Office Use Only

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Monash_2-RGB |  |  |  |  |

**Semester Two 2018**

**Examination Period**

**Faculty of Information Technology**

**EXAM CODES: FIT2101**

**TITLE OF PAPER: SOFTWARE ENGINEERING PROCESS AND MANAGEMENT - PAPER 1**

**EXAM DURATION:** 2 hours writing time

**READING TIME:** 10 minutes

***THIS PAPER IS FOR STUDENTS STUDYING AT: (tick where applicable)***

**🞏** Berwick **🗹**Clayton **🗹** Malaysia **🞏** Off Campus Learning **🞏** Open Learning

**🞏** Caulfield **🞏** Gippsland **🞏** Peninsula **🞏** Monash Extension **🞏** Sth Africa

**🞏** Parkville **🞏** Other (specify)

During an exam, you must not have in your possession any item/material that has not been authorised for your exam. This includes books, notes, paper, electronic device/s, mobile phone, smart watch/device, calculator, pencil case, or writing on any part of your body.  Any authorised items are listed below.  Items/materials on your desk, chair, in your clothing or otherwise on your person will be deemed to be in your possession.

**No examination materials are to be removed from the room.** This includes retaining, copying, memorising or noting down content of exam material for personal use or to share with any other person by any means following your exam.

Failure to comply with the above instructions, or attempting to cheat or cheating in an exam is a discipline offence under Part 7 of the Monash University (Council) Regulations, or a breach of instructions under Part 3 of the Monash University (Academic Board) Regulations.

**AUTHORISED MATERIALS**

**OPEN BOOK 🗹 YES 🞏 NO**

**CALCULATORS 🞏 YES 🗹 NO**

**SPECIFICALLY PERMITTED ITEMS 🞏 YES 🗹 NO**

**if yes, items permitted are:**

***Candidates must complete this section if required to write answers within this paper***

STUDENT ID: \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ DESK NUMBER: \_\_ \_\_ \_\_ \_\_ \_\_

**This page intentionally left blank**

This exam is 7 pages long and contains 7 questions. It is worth 74 marks in total.

Please write your answers in the script book provided. If you run out of paper, you may request another script book. You may answer the questions in any order, but please make sure you number each question clearly. To maximize your mark, we suggest that you attempt the questions you find easiest first.

Each question that requires a written answer has a suggested length. This suggestion is only there to help you understand how much detail we’re expecting – there is no penalty for writing more or less than the suggested length provided your answers are clear and complete.

Your markers request that you not use red pen.

***Question (2 + 3 + 2 + 5 + 4 = 16 marks)***

*This question is about* ***requirements****.*

User stories in Agile can be evaluated by the INVEST criteria. The V in INVEST stands for “Valuable”.

1. In your own words, what does “valuable” mean in this context? **(2 marks)**
2. What risks arise if user stories are not valuable? Write about a paragraph. **(3 marks)**
3. How can a team verify that their stories are valuable? Write a couple of sentences. **(2 marks)**
4. What is the difference between a *user story*, a *task*, and a *product backlog item* (*PBI*)? Do tasks and PBIs also need to be “valuable”? Why/why not? Write about half a page. **(5 marks)**

Here is a nonfunctional requirement expressed as a user story:

*“As a customer, I want the system to be extremely secure and highly responsive so that I can perform transactions very quickly without worrying that my bank account information will be compromised.”*

1. What is the problem with this requirement as currently written? Suggest an alternative version that does not have this problem. **(4 marks)**

***Question (6 + 4 = 10 marks)***

*This question is about* ***Scrum practices****.*

1. In typical scrum projects, sprint planning sessions are conducted at the start of each sprint so that the team will have a prioritized and estimated sprint backlog. One way to estimate tasks is to use Planning Poker.  
     
   When using Planning Poker, we usually give every member of the scrum team input into the estimation process. But that might mean that team members end up estimating tasks that are outside their experience – for example, a programmer might have to try to estimate how long a designer’s job is going to take. Some critics claim that this is totally unrealistic. What is your stand on this criticism, and why? Write about half a page. **(6 marks)**
2. What aspects of student projects make Scrum harder to manage compared to projects undertaken by typical industry teams? Write up to half a page. **(4 marks)**

***Question (4 + 3 + 2 + 3 = 12 marks)***

*This question is about* ***project inception****.*

Here is an idea for a product:

*Rovr is a service that helps dog owners find people to provide services for their pets. You can think of it as “Uber for pets”.*

*Service providers register themselves via a special app, verify their identity and contact details, and select which services they provide: bathing, clipping, dog walking, pet-sitting, etc. Pet owners can use the Rovr website or app to request a service. Rovr will offer the job to appropriate service providers in the local area. After the service has been provided, the owner gets to rate the provider’s service on a scale of 1-5, and the provider can rate both the owner and the dog – the system charges an extra fee for handling dogs that are known to bite.*

*All transactions are logged and audited once a month to verify that service providers are being paid on a timely basis. A small team of technical staff at head office will manage these logs as well as maintaining the servers and responding to tech support requests.*

