Scrum Project Management: Effective Software Engineering with a Small Team

CS4991 Capstone Report, 2023

Jacob Lear
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
jl7yst@virginia.edu

ABSTRACT

RevArt, an art service platform, wanted to modernize art management for artists by creating an online system, SAM, to organize artwork, payments, contacts and other important details. To create this system, I aided in adding a contact management service and implementing a financial subscription system. For the contact service, I implemented an online library of React UI components, AntD to make the page more user-friendly. The subscription system primarily utilized the Stripe APIs for handling online transactions. Weekly sprint meetings helped organize the team and get feedback on everyone's work to ensure efficiency and quality. By the end of the internship, the contact service was completed and the foundations were set for the subscription system. The next step was to complete the subscription system and polish the overall SAM application, which has now been launched.

1. INTRODUCTION

Have you ever wondered what helped bring about successful start-ups such as Uber and Airbnb? There was a time in software development history when each project was planned and executed by large teams. This plan-driven development is a strategy for producing predictable results, but it can become very costly if changes need to be made.

Smaller teams needed a strategy for software development that would allow for much more change. So, in 2001, seventeen software developers gathered together and wrote a manifesto for Agile Software Development (Beck, et al., 2001). This manifesto valued individual interactions and a shorter feedback loop from customers, allowing changes to be less costly, which resulted in teams using Agile methodologies to code, respond and change plans more quickly.

In the now-competitive software market, it is important for smaller groups to develop and get feedback more quickly to stay ahead. RevArt used an Agile methodology called Scrum. In Scrum planning, each person would get work for a sprint and finish it in one to two weeks. Each week I worked at RevArt, I would present the progress of the Contacts page or Stripe implementation, finding out my next steps, things to change, and bugs to fix.

2. RELATED WORKS

The Manifesto for Agile Software Development (Beck, et al., 2001) is especially important for changing the software landscape to what it is today. The document expresses four important values: "Individuals and interactions over processes and tools"; "Working software over comprehensive documentation"; "Customer collaboration over contract negotiation"; and "Responding to change over following a plan." Scrum planning was one of the earlier project

development methods that became popular since this manifesto was posted.

For more general information on Scrum, Lutkevich's article (2021) discusses the components to Scrum planning, and summarizes the basic components and vocabulary used. These include the term sprint, the time frame in which work is completed, and the sprint backlog, the collection of tasks yet to be finished. I used his explanations, diagrams, and conclusions to put my previous experiences at my internship in context.

3. PROJECT DESIGN

The overall project consisted of working on RevArt's Smart Art Manager (SAM) under a Scrum style of project management.

3.1 SAM Project

Together with a remote team of eight, we focused on developing the Smart Art Manager, or SAM, for RevArt's site. This system's goal was to provide digital management services for art. This included a digitized gallery, exhibition information, financial income and billing, documents storage, and more. Following the Scrum project management method, we used a website called Jira to track our project's progress and tasks. GitHub was used to manage the code and content for SAM. The ultimate goal was to launch SAM by the end of the internship. My specific goals were to complete the contacts and subscriptions pages.

3.1.1 Scrum Management for SAM

To use Jira, my internship advisor created a large number of tasks and added it to the site's backlog. Before a weekly sprint began, we would meet over Zoom to discuss these tasks or add new ones. Tasks included anything from bug fixing to creating new pages for features. Each team member would select their tasks to work on for the upcoming week's sprint. These tasks also had a number of points associated with them, which roughly translated to hours or difficulty. After the sprint ended, we returned to a meeting to discuss progress and show results.

At the beginning of my internship, I was assigned the contacts page of the SAM system. The Jira website had lots of features to organize tasks, so finding related tasks was always easy. After completing the contacts page, I was then assigned the subscriptions page. My manager did not have most of this page planned out, so a lot more of this section was spent experimenting with multiple solutions. This gave me a lot of freedom to throw in my own ideas.

3.1.2 Technical Side of SAM

In the team of eight, four of us were dedicated software engineers, one being our manager. We kept all of our programs on different branches of a GitHub repository. This allowed us to combine our code and keep track of any changes. Each of us ran the site locally on our computers for testing. Once our code was verified and added to the main branch, the changes would go out to the main RevArt hosted site.

The majority of the website was created using the React library for JavaScript. We also implemented Ant Design components, which were pre-made templates for sections of a webpage. Everything I did was on the front-end of the site, so I never altered anything that did operations on the database. The other interns and I had to use specific API endpoints created by our manager to interact with the back-end database, which was based on Django.

If we had any issues relating to our code or the API endpoints, we would use Slack to communicate with each other quickly. This was especially important with the API, since I had no access to its inner workings. All debugging was done through the browser console, with developer tools on Chrome, for instance.

For the contacts page, I ended up working on three modes: a table view, list view, and "business card" view. Each view had various filters and sorting methods. The "business card" view was my suggestion that I implemented using the most of our already created React components. I also added an add/edit page to handle new or existing contacts. To reach a user's data, I used API calls connected to buttons on the page. This ensured that contacts were maintained between sessions.

I worked on the subscriptions page in a very different way. Without much of a plan for making it, a lot of it was left to me. The primary requirement was that I use Stripe, a financial software service that could handle online transactions. Most of my work consisted of researching Stripe's API and understanding its functionality, rather than programming something concrete. Implementing it required back-end changes, so I could not do any coding until my manager made changes. This led to a slowdown in progress for SAM.

4. RESULTS

As a result of the initial planning and progress speed, I finished and polished the contacts page around halfway through my internship with minimal bugs. The subscriptions page, however, was not completed. I created multiple documents detailing strategies to connect the subscriptions page with back-end API in Stripe. For presentation purposes, I also created visual React components to show what the subscription page could look like once it was finished.

Because the work on the subscription page and polish on other work were not completed, the SAM system was not launched at the end of the internship as planned, though it did launch two months later.

5. CONCLUSION

Overall, my experience at RevArt gave me a lot of insight into working at a startup, working with Agile, and developing a website. From the very beginning, I used Scrum project development, an Agile methodology, and saw its effectiveness in our small team. It was much easier to organize work, even with a geographically spread out team. Working at a startup like RevArt gave me freedom to work on specific tasks in the way I preferred, and even allowed for my design suggestions to be implemented. On the technical side, my portion of the project utilized React, AntD components, and Stripe. From my programming experience, I learned the adaptability of React and usefulness of Stripe APIs.

6. FUTURE WORK

Since the SAM project was completed after my internship ended, the following are the steps that were needed to finish my personal work on the project. The next task I was to work on was fully implementing Stripe APIs into SAM's Django backend. This mainly consisted of setting up API listeners that would check the user in each page to acquire their subscription status. On the other side of this, the subscription UI on SAM's billing page needed to be polished and be functional by calling API endpoints.

To work on any website system based on a subscription model, I recommend Stripe and React. Stripe gives an adaptable and sizeable number of API endpoints for the handling of users' money and the website's responses. React is extremely useful for building UI for any website or JavaScript project. If starting from scratch, using an Agile methodology like Scrum would be optimal task management and responsiveness.

REFERENCES

Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifesto for Agile Software Development. agilemanifesto.org

Lutkevich, B. (2021, October 28). What is Scrum?. *Software Quality*. Retrieved September 29, 2023 from www.techtarget.com/searchsoftwarequality/definition/Scrum