How a Web Application can Better Serve Students during Lockdown
(Technical Paper)

How to Prevent Software Flaws in Finance Industry? Using ANT to Analyze the Relationship Between Software Engineers and Products
(STS Paper)

A Thesis Prospectus Submitted to the

Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements of the Degree
Bachelor of Science, School of Engineering

Ziyao Gao
Spring, 2022

Technical Project Team Members

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

ADVISORS

Hannah Rogers, Department of Engineering and Society

Rosanne Vrugtman, Department of Computer Science

Introduction

In 1994, Carmen Hermosillo published a widely influential essay online -- “Pandora's Vox: On Community in Cyberspace”. To illustrate, it began to be argued that “the use of computer networks had led not to a reduction in a hierarchy, but actually a commodification of personality and a complex transfer of power and information to corporations.” (Curtis, 2011). Since their introduction to the world in the 1980s, computers have been used widely in all walks of life. “In 1976, Steve Jobs and Steve Wozniak co-found Apple Computer on April Fool's Day. They unveiled Apple I, the first computer with a single-circuit board and Read-Only Memory.” (Williamson, 2021). Computers brought convenience, speed, and simplicity to all mankind. However, it has also brought an inevitable negative impact to the world: security vulnerabilities, due to computer software. Because finance companies know that a small mistake during a transaction may cause the company to lose millions of dollars, most finance companies hope and expect that, when there is a bug in the software systems or applications, software engineers can figure out a solution in a short time. In fact, software engineers are under a lot of strain at financial institutions, like banks, because of the high business dynamics–which means a technology that helps you look at your business scientifically and anticipate the future accurately. (Menzheres, 2021). If when hiring software engineers, the Human Resources department of a financial company can analyze the potential abilities of candidates more carefully and in more detail and, instead of just meeting the company’s recruitment needs, emphasize some of the security vulnerabilities to be faced in the future, then the company’s security problems, due to computer software vulnerabilities will be drastically reduced. In this research report, I discuss three possible ways to reduce the vulnerabilities of a software application in the finance sector by analyzing the social values, sense of responsibility, and ethics of software engineers.

Technical Project

The COVID-19 pandemic has caused endless harm and terrible inconvenience to almost everyone in the world. In 2021, I was one of five UVA students who built a web application for the Advanced Software Development class. As the group's Requirements Manager, I interviewed more than twenty individuals, who were students and faculty members of different majors. The interviews were conducted by Zoom and Google Form. Then, I analyzed the participants' responses to ensure that the major features in our web application could better serve UVA students during the COVID-19 lockdown. Using the Django framework, we implemented in Python the web application built with GitHub, Heroku, and Travis-CI platforms. The website has three major features: an online food ordering system, detailed information about Charlottesville public institutions, and a system to rate the local restaurants. These features helped local restaurants endure during the COVID-19 pandemic and provided students with more specific information about Charlottesville public service institutions. During the requirement and design stages, as the Requirements Manager of the team, my responsibility was to transfer the interviewers' ideas and needs about the features of the application to the other developers using a software development professional language. During the implementation stage, my job was to implement one of the three features, customer login, and payment APIs, and evaluate the overall quality of the implementation for the team. During the testing stage, the Testing Manager and I designed more than ten different types of test functions to ensure that the web application did not have any bugs or errors for deployment. Last but not least, during the maintenance stage, my job was to bring the actual web application demo to the interviewers, ask them to provide feedback and suggestions, and then make cogent updates and changes. The percentage of the change and optimization was about twenty. In summary, working on this project, I learned some knowledge

about web design, such as the basic skills for HTML, knowing how databases work, and hosting a web application on a web service host.

In conclusion, this group project was my first experience building a website using multiple platforms. The purpose of building the website was to provide more convenience to UVA students. However, I also improved my web designs and team collaborations skills. Although we met some challenges during the implementation stage related to the database, deployment, unit tests, and Google Calendar API. As a transfer student, I had no prior experience with Python programming language, which made the implementation stage even harder for me. For example, my goal was to come up with at least ten unit tests covering every functionality that I implemented. After wasting a couple of hours on debugging, I found the solution was to ask the Testing Manager to design several unit tests and walk through each of them with me first.

STS Topic

Due to the increasing amount of financial data, people no longer can thoroughly review and evaluate the data. Therefore, at incredibly low cost and high speed, machines step up to the task and perform financial data analysis (Laura, 2021). Although the financial industry has fully entered a very advanced and comprehensive technological process, various system vulnerabilities and security problems are still unavoidable. Once these disasters occur, they will not only have a huge negative impact on the company itself and its customers but also affect many people's views about the company and their investment concerns. For example, an automated trade execution system flooded the Chicago Mercantile Exchange's Globex electronic trading platform with a large sell order that caused the Dow Jones Industrial Average to plunge by almost 1,000 points in a half-hour, wreaking havoc on an already stressed market (Mearian, 2010). Because of the flaw in the implementation of the trading software, a large fundamental

trader initiated a sell order for 75,000 shares of stock worth about $4.1 billion. The trader's automated execution system sold 35,000 of those shares in just seven minutes (Mearian, 2010). Someone not familiar with investment or the stock market may not realize how serious the harm is that is caused by software security vulnerabilities. The following data represents the entire financial industry: Despite the fact that the financial services business does not have the largest security debt, one-third of the software used by financial firms (36%) includes high-risk defects. The most common vulnerability types detected within the sector are information leakage (66%), cryptographic flaws (61%), and code quality (58%) ("One-third of software (36%) used by banks has high-risk flaws, as reported in Veracode's State of Software Security Report", 2019). Although many of the software security vulnerabilities that were disclosed or discovered have been deleted, improved, or fixed by the software engineers, the cost was time consumption. In summary, I suggest that software engineers submit technical reports for every finished project, must provide documentation that records the detailed implementation plans, frequently design and update test cases for every single function or file, and write code formed only after their thought process.

