeling language UML,
8. Understand the requirements engineering processes and requirements validation,</td><td>(CS-SO-6) Apply computer science theory and software development fundamentals to produce computing-based solutions.
(IS-SO-6) Support the delivery, use, and management of information systems</td></tr>
<tr><td>9. Perform critical systems specification,
10. Use CASE tools for software requirements and specifications</td><td>(IS-SO-6) Support the delivery, use, and management of information systems</td></tr>
<tr><td>Course Outline:

1. The essential software requirement
2. Requirements from the customer’s perspective
3. Good practices for requirements engineering
4. The business analyst
5. Establishing the business requirements
6. Requirements elicitation
7. 7th Week Exam</td><td>8. Understanding user requirements
9. Documenting the requirements
10. A picture is worth 1024 words
11. Beyond functionality
12. Validating the requirements
13. Requirements management practices
14. Links in the requirements chain
15. Project Presentations
16. Final Exam</td></tr>
<tr><td colspan="2">Grade Distribution:

7th Week Assessment (30%):
Exam (25%) + Presentation 5%

12th Week Assessment (20%):
Project (20%)

Year Work (10%):
Quizzes (5%) + Homework Assignments (5%)

Final Exam (40%)</td></tr>
</table>

Policies:

Attendance:
AASTMT Education and Study Regulations (available at aast.edu)

Academic Honesty:
AASTMT Education and Study Regulations (available at aast.edu)

Late Submission:
Late submissions are graded out of 75% (1 week late), 50% (2 weeks late), 25% (3 weeks late), 0% (more than 3 weeks late)