2017-2018 Software Engineering (M) - Exam Feedback

This general feedback about the SE (M) exam should give you some insights into how the class performed in the exam as a whole. Overall, the average score was relatively high, with a large number of students achieving A grades. The rationales and justifications provided showed a good level of understanding and reflection on the software process. It was good to see that the class achieved a high level of performance on this exam and were able to show a good knowledge of software engineering.

Learning Outcomes

1. Develop clear, concise, and sufficiently formal documentation of extensions to an existing system, based on the true needs of users and other stakeholders;	Coursework only
2. Apply design principles and patterns while designing and implementing simple systems, based on reusable technology;	Q1,Q2,Q3b
3. Produce documentation appropriate for programs developed in practical exercises;	Coursework Only
4. Create UML class diagrams which model aspects of the domain and the software architecture;	Q1a,Q1c, Coursework
5. Carry out testing of programs and apply simple measurement techniques to software;	Q3b, Q3c,Q3d
6. Discuss the breadth of software engineering practices.	Q2a,Q3a

Question 1: Modelli

Most students did w ent get full or close to full marks for them. Some students failed to emphasise d to illegal modifications. Most students get full or close to full m e code for the Technician class.

Question 2: Design Principles

Most student generally did well on this question. Most students provide well formed arguments about the cohesion although a few students misunderstood the cohesion type considered (i.e. functional, not sequential or otherwise or did not provide any arguments to support their conclusion. Most students correctly identified examples of five types of coupling (Q2c), and generally provided good suggestions for how to correct two of them. Minor reductions were applied for not fully outlining the characteristics of the individual coupling or not specifically mentioning how it could be reduced in this case (with reference to the code). A subset of students confused data, stamp and content coupling.

Question 3: System Architecture and Testing

Students generally did well in explaining “test-driven development” (Q1). There were some uncertainties with the system architecture question; the presented diagram represents a basic multilayer architecture (with an association); while the system could be implemented using specific versions of MVP or client-server these system architecture patterns were not implied in the design (e.g. the TCP/IP communication was not indicated); partly credit was awarded depending on the validation of the arguments.

Some students struggled with questions c). Some failed to identify meaningful equivalent classes for the test. Some wrote test code that are disconnected with the provided UML and method signature. Most students correctly identify that the test should be a black box test.