Software Engineering II

Date: Thursday 15th May 2008
Time: 14:00 – 16:00

Please answer any THREE Questions from the FOUR Questions provided

Use a SEPARATE answer book for each Section.

This is a CLOSED book examination

The use of electronic calculators is NOT permitted.
Section A

1. a) Assume that you are a software project manager in a company that does not currently make use of any of the following software quality assurance (SQA) techniques:

- Software Process Improvement
- Formal Technical Review
- Software Test Coverage Analysis
- Checklists and Templates
- Defect Tracking
- Statistical Software Quality Assurance (SSQA)

For each of the following scenarios, state which *one* of the above techniques you would implement as a priority in order to rectify the problems described. In each case, give a justification of your decision in terms of the benefits and costs imposed by the introduction of the technique. Make clear any assumptions you make.

i) Several of the users of product A have complained about the variability of support offered for this product by its maintenance team. Sometimes problems are fixed very quickly, but other times there is not even an acknowledgment of receipt of the problem report until the user chases it. On other occasions, the problem seems to have been fixed, but then reappears in some other part of the system. On investigation, the maintenance team, which is based in both the UK and the US, are also unhappy. They report that work is often duplicated by both sites, and each site accuses the other of implementing sloppy, incomplete bug-fixes.

ii) System B is used to support the marketing team in targeting customers accurately for marketing campaigns. The collection of special offers and deals offered by the company changes rapidly, meaning that System B is subject to a continuous stream of requests for the addition of new functionality. The maintenance team for System B works to a regular release schedule, with new releases being made every fortnight. The team aims to implement requests for new functionality within 3 releases of the date when each request is approved, and for the most part they have been meeting this goal. Unfortunately, after each release, a large number of errors are reported. Upon investigation of the causes of these errors, the System B maintenance team presents convincing evidence that the problems lie with the specifications for the changes that are being passed to them from the Change Control Board, and not with the code that they produce.

(Question 1 continues on the following page)
iii) The development team for product C has recently had to deal with the loss of several key team members. Contract staff have been employed to replace them, but the turnover amongst these new team members has also been high because of the popularity of the technologies being used on this project. While some of the new team members have settled in well, there has also been a significant rise in the amount of rework being performed by the team. On investigation, this is found to be due to important omissions in deliverables as well as inconsistent use of design methodologies between team members.

iv) In recent years, several attempts have been made by project teams within the company to introduce more modern software development practices, including the use of software tools and management techniques such as risk management. The measurable gains from these efforts have been low, while the costs have been high. Senior management is now wary of devoting budget to such initiatives. At the same time, however, the company is also losing market share in favour of other companies advertising their up-to-date development processes. The company needs to find a cost-effective way to change the way it works. (12 marks)

b) For each of the above scenarios, describe one concrete aspect of software development in the company that you would measure, in order to gauge the success (or failure) of the SQA technique you propose to introduce. Briefly justify your choice in each case. (8 marks)
Section B

2. Imagine you are the leader of a programming team to develop a (small) library for a supermarket. The library is intended to provide the programming team with a basic set of modules for developing various applications in the retail business.

   a) In developing the library, why would you use Design by Contract?  

   b) What programming language would you choose, and why?  

   c) The supermarket provides an on-line ordering service for its customers. Using the language of your choice, outline a module for an on-line customer account, together with its contracts. You can assume that each customer is allowed a certain amount of credit. You should give a clear explanation of the outline code and the contracts.
Section C

3. Recall the project allocation example from the lectures. Here are the given sets and the state schema.

\[
\begin{align*}
\text{PERSON} \\
\text{TOPIC} \\
\text{ProjectAlloc} & \\
\text{studInterests, lecInterests} : \text{PERSON} \mapsto \text{iseq TOPIC} \\
\text{allocation} : \text{PERSON} \mapsto \text{PERSON} \\
\text{maxPlaces} : \text{PERSON} \mapsto \mathbb{N} \\
\text{dom studInterests} \cap \text{dom lecInterests} &= \emptyset \\
\text{dom allocation} &\subseteq \text{dom studInterests} \\
\text{ran allocation} &\subseteq \text{dom lecInterests} \\
\text{dom maxPlaces} &= \text{dom lecInterests} \\
\forall \text{lec} : \text{dom maxPlaces} \\
&\quad \bullet \#(\text{allocation} \supset \{\text{lec}\}) \leq \text{maxPlaces}(\text{lec})
\end{align*}
\]

The variable \textit{studInterests} records which people are students due to undertake a project, each mapped onto their list of interests. This list of interests are topics given in \textit{descending} order of interest to the student.

The variable \textit{lecInterests} similarly records which people are lecturers who may be supervising a number of students, each mapped onto their list of interests, given in \textit{descending} order of interest to the lecturer.

The variable \textit{allocation} records which students currently have a supervisor allocated, each mapped onto that supervisor.

