

Task 2

Select two user stories you just finished for your project; write **use cases** instead.

Would your team recommend use cases or user stories in these two examples? Why?

- Story #1 - Add Figma Photo Book Layout and Designs
 - Description: Create various photo book layouts and templates for our stakeholders to pick from. These layouts should offer some flexibility and customization in how the photo books will look like.
- Use Case #1 - Create Photo Book Layouts and Templates
 - Scenario: The scenario involves a graphic designer who wants to add various photo book layouts and designs to a digital design software system (Figma). The designer intends to provide stakeholders with a selection of customizable photo book templates.
 - Actors: Graphic Designer, Digital Design Software System
 - Actor's Intention: The Graphic Designer's intention is to design and create various photo book layouts and templates within the digital design software system. These templates should be visually appealing and offer flexibility to stakeholders who want to use them to create customized photo books with different looks and styles.
 - Observable Result: The observable result of this use case is the availability of a collection of photo book layout templates within the digital design software system.
 - Sequence of Actions: The System presents the designer with a blank canvas and a set of design tools, The Graphic Designer begins by adding design elements such as text boxes, image placeholders, backgrounds, and other graphical elements to create a photo book layout, The System stores the newly created photo book layout template in its database for future use, The designer can repeat this process to create additional layouts and templates, each with different designs and customization options.

In this case, we would want to use a user story and not a use case. The use case is helpful because it is structured and provides a step by step detailed description of what the designer should do in this situation. As this story is relatively simple for our designer to accomplish, we believe that using a story would be suitable for the situation. For a story that requires more complex instructions or descriptions, a use case would be more appropriate.

-
- Story #2 - Implement a Mock Database

- Description: The goal of this story is to create a mocked up database for user accounts in our software. The outcome should be being able to add all the user data that we need and creating a user account.
- Use Case #2 - Implement Mock User Account Database
 - Scenario: This scenario involves a software developer who aims to implement a database for user accounts within the software system (MongoDB). The objective is to create a functional mockup of the database that allows for the addition of all necessary user data and the creation of user accounts.
 - Actors: Software developer, Software System
 - Actor's Intention: The Software Developer's intention is to implement a user account database that serves as a mockup for the software system. This database should allow for the addition of all necessary user data and facilitate the creation of user accounts.
 - Observable Result: The observable result of this use case is the availability of a functional mockup of a user account database within the software development environment.
 - Sequence of Actions: The Software Developer begins by defining the structure of the user account database, including fields such as username, password, email, and user profile data, The developer configures the database to allow for the addition of user data and ensures that it can support the creation of user accounts, Upon completing the database setup, the Software Developer saves the database schema and configuration within the development environment.

In this case, we would want to use a user story and not a use case. As this is the first sprint of our project, we are aware that we are doing a lot of initialization and setup for the remaining sprints of this entire project. Many of the things we are doing in the first sprint are lower level tasks which will allow us to implement functionality later on. As this task is a lower-level requirement, we want to utilize a user story, as we do not require the high level description of actions to complete the task.

Try **planning poker** to estimate effort for your user stories. Let's say two developers on your team can't agree on an estimate. What are the options to resolve the problem?

- If two developers cannot agree on an estimate, we will be going with the average point if applicable, or the lower of the two estimates. Due to everyone's varying expertise and experiences, we do not expect planning poker to result in extremely accurate estimates. As a result, we will remain flexible to changing or increasing the points for the user stories if we find that the original estimate was

not accurate. We chose to do a lower point when stuck because pointing it higher may result in more inflated pointing.

Examine and highlight the distinctions between **Agile and plan-driven cultures** and the role of managers as discussed in Lecture 3 on slides 6 and 8. Which approach and details does your team prefer or find more favorable? e.g., Plan-driven: Managers assign teams (Only Himanshu likes it, he said it can avoid prolonged discussion); Agile Manager: Asking questions while allowing the team to create the solution (Luke, Renny and Annabelle like it, and agree that their scrum master should implement it)

- In the lecture slides, plan-driven cultures are usually characterized by clear and unchanging directions that are defined early on, and all decisions are made based on well-defined business cases. These cultures attempt to understand the entire problem and plan/analyze it before tackling it. On the other hand, agile cultures are very adaptable, embrace changes in a project's goals, and work in smaller increments to have a more rapid delivery. Agile does not work in all companies as teams can be unclear on what agile is, not have proper training, and can even be resistant to adopting its principles and practices. Our team prefers agile to plan driven cultures, as it allows us to be adaptable, focus more on our product for the client, and being able to deliver it quickly with high confidence.