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

SWE 333 – SOFTWARE QUALITY ASSURANCE 2 (2-0-1)

Required Course: 2 hours lecture and 1 hour tutorial per week

Course Instructor and/or Coordinator: Dr. Khalid Alnafjan (Kalnafjan@ksu.edu.sa)
Course Description (catalog): This course introduces fundamental concepts related to Quality Assurance and Measurements and Metrics in the software industry: Measurements of product, process and resource attributes – Planning a measurements program - Goal/Question/Metric - Collection and analysis of software empirical measurements - Building software metrics - Quality concepts – Software quality assurance - Software quality management - Quality planning and control – Quality manual – Product and process standards - Internal and external software quality attributes - Software reviews, walkthrough and inspection – Statistical software quality assurance – Software configuration management - Software reliability – International Software quality models, e.g. ISO 9000 Quality standards and ISO 9000-3, etc.. – Software process improvement – The Capability Maturity Model (CMM), balanced scorecards. Ethical responsibility to produce high-quality software is also discussed. Students participate in a group project on Software quality assurance.

Prerequisite(s): SWE 312 – Software Requirements Engineering
Co-requisite(s): None

Textbook(s) and/or Other Supplementary Materials:
Primary:
- D. Galin (2004): Software Quality Assurance: From Theory to Implementation", Pearson Education.

Supplementary:
- K. Naik and P. Tripathy (2008): Software Testing and Quality Assurance. John Wiley, 2008.

Major Topics Covered:

Topic	# Weeks	# Contact hours
Introduction and definitions	1.5	3
Components of software quality assurance	2	4
Software quality planning and control	1.5	3
Software quality metrics	2	4
Costs of Software Quality	1	2
International software quality models	2	4
Capability maturity model (CMM, CMMI)	2	4
Personal software process and team software process	2	4
Concluding remarks, review, and evaluation	1	2
Total	**15**	**30**

Specific Outcomes of Instruction (Course Learning Outcomes):
1. Understand the importance of quality in software development.
2. Learn how to develop systems with predictable and improved cost, schedule, and quality.
3. Understand the difference between quality control and software quality assurance.
4. Understand the quality management process.
5. Use and develop software metrics to measure the quality of software.
6. Implement SQA methods and techniques.
7. Understand and apply SQA international Standards.

8. Understand and apply personal and team software processes

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	L
(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	H
(d)	an ability to function on multidisciplinary teams	
(e)	an ability to identify, formulate, and solve engineering problems	
(f)	an understanding of professional and ethical responsibility	
(g)	an ability to communicate effectively	
(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	M

H=High, M= Medium, L=Low

COURSE INSTRUCTIONS AND POLICIES

- A sharp deadline for project and homeworks will be applied. Late submition will not be accepted

- Where a group of students work together on a task or subtask, all should be named on the work products. You should usually be listed in alphabetical order, but team leaders can be identified specifically.

- Except under special circumstances (serious illness, travel to a conference, etc.), there will be no chance for replacement quizzes and exams to absent students.

GRADING

MidTerm(s)	30%
Final exam	40%
Quizzes	15%
Homework and Projects	15%