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

A Research Paper submitted to the Department of Engineering and Society

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Ziyao Gao

Spring 2022

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

Hannah S. Rogers, Department of Engineering and Society

Abstract

Finance companies hire software engineers to develop, improve, and maintain software products to make profit. On one hand, the software engineers continuously update and optimize the software products to better serve the companies and public users. On the other hand, many of the software products have potential vulnerabilities due to the rapid development and fewer restrictions from the software engineers. This paper discusses the growth history of computers, the vulnerabilities of software programs in financial services, and the different ways to prevent computer software flaws in the financial industry, which will help to better understand the connection and relationship between software engineers and software products. Actor-Network Theory will be applied in the analysis part throughout the paper to comprehend that human and nonhuman forces in the social and environmental realms are linked in ever-shifting networks.

Introduction

In 1994, Carmen Hermosillo published a widely influential essay online -- “Pandora's Vox: On Community in Cyberspace,” which 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-founded 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). On one hand, computers brought convenience, speed, and simplicity to all mankind. On the other hand, security vulnerabilities and cyber attacks are the two biggest flaws brought by computers. 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. The Actor Network Theory method can be applied throughout the paper because the three major components of my paper are corresponding to the three principles of ANT: the social values correspond to the social, the sense of responsibility corresponds to the natural, and the ethics of software engineers corresponds to non-technological and technological. In this research report, I will discuss the development history of computers in order to show the irreplaceable role of computer programs and the increased demand for software engineers lay the basis for the points I will make, different software vulnerabilities, and three possible ways to reduce the vulnerabilities of a software application in the finance sector by analyzing the sense of responsibility and ethics of software engineers.

Actor-Network Theory Method

Humans have their own societies and communities; whereas, technology has its own network. Using Bruno Latour's article Actor-Network Theory (Latour, 2020) to analyze the connection and relationship between software engineers and software products is helpful and related to the research topic 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, to solve the research question, 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. 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 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 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, due to high demand, they cannot guarantee that every hired engineer is highly qualified. 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.

Rapid Development and High Demand of Software Products

Over a half-century ago, the development of computers started slowly. Throughout the last half-century from 1970 to 2020, the shift in computer development became more rapid. To illustrate this, on April Fool's Day in 1976, Steve Jobs and Steve Wozniak co-found Apple Computer. They unveiled Apple I, the first computer with a single-circuit board and ROM, Read Only Memory (Timothy, 2021). According to the image of "History of Computers," the evolution and revolution of computers brought convenience to the world because their size and weight became smaller, memory and screen quality became better, and the execution and internet speed became faster.

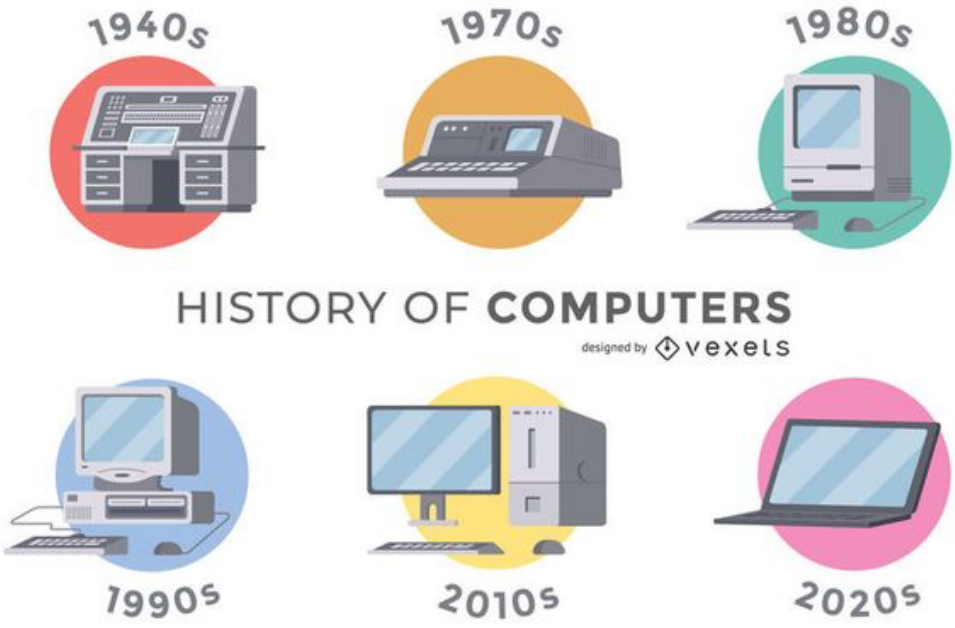

Figure 1: History of computers timeline design (2020, May 11)

