**HOMEWORK WEEK 5-6**

**Oleksandra Yankovska**

**TASK 1**

**SCRUM CEREMONIES**

**· Product backlog refinement**

- Product Backlog refinement is the act of breaking down and further defining Product Backlog items into smaller more precise items.

Often, Scrum Teams come together once per Sprint, or once per week to have their "Refinement Meeting". The Product Owner shares what Product Backlog items (PBI) need to be refined and the whole team discusses them. After the discussion (which can take a long time and often involves only a few), the planning poker cards are drawn to give an estimation to the PBI (Product Backlog Item)

.Every member of the Scrum Team is responsible for Product Backlog Refinement:

1. The Product Owner: building the right thing;
2. The Developers: building the thing right;
3. The Scrum Master: ensuring feedback and empiricism throughout these activities.

**· Sprint planning.**

Sprint Planning initiates the Sprint by laying out the work to be performed for the Sprint. This resulting plan is created by the collaborative work of the entire Scrum Team.

The Product Owner ensures that attendees are prepared to discuss the most important Product Backlog items and how they map to the Product Goal. The Scrum Team may also invite other people to attend Sprint Planning to provide advice.

Sprint Planning addresses the **following topics:**

1. Why is this Sprint valuable?

The Product Owner proposes how the product could increase its value and utility in the current Sprint. The whole Scrum Team then collaborates to define a Sprint Goal that communicates why the Sprint is valuable to stakeholders. The Sprint Goal must be finalized prior to the end of Sprint Planning.

1. What can be Done this Sprint?

Through discussion with the Product Owner, the Developers select items from the Product Backlog to include in the current Sprint. The Scrum Team may refine these items during this process, which increases understanding and confidence.

Selecting how much can be completed within a Sprint may be challenging. However, the more the Developers know about their past performance, their upcoming capacity, and their Definition of Done, the more confident they will be in their Sprint forecasts.

1. How will the chosen work get done?

For each selected Product Backlog item, the Developers plan the work necessary to create an Increment that meets the Definition of Done. This is often done by decomposing Product Backlog items into smaller work items of one day or less. How this is done is at the sole discretion of the Developers. No one else tells them how to turn Product Backlog items into Increments of value.

The Sprint Goal, the Product Backlog items selected for the Sprint, plus the plan for delivering them are together referred to as the Sprint Backlog.

Sprint Planning is timeboxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

**· Daily scrum.**

The purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work.

The Daily Scrum is a 15-minute event for the Developers of the Scrum Team. To reduce complexity, it is held at the same time and place every working day of the Sprint. If the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers.

The Developers can select whatever structure and techniques they want, as long as their Daily Scrum focuses on progress toward the Sprint Goal and produces an actionable plan for the next day of work. This creates focus and improves self-management.

Daily Scrums improve communications, identify impediments, promote quick decision-making, and consequently eliminate the need for other meetings.

The Daily Scrum is not the only time Developers are allowed to adjust their plan. They often meet throughout the day for more detailed discussions about adapting or re-planning the rest of the Sprint’s work.

**· Sprint review.**

A sprint review is an informal meeting held at the end of a sprint, in which the Scrum team shows what was accomplished during this period. This typically takes the form of a demonstration of new features, with the goal of creating transparency, fostering collaboration, and generating feedback.

The purpose of a Scrum sprint review is not to provide a status update or make a presentation to stakeholders; it is to collect and absorb feedback on the actual product increment—which is the sum of all backlog items completed during the sprint. If needed, the sprint review group will adapt the backlog going forward, in order to maximize efficiency in sprints to come.

**Sprint review meeting agenda.**

There are many ways to conduct a Scrum sprint review. But, in general, the sprint review agenda would include:

Reviewing the goal for the sprint

Demonstrating new features implemented during the sprint

Requesting feedback from the stakeholders

Discussing work not yet accomplished

Identifying risks and impediments

Reviewing project increment objectives

Looking ahead to the next sprint, using the top lines from the product backlog

Overall, the team is looking to get answers to questions, such as:

What did users like or dislike?

What didn’t they understand?

What things would they like changed or eliminated?

Are there features they would like added?

Do they have any tips about how the Scrum team could do their job better?

