Scrum Master Report for Team A-28 - Report

Introduction

This report documents our team's journey in developing a Cataloging and Lending App (CLA) for UVA students to borrow and lend textbooks. As the Scrum Master, I coordinated our team's efforts throughout multiple sprints, ensuring we maintained progress while addressing both technical and interpersonal challenges. Being a Scrum Master was not easy and required more hands-on efforts than other roles, as I had to not only manage four other people but also make substantial coding contributions to the codebase myself. This role pushed me outside my comfort zone and required implementing creative strategies to maintain excellent progress and proper communication between all team members.

Team Formation and Initial Organization

From the moment I was assigned to work with four previously unknown undergraduate students at the University of Virginia, I recognized the importance of taking initiative. Before meeting everyone in person, I sent an introductory email to all team members, requesting their contact information to establish communication channels. This proactive approach led to my unanimous selection as Scrum Master, a role that would prove both challenging and rewarding throughout our multi-month journey.

My first strategic decision was to create a comprehensive organization system through Google Drive, establishing dedicated folders for DevOps, Requirements, Scrum Master responsibilities, Sprint Reports, and Testing. At the project's outset, I downloaded all required templates and saved them in their respective folders, ensuring every team member had immediate access to necessary documentation. This early organizational effort set the tone for our project management approach throughout the development cycle.

Our team communication structure evolved to include both Discord and iMessage group chats, providing multiple channels for discussion and updates. We also finalized our team roles early on: Scrum Master: Zaid Contractor (myself), Requirements Manager: Rory Kretzer, Testing Manager: Surya Vemulapalli, DevOps Manager: Aidan Szilagyi, Software Architect: Dev Patel

This clear delineation of responsibilities helped establish accountability and allowed each member to focus on their area of expertise while still contributing to the overall codebase.

Initial App Design and Planning

Despite having to leave Grounds due to personal responsibilities early in the project, I maintained our momentum by scheduling a Discord call with the entire team. This 1-2 hour virtual meeting became instrumental in establishing our application's foundation. We collaboratively discussed potential layouts and user flows for our textbook lending application, with each member contributing perspectives on creating an intuitive platform for UVA students.

The session culminated in the creation of a wireframe diagram mapping user navigation between different pages. This visual blueprint outlined the primary journeys—from login through browsing textbooks, managing collections, and facilitating borrowing. The wireframe helped crystallize abstract concepts into concrete design decisions and revealed potential usability challenges early in the process.

By facilitating this planning session remotely, I demonstrated that effective leadership doesn't require physical presence. The shared blueprint minimized misalignments in later development stages and ensured each team member understood how their contributions would fit into the overall application structure.

Communication Strategies and Challenges

While our digital infrastructure proved effective, I now recognize a missed opportunity in not establishing regular in-person meetings. Throughout the entire project, our team met face-to-face only twice, which presented both advantages and challenges. On one hand, it demonstrated our ability to coordinate effectively across distances and manage parallel development efforts remotely. On the other hand, this limited physical interaction occasionally hindered deeper team bonding and immediate problem-solving that typically occurs during in-person sessions.

To compensate for the predominantly virtual nature of our collaboration, I implemented several key management strategies. Most significantly, I created a master task backlog Google Document where team members could claim specific responsibilities at the beginning of each sprint. I regularly reminded the team via our group chat to select tasks aligned with their interests and skills. This approach promoted ownership and clarity regarding individual contributions while maintaining overall project momentum.

Further enhancing our accountability system, I established a practice of sending individual direct messages to each team member throughout each sprint, checking on progress, offering support, and ensuring deadlines were met. This personalized communication proved valuable, particularly when team members became unresponsive in group settings. When messages went unanswered for extended periods, I escalated to phone calls rather than allowing progress to stall. This assertive yet supportive approach maintained our development pace despite occasional communication challenges.

One notable communication success was my role as the primary point of contact with our TA. I scheduled all Monday meetings and prepared comprehensive sprint reports beforehand, ensuring our team was properly represented and received valuable feedback. This structured approach to external communication complemented our internal systems and maintained alignment with course expectations.

Sprint Progression and Development Process

Our project followed a structured sprint cadence, with each sprint lasting approximately two weeks. Throughout our six sprints, we systematically built our application's functionality:

Sprint 1: Team Formation and Project Conceptualization

During this initial phase, our team met in person for the first time, established communication channels, and finalized roles. We discussed potential project ideas and ultimately decided on a textbook lending application. The sprint concluded with the initialization of our GitHub repository through GitHub Classroom and a clear understanding of our project's direction.