The variable \textit{maxPlaces} records for each lecturer who might supervise a project, the maximum number of students he or she is allowed to supervise.

a) Present a schema which defines the initial state of the system, in which no students or lecturers have been added into the system. Explicitly specify the value for every state variable, but also indicate which values might have been inferred from other values being specified, and why. (3 marks)
b) Present a schema which defines the normal case of the operation \textit{AddStudent} in which a person \( s? \), who must not already be in the system, is added to it as a student, together with his or her list of topic interests, \( ts? \). You should explicitly specify the precondition(s). Is the precondition(s) strictly necessary? Why or why not?  
(5 marks)

c) Present a schema which defines the normal case of the operation \textit{AddLecturer} in which person \( l? \), who must not already be in the system, is added to it as a lecturer, together with his or her list of topic interests, \( ts? \) and his or her maximum allocation \( maxAlloc? \). You should explicitly specify the precondition(s). (4 marks)

d) Here is the schema which defines the normal case of the operation \textit{Allocate}. Study it carefully, and then describe it. You may translate it into "English" if you wish, but this alone will not be enough for full marks. You should ideally show your depth of understanding of the Z concepts being used, and how they fit together to model the required behaviour of the \textit{Allocate} operation.  
(8 marks)

\[
\begin{align*}
\text{Allocate} & \quad \triangle \text{ProjectAlloc} \\
\Delta s? & : \text{PERSON} \\
\text{\( s? \in \) dom } & \text{studInterests} \\
\text{\( s? \notin \) dom } & \text{allocation} \\
\exists \sup & : \text{dom lecInterests} ; \ t : \text{TOPIC} ; \ i, j : \mathbb{N} \\
& \quad | \text{maxPlaces}(\sup) > \#(\text{allocation} \uplus \{\sup\}) \\
& \quad \land i \mapsto t \in \text{studInterests}(s?) \\
& \quad \land j \mapsto t \in \text{lecInterests}(\sup) \\
& \quad \land \big( \\
& \quad \quad (\forall \lec : \text{dom lecInterests} ; \ k : \mathbb{N} \\
& \quad \quad \quad | \text{maxPlaces}(\lec) > \#(\text{allocation} \uplus \{\lec\}) \\
& \quad \quad \quad \land \big( \\
& \quad \quad \quad \quad (k \mapsto t \in \text{lecInterests}(\lec) \Rightarrow k \geq j) \\
& \quad \quad \quad \quad \land \text{ran}((1..i - 1) \lhd \text{studInterests}(s?)) \\
& \quad \quad \quad \quad \land \text{ran}(\text{lecInterests}(\lec)) = \emptyset \\
& \quad \quad \quad \big) \\
& \quad \big) \\
& \quad \land \text{allocation}' = \text{allocation} \uplus \{s? \mapsto \sup\} \\
\text{studInterests}' & = \text{studInterests} \\
\text{lecInterests}' & = \text{lecInterests} \\
\text{maxPlaces}' & = \text{maxPlaces}
\end{align*}
\]
4. I am having a party! I have invited some of my friends and want to keep track of who has told me they are coming. Some of those invited have said they are coming. Because my parties are known to be terrible, when I invite a friend I start off assuming he or she is not coming. I do not need to distinguish between those who do not reply and those who say they are not coming.

Here is the given set and state schema for this specification.

\[
\begin{align*}
\text{[FRIEND]} \\
\text{MyParty} \\
\text{invited} : \mathbb{P} \text{FRIEND} \\
\text{coming} : \mathbb{P} \text{FRIEND} \\
\text{notComing} : \mathbb{P} \text{FRIEND} \\
\text{coming} \cup \text{notComing} = \text{invited} \\
\text{coming} \cap \text{notComing} = \emptyset;
\end{align*}
\]

a) Present a schema to define the initial state of the system after which all three state variables are empty. (2 marks)

b) Present a schema to define the normal case of the operation \textit{Invite}, in which a friend \(f\) is invited by me. He or she must not already be invited, and I start off assuming he or she is not coming. (3 marks)

c) Present a schema to define the normal case of the operation \textit{Uninvite}, in which a friend \(f\) is told they are no longer invited by me. He or she must be currently invited, but he or she may or may not have been coming. (3 marks)

(Question 4 continues on the following page)
d) Present a schema to define the normal case of the operation Confirm, in which a friend $f$, tells me he or she is coming. The friend must already be one of my invited friends, and must not have already been coming. (3 marks)

You are going to totalise your operations, using the style in which they will return an outcome value of type $RESULT$. For this, you will need the following free type definition.

\[
RESULT ::= \text{success} \mid \text{alreadyInvited} \mid \text{notInvited} \mid \text{alreadyConfirmed}
\]

You will also use the schema, $SuccessMessage$, defined as:

\[
\begin{array}{l}
\text{SuccessMessage} \\
\quad \text{outcome}! : RESULT \\
\quad \text{outcome}! = \text{success}
\end{array}
\]

e) Present a schema, $AlreadyInvited$, which takes a friend $f$ which must be one of my invited, and outputs outcome! with the value $alreadyInvited$. (2 marks)

f) Present a schema $NotInvited$ defined appropriately in a similar way. (2 marks)

g) Present a schema $AlreadyConfirmed$ defined appropriately in a similar way. (2 marks)

h) Present total definitions for $Invite$, $Uninvite$ and $Confirm$. (3 marks)