**· Sprint retrospective**

The Scrum sprint retrospective is a timeboxed meeting that takes place after the sprint review and before sprint planning. Its **purpose** is to:

Examine how the just-completed sprint went as far as people, relationships, processes, and tools.

Identify and order what went well.

Do the same with things that didn’t go well.

Identify potential improvements.

Create a plan for implementing improvements to the way the Scrum team accomplishes its work.

Everything that affects how the Scrum team develops the product is open to discussion and improvement, including processes, tools, artifacts, environment, and so on. It allows development teams to adapt Scrum to their particular circumstances.

Scheduling a Scrum retrospective at the end of every sprint ensures that needed changes are understood and implemented before they are lost in the rush of new work. It helps each Scrum team member to identify how they can improve the specific things they contributed to the sprint, asking:

What work has been done well in this sprint?

What work hasn’t been done well?

What should we start doing to improve?

During each Agile sprint retrospective, the development team focuses on increasing product quality by improving work processes or adapting the definition of “done.” This definition may vary from Scrum team to Scrum team. But the whole team must have the same understanding of “done” to assess when work has met expected standards.

As Scrum teams gain more experience, their definition of “done” will evolve, including more demanding criteria for higher-quality results.

**SCRUM ROLES**

**· ScrumMaster**

A Scrum master is responsible for ensuring that the Scrum team follows the processes that were agreed upon. Keeping obstacles and distractions out of the team's path is one of the responsibilities of the Scrum master. In this role, the individual acts as the interface between the Scrum team and other people or teams.

A Scrum Master holds a position that’s relatively narrow in scope, yet extremely broad in influence throughout any organization. In practice, however, a Scrum Master is working behind the scenes and is not involved in product ideation or strategy. They work more as a conduit between product/line-of-business owners and development teams as a project manager. Because agile processes are entirely dependent on people and collaboration, Scrum Masters must also marry soft skills with the latest tools and methods. After all, software projects have many moving parts, and individual programmers can quickly lose sight of the broader scope when heads-down in code. A Scrum Master, on the other hand, maintains a high-level view, helping teams to understand both organizational and technical dependencies while avoiding chokepoints. This creates a culture of accountability and enables teams to meet critical deadlines.

Scrum Master Responsibilities

1. Implement Project Management/Best Practices

2. Keep all Parties on Track and Informed

3. Introduce Agile Engineering Practices

4. Coach Team Members

5. Host Daily Stand-up Meetings

6. Assist the Product Owner With the Product Backlog

7. Remove Roadblocks

**· Product Owner**

A product owner is responsible for ensuring the success of a project in Scrum. The product owner is responsible for managing and optimizing the product backlog in order to maximize the value of the product. A Scrum framework is an Agile methodology that facilitates communication and self-organization within a team.

A Product Owner is part of the scrum team. The key responsibilities of a Product Owner are to define user stories and create a product backlog. The Product Owner is the primary point of contact on behalf of the customer to identify the product requirements for the development team. This product backlog will be a prioritized set of customer requirements. The Product Owner has the complete responsibility and ownership of defining and even prioritizing user requirements. The Product Owner must communicate with the development team to explain the product features to be implemented. Any queries that come from the development team must be addressed by the Product Owner on key user requirements. The role of the Product Owner is to maximize the value addition of the products that are developed by the agile scrum team.

The Product Owner must ensure that the user stories meet customer requirements. The role of the Product Owner is critical for companies that are keen to move to an agile-based product development methodology. The Product Owner has to collaborate and work closely with various stakeholders such as customers, business leaders, development teams, project managers, and other stakeholders.

One of the main roles of a Product Owner is to manage the product backlog.

**· Development Team.**

The Development Team forms an integral part of a bigger Scrum team. It is comprised of professionals who deliver a potentially releasable Increment of “Done” product at the end of every Sprint. At the Sprint Review, a “Done” increment is required. Typically, only the members of the Development Team create this Increment.

Development Teams are structured well and empowered by the organization to effectively manage their own work. This results in a unique synergy that optimizes the overall efficiency of the Development Team.

Ideally, a Development Team should be small enough to remain ‘agile’ and large enough to complete a significant amount of work within a specific Sprint. This will result in a product of the best possible value.

