Appendix for Q1 and Q2 attached

Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Software Evolution

Date: Thursday 3rd June 2010
Time: 09.45 – 11.45

Please answer any THREE questions from the FOUR questions provided

This is a CLOSED book examination

The use of electronic calculators is NOT permitted.
1. a) Researchers have identified four different types of software change: corrective, adaptive, perfective and preventive. State which types of change are required in each of the following scenarios, giving a brief justification for your answer in each case.

i) After installation of a major new release of the marketing software for your company, customers have started to complain that they are receiving marketing material sent to their address with incorrect names and regarding upgrades to products that they do not own.

ii) The company which supports the PIPA operating system on which your software runs has recently announced that it will no longer be providing technical support for PIPA.

iii) At present, your company’s e-commerce software allows customers to browse items in the online catalogue, but does not allow customers to pay for their items online too. Your manager makes a strategic decision that this functionality must be included in the next release.

b) Draw a data dependency graph for code fragment A (given in the Appendix at the end of the examination paper).

c) Give all control dependencies for code fragment A (given in the Appendix at the end of the examination paper). You should include transitive dependencies explicitly. You may either draw a control dependency graph or (quicker) just list the dependencies you find in the form $S_1 \rightarrow S_2$ (meaning statement $S_1$ is control dependent on statement $S_2$).

d) Control flow graphs provide a way of assessing code quality in an objective manner, since they can highlight the presence of “spaghetti” code (i.e., code in which the structure is highly complex and woven together, usually through inappropriate use of goto statements). Spaghetti code shows up in a control flow graph when edges cross over one another, regardless of how the nodes are placed. Data flow graphs cannot show the presence of spaghetti code, but they can indicate other software quality issues. Describe one such quality issue, and point out the features of a DFD which occur when software suffers from this issue.
2: a) Suppose that you have been using an open source programming tool intensively for some months, and that you have got to know its operation pretty well. You have an idea for a new feature that would be extremely useful. You decide to join the open source project, in order to work on the implementation of this feature. However, the source code is very large and you do not at present know much about the organisation of the code, or where this feature should be implemented.

What code reading strategy would you adopt for your first encounters with the source code of the system you are interested in contributing to? Justify your answer. (3 marks)

b) Which of the software idioms covered in the lectures are present in code fragment B (given in the Appendix at the end of the examination paper). For each instance of an idiom you find, state the name used for the idiom in lectures, the lines of code in which it occurs, and the main components of the idiom as they appear in the code (e.g. say what the collection operated on is, if you spot one of the idioms that process a collection). (10 marks)

c) Based on the idioms you mentioned in your answer to part b) of this question, formulate a hypothesis as to what this function is supposed to achieve. Note any questions or conjectures you formed in order to reach your conclusion. (3 marks)

d) Examine code fragment A (given in the Appendix at the end of the examination paper). It contains an example of another standard idiom, one which we did not cover in the lectures for this course. Point out this idiom, giving a high-level description of it and indicating clearly the lines of code involved in its implementation. (Note: answers which merely describe what this specific piece of code is doing will score no marks. You must describe the general case idiom, i.e. the generic code pattern that this specific piece of code implements, to earn marks.) (4 marks)
3. Mud Unlimited is a national chain of gardening stores and plant nurseries. Traditionally, their sales have been through brick-and-mortar stores, although they have always maintained a small mail-order business supplying specialist seeds and equipment to expert growers around the country. In recent years, the rapid increase in the number of active allotments (plus other factors) has led to an unforeseen growth in the requests for mail order products. Mud Unlimited were unable to cope with the demand and were turning customers away.

A bright young executive at Mud Unlimited spotted an opportunity when it became known that InterFlower, a north of England/Scotland-based flower delivery service, was in financial trouble and looking for a way to survive. With its extensive network of trained delivery staff, it seemed to be the perfect partner for Mud Unlimited’s planned move into e-commerce.

InterFlower was purchased by Mud Unlimited. The next step is to examine the computer systems of both companies and to re-engineer them to create a merged system offering the best of both company’s abilities. The new system must support the existing Mud Unlimited brick-and-mortar stores, while adding a new revenue stream consisting of sales direct to the public through the Web. InterFlower’s sales are currently made exclusively over the telephone. This will continue in the short term, but Mud Unlimited plan to gradually phase this revenue stream out, as customers are converted to the new online sales route. The delivery service will now apply to sales from all revenue streams: within stores, online and (in the short term) over the telephone.

