Strathmore
UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

END OF SEMESTER EXAMINATION

ICS 2201: SOFTWARE ENGINEERING

DATE: 21 November 2018 Time: 2 Hours

Instructions

1. This examination consists of **FIVE** questions.
2. Answer **Question ONE (COMPULSORY)** and any other **TWO** questions.

QUESTION ONE (30 marks)

PART A

Read the description below and answer the questions that follow:

The software described here is software that is used in the controller of various digital cameras. The software should be able to provide different levels of functionality depending on the price segment of the camera and the software should be able to be used for various kinds of hardware configurations (buttons, screen, data storage and optical components). The digital camera consist of these hardware components:

· Controller (CPU, memory): managing the other components, provide interface with the user etc.

· Controller buttons (vary from camera to camera). Typically buttons for on/off, flash, take picture, menu, navigation, zoom etc.

· Digital screen (can vary from camera to camera in size, colour depth etc).

· Permanent data storage (typically flash-memory, memory stick, SD-cards etc).

· Optical component with an interface to control zoom, focus, etc....

Here is a list typical functionality provided by the camera:

· Turn on/off camera.

· User controlled optical functionality (zoom out, zoom in, flash).

· Camera controlled optical functionality (auto focus, lens opening).

· Power save functionality (shut down camera if not used).

· Storing, retrieve and delete pictures.

· Processing images (rotate, enhance).

· Display pictures, information to the user on the camera's screen.

· Camera set up (storage options, GUI-options, language options etc.)

Case study adapted from: The Requirements Engineering Handbook by Ralph R. Young

a. Identify the **three** most important quality attributes of the digital camera system described. Explain why each of the attributes identified is important for the system **(6 marks)**

b. **Create** architecture views of the digital camera system. The architecture must be described in two views according to the 4+1 view model:

I) Scenario view **(5 marks)**

II) Logical view **(4 marks)**

PART B

c. Explain the difference between verification and validation. Give an example of a technique that can be used for each. **(4 marks)**

d. You are hired by Strathmore University to create for them an online voting system for their student elections, to be able to automate their traditional manual voting process. What are the key steps that you would follow to gather the requirements of the system. Explain what you would do at each step. **(5 marks)**

e. Suppose you are managing a project which is getting behind schedule, possible actions include:

   - Renegotiating the time schedule
   - Adding people to the project
   - Renegotiating the software quality requirements

   i) In which ways can these actions help you finish the project on time? **(3 marks)**

   ii) Which action would you prefer and why? **(1 mark)**

f. Consider the following scenario. Angel and Beatrice are working a software system and are using git, a distributed version control system to manage their code. Beatrice pulls from the master repository and updates her working copy. She realizes there is a bug in the code and works to fix it. When she is finished, she commits her code. The next day, Angel begins working on the code by pulling from the master repository and updating his working copy. He too realizes the code has a bug (the same one Beatrice saw) and spends the whole day fixing it. What went wrong? Why didn't Angel benefit from Beatrice's work? **(2 marks)**

Question TWO (15 marks)

Read the case study and answer the questions that follow

The customer, a military organization, handed the contractor a shelf-load of rules and regulations (“regs”) saying, “These are the requirements.” The programmers, all on-site employees of a contractor, were ready to start, and they did. Representatives of the military organization were aware of the project, but they did not participate in it until it was time to review the finished code. While the code was being written, the contractor undertook to convert the regs to a set of shall statements. This was faithfully and painstakingly done, but as the code emerged, it was found that the verification of the “shall” statements—matching them to parts of the different code modules—was virtually impossible. They simply did not map. Another complication encountered was a complete breakdown in communication between the contractor and the subcontractor producing the code. It wasn’t that they did not understand each other; they simply did not communicate. The subcontractor viewed any inquiries as interference, and in an atmosphere of hostility, communication simply died. As the code came into review by the customer, the military representatives were heard to say again and again, “No, that’s what we do, but that’s not how we do it.”

And as the modules came into test by the contractor, they failed repeatedly. The right things had been written the wrong way, and they did not work. After months of struggle, the first of about 20 modules was almost ready for release. It was still a little shaky, and lots of people were unhappy with it, but it was close to acceptance from the contractor’s perspective. However, the customer gave up because relationships between the two parties had become broken during the code development period. Not only was the process broken, but also the platform and the operating system were outdated and inadequate. Three million lines of code were abandoned, the hardware was scrapped, and the whole project was started all over again.

Case study adapted from: The Requirements Engineering Handbook by Ralph R. Young

Answer the following questions from the case study given above:

a. From the case study, identify the two key stakeholders of the project. What was the interest of each of the identified stakeholders to the project? **(2 marks)**

b. Identify and explain 5 key factors that led to the failure of the project. For each factor suggest what could have been done better **(5 marks)**

c. Identify and explain 5 key steps that would have been followed at the Requirements Inception Stage , to analyse the key problem that was to be solved by the system **(5 marks)**

d. Identify 3 elicitation techniques that could have been used to gather detailed requirements. For each technique identify the most suitable stakeholder and explain why the technique would be suitable for them. **(3 marks)**

Question THREE (15 marks)

a. How can different stakeholders of a software system influence the software architecture? Give examples for Developer's organizations, Marketing, End-user, Maintainer, and Customer **(5 marks)**

b. Layered architectures are designed to reduce coupling between components of a software system. Why is this reduced coupling useful? Describe a typical layered architecture, and explain the role of each of the layers. **(5 marks)**

c. Indicate the most appropriate architectures for the following short descriptions of systems:

i) An application for cooking recipes that can run on various mobile applications with various screen configurations (in size) and input devices (keys, touch-based screens, joysticks, etc). **(1 mark)**

ii) A game engine that provides a high-level API to the programmer. The programmer can also access medium-level and low-level APIs to get a richer set of functionality if required. **(1 mark)**

iii) A collaborative application for sharing various information stored in a common database (repository). **(1 mark)**

iv) A distributed application for sharing information across networks. **(1 mark)**

v) A PC application for analysing weather data through a set of data transformations **(1 mark)**

Question FOUR (15 marks)

a. Identify 3 goals of a software test **(3 marks)**

b. Give two advantages of writing tests before writing code **(2 marks)**

c. What are the differences between “White Box Testing” and “Black Box Testing”? **(2 marks)**

d. Give 3 distinct benefits of performing incremental code reviews **(3 marks)**

e. A program is seeded with 25faults. During testing, 18 faults are detected, 13 of which are seeded faults and 5 of which are indigenous faults. What is Mill’s estimate of the number of indigenous faults remaining undetected in the program? **(3marks)**

f. Give two characteristics of well written code **(2 marks)**

Question FIVE (15 marks)

a. Briefly discuss the difference between quality control and quality assurance. **(2 marks)**

b. “Software maintenance” is a bit of a misnomer since software does not wear out nor require lubrication/adjustment. State three tasks that are part of software maintenance. **(3 marks)**

c. Identify and explain three key challenges faced during maintenance of a system **(3 marks)**

a. The developer's view of user-friendliness is likely to be determined by technical properties of the interface: use of windows, buttons, scrollbars, pop-up menus, etc. The developer is inclined to look at user-friendliness from the inside. On the other hand, the user's main concern is to get his job done in the most effective way. User interfaces should not be friendly; they should effectively support the user at work. What are the four important ways to measuring the usability of a system? **(4 marks)**

d. Identify and explain three general techniques used in software maintenance **(3 marks)**