There are some salient characteristics of a development team. The most important ones are listed below-

They are self-organizing teams. No one (not even the Scrum Master) directs the Development Team on how to convert Product Backlog into Increments of potentially releasable functionality.

Development Teams are generally cross-functional. It comprises of members with varied skills. As a team, these combined skillsets are necessary to create a product Increment.

Development Team members do not have individual titles. Each member is identified only as a part of the Development Team, irrespective of the work being performed by the person.

Scrum recognizes no sub-teams in the Development Team, although it may be formed of domains like testing, business analysis, operations, or architecture.

The Development Team as a whole bears accountability of a project, not individual team members.

What are the responsibilities of a Development team?

1. Perform Sprint Execution:

2. Inspect and Adapt:

3. Groom the Product Backlog:

4. Plan the Sprint

5. Inspect and Adapt the Product and Process:

**—------------------------------------------**

**Question 2**

**Function**: Yoga booking system.

**Features**: Very simple interface to accept user input (bookings) and display classes information.

**Data storage location**: SQL database named “Yoga classes”

**Requirements**:

One person works on one task.

TASK 1

1) Оn the main page, show the schedule of yoga classes in a weekly format: including the days, hours, and minutes of these classes, and the teacher who will conduct the classes.

2) Аdd a "book a class" icon to the site. Prompting the user to enter the desired class booking date.

3) Write the code in Python to check the availability of a free slot for the desired booking time.

When writing the code, consider the following:

1. each class can have a maximum of 10 people;
2. the program must return the number of free slots for the selected hour to the user
3. add only possible options from the database, in case there are no more places left for the class - do not show this date for booking
4. when booking a class, the program must ask "will the user be alone or with a guest";
5. to book a class, the user must enter his full name and email, in case the user will be with a guest - in addition, the name and email of the guest;
6. After completing the user registration, the program should automatically accept user data(and guests) in the order "last name, first name, email" into the SQL database and return to the user further information "Your reservation is confirmed. We are waiting for you (address, booking time)"
7. After confirmation the program should subtract the number of persons entered during booking from the number of free slots in the database and collect this data into the database.

4) Accept user data into the database in the separate table of all users and sort them in alphabetical order.

5) Develop code to remind the user about the class via email.

TASK 2

In case the user did not have time to reserve a place for the class, return to the user "sorry, unfortunately you are late, there are no free places for this hour. You can choose another class”.

Write code to return free slots to the user for a user-selected hour but on a different day or for classes on the same day at the closest time to the user's input. For example, if the user wanted to attend the class on September 25 at 8:00 p.m., offer him to book a class for September 26 at 8:00 p.m. or September 25, 7:00/9:00 p.m. If there are free slots.

TASK 3

test the ability of the program to count free slots correctly;

test the ability of the program to correctly accept user data in the order "last name, first name, email";

test the ability of the program to enter user data into the database and sort them in the database of all users in alphabetical order;

test the ability of the program to send an email with a reminder.

test the ability of the program to find the closest available time to the user's input.

**Notice**:

Tasks 1,2 can be worked on in parallel.

Task 3 needs to be worked on in particular order as soon as tasks 1,2 are ready.

**Activities**:

To handle the logic and manage the data flow, write a code in Python for above requirements.

**Duration of Sprint**: 2 week

**TASK 2**

| **requirements** | **challenges** | **tools** |
| --- | --- | --- |
| the schedule of movies in a day-by-day format | make sure that schedule data is updated every day due to possible changes and new premieres | User Interface (UI) |
| the possibility of booking a place to watch a movie through a separate "book" icon | quick update of the database with reservation confirmation to avoid situations of uncertainty for the last available seats | SQL DB |
| show the chart as a viewing hall with free seat icons | makе sure that the bar received a total list of previous orders | Python |
| the possibility of choosing a seat from three possible options: (VIP seats, seats in the middle of the hall, and back seats) | maximum visual identity of the scheme of the hall to the hall in reality | Secure payment system |
| request registration data from the user about user (and company): last name, first name, phone number, email | variety payment methods | Tableau/Power BI/Excel diagrams/Google charts |
| add the option of ordering popcorn and drink by default |  |  |
| offer the possibility to add an event to Google Calendar |  |  |
| offer the opportunity to receive emails about premieres |  |  |
| after registration, display the link for payment |  |  |