In 1949, Jay W. Forrester and his team at MIT constructed the "Whirlwind" for the US Navy's Office of Research and Inventions. The Whirlwind is the first computer designed for real-time work; it can do 500,000 additions or 50,000 multiplications per second. This allows the machine to be used for air traffic control. In 1977, Mitsubishi, Sony, and Hitachi showed the first digital audio disc prototypes at the Tokyo Audio Fair. In 1997, there were 650,000 web servers in existence ("A Computer Timeline"). The reason why computers have developed faster and faster over the past half-century is the huge increase in the demand for computer software in many industries. To illustrate this, for industries such as finance, companies are looking for developers and engineers with just the right mix of skills and experience. At financial firms, those skilled in machine learning, A.I, and other disciplines can expect to make far more than the baseline, even in relatively entry-level positions (Nick, 2019). The average salary for software engineers is already above the average salary compared to most other jobs. However, due to the high demand for software designs and products, the software engineers who work in the financial industry have a higher salary rate compared to the other software engineers who work in other industries such as healthcare, retail, and agriculture.

Finance is one of the highest-demand industries for software developers. Notably, many financial professionals rely on programming languages for risk management, pricing, and trade management programs. With a high need for security and mathematics, the finance industry is in high demand for software developers ("Top Industries with High Software Development Demand", 2021). 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). The next section

will be discussing the potential software flaws in the financial industry with some specific examples.

Software Vulnerabilities in the Financial Industry

Although the financial industry has fully entered a very advanced and comprehensive technological process, various system vulnerabilities and security problems are still unavoidable. I believe that it is the nature of this technology because the fast development of computer programming can often ignore some potential issues during the development phase. In 2018, the finance industry experienced 19% of all cyber-attacks and incidents, making it the most targeted industry in the world ("5 Cybersecurity Weaknesses in Banking and Finance", 2019). Bugcrowd (a crowdsourced security platform) released its 2022 Priority One report to spotlight the key cybersecurity trends of the past year, including the rise in the adoption of crowdsourced security due to the global shift to hybrid and remote work models, and the rapid digital transformation associated with it. In particular, financial services companies on Bugcrowd experienced a 185% increase in the last 12 months in high-risk critical vulnerabilities ("185% increase in high-risk vulnerabilities within the financial sector", 2022). A lot of financial companies require their software engineers to build software in a short amount of time in order to make a fast promotion to their clients, which leads many of them to use open source (which means anyone can see, modify, and distribute the code as long as they are publicly accessible) from any random place on the internet. In the highly regulated financial industry, technologies are increasingly reliant on open source components. As open-source reshapes both the development processes and the products, we are creating a new set of security risks (Ayala, 2021). The actor network theory treats the human and non-human equally as agents since the separation between those elements is difficult. To illustrate, when talking about software vulnerabilities we could wonder if they result

from the human interactions (software engineers) or from the technology (software system). It seems difficult to differentiate between the technical and human aspects from the way in which a software development team responsible for the technique is influenced by our social-cultural background. 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 physical damage and different ethical issues caused to the society seems to be harmful and endless.

Software Vulnerabilities Prevention Strategies

In reality, all software vulnerabilities were created by software engineers, whether unintentional or with less care. Although some of the issues were due to hacker attacks, most of the flaws were exposed due to the poor design of the software engineers. The following image suggests some of the potential prevention ways that software engineers can take. The six software security practices are used to detect a breach or to respond to the detection of vulnerabilities once the product is deployed, and these six practices are used, on average, by 48 percent of the firms (Michael, 2022).