Sprint 2: Requirements Elicitation and Infrastructure Setup

This sprint was crucial for understanding user needs and establishing our technical foundation. We developed a comprehensive Google Form that was distributed to over 30 UVA students for requirements elicitation. Additionally, team members conducted in-person interviews to gather detailed feedback on desired features and constraints.

The requirements elicitation process proved invaluable, making necessary features much clearer by hearing directly from potential users. As one student commented in our survey, "I would love to be able to search for textbooks by class number rather than just title," which directly influenced our implementation of the Required Materials page later in development.

While the elicitation process was somewhat tedious, the dedication of all team members ensured we collected sufficient data to inform our user stories, which the Requirements Manager carefully documented in GitHub issues. Concurrently, our DevOps Manager successfully configured Heroku and deployed our initial Django application.

Sprint 3: Core Functionality Development

This sprint marked the beginning of substantive code development. Our Software Architect created the database structure to support different user roles (librarian and patron), our Testing Manager designed the librarian interface, our DevOps Manager implemented Google Authentication, and our Requirements Manager established the homepage template and placeholder pages for other views.

As Scrum Master, I managed overall planning and progress while ensuring alignment with our established user stories. I also examined all new commits on GitHub and our hosted web app to verify we were meeting sprint requirements and addressing the user stories from Sprint 2.

Sprint 4: File Storage and Enhanced Features

During this sprint, we expanded our testing suite, implemented GitHub Actions CI for continuous integration, and integrated Amazon S3 for file storage. This allowed patrons to upload profile pictures and librarians to upload images for items in the CLA. We also enhanced user accessibility, improved the UI, and added the ability for librarians to promote patrons to librarian status.

Sprint 5: Collections and UI Refinement

Sprint 5 involved implementing the Required Materials page, populating our database with actual UVA classes, expanding our testing suite, and fixing various bugs related to Google sign-in and profile pictures. We also significantly improved UI elements and developed public and private collections functionality, allowing users to organize items and control access permissions.

Sprint 6: Beta Testing and Final Features

Our final sprint focused on completing key features and preparing for beta testing. We finished the collections implementation (both backend and frontend), improved user profiles, enhanced item management, implemented the borrowing process, and created an anonymous user access mode. We also refined the UI with consistent footers and better formatting throughout the application.

Technical Challenges and Solutions

Throughout development, we encountered numerous technical obstacles that required creative problem-solving:

Database and File Storage Issues

One of our most significant challenges occurred when our database was accidentally deleted. While we were able to restore the database structure, all previously uploaded images were lost, requiring us to repopulate the

system. This incident highlighted the importance of regular backups and careful access control for production environments.

We also faced issues with Amazon S3 key rotation and restricted access. Since only our DevOps Manager had credentials for S3, other team members experienced friction when implementing or testing features that involved file uploads. This created a bottleneck in our development process and occasionally delayed progress.

Another persistent issue was the inability to access S3 files when testing locally. This forced us to develop and test certain features in isolation from the file storage system, introducing additional complexity to our integration process.

Version Control and Deployment Challenges

We experienced functionality differences between our local development environments and the live server hosted through our main branch. Team members with the latest code could run features successfully locally but encountered errors on the deployment server. Resolving these discrepancies required careful examination of GitHub commits and proper branch merging.

To address these issues, I led efforts to establish disciplined GitHub procedures, emphasizing proper branch creation for new features, consistent issue tracking, and adherence to git versioning protocols. We created a separate development branch where all new code was pushed before being merged to main, helping ensure the stability of our production environment.

Login System Implementation

A minor disagreement arose regarding whether we needed a dedicated login page at all. Some team members questioned its necessity, suggesting users could simply be redirected to Google authentication directly. As Scrum Master, I made the executive decision to implement a proper login page after observing successful implementations by other teams. This decision ultimately enhanced the user experience and provided a more professional entry point to our application, demonstrating the importance of occasionally making decisive leadership choices.

Beta Testing and User Feedback

Our beta testing stage proved crucial to the application's final quality. Each team member conducted comprehensive testing sessions with UVA students, following a structured script to evaluate the functionality and usability of our app. This process generated extensive feedback that guided our final refinements.

