Tangelo Project Evaluation

This document is divided into two sections: first, a holistic evaluation of the positive and negative experiences of the project, including: team, skill growth, development, conflict resolution, and organization and structure; second, a reflection on key learnings from the project that the team hopes to carry forward.

Project Experience

Overall, the Tangelo team had a positive experience building the application out.

Team. One of the strong parts of the project experience was the overall sense of comradery between the team members. Meetings were usually enjoyable, with people cracking jokes and cheering successful feature launches together. Team members were able to draw upon their backgrounds and uniquely contribute to the project overall. Further, team members were able to effectively and fluidly respond to unexpected events, such as a member missing an important meeting or a critical bug emerging.

Skill Growth. At the start of project development, each team member described the types of work they wanted to do and the associated skills they hoped to gain; luckily, each team member was able to work on a part of a project that also developed a skill set they expressed interest in developing. The front-end presented some unique learning opportunities given the design direction we selected, such as having all elements appropriate scale/resize/stack depending on the screen and working with the Gridstack API. The backend provided opportunities to explore several new libraries and how to integrate them, such as SQLAlchemy, Celery, Redis, and data sourcing (BeautifulSoup, feedparser, etc.).

One negative experience that arises from the project is the amount of time that can be lost to debugging. While it is always valuable to learn via the debugging process, often too much time is spent trying solutions that don't work, only to find a fix after literal hours. Another that arose would be merge conflicts: with the nature of multiple people working on the same codebase they're bound to arise, but the process is no less frustrating.

Development. Development occured at a fairly consistent rhythm, leading to constant progress on the application. It was always exciting to pull the code and see what improvements await you upon launch. When the team stopped coding to prepare for the presentation/written component of the project, it felt strange to not have any additional updates. However, perhaps the markably weakest component of the project was the unequal contribution to development across the group. This disparity was felt in mastery of the application's function, in the time invested, and is even viable by the number of lines contributed overall (not that any can accurately measure one's contribution in a codestack, but rather a possible indicator).

Conflict resolution. Conflict inevitably arose, and the team's ability to speak openly about these conflicts created a much more positive experience. Team members were able to express their opinions and wants surrounding the project, with a compromise or common understanding being reached. The contribution disparity was one such example, as was issues regarding stylistic or structural directions. Conflicts were never personal and always conducted professionally.

Organization and Structure. Tangelo greatly benefited from strong leadership and organizational structure throughout the process. Meeting times and dates were decided upon well in advance with consideration for others conflicts; meetings were generally productive, and were sometimes cancelled if determined necessary for the time. Tangelo structured work by assigning goals to each member with an expectation to complete them by the next meeting. Frequent and open communication was another positive experience, leading to a strong awareness of project status and core issues that were being tackled at any given time. One negative component of the experience here was the misalignment between leadership title and function: namely, that one member of the group was officially "team leader" while another actually served as leader for a given period of time. This was addressed and remedied halfway through the project.

Core learnings

Some of the major lessons learned include:

Communicate Needs Early. Some conflicts that arose in the project could have been addressed earlier if individuals had raised them at their first instance, instead of ignoring them and letting conflict fester.. This would have made for an overall more positive experience.

Reach Out for Help. Given the scope and size of the codebase, it's practically impossible to be totally on top of all functionings. While it can always be beneficial to try solving the problem yourself, reaching out to teammates and getting their thoughts can dramatically accelerate development.

Set Clear Goals. Early on the Tangelo team didn't have clearly defined goals between checkpoints, which led to less productivity overall. Transitioning to our later, more structured work cycle allowed for better pacing overall.

Working Together. Tangelo realized that working together has some incredible benefits. You can troubleshoot code together, get a better sense of what your teammates are working on, and get a boost to overall productivity.

Planning and Designing Ahead. Tangelo designed and built much of the product as we went along. However, many pitfalls of the development process could have been streamlined or centralized if the team utilized wireframes, full database schemas, and/or fullstack process/data flows prior to starting to code.

As our team continues to evaluate Tangelo's potential to serve the Princeton community, and engage in discussions as to the platform's future development, we hope to use these learning moments as stepping stones to a more effective and joyful work environment, as well as opportunities for personal growth as we pursue our own academic and career goals.