

Two hours

Question ONE is COMPULSORY

**UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE**

Agile Software Engineering

Date: Friday 29th January 2016

Time: 09:45 - 11:45

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. a) The goal of many agile practices is to reduce waste during software development. As we saw in lectures, this is achieved by embracing the concepts of:

- i) simplicity
- ii) trust
- iii) feedback

into our team's processes. For each of these concepts, explain briefly how it contributes to reducing waste during development. Illustrate your answers by naming one agile practice that reduces waste in the way you describe for each concept. (9 marks)

- b) Although the minimisation of waste is important in a software project, there is another (arguably even more important) responsibility that we have when we choose a software process for our team. What is this responsibility? (1 mark)

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, an agile team is breaking one of the four values from the Agile Manifesto. For each scenario, state which value is being disregarded and give a brief justification of your answer. (The text of the Agile Manifesto is given in Appendix A, for you to reference.)
- i) A few weeks into the project, senior management in the customer organisation issue a memo stating that all projects must comply with a particular international engineering standard. Amongst other things, the standard mandates that all requirements be signed off through a formal review process before development can begin. To protect the team, the business analyst and the customer representative spend a week in each release period, converting the user stories for the next release into a lengthy document that can be sent for review several weeks before the release begins.
 - ii) A distributed team finds it difficult to get all the technical leads together in meetings, due to extreme time zone differences. Whenever key architecture or detailed design meetings are held by one sub-team, a team member takes minutes which are then sent to the other sub-team for review. Minutes of the meeting in which the design decisions are reviewed and amended are then e-mailed back to the first team, and so on, until consensus is reached.
 - iii) A member of the bank fraud section is seconded to work with the development team as their customer representative. Although now very experienced in bank fraud, the customer rep. holds a degree in Computer Science and started out as a COBOL programmer. The team spends the first two weeks of the project helping the customer rep. to learn to read and compile C# code, as well as how to get the code out of their source code repository and into their IDE.
- (9 marks)
- b) You are the business analyst on a new agile team, which has just started its first project. You have held a group story writing workshop, and your next job is to set up a task board for the team, to track their progress on the project and to use as a decision making tool. Although your team expect to do a lot of testing themselves, the customer has mandated that all code must be checked by an independent QA team before it can be deployed to the customer's machines, even for testing. Because of the mission critical nature of the project, all code must also be reviewed by a security expert, before it can be released to the QA team.

Assuming that otherwise your team uses a fairly typical, though simple, agile process, design a task board structure to track the progress of the team's work on this project. For each column, give the title you would give to it, and describe the kind of card that would be placed in it. Your description should state clearly what causes a card to be put into the column or to leave it.

(5 marks)

(Question 2 continues on the following page)

(Question 2 continues from the previous page)

- c) An agile team in your organisation has been experiencing some difficulty in filling an empty developer post. The business analyst and three developers have been members of the team since its formation, three years ago. The fourth developer role has been filled by a number of staff from your organisation over this period, including some very talented programmers. But no one seems to have been able to gel with the long-established team members, and the pattern is that after 3 or 4 months, the fourth developer requests a transfer to another team. Overall, the team's performance is reasonable; they develop a lot of software for their team size and customers seem to be happy with the results. But it is rare for customers to request the same team, even if they come back to the organisation for another project. They are not building the kind of customer loyalty that other teams in your organisation can command.

You have been asked to join this team as the technical lead, to see if you can diagnose the problem with this team. On joining, you are surprised to discover that the team is using the same set of practices, in much the same way, as when the team was formed. For example, the team is using the same set of task board columns as on their first project. The team is enthusiastic about regular retrospectives (which they hold in a nearby pub), but keeps them informal and unstructured. You can find no evidence that any changes have been made as a result of any retrospective.

Design a more structured retrospective process for this team, that can help them to make real improvements in their team performance. You should include suggestions for some prompts the team can use to focus their discussions on the problematic areas. (For full marks, the prompts should be tailored to the specific issues the team are facing, rather than just being generic retrospective prompts.) (6 marks)

3. a) Explain why the following statements are unlikely to be made by members of a well-functioning agile team.
- i) *Customer speaking to developers*: “We need to demo these stories at our trade fair in two weeks time. So, make sure you give them estimates that add up to no more than two weeks worth of story points.”
 - ii) *Developer speaking to another developer*: “These stories at the top of the list will be easy. They’re just some reports. We can do them whenever. Let’s start with these stories about the REST interface down here. They sound like they’ll make much more of a splash.”
 - iii) *Developer speaking to QA manager*: “Yeah, the unit test coverage is pretty low at the moment. But don’t worry. Soon we’ll be handing the code over to the independent testing team, and they’ll get the test code coverage right up.”
(6 marks)
- b) Examine each of the following user stories and state whether the user role mentioned in the story does indeed describe the kind of person who will gain significant value from the story when implemented and deployed. Give a brief justification for your answer in each case. Note any assumptions you make in coming to your decision.
- i) As an undergraduate, I want weekly e-mail reminders of any debts that I owe the University I am registered with.
 - ii) As a shopper, I want to register my credit card details before I can place products into my online basket.
 - iii) As a store manager, I want to be warned when stock is nearing its sell-by date.
 - iv) As a waiter, I want the menu that is published on the restaurant’s website to always match the menu being served in the restaurant.
 - v) As a frequent customer, I want to be shown a list of products that are on offer while I am waiting for my payment to be processed.
(5 marks)
- c) Select any two of the user stories from question 3b and add a convincing business value. If you argued in your answer to 3b that the user role does receive value from the story, then you should keep the same user role. If you argued that a different user role receives more value from the story, then you should rewrite the story from that role’s point of view.
(4 marks)

