Journal Entry Two – Product Owner

Melissa Chessa

Southern New Hampshire University

CS 250: Software Development Lifecycle

Professor Hicham Elhayboubi

03-23-2025

Journal Entry Two – Product Owner

Reflecting on this module’s assignments, stepping into the role of the Product Owner gave me a clearer understanding of what it means to advocate for user needs throughout the software development process. While I wasn’t officially functioning as a Product Owner in a real project, the experience of creating and prioritizing user stories helped me understand the importance of gathering feedback and translating it into features that provide real value.

One of the biggest takeaways for me was how important it is to engage directly with users or stakeholders. It’s not just about collecting ideas—it’s about building trust and showing that their input is valued. Listening to feedback early on allows the development team to focus on what really matters. For example, in the user meetings we reviewed, one user wanted more cruise options while another wanted to avoid tropical locations. These preferences seemed small at first, but they completely reshaped the way we thought about the destination list feature. Instead of just showing a static “Top 5 Destinations” list, we prioritized features like being able to hide destinations that don’t match a user’s preferences. That change made the software feel more personal and flexible, which is something many travelers expect now.

User stories played a big part in making that shift clear to the team. Writing the stories in a structured way—*“As a user, I want to... so that I can...”*—forced me to think about what people actually want and why. For instance, one of my stories focused on letting users hide vacation types they don’t like. It was simple, but that format helped me explain it clearly to the team, and it gave them a target to build toward. Well-written stories also made sprint planning easier because we could break them into smaller tasks and set clear expectations. As Mountain Goat Software (2024) points out, user stories aren’t just for capturing requirements—they’re tools for team communication and collaboration.

The mock interviews and user meetings from the assignment were especially helpful for writing more realistic stories. It’s one thing to assume what a user might want, but hearing their words directly gave me a much stronger sense of what to include. Without those conversations, I might’ve stuck with basic features instead of exploring ways to customize the user experience more deeply. Scrum.org (2020) reinforces that early user engagement helps teams avoid wasting time on features that won’t meet real-world needs.

Moving forward, I’d also recommend adding more feedback methods beyond interviews. Usability testing, for example, can show how people actually use the software, which sometimes reveals issues that don’t come up in conversation. Surveys can help gather feedback from a wider group, and analytics can tell us which features people use the most—or not at all. Concepta (n.d.) notes that using a mix of feedback tools gives teams a better chance to build features that are both useful and relevant.

Overall, even though this was a simulated Product Owner experience, it helped me better understand how to gather user input, write actionable stories, and contribute to building software that solves real problems. I’ve learned that the best user stories come from listening first and staying focused on what actually improves the user’s experience.

**References**

Concepta. (n.d.). How to define stakeholders for your software development project. Retrieved from https://www.conceptatech.com/blog/how-to-define-stakeholders-for-your-software-development-project

Mountain Goat Software. (2024). User stories and user story examples by Mike Cohn. Retrieved from https://www.mountaingoatsoftware.com/agile/user-stories

Scrum.org. (2020). User story or stakeholder story? Retrieved from https://www.scrum.org/resources/blog/user-story-or-stakeholder-story