SOEN390 - Software Engineering Team Design Project
Team 6 - Deliverable 2

Sprint 2 Retrospective: Reflecting on our CMS Project

Winter 2024

Done by:
Hoang Minh Khoi Pham 40162551
Michaël Gugliandolo 40213419
Jessey Thach 40210440
Mahanaim Rubin Yo 40178119
Vanessa DiPietrantonio 40189938
Ahmad Elmahallawy 40193418
Clara Gagnon 40208598
Khanh Huy Nguyen 40125396
Jean-Nicolas Sabatini-Ouellet 40207926
Mohamad Mounir Yassin 40198854

Professor Junqiu Yang
Department of Computer Science and Software Engineering
Gina Cody School of Engineering and Computer Science

Concordia University

Table of Contents

1. Introduction

In our ongoing venture, we're actively developing a Condo Management System, a digital solution crafted to streamline and integrate various facets of condominium living. Leveraging the SERN (SQL, Express.js, React, Node.js)stack and integrating Docker for deployment, our team is dedicated to forging a seamless platform for efficient condo management. Our primary objective remains to deliver a user-friendly and interconnected system that elevates the overall condo living experience.

Throughout the project, we're encountering a mix of successes and challenges, and this document will outline what's working well, areas posing difficulties, and opportunities for refinement in the final outcome. In this document, we will discuss what went wrong (why did it happen, the impact of it, addressing the issue and what can be improved. We will also discuss what went right (Why it happened, What was the impact, How we addressed it and what we think could have been better).

1.1 What went wrong

1.1.1 Backend Overhaul

- **Issue:** The backend team decided to revamp the entire codebase early in the project.
- **Why did it happen?** Lack of proper planning and communication regarding the initial architecture.
- **Impact:** Delayed development progress and increased workload for both frontend and backend teams.
- **Addressing the Issue:** Implemented stricter planning protocols for major architectural decisions and enhanced communication channels between teams.
- **What can be improved:** Establish clearer guidelines for architectural changes and involve all relevant stakeholders in decision-making processes.

1.1.2 Inflexible Task Prioritization

- **Issue:** Task prioritization was rigid and did not allow for flexibility in addressing emerging issues or opportunities.
- **Why did it happen?** The Lack of a dynamic task prioritization process and adherence to a fixed roadmap.
- **Impact:** Hindered the team's ability to adapt to changing requirements or user feedback promptly.
- **Addressing the Issue:** Introduced a more agile approach to task prioritization, allowing for frequent reassessment and reprioritization based on evolving project needs.
- **What can be improved:** Implement a continuous feedback loop and embrace a more iterative approach to task prioritization to enhance responsiveness and agility in project execution.

1.1.3 Communication Gap

- **Issue:** Insufficient communication between frontend and backend teams.
- **Why did it happen?** Lack of regular sync-up meetings and coordination efforts.
- Impact: Inconsistent integration of frontend and backend components, leading to compatibility issues.
- **Impact:** Introduced regular cross-functional meetings and established communication channels for real-time collaboration.
- **Addressing the Issue:** Introduced regular cross-functional meetings and established communication channels for real-time collaboration.
- **What can be improved:** Implement automated integration testing to identify compatibility issues early in the development cycle.

1.1.4 Task Division Imbalance

- **Issue:** Imbalance in the division of tasks within the web application development**.**
- **Why did it happen?** Inadequate assessment of individual skill sets and workload distribution.
- **Impact:** Uneven progress across different components of the system, leading to bottlenecks.
- **Addressing the Issue:** Conducted regular retrospective meetings to reassess task allocation and redistribute workload accordingly.
- **What can be improved:** Implement a skill-based task allocation system to ensure equitable distribution of responsibilities.

1.1.5 Difficult to organize a meeting where every single member is present

- **Issue:** We could not get everyone in 1 meeting to make sure everyone is on the same page in sprint 2.
- **Why did it happen?** Scheduling meetings proved challenging due to varying personal commitments, and diverse schedules among team members.
- **Impact:** The inability to have every member present hindered effective communication and decision-making during crucial project discussions.
- **Addressing the Issue:** Implementing a collaborative scheduling tool to identify overlapping availability and streamline meeting coordination.
- **What can be improved:** Establishing a standardized meeting schedule at the project's outset to accommodate diverse time zones and commitments, ensuring optimal attendance.

1.2 What went right

1.2.1 Consistent Naming Conventions

- **Why did it happen?** Adhered to consistent naming conventions across the project
- **Impact:** Enhanced code readability and maintainability, facilitating smoother collaboration among team members.
- **Addressing it:** Enforced naming conventions through code reviews and automated linting tools.
- **Improvement:** Continuously update and refine naming conventions to align with evolving best practices.

1.2.2 Test-Inclusive Pull Requests (PR)

- **Why did it happen?** Ensured that all PRs included comprehensive test coverage with a coverage above 80% in all metrics
- **Impact:** Reduced the likelihood of introducing regressions and improved overall code quality.
- **Addressing it:** Implemented automated code quality checks and mandated test coverage thresholds for PR approvals.
- **Improvement:** Enhance test coverage metrics and provide additional training on writing effective unit and integration tests.

1.2.3 Sprint Goal Achievement

- **Why did it happen?** Successfully covered all user stories outlined for the sprint
- **Impact:** Demonstrated team efficiency and progress towards project milestones.
- **Addressing it:** Maintained a prioritized backlog and conducted regular sprint planning sessions.
- **Improvement:** Refine estimation techniques and conduct more granular user story breakdowns for improved sprint planning accuracy.

1.2.4 Good Progression of Work from the Beginning

- **Why did it happen?**The project exhibited a consistent and steady pace of progress from its inception.
- **Impact:** Ensured timely completion of milestones and maintained momentum throughout the development cycle.
- **Addressing it:** Regularly monitored progress against project timelines and adjusted resource allocation as needed to maintain productivity.
- **Improvement**: Continue to emphasize the importance of pacing and consistency in work progression, ensuring that momentum is sustained across all stages of development.

1.2.5 Quick Decision-Making as a Team

- **Why did it happen?** Swift decision-making was adopted due to academic constraints and project timelines.
- **Impact:** Minimized delays, maintained momentum, and increased overall productivity.
- **Addressing it:** Established clear decision-making protocols, empowered team members, and maintained transparent communication.
- **Improvement:** Make sure that we fully understand the requirements as soon as the sprint starts to make decisions quickly.

2. Conclusion

In the process of developing our Condo Management System (CMS) in sprint 2, we navigated through a dynamic landscape of successes and challenges, each contributing to our understanding of collaborative software development. Our commitment to proactive team collaboration was the basis of our achievements. By valuing diverse contributions, we fostered an environment that enriched our learning experience and produced positive outcomes for our project. This collaboration proved to help us the most at achieving effective problem-solving and maintaining a positive team dynamic.

A critical lesson learned was the importance of a continuous and steady workflow in an academic setting. Recognizing the need for a disciplined approach, we successfully reduced stress, improved comprehension, and ensured a smooth progression of project tasks. Time management strategies became essential in navigating the academic nature of our project, and showed the significance of maintaining consistency and diligence in our work.

Effective communication, both within our team and between frontend and backend teams, played a big role in streamlining project outcomes. Utilizing university-provided tools and scheduling cross-disciplinary meetings facilitated enhanced understanding and improved collaboration. However, challenges in organizing meetings highlighted the need for better coordination, prompting us to explore better solutions to ensure optimal team attendance in future projects. As we conclude sprint 2, our team will shape our approach to future projects based on our strengths and challenges throughout this experience.