

Two hours

Question ONE is COMPULSORY

**UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE**

Agile Software Engineering

Date: Monday 23rd January 2017

Time: 14:00 - 16:00

Please answer Question ONE in Section A and TWO Questions from Section B.

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

[PTO]

Section A

The (single) question in this section is compulsory. It is worth a total of 10 marks.

1. Appendix A lists the 12 agile principles. For each of the agile practices given below, state one agile principle that the practice upholds (bearing in mind that most agile practices aim to uphold several of the agile principles). Justify your answer with a brief explanation of the specific aspects of the practice that are relevant to the practice and how those aspects uphold the principle.
 - i) Self-organising team
 - ii) Task board
 - iii) On-site customer
 - iv) Daily stand-up meetings
 - v) Test driven development

(10 marks)

Section B

You should answer any TWO questions from the FOUR questions provided in this section. The questions in this section are all worth 20 marks.

2. a) In each of the following scenarios, a software team is attempting to make the change from a “big up front” process to a more agile process. In each case, something is going wrong for the team. Suggest one possible cause of each problem, and propose a corrective action that can help get the team back on track.
- i) Team A struggle at first with writing user stories instead of the large UML models and documents they used to produce. But they persevere, and with the help of some more experienced agile practitioners from other teams, they manage to create a story map covering the most important areas of the system to be built, and to create detailed (very thin) stories for the whole story map. They end up covering almost one whole wall of their office with stories. But after that progress seems to stagnate. The developers confess to being overwhelmed by the wall full of tiny details. They don’t know how to move forward from this point.
 - ii) Team B includes some experienced agile practitioners, and they are quickly able to identify a number of minimum viable product options that they wish to explore, to deliver value for their customer. They decide to prototype some options and to use low-cost channels to put the prototypes in front of real customers to gather statistics on take-up. They put in a request for the budget needed to achieve this, supported by a 2-page document describing the options to be tested and the rationale for the overall approach. 2 weeks later they get a response from the senior strategy board saying that the funds have been approved, subject to the team passing its first major project review in 2 months time.
 - iii) Team C is undertaking its first agile project, but is lucky enough to be working with a client who has commissioned software from agile teams before. Two experienced customer representatives are assigned to work with the team, and they lead several very successful story writing workshops. The team now has a story map covering the core business processes that the software must support, plus a collection of stories of varying sizes describing the functionality that is needed by the client. Unfortunately, at this point, the problems begin. The developers create a plan for the work without reference to the customer. When the customer requests that the most valuable work be scheduled up front, the team responds that it’s impossible to do the work in that order because the data needed won’t be in the database.
- (12 marks)
- b) Your team uses a story board with the following columns: Backlog, Next Iteration, This Iteration, Doing, To Verify, Done. Explain to a new member of your team what key events cause stories to move from one column to the other on this board, and what agile practice is being carried out when these events occur. (5 marks)

(Question 2 continues on the following page)

(Question 2 continues from the previous page)

- c) The team in part b) is using an iteration based approach, in which stories are pushed across the board, based on what the team decide should be implemented in each iteration. An alternative approach, not covered in the course, is to use a pull-based strategy, in which stories are moved from left to right whenever a space becomes free in that column. So, a team with 2 testers would have 2 spaces in the “To Verify” column, and stories could only be moved into that column if it contains 1 or 0 stories. This kind of board is particularly well suited for projects in the maintenance/support phase. Why might this be the case? Why are iteration based (push-based) story boards less suitable for projects in this phase?

(3 marks)

3. a) For each of the following statements, indicate whether it describes value that can arise from the development of new software or not. In each case, give a brief justification for your answer (rather than just saying “is value” or “is not value”).

- i) The software must be in use by the end of the financial year.
- ii) Fewer housing benefit claims are wrongly classified.
- iii) Fewer bugs remain in the deployed product.
- iv) The customer can choose whether to browse the full product description or not.
- v) More students choose to study at Manchester University.
- vi) A history of purchases can be associated with each customer.
- vii) It will become impossible for students to cheat on their coursework.

(7 marks)

- b) You are a member of a team writing an app to act as a front-end to a “smart fridge” device. The device is able to detect when products are taken from the fridge and when they are replaced. It can acquire information about “use by” dates, as well as product information, from the bar code on the product. The aim of the app is to provide the users of the fridge with warnings about products that are about to go past their sell-by-date, to add suggested purchases to the users’ online shopping accounts when stocks run low and to make suggestions for small achievable changes in diet to improve the health of the household in which the fridge is based.

Suggest an initial minimum viable product (MVP) version of this app/device combination. Be careful to describe the value the customer will receive if just this proposed functionality is delivered. State any assumptions you make about the device or the intended users. For full marks, you must state clearly how the MVP would work, avoiding vague statements such as “the device detects when the users are running low on kimchee”. You must give some indication of how the device knows that that is happening.

(5 marks)

- c) Write a user story for your suggested MVP, using the Connextra format (“As a ... I want ... so that ...”).

(3 marks)

(Question 3 continues on the following page)

(Question 3 continues from the previous page)

- d) Since this product represents something of a departure from the traditional web applications that your company normally builds, management are keen to get some evidence of the likely rate of return from it, before much investment is made.

