March 14th, 2018

From: Spoiled Tomatillos Development Team

To: Spoiled Tomatillos Board of Trustees

Jose Annunziato

Nathaniel Derbinsky

Michael Weintraub

Re: Spoiled Tomatillos Project Team Assessment

We write to you, the Spoiled Tomatillos Board of Trustees, today in order to asses our team's experience and skills against this project. Below, the five members of our software development team each detail their experiences working on the project so far and any gaps they have identified in the implementation of the project along with an associated plan to address these gaps.

Daniel Chen:

I am most comfortable working in Java, as is true for most of the team. Choosing to use Python for this project was a conscious decision to put us in the situation to develop our adaptability and flexibility. Though I've had some experience in Flask through hackathons where I have to maximize fast implementation and deployment over "good design", this is my first time working on a serious project with it. This means I'll have to make sure I am putting out high quality code: tested, and documented; in a language I am not too familiar with.

Over the duration of this project I've forced myself to work outside of my comfort zone. I decided to take over a large part of the sysadmin tasks through running our production server. This has forced me to develop my comfort with the shell, as well as understanding web servers properly and how to set it up and maintain a healthy web server. I have generally taken over the tasks of making sure our production server is always up and has minimum downtime between build or otherwise.

I've also had the chance to work on my front end development. Since I've never formally been asked to do front end, I've always been comfortable working solely on the backend. However for this project, during this last sprint I had the opportunity to redesign our UI, trying to maximize the user experience and improve the aesthetic appeal of our site. I developed a few extra pages on that and still have been able to contribute to our backend and slightly in the design of our server too.

Our current code base is getting slightly out of hand and I think we need to go through and clean it slightly. We should also follow a consistent style guide across the team as I've noticed that while editing code written by a teammate, I run into too many style related issues. Overall we work extremely well as a team all seem eager to help out when a teammate is running into issues.

Jay Lok:

Coming from a more research oriented development background, many of the concepts and practices followed in contemporary software development were new to me as well as other members of the team. The majority of our experiences were academic in nature and our diverse backgrounds made it hard to converge on a common skill set.

The languages I am most comfortable with are C and ruby however this was not useful to the group due to those languages not being covered in a classroom environment. While we all knew java none of us were extremely comfortable with using Spring as the documentation made it more challenging. Thus we chose python and flask as our frameworks to build our project on. This created a whole new set of challenges that we did not anticipate, mainly that the courses support for infrastructure and development would revolve around java. As a result we had to dedicate time into learning how best to support our tools and frameworks of choice.

Matt Morgan:

For this project I found myself quite out of my element. While I have some fundamental experience with python for data analysis purposes, I had never written any legitimate software in it. We chose to write this project in python as a learning experience, and we recognized that this would provide many additional challenges. While I have never use flask before, I have put in extra time to research how to use the framework efficiently. A goal of ours is to use this opportunity not only to learn about the software development life cycle, but also as an opportunity to learn the flask framework and get some hands-on python development experience.

Another challenge we find ourselves facing is the use of a relational database to store our movie data. While a couple of us have take database design, we don't have much experience modeling real, messy, extremely large data like that provided by the IMDB database. I took it upon myself to tackle this issue because I'm interested in learning how to store such complex data. The process what complex and difficult, but I ended up creating a MySQL database (foreign keys and all) using that data provided by the IMDB database. I learned how to use php scripts to import massive files to sql, and I also learned how to use SQLAlchemy to model the data in Python. I think the hardest part was learning how to fit all the pieces together, but has been rewardings to tackle new technologies and start to see progress.

Veronica Shei:

As someone who has never had formal software development experience, I found the Spoiled Tomatillos project to be a challenging and exciting project for me to work on. I primarily worked on the front end aspect of this project, especially with user login pages and the user profile page.

While I have some experience developing in a Python/Django environment due to a prior project, I had never before used Flask. Luckily, the benefit of Flask is that it is lightweight and easy to build up from. This means that it was easy for us to get started on the project and to create our initial homepage. However, once we hit sprint 3, where we were all developing different features and pages like login, logout, index, user_profile, movie_page, and more, Flask's lack of structure made it harder to keep track of all these changes in comparison to a framework like Django.

I chose to do the front end aspects of this project so that I could relearn a lot of the skills that I had lost and further develop them. The main difficulty for me during this project was remembering my HTML/CSS skills. I had to relearn many aspects of UI/UX design that I had forgotten since my last web development project, meaning that my progress was a lot slower than it could have been. Luckily, my teammates all have strengths in different areas and were able to assist me with any of my inquiries.

Some of the main gaps that are left in our project is the ability to edit the user page, the ability to add in more data for the user like user description, user profile picture, user's favorite movies, and more. These will most likely be implemented through some sort of button/form on the user_profile page. This will serve as a good challenge for me to further flesh out my HTML/CSS abilities and the ability to link the backend to the front end. All of these tasks are currently reflected in our JIRA backlog.

Abel Shin:

As a second year student, I haven't had my co-op yet; I started with almost no experience with many of the tasks we dealt with in class (and for project) such as Jira and Jenkins, which my teammates seemed to know at least a little about it. Initially, I felt discouraged and wasn't sure how I could benefit the team. I had to put lots of extra hours at the beginning to make something that would take my teammates couple hours to finish. But as it went, I realized I was getting more comfortable with my team and the tasks I was dealing with, and I was actually making something that was beneficial to the team.

I chose to work mostly on back end / little front end regarding end-user since I wasn't too familiar with front end tools such as HTML/CSS. I was comfortable with Python but I have never really worked with Flask on a bigger project before, so I was worried at the beginning but fortunately, Flask is known for microframework where it was easier to learn. In this project I've really learned a lot on how actual software application was made and how everybody had to cooperate / work on different tasks.

One of the big challenges I’ve noticed was that it was getting harder to keep tracks of ideas and functions using Flask since Flask is primary aimed for small app development. I believe with issue trackers such as Jira and communication (using Slack, etc.), we could overcome this problem.