Practice	Usage (%)
Create or interface with incident response.	84
Track software bugs found in operations through the fix process.	76
Have an emergency code base response.	72
Use application input monitoring.	45
Use application behavior monitoring and diagnostics.	4
Fix all occurrences of software bugs found in operations.	4
Average	48

Figure 2: Vulnerability response practices (Michael, 2022)

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. Teamwork is a key aspect for software engineers. If the engineers who work on the development team could more frequently collaborate with the security team, it will help them find and fix more flaws before a product is deployed to the public. The frequent, incremental changes brought forth by DevSecOps (means development and security teams) make it possible

for these teams to fix flaws lightning fast compared to a traditional development team (Paul, 2019).

Another alternative 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).

Last but not least, being an ethical software engineer when implementing and designing software products is another strategy to prevent software vulnerabilities. If we take a closer look at actor network theory, a software engineer as an agent for ANT isn't just an actor but an association of heterogeneous elements themselves constituting a network to look like a single point actor. For example, all of the software engineers can be simplified as an agent that is opened and the contents reconsidered. This fits the idea that networks are always reliable because they can influence the outcome and the power systems. Software engineers are in a needed profession and have the ability to make a stand and be heard (Rotem, 2019). Ethics is a key principle that every software engineer should follow because it tells the difference between

good or bad and right or wrong when designing. Being an ethical software engineer means that he/she is an honest, trustworthy, and responsible software developer. Profession software engineers shall advance the integrity and reputation of the profession consistent with the public interest, it began as an unrealistic attempt to define bugs as unethical (Ritva, 2020). For example, if the software engineers ask themselves if this code makes a negative impact on the company or its users or if this piece of code has a potential issue that might place some people in a disadvantaged situation after they have designed a software product. Then, they are ethical software engineers.

Counter Argument

Computers, having been developed and improved faster and faster over the last fifty years, are being applied to every industry. Finance industry, one of the industries that has the highest demand of software engineers, brought many more software vulnerabilities than any other industry. Software engineers must be more careful and follow ethical rules when designing software products. However, people are criticizing that the vulnerabilities are not the software engineer's fault, due to various reasons. First, the high demand of software products makes finance companies require or even force their software engineers to build new products in a short amount of time, which means giving them less time to design and even no time to check for flaws after designing. So shouldn't the company take the responsibility when finding a software product that has flaws? Second, all software engineers are being hired through the human resource department and being placed into different software teams. A software product needs collaboration between all of the teams to make it to work. If it's only one person's fault, but we couldn't find out from the whole team, shouldn't we criticize the hiring team to be more careful and make stricter rules when hiring software engineers? Third, even though software engineers

have studied and are trained in software development knowledge professionally, there are also some great hackers who have been trained professionally. Engineers must work on multiple software projects, but hackers just need to be focused on attacking one specific project. Therefore, why should the software engineers be criticized for not being able to prevent an attack?

Conclusion

The intent of this research paper was to give a broad understanding of the rapid development history of computers, notice the different types of software vulnerabilities in the financial industry, and analyze how to prevent software flaws in the finance industry. The three important factors for preventing software flaws are: software engineers should improve their social responsibility and be aware of potential security vulnerabilities when writing software programs, widely use the Python program as the main language for finance industry software systems, and be an ethical software engineer when implementing any software product. Although the demand for software engineers will continually increase for the next era in the finance sector, the ethical issue awareness and social responsibility of a software engineer will be much stronger and stable than now. The Actor Network Theory helped me come to that conclusion, because the fundamental objective of the theory is to investigate how networks of relations are built, maintained, and become more durable through time.

