Comments  Please see the attached report.
COMP33812 Examination Performance Feedback

2015–2016

The fact that majority of students chose not to attend lectures was very evident in answers given in the examination.

Question 1
a) The question asks why change must occur. Thus, of the four types of software change, preventive (refactoring) is not valid. A number of answers included preventive. The fact that the question asks for outline of the reasons, means that more than the name of a type of change is required.
b) Question is looking for four reasonable points (four marks). Many answers did not cover four points. Some answers gave variants on the same points rather than distinct points.
c) Question was looking for three sources such as users, test cases and logs. Many answers identified three such sources. The description says that there is no documentation and implies a small (single) development team. Thus, answers suggesting using design diagrams, or asking other members of the development team or testers, show a lack of appreciation of the scenario. The question explicitly excludes the original developer or source code, yet a number of answers suggested using these.
d) An open-ended question for which answers giving any conventions would be acceptable. However, the question explicitly asks for a statement of why the conventions are helpful, many of the answers failed to address this part of the question.
e) Many answers gave a good post-dependency table. However, a significant number failed to add the START and STOP nodes and include them in the dependencies.

Question 2
a) Question is asking for reasons why the size of software changes not reasons why software is changed. Answers needed to describe why a type of change could cause the size of software to change. Many answers lacked this reason element. Notably, adding new features will also certainly change the software size, there is no reason why a bug-fix should increase the size of software.
b) Question asks what a code smell is and why it is important. Generally, what a code smell is was successfully answered. The reason why there are important was less successfully answered. In particular, although a code smell can result in refactoring; refactoring is not the reason that they are important. A code smell indicates code that is difficult to understand.
c) Question was asking for identification of code smells and how to address them. Generally three types of code smell were identified. Some answers, gave variants of the same type of smell. The weaker part of answers was how to address the ode smell described.
d) Reasons why refactoring has not been historically done include costs time and money, no direct financial benefit to company and risk of breaking working system. Refactoring is now done because code review is part of modern development practices. Generally answers lack some depth to reasons.
e) i) An open-ended question looking for characteristics/questions. Generally plausible characteristics/questions were identified. However, asks for these to be set in the context of generally programming experience. This part of the question was less successful done.

Question 3
a) Question was looking for assessment of options and justification for a recommendation. The quality of answers varied. Answers that lacked something, for example, failed to appreciate the long-term cost of an
option. A significant number of answers failed to jive a recommendation or a justification for a recommendation.

b) A number of answers just migrated data from the convenience store system to the supermarket system. Before this can happen, the existing supermarket system must be updated to allow this data to be migrated.

c) Again the main problem encountered in answers was not altering the supermarket system to include the on-line operation of the convenience stores. There was also a lack of justification for the proposed architecture.

d) Many answers failed to clearly indicate the order in which elements in the architecture would be migrated or failed to appreciate business value in the proposed order.

**Question 4**

a) Answers were generally good. However, some lacked some depth to their consideration. A number of recommendations failed to appreciate the absolute need for the business to have a system that was guaranteed to work from day one.

b) Similar issues to the previous part. Some answers lacked depth and others failed to ensure that the business could be successfully run.

c) An open-ended question, many answers failed to give depth to their points and, in particular, failed to mention separation of concerns.