Vimon Sok

02/24/2025

SNHU – CS 250

Professor: Ryan Woolever

Module Eight Journal

Directions

For this course, submit the **Sprint Review and Retrospective** deliverable from your final project. This deliverable pulls together the Agile and Scrum practices that you have applied throughout this course.

Once you have submitted this artifact, **update the README file** in your repository and include your answers to each of the questions below. These prompts are some of the “essential questions” for the Computer Science program. You will encounter these questions in different courses throughout the program. They are open-ended and meant to help you engage with “big ideas” in the field. You could include the questions and write a few sentences in response to each one, or you could write a paragraph or two weaving together all of your answers.

* How do I interpret user needs and implement them into a program? How does creating “user stories” help with this?

Interpreting user needs requires gathering requirements through user research, stakeholder interviews, and feedback loops. User stories help by breaking down complex requirements into smaller, manageable tasks that describe the needs from a user’s perspective. This ensures that development stays user-focused and aligned with real-world use cases. By implementing user stories, developers can create features iteratively and validate them with users throughout the process.

* How do I approach developing programs? What Agile processes do I hope to incorporate into my future development work?

My approach to developing programs follows an iterative and incremental methodology, starting with requirement analysis, followed by designing, coding, testing, and refining. Agile processes such as Scrum, Kanban, and continuous integration/deployment help in adapting to changes efficiently. I plan to incorporate daily stand-ups for better team communication, sprint planning for structured iterations, and retrospectives to continuously improve workflows in my future development work.

* What does it mean to be a good team member in software development?

Being a good team member in software development means collaborating effectively, communicating clearly, and contributing to a positive work environment. It involves actively participating in discussions, providing constructive feedback, and being open to feedback from others. A good team member also respects deadlines, supports teammates when needed, and embraces a mindset of continuous learning and adaptability.