References

(n.d.). Computer Timeline. Retrieved March 28, 2022, from http://webspace.ship.edu/cgboer/computertimeline.html

., L. M. (2021, September 10). *Using Python For Finance: Analyze Financial Data the Smart Way*. BitDegree. Retrieved March 28, 2022, from https://www.bitdegree.org/tutorials/python-for-finance/

Arias, O. C. (n.d.). *Balancing the Risks and Rewards of Python*. FINCAD. Retrieved March 28, 2022, from https://fincad.com/blog/balancing-risks-and-rewards-python

Barcenas, R. (2020, February 29). *Why ethics is very important in Practicing software engineering in the industry?* AskingLot.com. Retrieved March 28, 2022, from https://askinglot.com/why-ethics-is-very-important-in-practising-software-engineering-in-the-industry

Being an Ethical Software Engineer. (2019, May 27). InfoQ. Retrieved March 28, 2022, from https://www.infoq.com/articles/ethical-software-engineer/

Curtis, A. (n.d.). *All Watched Over by Machines of Loving Grace (TV series)*. Wikipedia. Retrieved March 28, 2022, from https://en.wikipedia.org/wiki/All_Watched_Over_by_Machines_of_Loving_Grace_(TV_series)

Farrington, P. (2019, June 20). *How banks can use more secure software to protect their future*. ITProPortal. Retrieved March 28, 2022, from https://www.itproportal.com/features/how-banks-can-use-more-secure-software-to-protect-their-future/

5 cybersecurity weaknesses in the banking and finance industry. (n.d.). Swivel Secure. Retrieved March 28, 2022, from https://swivelsecure.com/solutions/banking-finance/5-cybersecurity-weaknesses-threats-in-banking-and-finance-industry/

Goldstein, A. (2021, March 25). *Top 3 AppSec Challenges to the Financial Industry*. WhiteSource. Retrieved March 28, 2022, from https://www.whitesourcesoftware.com/resources/blog/top-3-appsec-challenges-to-financial-industry/

History of computers timeline design #AD , #AFFILIATE, #affiliate, #computers, #timeline, #design, #History | Computer history, Timeline design, History education. (n.d.). Pinterest. Retrieved March 28, 2022, from https://www.pinterest.com/pin/749779037958644805/

Kolakowski, N. (2019, December 17). *What Software Engineers, Analysts Earn at JPMorgan, Big Finance Firms*. Dice Insights. Retrieved March 28, 2022, from https://insights.dice.com/2019/12/17/software-engineer-analyst-pay-jpmorgan-finance/

Latour, B. (2020, 9 28). *Actor-Network Theory*. Oxford Research Encyclopedia of Literature. Retrieved 3 28, 2022, from https://oxfordre.com/literature/view/10.1093/acrefore/9780190201098.001.0001/acrefore-9780190201098-e-965

Martinez, M. (2022). *50 years of Software: How It Began, Where It's Going*. IEEE Computer Society. Retrieved 3 28, 2022, from https://www.computer.org/publications/tech-news/trends/50-years-of-software

Mearian, L. (2010, October 1). *Regulators blame computer algorithm for stock market 'flash crash'*. Computerworld. Retrieved March 28, 2022, from https://www.computerworld.com/article/2516076/regulators-blame-computer-algorithm-for-stock-market--flash-crash-.html

Menzheres, A., & Grytsai, V. (2018, June 8). *What Software Development in the Financial Sector is Like*. eteam.io. Retrieved March 28, 2022, from https://www.eteam.io/blog/software-development-in-financial-sector

185% increase in high-risk vulnerabilities within financial sector. (2022, January 18). Security Magazine. Retrieved March 28, 2022, from https://www.securitymagazine.com/articles/96926-185-increase-in-high-risk-vulnerabilities-within-financial-sector

One-third of software (36%) used by banks has high-risk flaws, finds Veracode's State of Software Security Report. (2019, December 3). IT Supply Chain. Retrieved March 28, 2022, from https://itsupplychain.com/one-third-of-software-36-used-by-banks-has-high-risk-flaws-finds-veracodes-state-of-software-security-report/

Top Industries with High Software Development Demand – Social Trends. (2021, September 6). Sites at Penn State. Retrieved March 28, 2022, from https://sites.psu.edu/socialtrends/2021/09/06/top-industries-with-high-software-development-demand/

Williamson, T. (2021, December 1). *History of computers: A brief timeline*. Live Science. Retrieved March 28, 2022, from https://www.livescience.com/20718-computer-history.html

Zobel, J. (2016, May 31). *The history of computing is both evolution and revolution*. The Conversation. Retrieved March 28, 2022, from https://theconversation.com/the-history-of-computing-is-both-evolution-and-revolution-57126