Another alternative way to prevent software failure in the finance industry is to use Python program language when implementing code. Using Python can reduce operational risk by introducing automation to areas that previously involved manual handling of data. It is particularly useful as it allows users "visualization at every step in the development process" (Arias, 2018). Python, as a modern computer language, has not only multiple advantages, which are reflected in a more comprehensive data analysis software package, but also has excellent flexibility, readability, scalability, portability, and execution speed that are better than any other language. These advantages of Python allow software engineers and data scientists to analyze

efficiently and contribute more code. Python in finance is the leading programming language for performing quantitative and qualitative analysis. This language is involved in the development of payment and online banking solutions, in the analysis of the current stock market situation, in reducing financial risks, and in determining the rate of return of stocks (Laura, 2021).

My research question is “How to prevent software flaws and security vulnerabilities in the finance industry?”, which can be using Bruno Latour’s article Actor-Network Theory to analyze the connection and relationship between software engineers and software products. Bruno Latour’s article, “Actor-Network Theory,” is helpful and related to my research question because it emphasizes and focuses on the connections and relationships that are being made and remade between human and non-human actors. I chose this method because it is required to study all of the involved actor's perspectives and relationships with one another such as the software engineers, the software products, and finance companies in order to solve the research question. Moreover, because it is a novel approach that attempts to redefine actors not so much as willful or intentional agents, but instead as any entity—human or nonhuman—that in some way influences or perturbs the activity of a techno-social system – a system that is most effective when examining limited systems, such as ship navigation, electrical network failures, which are a similar issue to my topic (Latour, 2020). At the beginning of the 21st century, scholars from various fields, including anthropology and material practices, began to reconsider the nonhuman's agency. Jane Bennett's book, *Vibrant Matter,* which promotes an ANT mindset (Latour, 2020), is significant in this trend. In my research topic, the nonhuman factor is software programs which is a key component in the finance industry. ANT will apply to the subject because my research emphasizes the relationships and connections between the software engineers, software programs and finance companies are mutually influential and inseparable.

On one hand, the nonhuman actor software programs can only be created by the software engineers, the human actor software engineers rely on finance companies for living, and finance companies need software programs to be operated in order to make a profit. On the other hand, when software engineers implement software programs, they might also create software vulnerabilities. When finance companies hire software engineers, they can not guarantee that every hired engineer is high qualified due to the high demand. When software programs are operated by finance companies, the companies should understand the enormous impact of any small vulnerabilities. In summary, this article supports my research topic by showing that humans-nonhumans, in some way, influence or perturb the activity of a techno-social system.

Conclusion

The intent of this STS research paper is to analyze how to prevent software flaws in the finance industry and summarize my web application group project experience. The two important factors are: the software engineers should improve their social responsibility and be aware of the potential security vulnerabilities when writing software programs, and widely use Python program as the main language for finance industry software systems. My technical project helped me gain a broader knowledge of web design and API implementations. By completing the web application project, I gained greater skills in Python and HTML structures and platforms, such as Django, GitHub, Heroku, and Travis-CI.

References

All Watched Over by Machines of Loving Grace (TV series). (2021, May 17). Wikipedia. Retrieved October 16, 2021, from https://en.wikipedia.org/wiki/All_Watched_Over_by_Machines_of_Loving_Grace_(TV_series)

Arias, O. C. (2018, 10 23). *Balancing the Risks and Rewards of Python*. FINCAD. Retrieved 10 16, 2021, from https://fincad.com/blog/balancing-risks-and-rewards-python

Latour, B. (2020). *Actor-Network Theory*. OXFORDRE. https://doi.org/10.1093/acrefore/9780190201098.013.965

M, L. (2021, 1 5). *How Can You Use Python for Finance: Explained*. BitDegree. Retrieved 10 16, 2021, from https://www.bitdegree.org/tutorials/python-for-finance/

Mearian, L. (2010, 10 1). *Regulators blame computer algorithm for stock market 'flash crash'*. Computerworld. Retrieved 10 16, 2021, from https://www.computerworld.com/article/2516076/regulators-blame-computer-algorithm-for-stock-market--flash-crash-.html

MENZHERES, A. (2019). *What Software Development in the Financial Sector is Like*. eTeam. Retrieved 10 16, 2021, from https://www.eteam.io/blog/software-development-in-financial-sector

Timothy Williamson. (2021, 11 30). History of computers: A brief timeline. LiveScience. Retrieved 12 01, 2021, from https://www.livescience.com/20718-computer-history.html

Veracode. (2019, 12 3). *One-third of software used by banks has high-risk flaws*. IT in the SUPPLY CHAIN. Retrieved 10 16, 2021, from

https://itsupplychain.com/one-third-of-software-36-used-by-banks-has-high-risk-flaws-finds-veracodes-state-of-software-security-report/