What early stage trials could you run with the MVP you proposed in your answer to part 3b)? What resources would you need, and what kind of trial participants would you need to recruit? What measurements would you gather from the trial, to help you judge the likely return on investment from the product in the future? (5 marks)

4. a) You have been asked to coach a team of experienced software engineers, working for the first time in a cross-functional team. They have been used to working with other developers, farming out all responsibility for testing their software to dedicated testing teams. Now they have to take on a testing role from day 1, as their new team will be using a number of modern automated testing techniques. Your job is to make the transition to this new way of working as smooth as possible.

What would you say to this team to convince them of the benefits of having developers write their own unit tests, and having the whole team take responsibility for the suite of acceptance tests (and maybe ATDD/BDD test suites)? (4 marks)

- b) Design an acceptance test table for the following story, and populate it with 8 rows describing 8 contrasting acceptance test cases:

As a piping engineer, I want to know when my insulation plan for a pipe does not comply with statutory regulations, so that our company becomes known for its excellent fire safety record.

On the back of the index card bearing this story, the following conditions of satisfaction have been written:

- A piece of piping usually comes in 2 diameters: 5mm and 10mm.
- Piping is constructed from 1m long segments, so a section of piping is always a multiple of whole metres in length.
- We currently supply 3 types of pipe insulation: glass fibre, eco-wool and polymex.
- Eco-wool is only suitable for piping of 5mm in diameter and below.
- Eco-wool cannot be used for continuous lengths of piping longer than 10m, but it can be used in combination with polymex to cover longer distances (since it is the cheapest of all the options).
- Only glass fibre is suitable for extreme temperatures ($\geq 80^{\circ}\text{C}$ and $\leq -10^{\circ}\text{C}$)

Take care to indicate clearly in your answer which of your columns are inputs and which are outputs. (8 marks)

(Question 4 continues on the following page)

(Question 4 continues from the previous page)

- c) Write a SLiM FitNesse fixture for the acceptance test table you designed in your answer to question 4b), using programming-by-wishful-thinking to design the domain objects you think the service level API should implement, in order to provide the functionality described by the user story.

State clearly the table type you are assuming, if it is not already obvious from the test table design (e.g. decision table, query table).

Give a brief description of the domain objects that you invent for your fixture code. Explain the behaviour of any fakes that you create, and state why a fake was needed.

Note: you will not be penalised for simple syntax errors in your solution, or for minor and insignificant deviations from the methods and fields that SLiM expects of fixture classes for your chosen table type. (8 marks)

5. a) An agile coach joins a team that has recently adopted agile methods and is struggling. After observing the team at work for a couple of iterations, the coach makes several recommendations (shown below). All the recommendations are sensible, but the team is struggling to accept them. Explain briefly what the coach's thinking is in the case of each recommendation, using Mike Cohn's Test Automation Pyramid:

- i) The number of unit tests should be greatly increased
- ii) The number of GUI tests should be greatly reduced
- iii) A service level API should be created and tested thoroughly

(8 marks)

- b) You are using TDD to develop a role playing game for a client. The project is at an early stage. A simple grid-based game world exists, and a player can be created and moved around the world. Players have health points, which they lose if they step onto an electrified grid square. All other grid squares (at this stage) have no effect on the player.

The next task is to create grid squares that can heal the player when stood on. You speak to your client about the requirements. This is what she says:

We want to add three new types of grid square, which heal the player when she or he stands on them. Healing takes place on each game tick. The longer the player stands on a healing square the closer to full health they should get. We want to have three types of healing square: yellow glow (which heals at 5 HP per tick), orange glow (which heals at 10 HP per tick) and a very rare magenta glow (which heals at 100HP per tick).

(Question 5b continues on the following page)

(Question 5b continues from the previous page)

- i) Give 9 examples of tests that you would expect to write when implementing this functionality using TDD. At this stage, you can describe the test using a brief sentence or two. (The code will come later.) (4 marks)
- ii) What is the first piece of code that you would write when beginning to code this feature using a *strict* TDD approach? Explain why you chose to begin the development in this way.

Note: you should assume that simple game classes following the above description exist. You do not need to write out the definitions of any of these classes, apart from the class/methods that are being actively developed for this feature. A brief explanation of what the classes do can be given if you assume some aspects that are non-obvious. (4 marks)

- iii) Describe the thinking and coding you would carry out to complete this and one further TDD cycle. You should write out the state of the code for each step, highlighting the changes you have made for that step and giving a brief (one or two sentence) explanation of what you have done. Since each TDD cycle consists of 3 steps, you will need to write out the code and thinking for a further 5 steps to complete this question. You need only show the definitions of methods that have changed at each stage, and should feel free to break with good practice and use short method and class names to keep the amount of rewriting needed down, provided the meaning of the code at each step is clear. (4 marks)

[PTO]

APPENDIX A: The Agile Principles

This appendix lists the twelve Agile Principles that accompany the Agile Manifesto (agilemanifesto.org/principles.html), for use in answering question 1.

Principles behind the Agile Manifesto

We follow these principles:

- A) Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- B) Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- C) Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- D) Business people and developers must work together daily throughout the project.
- E) Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- F) The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- G) Working software is the primary measure of progress.
- H) Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- I) Continuous attention to technical excellence and good design enhances agility.
- J) Simplicity—the art of maximizing the amount of work not done—is essential.
- K) The best architectures, requirements, and designs emerge from self-organizing teams.
- L) At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

END OF EXAMINATION