Some key improvements resulting from beta testing included:

- Redesigning the homepage to incorporate marketplace functionality with a tile for each item
- Enhancing the UI with more color, consistent navigation elements, and better visual hierarchy
- Adding specific user profile fields relevant to textbook lending (such as "major" instead of generic "interests")
- Improving the item view to show owner information and detailed textbook descriptions
- Implementing a more intuitive selection/deselection mechanism for items in collections
- Adding search and filtering capabilities to the Required Materials page based on academic major
- Refining the borrowed items display to show pending requests and clear "None" indicators when appropriate

These user-driven improvements significantly enhanced the application's usability and appeal. As one beta tester commented, "The color coding for item status makes it much easier to quickly see what's available," validating our attention to interface design.

Interpersonal Challenges and Team Dynamics

Working with previously unfamiliar team members presented natural interpersonal challenges. The two-week sprint structure often resulted in a pattern where significant progress occurred during the second week rather than being evenly distributed. This created periodic intensity that could have been mitigated with more consistent engagement throughout each sprint. While understandable given our busy academic schedules, this pattern occasionally created stress near deadlines.

Maintaining accountability across a distributed team required constant attention. My system of individual check-ins helped address this challenge, but I occasionally needed to be more assertive with team members who were slow to respond or fall behind on tasks. Finding the right balance between supportive leadership and necessary firmness was an ongoing learning process.

Our predominantly remote collaboration also presented challenges for building team cohesion and ensuring everyone understood each other's code contributions. Without regular face-to-face interactions, we needed to be exceptionally clear in our written communication and code documentation. This experience highlighted the value of occasional in-person meetings, even in teams that operate primarily remotely.

Recommendations for Future Teams

Based on our experiences, I offer the following recommendations for future teams:

Establish Regular In-Person Meetings

While remote collaboration can be effective, scheduling a weekly in-person work session would facilitate clearer communication, faster problem resolution, and stronger team cohesion. I recommend using a scheduling tool like When2Meet at project initiation to identify a time that accommodates everyone's availability. These sessions could incorporate brief stand-up meetings where each member shares progress, plans, and blockers, enhancing accountability and coordination.

Distribute Critical Knowledge

Ensure that mission-critical systems and access credentials are shared among multiple team members. In our case, having all DevOps responsibilities managed by a single person created potential risks. Future teams should ensure that at least two members have access to deployment platforms, database credentials, and other essential infrastructure. This includes providing shared access to the Heroku dashboard so that multiple team members can deploy and inspect CI/CD pipelines.

Implement Consistent Work Cadence

Rather than concentrating work in the final days of each sprint, establish expectations for regular, incremental progress. Even small daily contributions can maintain momentum and prevent last-minute rushes that may compromise quality. Consider implementing mini-milestones within each sprint to encourage steady progress.

Prioritize Requirements Elicitation

Invest substantial effort in requirements elicitation at the project's outset. Our survey responses and in-person interviews provided invaluable guidance for feature prioritization and design decisions. Gathering diverse perspectives from potential users helps ensure the final product meets real needs rather than assumed ones.

Consider Test-Driven Development

While our approach of building features before tests worked for our circumstances, future teams would benefit from adopting test-driven development practices from the beginning. This would improve code quality, reduce regression issues, and provide clearer documentation of expected behavior.

Conduct Thorough Beta Testing

Allocate sufficient time for beta testing with actual users and be prepared to implement feedback quickly. Our beta testing phase revealed numerous usability issues and improvement opportunities that weren't apparent to us as developers. Creating a structured testing script ensures consistent feedback across different testers.

Conclusion

Our team successfully delivered a functional and aesthetically pleasing textbook lending application that met all project requirements. Our final implementation includes robust login functionality with different user roles (anonymous, patron, and librarian), comprehensive item management, public and private collections, user profiles with customizable information, a borrowing request system, and required materials listings for UVA courses.

Throughout the project, the organizational systems I established as Scrum Master—including structured documentation, regular check-ins, and clear task allocation—contributed significantly to our project's success. I'm particularly proud of our team's ability to adapt to predominantly remote collaboration while still delivering high-quality results, often working late nights to ensure we met our commitments.

This experience has demonstrated the importance of proactive leadership, clear communication channels, and consistent follow-through in managing distributed software development teams. It has challenged me to develop creative solutions to coordination problems and to remain adaptable as project requirements evolved. The lessons learned throughout this journey—technical, interpersonal, and organizational—will undoubtedly serve as valuable resources for both our individual professional development and for future teams undertaking similar projects.

Most importantly, this experience has reinforced the value of balancing individual autonomy with team accountability, allowing each member to contribute their unique talents while maintaining collective progress toward our shared goal of creating a useful tool for the UVA community. As our project concludes, I am confident that the skills we've developed through this collaborative effort will continue to benefit us throughout our academic and professional careers, particularly in environments requiring cross-functional teamwork and complex project management.