HOMEWORK WEEK 5-6

Lorelle Brownlee

TASK 1 (Agile Techniques)

Question 1

Complete definitions for Scrum related key terminology provided below.

SCRUM CEREMONIES

- *Product backlog refinement* - this is the process of adding detail, order and estimates to items in the Product backlog. The product backlog is an ordered list of what steps need to be completed in order to improve a product being developed using the Scrum methodology (effectively it is a product roadmap). The Product Backlog Refinement is an ongoing, collaborative process shared between the development team and the product owner.

- *Sprint planning*.- this is an event in the Scrum methodology framework where the team meet and decide upon which items in the product backlog they will focus on during the Sprint, and plan how to complete them.

- *Daily scrum*. - This is a daily scheduled meeting, usually 15-minutes in length, that is part of the Scrum framework and is held with the team developers. The aim of the meeting is to explore and update the team on progress towards the goal of that Sprint. Each scrum involves updates on progress towards the goal as well as creating a plan for the upcoming day's work.

- *Sprint review.* - this is also known as the "demo". It is a meeting between the team and stakeholders or end users, that takes place at the end of a

Sprint. At the Sprint Review, the team showcase their work. It is an opportunity for the team to receive constructive feedback on the product.

· *Sprint retrospective* - this is a meeting held at the end of a Sprint, where the team can reflect upon the Sprint, discussing what went well during that Sprint and what can be improved upon during the next Sprint. It is an opportunity for the team to identify ways to improve its effectiveness going forward.

SCRUM ROLES

· *ScrumMaster* - this is a team member who facilitates a development team through Scrum, adhering to the Agile product management methodology. A scrum master helps the team to self-organise and optimise productivity. They are responsible for creating the conditions that accommodate effective item delivery through facilitation and coaching of team members.

· *Product Owner* - this is the team member who is responsible for maximising the value of the product through optimising the work of the Scrum team. They are responsible for effective Product Backlog management, including strong communication of product goals and product backlog items. This role exists to represent the needs of both the users and the business stakeholders within the team.

· Development Team. - an Agile development team is, broadly speaking, a group of people working together to create a software product. They are accountable for product quality. Under the agile methodology, they should be self-organising and cross-functional, meaning

Question 2

You are leading a development team that was given a task to create a new yoga booking system.

High level description of the system is as follows:

- It has a very simple interface to accept user input (bookings) and display classes information
- All bookings, appointments, schedules etc should be stored in a SQL database.
- There is a ‘backend’ system that should be written in Python to handle the logic and manage the data flow.

Your team has two weeks to build a simple prototype that will be shown to the client to seek their feedback and discuss further enhancements.

TASK

- Break this task into smaller stories (chunks of work) for the team to work on.
- Assume that one person works on one task.
- Mark tasks that can be worked on in parallel and perhaps those that need to be worked on in particular order.

Tasks:

Phase 1) Design and build simple UI.

Must show user available class times, yoga types and location. Must allow easy booking. User can create an account and log in.

Phase 2) (These can all be performed in parallel)

- Design and build SQL database of user accounts, bookings and classes.
    - Must contain a table of user account information, a table of bookings and a table of classes.
- Design and build backend system to link frontend to database.

    - Must be built in Python and logically move relevant data from frontend to backend and perform data manipulation or calculations as required.
- Build secure online payment system for paying for classes

Phase 3) Unit testing of prototype and developer responses to unit tests.

TASK 2 (SQL)

Question 1

Design a cinema booking system.

Think how you would approach the problem and what are potential ways of solving it?

You do not need to write actual code, but describe the high-level approach:

- **Draw a list of key requirements**

- Needs a simple user interface, displaying movie blurbs and viewing times for the selected day.
- Needs to have a feature to create an account and log in
- Need to be able to filter by date or by film, to see what films and show times have seats available.
- Needs to be able to store user records (for users with accounts) including upcoming and previously booked tickets.

- Needs a secure online payment system

· **What are your main considerations?**

- Keep record of how many seats have been booked and how many are still available, to avoid overbooking.
- Need to warn users of age restrictions on certain films before booking is complete.

· **What would be your common or biggest problems?**

- Server going down, preventing users or the cinemas from accessing bookings
- Payment system failure
- Double-bookings or overbooking

· **What components or tools would you potentially use?**

Tools:

- Website design using Python/Flask for the web framework, and for the backend there will be an SQL database for user records, bookings and film showings.
- If the cinema is busy, may need to write bookings to the database in large chunks e.g. after 1000 bookings (batching).
- The database will need to be backed up regularly.
- If a booking is terminated before it is completed, how do we prevent payment from going through?
- Keep record of how many seats have been booked and how many are still available, to avoid overbooking.

· You are welcome to draw a diagram (a very simple one) for the process flow to explain how it is going to work.