Team 17 - SurfsideSoftware

GitHub repository link

https://github.com/cse110-sp24-group17/warmup-exercise/

Video link

https://youtu.be/Tu51tL7PB4U

SWOT Analysis

Process Summary

Overall, our team is happy with what we were able to accomplish in the short amount of time given. Admittedly, the process was in some ways a little rushed, but we are glad that the warmup gave us the opportunity to identify this issue, among others, as areas of potential future improvement before we begin working on the bulk of the main project.

The main shortcoming of our process this time around was the ordering of our tasks. We did not spend as much time researching, brainstorming, and planning before jumping into the coding and raw development as we probably should have. Rather, many of those important pre-coding tasks, such as creating user stories or wireframing, were worked on in parallel to the development of the widget. That is to say, these things were happening at the same time – we worked on design tasks at the same time as development tasks. Having finished the first iteration of widget now, it is clear looking back that better planning beforehand would have made the process much smoother. We believe the main difficulty in planning came from our lack of meetings earlier on during the process. Knowing this now, one of our top priorities is to look into scheduling more meetings throughout each week. Online meetings are an option, and it is alright if not all members can be present as long as thorough meeting minutes are taken to help absent members catch up.

Though our parallel design-coding process certainly did not meet the ideal we now strive for, our team's strong communication during our work period mitigated much of the issue, allowing for us to effectively complete a satisfactory minimum viable product. We spent some time determining what specific features our product should have, and once that was decided, our team split into three main subgroups. One group worked on the design aspects, creating user stories, wireframes, and mock-ups in order to pass along to the other two groups, which would be used to guide and instruct their work. These two groups respectively worked on the main calendar display on the left side of the page, and the daily events list on the right side of the page. Once groups were finished with their sub-tasks, we were able to quickly combine everyone's work into a cohesive final product.

Our GitHub Actions and Testing pipeline aided in how smoothly the process went in the end. Before any coding work was begun, a comprehensive list of GitHub issues was compiled based on all the requirements we decided upon in our specifications document. Each sub-team used separate branches and made pull requests at significant milestones to signal their progress, with issues being closed alongside their respective pull requests. This approach helped prevent the chaos that may have been caused if everyone simply pushed to main, and incremental pull requests at smaller steps rather than a single bulky push also made the merging process easier. As part of ensuring technical robustness, our team practiced employing unit tests. Whenever a pull request is opened on the repository now, JavaScript unit tests are run automatically via GitHub Actions. The PR is marked "red" if any of the tests breaks, signifying an issue in the PR, as illustrated in an example of a bad PR here. This also helps smoothen out the workflow, especially when multiple teams have to work on separate tasks at the same time.

We were ultimately able to accomplish what we wanted to, and have a much better feel for what does or does not work well after completing this warm-up. Both the completed features for the calendar widget and the planned future improvements may serve as good points of continuation for our development on our final work journal, and we are excited to see what we are able to accomplish going forward.

Strengths - What we did well

- We managed to work very efficiently during our designated meeting time. We split the team into smaller sub-groups that each focused on a key part of the calendar widget or documentation/github management side of things. This way we were all focused on one medium sized task that did not depend on the others and could freely work.
- All members were able to stay open and flexible during the development process, allowing us to bounce ideas off each other to figure out the path we wanted to take as a team. Disagreements were not common during this warm-up, but when they came up they were resolved quickly through direct communication with each other.

- Our whole group was present at the designated meeting, allowing us to freely contribute as a group. Every member was able to productively contribute to the task, and no member was left out or idle during our work period.
- Our proactive risk management helped us stay on track by focusing on essential features first and adding optional ones later with any extra time we had. This was incorporated into our specs document – priority of sub-tasks was clearly defined, and we worked to complete tasks in order of such priority.
- Our communication throughout the process has been very effective. Everyone responded to messages promptly, and communication was clear.
- We worked under a structured workflow with GitHub Actions and the like, and all tasks were guided by a main requirements list we agreed upon before any other work was done.

Weaknesses - What we didn't do well

- Lack of frequent meetings resulted in a slow/late start. We had one designated meeting towards the middle of the week, but it was harder to coordinate before then. This meant not much work was accomplished before the middle of the week, either.
- Lack of a predefined development model and schedule made the research and development process inefficient.
- We could have focused more on research and user stories at the beginning of our process to really build from the ground up.
- We could conduct a more comprehensive post-project evaluation to discuss the testing details and draw conclusions on what we can improve in the future.
- We lacked a coordinated working schedule outside of our weekly meeting, which slowed our progress as working on an individual basis became more difficult.
- The requirements document, while successful in structuring our workflow, could have been more robust

Opportunities - Areas of potential improvement

- If we increase the number of times we meet per week, we can ensure that we can start earlier, reducing stress and giving us more time to develop the project further. Earlier meetings could further improve the brainstorming, design, and research aspects of the development cycle.
- Creating a solid foundational model/schedule for our development process could greatly improve the efficiency of our development cycle.
- We hope that conducting a comprehensive post-project evaluation – this SWOT analysis being part of such an evaluation – to discuss the development process, testing details, project performance and similar analysis will help identify further points of future improvement.

Threats - Areas that may cause issues going forward

- We currently only have one dedicated meeting time set. Though we were able to pull through on this warm-up, scheduling more meetings – and really we mean *scheduling*, not just having spontaneous meetups when convenient – will be crucial to our team's success going forward.
- Individual members' specific roles are still unclear. Going forward, it could be difficult to effectively delegate tasks to members without some kind of structured role system, which may then lead to bottlenecks in our development process. A meeting to address this should be held soon.