University of Glasgow

Wednesday 28th April 2021
Available from 14:00 BST
Expected Duration: 2 hours
Time Allowed: 4 hours
Timed exam within 24 hours

DEGREES of MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

Professional Software Development H
COMPSCI 4015

(Answer all 4 questions)

This examination paper is an open book, online assessement and is worth a total of 80 marks.

1. Project Management

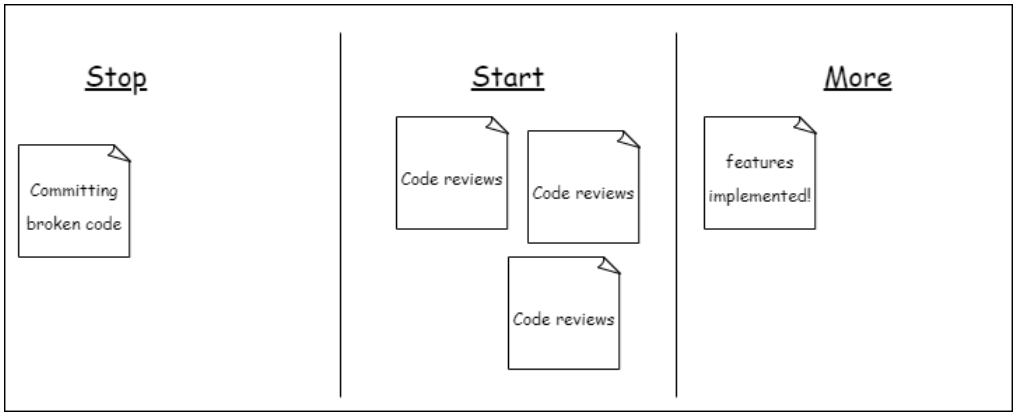
Reading the following scenario and then answer the questions below.

You have started work as an agile coach with a team who have recently adopted Scrum. The team comprises Lesley (the project manager and product owner), Kerry (a senior developer) and four developers, Robert, Jody, Jamie and Alex. The team conduct a retrospective late on a Friday after the end of their second sprint. You are invited to participate as an observer.

Lesley begins the retrospective by explaining that it will focus on resolving the delays in the project's schedule. All the information recorded in the three overdue tickets are as follows:

- *Configurable notifications* As a user I want to be able to have notifications for all the interesting system events (32 hours).
- *Broadcast message* As a user I want to broadcast a message to other users who are interested in the topic (25 hours).
- *Encrypt messages* As a user I want to be able to choose to encrypt messages to certain other users (12 hours).

Lesley also states that she is relieved that Robert won't be attending the retrospective (he doesn't work on Friday) as the overdue features were all assigned to him. She says that this means the rest of the team can discuss what Robert did wrong without embarrassment. Lesley then initiates a stop, start, continue data gathering exercise. She completes her sticky note and adds it to the board under 'More'. Lesley then gives a sticky note to Kerry (a senior developer) who completes it and then adds it to the board. She then repeats this process for each of the other developers in turn.



Once the data gathering is complete Lesley reminds the team of the set of features that have been planned for them to complete in the *next* sprint and then gives Jamie, Jody and Alex one overdue feature to complete as well, so that the team can catch up.

Kerry comments at this stage that 'committing broken code' was raised under 'stop'. Lesley agrees that this issue can also be addressed in the sprint. An extra

issue, is added to the team's issue tracker with the title "Stop committing broken code" and assigned to every team member. Speaking to Jamie, Jody and Alex afterwards you learn that they are pleased the meeting only lasted 10 minutes so they could resume work on the late features before the weekend.

(a) Explain the purpose of a Scrum *retrospective*, contrasting the purpose with that of a *sprint review meeting*.

[3]

(b) One problem with the retrospective is that the team are not performing root cause analysis on the issues they identify. For example, the reason that "broken" code is being committed is not investigated to find a constructive solution. Identify *six* other problems with the way that the team conducted a retrospective. For each of the problems propose an alternative approach that would result in a better outcome.

[12]

(c) Choose a root cause analysis method and apply it to the overdue project schedule, based on what you know about the software team process from the scenario.

[5]

2. Requirements Engineering

Consider the following user stories, concerning the development of a smartphone application to allow patients with breathing difficulties to monitor their 'peak flow' capacity. The application can have a Bluetooth link to a peak flow meter, which they blow into as hard as they can to create a reading, in litres/minute. Peak flow readings of less than 250 l/min are potential signs of problems, such as an asthma attack.

> As a user I want to connect my peak flow meter to my smartphone so that I can automatically record readings.
>
> As a user I want to first follow the procedure for linking my device to the meter (only need to do this the first time, unless it doesn't work), before then blowing into the peak flow meter, which will display a reading . If my smartphone is linked to the device then the reading will be recorded by the app automatically, otherwise I should enter a new reading on the app user interface. Once the reading has been entered the time of the reading will be automatically recorded. Once a basic reading has been taken I can optionally record whether the reading was taken before or after medication and before or after exercise. I can also leave an optional note. I will also be able to view the history of my peak flow readings in both tabular and graph form so that I can monitor my breathing over time. There should be buttons across the top of the display to access these functions. Periodically, the data recorded should be sent to the clinical team for review (with my permission) and the clinicians should also be able to view data on their patients on the app. If my peak flow is recorded below 300 for two days in a row then a notification should be sent to my clinical team. Performing this workflow every day is critical so that an accurate record of my breathing difficulties as a patient can be reviewed by my clinical team and myself.