1. Who should be invited to a story-writing workshop for this project, and why? Write about half a page. **(4 marks)**
2. Suggest at least three personas that could be used in this workshop. **(3 marks)**
3. Consider the information that Rovr’s system will need to store.
   1. List at least *two* security-related risks related to the storage of this information. **(2 marks)**
   2. Suggest a mitigation strategy for at least one of these risks. **(3 marks)**

***Question (4 + 4 = 8 marks)***

*This question is about* ***Agile and traditional process models****.*

1. What problems with traditional processes do Agile process models help to solve? Write about half a page. **(4 marks)**
2. Give an example of a situation in which an Agile approach would *not* be suitable. Explain why. Write about half a page. **(4 marks)**

***Question (4 + 4 + 6 = 14 marks)***

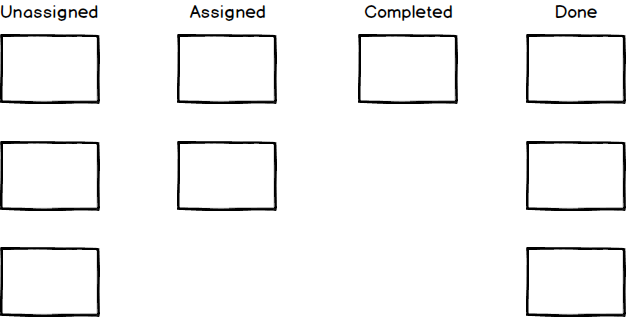
*This question is about* ***Agile practices****.*

A team is developing web services and web applications. Five years ago, the team used Scrum as documented in the Scrum Primer. Since then, their methodology has changed.

The team no longer uses user stories. Instead, requirements are managed by tasks. The Product Owner sits where she can see the master task list and notifies the Scrum Master if she sees anything that doesn’t seem to be adding value from her perspective, and the Scrum Master investigates.

A master task list is kept in an online system. The task list is displayed at all times on a screen in the office that is visible from all team members’ workstations, and is also accessible to team members when they are working offsite or from home.

Here is an image to show you the structure of this task list (note that this image is too small to let you read task details, but they would be easily readable on the screen.)



Whenever a team member is looking for something to do, they can either develop or QA. If they want to develop, they assign themselves a task from the “unassigned” column by putting their name on it and moving it to the Assigned column. When they feel it’s finished, and it passes all unit tests, they move it to the Completed column. Team members who want to do some QA take a task from the Completed column, informally review the code, verify that it passes all automated tests and the acceptance tests as documented on the card, then move it to the Done column.

The team no longer conducts sprint planning or backlog grooming. Instead, team members do their own estimation at the time that they assign tasks to themselves. The Scrum Master and Product Owner check on an informal basis that the task list looks okay.

Instead of running a daily scrum, the Scrum Master makes sure to communicate one-on-one with each team member every day or two to informally check that everything is running smoothly.

*CONTINUED ON NEXT PAGE…*

Sprint reviews are no longer conducted because the team no longer feels that they are necessary. Instead, software is deployed to a staging server as soon as it passes unit tests and is manually pushed to production once it has passed user acceptance tests. This means that they are shipping new features much more often than they used to. The team does make sure to conduct a retrospective every two or three weeks to consider how their process can be made more efficient and effective.

1. Is this still Scrum? Why/why not? Write up to half a page. **(4 marks)**
2. Is this still Agile? Why/why not? Write up to half a page. **(4 marks)**
3. Have the changes to this process increased or decreased the risks involved? What are the tradeoffs (i.e. costs and benefits)? Write up to a page. **(6 marks)**

***Question (2 + 2 + 4 = 8 marks)***

*This question is about* ***team structure****.*

1. An Agile team may have as many developers as you like, but should only ever have one Product Owner. What risks arise if a team decides to appoint two Product Owners? Write about a paragraph. **(2 marks)**
2. What is *siloing*, and why is it considered a bad thing? Write a couple of sentences. **(2 marks)**
3. Explain how DevOps approaches help overcome siloing, and how this improves outcomes for teams. Write about half a page. **(4 marks)**

***Question (6 marks)***

*This question is about* ***risk management****.*

A team member has recently attended an Agile conference. During your team’s retrospective, he makes this suggestion:

*“At the conference, I learned about the concept of YAGNI: You Ain’t Gonna Need It. That is, you shouldn’t design more functionality into your software than specified by your requirements, because doing so is likely to need to wasting effort, and you can always conduct a refactor if the extra functionality turns out to be necessary later on. That makes sense, right? But then I got to thinking, we spend ages thinking about risks, and they hardly ever happen. It’s impossible to know what will happen until after it’s happened, so why don’t we apply the YAGNI principle to risk management and not write a risk register until after each risk has happened?”*

What do you think of this suggestion – do you agree or disagree? Give reasons. Write half a page to a page. **(6 marks)**

🙟 *END OF EXAM* 🙝