School of Information and Physical Sciences

SENG4430 – Software Quality

Assessment 1: Group Report (30%)

Due Date: 11:59 pm on 6 Jun (Week 13)

Design Problem

The following is the design problem that each group will undertake:

Computer systems and their supporting software are embedded in all aspect of life in the modern world. Expectation of access have changed to expectations of dependability across security, safety, availability, maintainability, and quality user experiences. To meet these demands, software development processes have also evolved with more agile approaches being the norm and a software lifecycle supporting ongoing requirements change, interdisciplinary teams, asynchronous development, and regular software updates.

Managing software quality in such an environment is challenging. However, tool support can considerably reduce the risks of poor software delivery and poor software quality management by automating common tasks and providing evidence for decision making in terms of resources allocation.

You are part of a small software engineering start-up, and your company has been approached to develop a large software quality support tool. To win the contract, the client would like you to demonstrate your expertise in software engineering and software quality assurance. You will develop a scalable software quality tool prototype for a predetermined industry organiztion and produce the supporting "quality case" for this prototype.

The scope and aspects of software quality to be considered are open but must, at a minimum, include **usability, reliability, security,** and **maintainability**. System scope, prototype scalability and software quality reporting all need to be justified in a formal design document. A mapping from this formal design to the final deliverable will determine the grades for the project deliverables. The client has also specified that the tool will need to be in **Java**, that you will need to identify your own datasets to test your tool and that the **tool can evaluate itself**, i.e., the software tool you develop must be able to test itself for software quality.

Project Organisation

Your parent company has split you into 5-6 person teams with initial allocation done at the start of the project (Week 1 workshop). Within the company structure, you can swap between teams, but you must find someone to swap with. Company management will not broker trades between teams. Teams are fixed at the time of the first progress meeting (Week 2 workshop).

You will use an agile software development methodology and work towards weekly sprints. Sprints will be reviewed across the project by the client. Both the prototype code and documentation will be developed and maintained in an agile fashion.

The client's company uses IEEE standards and requires that the IEEE Std 1016-2009 (IEEE Standard for Information Technology – Systems Design – Software Design Descriptions) being used as the basis for the system design document. The client is also interested in how your company can work with standardised templates. Thus, they have provided teams with an example Test Plan template from an Enterprise Project Management Methodology (EPMM) for use on this project.

Each team member will identify and take ownership of **two** software quality aspects. They will code and document their aspects throughout the project. Quality of work will be evaluated on an individual basis.

Other aspects of the system, including requirements, system architecture, interface design/implementation, testing design/implementation, evaluation design/implementation will be reviewed at a team level (through peer review). Teams are responsible for producing a meeting agenda for each sprint meeting and taking minutes to document progress and actions.

Deliverables

In this assessment, you are required to prepare a **working prototype** and the following documentations: **system design document, test plan, quality case document, evidence of project management,** and **peer evaluation form**.

Working Prototype

The software quality support tool should be written in Java and at least cover quality characteristics such as usability, reliability, security, and maintainability. It must also be capable of self-testing.

System Design Document

The system design document should conform to the IEEE Std 1016-2009. Detail your design process, algorithm, rational, observations, testing method.

Test Plan

Using the provided EPMM test plan template, document the testing process throughout the entire project development.

Quality Case Document

The quality case document should include example test sets of software used in testing the software quality tool. Evidence of successful quality tests as well as verification and validation should be provided.

Evidence of Project Management

The submission of minutes of meeting is compulsory. Each minutes of meeting should contain attendance, action lists, tasks completed, major project decisions and a commentary on group dynamics. Additional evidence should also be included, but not limited to project action plan, Gantt charts, Kanban, etc.

Peer Evaluation Form

Every team member should submit a peer evaluation form to fairly judge the team dynamics. In case all members have a general consensus on equal contribution, a peer evaluation form may be omitted.

You should hand in your code as well as all documentations to Canvas by **11:59 pm on 6 Jun (Week 13)**. All documentations (except peer evaluation form) should be submitted in PDF format and one copy per group is sufficient.

The marking rubric for this assignment is available with this assignment specification on Canvas. Note that there is no target page length for all documentations. Quality, criteria coverage, and team effort are better than quantity. Be sure that you address each of the assessment criteria.

Late Submission and Academic Integrity

1. Assignments submitted after the deadline will be penalized by 10% per day late (including weekend). For example, an assignment worth 20% marked at 78% may be graded as:
   - On time: 0.78 * 1.00 * 20 = final mark = 15.6
   - 1 day late: 0.78 * 0.90 * 20 = final mark = 14.04
   - 3 days late: 0.78 * 0.70 * 20 = final mark = 10.92
   - 5 days late: 0.78 * 0.50 * 20 = final mark = 7.8
2. The assignment report will be checked for plagiarism. A plagiarized assignment will receive a zero score and be reported to SACO.
3. As per the Student Conduct Rule, any work submitted for assessment must be your own original work, and as such, the use of AI LLMs such as Microsoft Copilot, ChatGPT and other similar tools **cannot** be used in the writing or drafting of any work submitted for assessment. By breaching this rule, you will be at risk of being reported to SACO and receive a zero score. The use of AI LLMs is **only acceptable** as an aid in helping you understand the assessment requirements, but be wary about errors and hallucinations.