In addition, the target computer system must also provide new planning and strategic management functionality, since the new combined business is too large and complex for the ad hoc family-based management style in place in Mud Unlimited to date. Software support is urgently needed if the management are to quickly understand their new business and to make appropriate purchasing, stocking and marketing decisions in both the short and long term.

a) You have been put in charge of the team responsible for planning and undertaking this re-engineering project. Your task is to bring the new e-commerce and planning aspects of the business online while allowing the store-based sales of both businesses to continue unhindered. Given the mission-critical nature of the software systems, you decide to use the Chicken Little re-engineering methodology. As a first step, therefore, you ask your team to evaluate the architecture of the current computer systems. The result is shown below.

(Question 3 continues on the following page)
How would you characterise these architectures in terms of their decomposability? Briefly explain your answer in each case. (3 marks)
b) As the next step, you ask your team to produce an architecture for the target system. They produce the following design:

As can be seen from the diagram, the new system will operate over two databases, one storing the more static catalogue of products sold by the merged company and the other storing the rapidly changing (and growing) data on individual sales and their associated deliveries. The new Sales and Deliveries must combine information about store sales and online sales, whereas the original Mud Unltd database only stored the former.
and the original Interflower DB only stored the latter. The only information about deliveries stored by MudUnltd DB is the destination address, while InterFlower DB stores an entire delivery plan, including stages in the journey of the goods and the staff responsible for enacting those delivery stages in real life.

Above the level of the databases, new functionality must be provided for a new class of users: the management team. This functionality consists of analyses and reports over the sales data and delivery tracking data. As regards sales, the combined catalogue of products from both companies (gardening equipment and plants from Mud Unlimited, flower arrangements and accessories from InterFlower) must be available for sale through stores and online. (The telephone interface is not shown, as it is assumed that this will be phased out soon.) The presentation logic differs for each type of user, but the core business logic is the same for all sales routes (since each product has its own associated set of business rules).

Based on these considerations, discuss the suitability of each of the three migration strategies (i.e. forward migration, reverse migration and general migration) for this re-engineering project. State clearly which is your preferred strategy based on your analysis of the business case for each one. (7 marks)

c) Using your preferred migration strategy from part b) of this question, design a migration path for the system that brings the most important functionality on-line soonest, while maintaining existing mission-critical functionality as necessary. For each migration step, you should:

- Draw a diagram of the architecture of the system at that point in time, showing clearly any gateways in place, and distinguishing legacy from target components and reverse from forward gateways. (If the change is small, you can save time by drawing only the affected part of the architecture, provided that the details of the migration step are clear.)

- Justify your choice of component to migrate in terms of the benefits, costs and risks of the step. Reference must be made to all three aspects of the choice for full marks to be awarded. (10 marks)
4. Some years ago, the government of the Kingdom of Barataria set up a project to create a national health database for the whole country. The goal was for each citizen to have an electronic patient record within the national database that would contain details of allergies, current medication, past medical history and other facts that might be relevant to any future medical treatment.

In anticipation of the delivery of this system, hospitals in and around Manport, the capital city of Barataria, formed a consortium to build a set of applications that could exploit the planned national database. These applications support GPs in prescribing medicines, hospitals in administering drugs to patients and pharmacists in fulfilling prescriptions. A lot of resource was given over to these new applications, and the most important ones were ready a few months after the national patient record database (NPR DB) was supposed to go live.

Unfortunately, the NPR DB project is running very late. It is suffering from all the usual problems of large integrative software projects, with the additional complications of the more stringent security and reliability requirements of a national health application. The Baratarian government has just announced a delay of a further two years before delivery of the first version of the NPR DB.

This left the Manport hospital consortium with a major problem. They have been busy reorganising their own computer systems on the assumption that the government data would be available. Now they face a further two year wait.

Rather than dump their own applications for two years, the consortium decide to pool their own data resources, to try to simulate the presence of the NPR DB. They will create their own shared database, with a schema similar to that being proposed for NPR DB, which will then be populated by regular feeds from the internal databases of each hospital in the consortium. Once sufficient data has been migrated, the new applications can be run on top of it.

