Daniel Beale

Professor Konishi

CS 250

23 February 2025

Final Project

Transitioning from a waterfall development model to Agile with Scrum was a significant shift for ChadaTech. Our team was responsible for developing a booking system for a travel agency aiming to expand its customer base by offering trendy vacation packages. The client required the system to be functional within five weeks to capture early vacation planners. The goal was to create a streamlined booking experience for customers while helping the agency reach a broader audience.

Each role in the Scrum team played a key part in keeping the project on track, and their collaboration was essential to ensuring a smooth development process. The Product Owner worked closely with stakeholders to define business requirements and prioritize tasks in the backlog. The Scrum Master facilitated communication, removed obstacles, and ensured the team followed Agile best practices. Developers and testers collaborated continuously to ensure incremental delivery, validating each feature early in the process. Rather than working in isolation, these roles operated as a unified team, frequently aligning their efforts. The Product Owner and Scrum Master ensured that user stories were well-defined and prioritized effectively. Developers and testers worked together to break down user stories into smaller, testable components, ensuring clear acceptance criteria and test cases. Instead of waiting until the end of development to test the software, testing was integrated throughout the process, allowing for early bug detection and continuous feedback.

One of the biggest advantages of Agile is its ability to handle changing requirements without disrupting progress. In our project, we faced a major pivot when the client decided to integrate a third-party payment system midway through development. The Product Owner reassessed priorities, worked with stakeholders to define the new requirements, and adjusted the backlog accordingly. The Scrum Master facilitated discussions to ensure that the team remained aligned and that potential roadblocks were addressed early. Developers reworked parts of the existing codebase to accommodate the new API, collaborating with testers to ensure compatibility. Testers designed new test cases to validate payment transactions and ensure security compliance, working closely with developers to verify edge cases. Since Agile follows an iterative approach, this change did not derail our progress. Instead of waiting until the end to implement the payment system, we incorporated it into an upcoming sprint. The ability to reprioritize user stories and break down the change into manageable tasks allowed us to integrate the feature without missing deadlines.

A key part of Agile development is the creation and refinement of user stories, which represent features from the end-user’s perspective. A well-structured user story follows the format: “As a [user], I want [goal] so that [reason].” For example, one of our user stories was: “As a traveler, I want to filter vacation packages by budget so that I can find affordable options quickly.” To handle requirement changes effectively, the Product Owner and development team frequently refined user stories based on client feedback. Some user stories lacked sufficient detail initially, making it harder to define precise test steps. To address this, we held backlog refinement meetings where developers, testers, and the Scrum Master clarified requirements to ensure user stories were actionable.

Internal communication was essential throughout the project. Agile depends on frequent interactions to quickly resolve issues and adapt to change. Daily stand-ups kept everyone aligned on their tasks, surfaced roadblocks early, and allowed the team to adjust workload distribution when necessary. Sprint reviews and retrospectives facilitated continuous improvement by allowing the team to reflect on completed work and discuss potential refinements. Backlog refinement meetings ensured that user stories were well-defined, removing ambiguity before developers began implementation. Pair programming and code reviews further strengthened collaboration, improving code quality and ensuring that test cases aligned with business requirements. This culture of continuous feedback ensured that issues were caught early rather than late in the development cycle, significantly improving efficiency and product quality.

Scrum teams rely on software tools to manage progress and keep tasks organized. Jira plays a crucial role in tracking user stories, managing sprint backlogs, and monitoring development. The Kanban board in Jira can provide visibility into task statuses, helping the team identify bottlenecks early. Burndown charts allow you to monitor sprint velocity, ensuring that you stay on track to meet deadlines. Using story points instead of fixed time estimates improves planning accuracy and encourages discussions focused on effort rather than arbitrary deadlines. Jira streamlines task management and helps teams quickly adjust priorities when requirements change, making it a vital tool in Agile project management.

While Scrum offers many advantages, it also comes with challenges, particularly in balancing workloads and keeping up with constant feedback. Since Agile relies on continuous communication, scheduling meetings across time zones and availability could sometimes be difficult. As a tester, staying engaged throughout the process helped keep test cases relevant and effective. Automating repetitive tests freed up time for exploratory testing, which uncovered more unexpected issues. One of the key factors in the project’s success was focusing on Agile principles rather than rigid processes. Prioritizing adaptability and collaboration allowed the team to respond efficiently to changes. Transparency also played a crucial role in keeping everyone aligned and preventing miscommunication.

Ultimately, Scrum proved to be the ideal methodology for this project. Its iterative nature allowed for refinements, improved stakeholder engagement, and more efficient problem-solving. Compared to waterfall, where late-stage changes can be costly, Agile provides the flexibility necessary for a fast-moving development environment. This experience reinforces the value of Scrum for future projects that require continuous adaptation and collaboration. The ability to pivot and reprioritize tasks while maintaining development progress highlights why Agile is well-suited for modern software development.