HOMEWORK WEEK 5-6

Aurora Sanchez Diaz

TASK 1 (Agile Techniques)

Question 1

Complete definitions for Scrum related key terminology provided below.

SCRUM CEREMONIES

- **Product backlog refinement**

It is a process in which the Product Owner and the Development Team collaborate to create a shared understanding of the following:

1. What the Product will and will not do.

2. How long and how much effort it will take to implement the product.

3. The order of the tasks.

Definition adapted from the following source(s): https://www.digite.com/agile/backlog-refinement/

- **Sprint planning.**

Scrum Master, Scrum Product Manager and Scrum Team meet to decide which backlog items[1] will be tackled during the following sprint. Team members can influence when and how work is done thanks to this sprint planning ceremony.

Definition adapted from the following source(s): https://www.workfront.com/project-management/methodologies/scrum/sprint-planning

- **Daily scrum.**

It is a 15-minute meeting for the development team and the scrum master in which the progress towards the sprint goal is reviewed and a goal for the next day of work is set.

[1] Backlog items are ideas that could add value to a product being developed.

Definition adapted from the following source(s): https://www.scrum.org/resources/what-is-a-daily-scrum

- **Sprint review.**

It is an informal meeting held at the end of each sprint in which the job completed during said sprint is celebrated. ScrumMaster, Product Owner and Development team attend this meeting.

Definition adapted from the following source(s): https://www.mountaingoatsoftware.com/agile/scrum/meetings/sprint-review-meeting

- **Sprint retrospective**

It is a meeting held at the end of the sprint in which the team reviews the sprint's outcomes. Usually, the following questions are discussed:

- What worked well?
- What could be improved?
- What will we commit to doing in the next sprint?

Definition adapted from the following source(s): https://www.scrum.org/resources/what-is-a-sprint-retrospective

SCRUM ROLES

- **ScrumMaster**.

A ScrumMaster is a worker that makes sure scrum principles are followed in their team. The Scrum Master is responsible for the Scrum Team's success. A Scrum Master serves their team, the Product Owner and the organisation they are a part of.

Definition adapted from the following source(s): https://www.scrum.org/resources/what-is-a-scrum-master

- **Product Owner**

A Product Owner is the person responsible for increasing the value of the products produced by a scrum development team. A Product Owner takes on different roles, such as business strategist, product designer, market analyst…

The Product Owner also creates a product roadmap, which is a visual representation of the vision and development plans for the product. It is a plan for execution AND a guide for stakeholders to reference.

Definition adapted from the following source(s): https://www.lucidchart.com/blog/product-owner-roles-and-responsibilities

· **Development Team.**

They are the members responsible for building the actual product increment and meeting the sprint goal.

Definition adapted from the following source(s): https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf

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.

USER STORIES

1. As a user, I want to see the classes available on a given day.

2. As a user, I would like to see those classes on a weekly grid in chronological order.

3. As a user, I need to see the details for each class when clicking on its title.

4. I should be able to click on a clickable button to book the lesson I am interested on.

5. Additionally, I need to introduce my first name, last name, phone number, email and card details/Paypal details to confirm my booking.

6. Finally, I should receive a confirmation email with my booking reference and the details I have provided.

7. The system should be created around a database that contains all the information about classes and schedules.

8. After having completed the booking process, my booking details should be stored on a SQL database that is connected to the booking system.

· 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.**

Assuming that we have been given a design, backend and frontend teams could work in parallel using placeholders and hardcoded values for user stories 1, 2, 3 and 4. They would also work on just the design of user story 5 and 6. Let's say this process would take 2 days.

In the meantime, backend should work on the confirmation email and the database with several tables that will hold information about date, time, number of participants, type of lesson, participants' names and card details (user story 7). This information would have an impact on user stories 1, 2, 3 and 5. This process would take around 2 days based on the complexity of the system.

Afterwards, the front-end team should make sure the design is finished and that the website looks as close as possible to the wireframe and initial designs as possible (this would fulfil the requirements outlined in user stories 1-7). The process should take 2 days if we have three team members doing this.

Lastly, days 9 and 10 should be dedicated to testing and making sure the system works.

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?

- Draw a **list of key requirements**

  1. Being able to add new cinemas to the booking system while also taking into consideration the number of pre-existing cinemas in the system.

  2. Being able to pick the nearest branch out of all the cinemas in the country.

  3. Once the nearest cinema has been picked, displaying all movies shown in two-week intervals.

  4. Ability to prebook seats for major releases outside of the two-week interval.

5. Search bar for finding movies.

6. Being able to pay with major card brands, PayPal and ApplePay. Also, being able to prebook on the website and pay in person in the actual cinema.

7. Age verification for R rated movies.

8. Accessibility information for each movie and cinema room should be displayed next to the schedule for each movie.

9. Trailers for each movie inside of the movie info section.

10. No more than 10 people can book seats in the same transaction. They should get in touch with the branch directly to book seats.

11. Refund options (different in case of changes or cancellations of the showings)

- What are your **main considerations?**

1. Constant need for updates, particularly when movies are not shown in cinemas anymore or when new movies are being shown in cinemas.

2. Different branches will need different information as some cinemas are bigger than others/are showing different movies.

3. It should be easy and quick to pay for tickets right after selecting seats. Seats should only be reserved for 15 minutes before finishing with the payment.

4. No need to register to book seats, print tickets or download a specific app to display tickets, as these features are inconvenient and can deter potential customers from using our booking system. Tickets are sent via email in pdf format and QR codes.

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

1. Users might be overwhelmed if there are a lot of different showings in their local branch. This can be solved with a simple layout regarding front end development.

2. Users might not want to book a seat and watch a movie in their nearest branch, but other branches (for example, if they are travelling domestically, or if they live in a city like London, with many different branches of the same company).

3. User might want to book further in advance than 2 weeks.

4. Information related of movies that are no longer being shown should not appear on the website. This process should be automated without forgetting that different branches stop showing movies in different dates.

5. Some users could try to book two different movies that are shown simultaneously (partially or totally). Our booking system should prevent users from doing this.

6. Seats that are reserved before being paid for should not be able to be selected. Not doing this would result in two or more people booking the same seat. A queuing system can be

7. Underage users might lie about their age when booking seats for R rated movies. Our booking system should ask for age verification and include this information on the ticket. Staff at the branch should also ask the user to show proof of ID to verify the information given is truthful.

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

1. Python for backend.

2. Frontend: HTML/CSS and some Javascript (for interactivity). Or a frontend framework to streamline the process and better display the information.

3. Database with information about bookings.

4. Connection to a movie API that can display key information about each movie.