(Question 3 continues on the following page)

(Question 3 continues from the previous page)

- d) Consider the following epic story:

As a recruitment manager, I want applicant CVs to be ranked according to their participation and success in hackathons, so we can quickly identify candidates who may be suitable for entrepreneurial/innovation roles.

Refine this story into a sequence of simpler stories that would allow you to incrementally implement the full epic over a series of 2-week iterations, putting in place a small part of the functionality in each iteration. Make sure the order in which the simpler stories should be implemented is clear from your answer. Assume a 6 person team, which includes 4 competent developers.

(5 marks)

4. a) Your team has been awarded a contract to deliver software for a regional chain of cinemas. The owner of the chain has a decade of experience of commissioning software for her own and other, larger organisations. She knows the pitfalls and takes a hands-on approach, having learnt that she needs to work closely with the development team, to make sure she gets software that meets her needs. She's worked with one other (Scrum-based) agile team, and was not impressed. They created lots of post-it notes, but the software that was eventually produced didn't seem to have much to do with what was written on them.

It's clear from the outset that you'll need to carefully tailor your agile processes to fit the needs and prejudices of your customer. Name two practices that you will adopt that will help your team to avoid the problems this customer has experienced in the past. Briefly explain why you think these practices will help this customer get the software she needs.

(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 cinema sales manager, I want the ticket prices displayed as being available for each film to be appropriate for its showing type and classification, so that we always comply with government regulations on selling cinema tickets.

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

(Question 4 continues on the following page)

(Question 4 continues from the previous page)

- Film classifications available are U (unclassified), PG (parental guidance) and A (aged 18 and above).
- Showing types available are: “first run”, “repeat run” and “weekend family”.
- Basic ticket prices available are: adult (currently £10) and child (currently £5).
- Child tickets are only available for films with a U or PG classification.
- First run films are charged at 100% of the basic ticket price.
- Repeat run films are charged at 75% of the basic ticket price.
- Weekend family adult tickets are charged at 25% of the basic ticket price.
- Weekend family child tickets are charged at £1.00.
- Only films with a U classification can be shown as weekend family films.
- The basic ticket prices change regularly.

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

- 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)

[PTO]

5. a) Examine each of the program statements below, and decide whether it is part of the code for a unit test, whether it is part of the production code or whether it could be either. Briefly justify your answers.

- i) `fail("Expected an exception to be thrown we attempt to pay for a discontinued product.");`
- ii) `FakeCustomerAccount fakeAccount = new FakeCustomerAccount();`
- iii) `this.accountNumber = accountNumber;`
- iv) `assertThat(fakeAccount.balance(), is(TWELVE_POUNDS));`

(4 marks)

- b) You are developing software for a pharmaceuticals company to help it manage the data from its drug trials. Your next task is to write code that will convert a list of records from a legacy data source into the format used by the company's main drug trials database.

You decide to begin by writing code to convert the way the sex of the subject is represented. The records are passed to your code as an object of type `List<TrialSubject>`, and the class `TrialSubject` has methods `getSex()` and `setSex()` defined on it. This same class can be used to represent legacy and target records.

The company database uses the ISO/IEC standard 5218 while the legacy database uses a proprietary representation. The following table describes the conversion your code needs to perform:

Representation in legacy database	Representation in ISO/IEC 5218
"F"	2
"f"	2
"M"	1
"m"	1
"O"	0
" "	9

You decide to use detailed TDD steps to implement this code, because of its mission critical nature. What piece of code do you write first? Write a brief explanation for why you chose to start with this particular chunk of code.

(4 marks)

- c) Describe the thinking and coding you would carry out to complete this and two further TDD cycles. 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 8 steps to complete this question. You may abbreviate the method and variable names used in the code to keep the amount of rewriting needed down, provided the meaning of the code at each step is clear.

(12 marks)

APPENDIX A: The Agile Manifesto

This appendix contains the text of the Agile Manifesto (agilemanifesto.org), for reference when answering question 2, part a).

The Agile Manifesto (2001)

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

1. Individuals and interactions over processes and tools.
2. Working software over comprehensive documentation.
3. Customer collaboration over contract negotiation.
4. Responding to change over following a plan.

That is, while there is value in the items on the right, we value the items on the left more.”

END OF EXAMINATION