Data to be included in the feeds is:

- Diagnoses entered by GPs or consultants
- Drugs prescribed by GPs or consultants
- Hospital procedures performed
- Test results, including from screening clinics and well-being clinics
- Any allergies or bad reactions to medicines already prescribed

a) You have been put in charge of the team who will create this simulation of the NPR DB. One of your first tasks is to determine the frequency with which the feeds into the shared database must be made. What questions would you ask of the appropriate stakeholders in order to discover the optimal frequency for each hospital feed? (4 marks)
(Question 4 continues from the previous page)

b) As well as deciding the frequency and timing of the feed, it is also necessary to select a strategy for locating the Extract, Transform & Load (ETL) work involved. Under what circumstances would a push strategy be most appropriate for this application? How would the circumstances have to be different in order to make a pull strategy the best choice? In each case, you should relate your answer as closely to the given application as possible, rather than talking in general terms. (8 marks)

c) Sometimes, it is not possible to achieve the desired frequency of data refresh, because the technical capabilities of the source or target systems prevent it. Assuming that the key application for the integrated data is to check the safety of drugs being prescribed for patients based on their previous medical history, propose a refresh strategy for the various types of data listed in the introduction to the question, stating which data types you would endeavour to keep most up-to-date and which could be safely allowed to lag behind. Give the justification for your answer, and any assumptions you make about the application domain. (4 marks)

d) Given that the data feeds from multiple hospitals must be integrated, it is extremely likely that inconsistencies in data from different source databases will prevent them from passing correctly through the ETL feed pipeline. Name two data inconsistency problems that can occur when integrating data from independently produced databases, illustrating each one with an example of the kinds of inconsistencies that might occur in this health-based scenario. (4 marks)

Appendix follows
Appendix

The following code fragments are all taken from the source code for the open source Android project (source.android.com). Android is a mobile computing platform, including an operating system, key mobile applications and developer APIs for application programming. See source.android.com/license for licensing information.

Code fragment A, used in Question 1(b), Question 1(c) and Question 2(d)

This code is a part of the stress test suite provided for Android. Minor modifications have been made to the code, in order to make it more readable with the short line length allowed by this page size.

```java
public class CameraStartUp extends InstrumentationTestCase {

    public void testLaunchVideo() throws Exception {
        String individualStartupTime;
        individualStartupTime = "Individual Video " +
            "Startup Time = ";
        long totalStartupTime = 0;
        long startupTime = 0;
        for (int i = 0; i < TOTAL_NUMBER_OF_STARTUP; i++) {
            if (i == 0) {
                // Capture the first startup time individually
                long firstStartUpTime = launchVideo();
                writeToOutputFile(firstStartUpTime, "na", true,
                    "Video");
            } else {
                startupTime = launchVideo();
                totalStartupTime += startupTime;
                individualStartupTime += startupTime + ",";
            }
        }
        Log.v(TAG, "totalStartupTime =" + totalStartupTime);
        writeToOutputFile(totalStartupTime,
            individualStartupTime, false,
            "Video");
    }

    public void testLaunchVideo() throws Exception {
```
Code fragment B, used in Question 2 (b) and (c).

This code is a common library function used by the Android implementation of the Bluetooth protocol. It is written in C. The `strncasecmp` function provides case-insensitive string comparison. It returns 0 (i.e., `FALSE`) if the strings given as arguments are identical and a non-zero number (i.e., `TRUE`) if they are different. The double underscores at the beginning of the '__svc' array name merely indicate that the implementers of the code consider 'svc' to be a reserved name for their own use. They can be ignored for the purposes of understanding the code.

```c
uint16_t bnep_service_id(const char *svc)
{
    int i;
    uint16_t id;
    /* Friendly service name */
    for (i = 0; __svc[i].name; i++)
        if (!strncasecmp(svc, __svc[i].name)) {
            return __svc[i].id;
        }
    /* UUID 128 string */
    for (i = 0; __svc[i].uuid128; i++)
        if (!strncasecmp(svc, __svc[i].uuid128)) {
            return __svc[i].id;
        }
    /* Try convert to HEX */
    id = strtol(svc, NULL, 16);
    if ((id < BNEP_SVC_PANU) || (id > BNEP_SVC_GN))
        return 0;
    return id;
}
```

END OF EXAMINATION