Two hours

UNIVERSITY OF MANCHESTER
DEPARTMENT OF COMPUTER SCIENCE

Agile Software Engineering

Date: Wednesday 15th January 2020

Time: 14:00 - 16:00

Please answer BOTH Questions
Each question is worth 25 marks

Use a SEPARATE answerbook for each QUESTION

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

Question 1

You are a member of an agile team working for a company that provides web services for magazine publishers. The company has successfully tendered for a project to manage one publisher's magazine subscriptions. The subscription system would allow for digital and print subscriptions for all magazines provided by the publisher, with a variety of payment methods.

a) Compare and contrast two types of contract for software development that could be used for this project. Your answer should:

   i. Name and describe the two approaches (2 marks)

   ii. Describe how each approach does or does not fit with an agile software development approach. (2 marks)

b) Suggest an initial minimum viable product (MVP) for the subscription system. Be careful to describe the value the customer will receive if just this proposed functionality is delivered. State any assumptions you make about the magazines, publishers, readers, subscriptions or other entities. For full marks, you must state clearly how the MVP would work, avoiding vague statements such as "magazines are distributed to all subscribers". You must be sure to give some indication of the mechanisms, format, or other characteristics of the MVP. (5 marks)

c) Write one (1) epic and two (2) user stories for the project, using the Connextra format ("As a ... I want ... so that ..."). Your stories should differ in the stakeholder who will receive the value. The expected value (or behaviour change) that the stakeholder will receive must be clearly described. (9 marks)

d) The Three C's formula (Card, Conversation, Confirmation), developed by Ron Jeffries is one means to help ensure shared understanding of the user story.

   Card: The user story starts out intentionally brief, with a simple statement that could fit on an index card, similar to those you have previously written using the Connextra format.

   Conversation: Details must emerge before the team starts working on the story. To avoid extensive paper documentation, a conversation between the customer and the development team ensures that everyone has a clear understanding of what's being asked for and the value being provided. The written card will often be adjusted as a result of the conversation.

   Confirmation: The development team needs confirmation of the criteria that must be met for the story to be considered done. This criteria clearly describes how the feature will work in different situations, including error conditions.

(Question 1 continues on the following page)

(Question 1 continues from the previous page)

Choose one of your user stories from (c), and

i. Give two example questions that the development team might ask during a Conversation. (2 marks)

ii. Suggest how your user story might be updated as a result of such a Conversation. (2 marks)

iii. Provide three example acceptance criteria for your user story. (3 marks)

Question 2

You are leading an agile team that has been commissioned by a major international charity to create a web-based system for running on-line petitions.

The system will allow charity members to create their own petitions, which will be automatically advertised to charity members who have agreed to receive contact from the charity for this purpose. Supporters who follow the link to the petition should be able to sign it without entering their own details. Petitions should also be available from the charity's public web pages, where members of the public can sign the petition (after first entering their details).

The system must support two kinds of petition: "Strong" and "Normal". A strong petition is one that is either intended to be delivered to government or is in an priority area for the charity. These petitions must store the full name, the post code, and the e-mail address of all people signing them. All other petitions are considered to be normal. It is only necessary to story the name of the person for these.

a) The charity has arranged for a group of 7 of their supporters to come to their offices to help in testing an early version of the new system. Unfortunately, there have been some delays in recruiting your team, and the 7 supporters arrive unexpectedly on the first day that your team is in place. Describe how you would spend the day and explain how this would help you to gain the maximum value from the supporters' time, at this early stage of the project. (3 marks)

b) After a busy first day, you and your team start to settle into the process of developing the system. You decide to use behaviour driven development using Cucumber, and therefore set about creating a set of scenarios using the Gherkin language. One of your developers has no experience with Cucumber but is keen to learn. You ask him to write a scenario for the feature that allows petitions to be signed. He comes back with the following (very poor) scenario:

```
Scenario: a normal petition is signed by a supporter

Given a supporter, Jo
When Jo views the web page showing a named petition
And the petition is not one that needs post codes
Given Jo clicks on the "sign" button for the petition
When the names on the named petition are checked
Then Jo's name should be there
```

Point out 3 errors with this scenario, either in terms of conformance to the Gherkin syntax or in the semantics of the scenario as written. For each error, give an explanation of why it is a problem to write scenarios in this way, for the benefit of your developer. (6 marks)

c) Write a new version of the scenario specifying the behaviour expected when a support signs a normal petition, using good practice and correcting the mistakes you pointed out in your answer to part b). (6 marks)

(Question 2 continues on the following page)

(Question 2 continues from the previous page)

d) Write 2 further scenarios for the "Sign petitions" feature, covering different cases from that covered in your answer to part c). (4 marks)

e) Write the glue code that would be needed to run the simplest of the three scenarios you have written in your answers to parts c) and d) of this question. Indicate clearly which scenario you have selected to implement (e.g. by putting an asterisk next to it).

Give a brief description of any domain objects that you assume the existence of (1-2 sentences), to allow the intent of your glue code to be understood.

You should assume that the scenarios will be run using the Cucumber-JVM, and therefore should write your glue code in Java. Minor syntactic errors will not be penalised, provided the overall intent of the glue code is clear. (6 marks)

END OF EXAMINATION