(a) Evaluate the second user story with respect to the six INVEST criteria.

[6]

(b) Taking account of your critique, propose six new high priority user stories to improve the requirements specification for the mobile application.

[6]

(c) Identify and fully describe two personas in the original two user stories.

[4]

(d) Assume that you only have between 25% and 50% of the budget necessary to implement all the user stories you have identified. Annotate *all* the user stories (including the first user story in the scenario and the ones you created) according to the MOSCOW criteria so that you define a minimum viable product. Justify the choices and explain any assumptions you have made.

[4]

3. Software Architecture

Consider the following scenario and then answer the questions below.

> You are part of a team working on a framework for studying software repositories on the GitHub hosting site. GitHub has an API which can be queried for various different types of metadata about a repository, such as contributors, commits, issues and pull requests. A Python library already exists that wraps this information in a `GithubRepository` class. Repositories can, of course, also be cloned for further analysis using the git version control tool. The purpose of the framework is to allow researchers to create large samples of source code repositories and perform different analyses on them in a systematic and consistent way. For example, one of the researchers the team works for wants to be able to:
>
> 1. Search for repositories that use the Python programming language.
> 2. Filter the repositories to ensure they contain a `requirements.txt` files that contain a list of Python package dependencies from both GitHub and GitLab.
> 3. Filter the sample of repositories to ensure they have a minimum number of commits.
> 4. Filter the repositories to ensure they have a minimum number of contributors.
> 5. Filter the repositories to ensure they have a minimum code base size in the latest commit.
> 6. Clone all the repositories into a local directory.
> 7. Run the `dependency-check` tool on the `requirements.txt` file in each of the repositories.
>
> Other researchers may want to create different processes for filtering and analysing the repositories on GitHub. For example, rather than searching for repositories based on language, they may want to generate a random sample of repository IDs. The researchers may want to insert or remove filters, filter a single source sample in two or more different ways, or they may want to apply different analyses, focusing perhaps on the unit tests in the repository, or exploring the relationship between issues and pull requests. Additional features such as being able to generate graphs may also be useful, or being able to flexibly record information about repositories as they are filtered or analysed.
>
> The goal of the project is to provide a framework that can be configured to automatically perform the different stages of an experiment.

(a) You are working with a colleague to decide on the overall architecture for the system. A colleague proposes that the overall architecture for the system should be based on the observer-observable design pattern. Explain the difference between an architectural and design pattern to your colleague.

[2]

(b) From those you have seen in the course, choose an appropriate architectural pattern for the GitHub analysis system. Explain why the pattern is a suitable solution for the design problem.

[3]

(c) Explain how you would *tailor* the pattern to fit the needs of the scenario.

[4]

(d) Sketch a component diagram to illustrate the architecture that you have chosen. Include the key components and interfaces.

[6]

(e) The product owner reports that a new requirement has been proposed. The research team would like to be able to perform the same sampling technique on the GitLab repository hosting platform concurrently with GitHub. Explain how this could be accommodated within the architecture you have proposed.

[3]

(f) A colleague has proposed using the plugin architecture to allow third party tools to be included in the repository analysis system. State whether you agree with this proposal and explain your reasoning.

[2]

4. Consider the following gherkin feature file, describing a system for managing reminders and then answer the questions below.

Assume that:

- Every distinct step in the feature file has a distinct step implementation function.
- Any wrapped steps are soft wraps applied by the editor (they aren't in the source code).

```
Feature: Configure reminders

  Scenario: Reminder on day in next week
    Given a RemindMeBot in the channel General
    And the date is 22nd January 2021
    When I say remind me next Wednesday to buy milk
    Then RemindMeBot responds I will remind you to buy milk on
   2021-01-27

  Scenario: Reminder on day next week, re-word reminder text
    Given a RemindMeBot in the channel General
    And the date is 22nd January 2021
    When I say remind me next Thursday to collect my parcel
    Then RemindMeBot responds I will remind you to collect your parcel
   on 2021-01-28

  Scenario: Reminder on day in next week, seven days ahead
    Given a RemindMeBot in the channel General
    And the date is 22nd January 2021
    When I say remind me next Friday to buy milk
    Then RemindMeBot responds I will remind you to buy milk on
   2021-01-29

   Scenario: Recurring reminder
     Given a RemindMeBot in the channel General
     And the date is 22nd January 2021
     When I say remind me every Monday to attend the standup
     Then RemindMeBot responds I will remind you every Monday to attend
    the standup beginning 2021-01-25

   Scenario: Reminder in a different channel
     Given a RemindMeBot in the channel General
     And the date is 22nd January 2021
     When I say remind me next Wednesday in Personal to buy milk
     Then the channel changes to Personal
     And RemindMeBot tells me I will remind you to buy milk on
   2021-01-27
```

(a) How many step functions will be required to execute the feature as a test suite using a BDD framework such as Behave? Give a short explanation of your answer.

[2]

(b) Write a scenario to test whether a reminder can be set in the General channel for Kerry to attend the standup every Monday.

[2]

(c) Including the scenario you have created, explain 5 ways that the feature file could be changed to make it easier to maintain without changing the behaviour of the test suite.

[12]

(d) You and your colleague are discussing the merits of applying code coverage analysis to your suite of behave tests. Your colleague argues that code coverage is not relevant to acceptance tests written for BDD frameworks such as *Behave* or *JBehave*. Explain whether you agree with this statement.

[4]

END OF QUESTION PAPER