How Compliance Standards have affected the Software Industry

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

Vance Elliott

Spring 2025

Advisor
Rider W. Foley, Department of Engineering and Society

Introduction

As the world shifts to be increasingly reliant on computers, the need to ensure software quality is spotlighted. While testing strategies continue to develop, and the number of quality assurance employees grows at a steady rate, now almost at 200,000 employees, software is still plagued by defects and vulnerabilities (Zippia, 2024). When a piece of software plays a crucial role in society, this can be a very dangerous problem (Joseph et al., 2024). One example of this is the most recent Microsoft outage, caused by a defect in a security system called CrowdStrike. Due to a lack of sufficient testing architecture, a faulty update was allowed to be sent out globally, resulting in the crashing of many windows systems (Cohen, 2024). This error is estimated to have cost businesses billions of dollars (Davis, 2024). Thus, software has a large impact on the world, but a lack of a significant testing infrastructure that allows this technology to succeed.

Particularly when it comes to electronic health records (EHRs), poor testing frameworks can even lead to the loss of life (Bowman, 2013). The National Institute of Standards and Technology reported 18,738 defects that they alone discovered in 2021, and as the amount of software increases, the number of vulnerabilities will only trend upwards (National Institute of Standards and Technology, 2022). Software defects are often reported as the cause of an accident in hospital incident reports (Howe et al., 2018). Thus, there is an urgent need for healthy testing practices.

When it comes to testing software, there are many important principles and procedures that lead to quality software. One of these terms is Test Driven Development (TDD), which refers to a method of developing software that incorporates testing at every stage of the development process (Baresi et al., 2006). When using TDD, one will write a suite of test cases, essentially the requirements that the developer wants their code to complete, and then writes code until all of those tests pass. Rather than just testing after the software is finished, a technique called Unit Testing, TDD ensures that software is working from the very beginning of the development process. This process creates more efficient and effective software with one study from North Carolina State reporting that 92% of developers believe that TDD yields better code, 79% thought TDD results in simpler design, and 71% thought that it was noticeably effective (George & Williams, 2003).

While most developers recognize that this is a more effective strategy, few actually put it into practice. One survey conducted in 2020 by Vanson Bourne, a research agency, found that only 41% of developers practice TDD. When asked whether they write test cases before they write code, the very definition of Test Driven Development, only 8% of developers said yes (Diffblue, 2020). This is due to TDD being a timely and difficult process (Parsa et al., 2025). Software developers are either too lazy or not educated enough about the benefits of TDD to implement this testing strategy, causing their code to have a significantly larger number of defects (Makinen & Munch, 2014). Due to a lack of effective testing strategies and architecture, crucial software to society, specifically healthcare software, continues to fail, harming the people who rely on these technologies (Howe et al., 2018). Through analyzing software through the framework of the Social Construction of Technology, the impact that ineffective testing practices have on the quality of healthcare software will be studied.

Sociotechnical Analysis

Analyzing the testing processes of a company, one can see how the influences of different social groups guided the creation of different tests and the extent to which they were conducted (Baresi & Pezzè, 2006). This idea of social groups informing technology was proposed first by Wiebe Bjiker, who alongside Trevor Pinch, authored this framework “The Social Construction of Technologies” (Bijker et al., 1987). Using the invention of the bicycle, Bjiker proposed that technologies can be described by the influence of the social groups surrounding them. One can look at the forces of different stakeholders and see how their interests and desires molded the technology. This is a useful framework to keep in mind as more testing processes are analyzed. Poor testing strategies are often a result of a lack of social pressure to test, causing companies to cut corners as they do not have ample incentives to test well. Because of deficient motivation for quality assurance employees, and the growth in complexity of testing software, there is an alarming rise in software defects and vulnerabilities that negatively impact people, a problem that requires an urgent solution (Howe et al., 2018).

Bjiker’s Social Construction of Technology will allow an in-depth, thorough analysis of the effectiveness of a testing system. By looking at the surrounding stakeholders of a company, one will be able to determine the quality of a company’s software. For example, at a healthcare software company, one of the surrounding social groups of that business is the United States government. This is because that company’s software handles sensitive medical information, and the government desires to protect that data, passing many laws and regulations to promote information privacy such as the Health Insurance Portability and Assurance Act in 1996, otherwise known as HIPAA (U.S. Department, 2013). Because of the stricter policies surrounding healthcare companies, they must test more rigorously and thoroughly to ensure compliance with these standards. Thus, using the social construction of technology, one can look at a company’s invested stakeholders, determine whether testing is an important value amongst any of the groups, and then accurately predict the emphasis that business places on testing.

Biker’s theory will help with the correct analysis of other companies’ testing systems, particularly when the exact internal structure of a company is unknown. Three analytical components will be used to analyze the results of this thesis: relevant social groups, interpretive flexibility, and stabilization. As relevant social groups invest in and create technology, there start to become certain guidelines and norms with how that technology is made, a phenomenon Bjiker calls stabilization. With that stabilization of technology, there also comes certain flexibility and uncertainty with how companies will interpret these regulations, another aspect of Bjiker’s framework that he calls interpretive flexibility. We will use these two concepts, along with relevant social groups, to determine how to best encourage test driven development. By looking at the social forces surrounding software testing, as well as the ways they determine and interpret testing regulations and standards, one can begin to discern the health of current testing strategies. Therefore, by using Bjiker’s Social Construction of Technology, we can analyze a testing system in a helpful way, not ignoring the outside stakeholders, but rather incorporating them into the analysis to create more robust, effective solutions.

Case Context

There are many potential solutions to software developers’ lack of desire to implement proper testing practices, but this thesis will be looking specifically at code regulation documents. The six most common software compliance standards include: the General Data Protection Regulation (GDPR), ISO 27001, HIPAA, the Payment Card Industry Data Security Standard (PCI DISS), FedRAMP, and the Federal Information Security Management Act (FISMA) (Ali 2024). Each of these outline a different set of rules and guidelines for code, and have a significant impact on how often and effective software is tested.

Firstly, the General Data Protection Regulation (GDPR) is a rigorous privacy and security law enacted in 2016 by the European Union (EU). This standard of data privacy applies to both businesses that are located within the EU, and businesses that are not located within the EU, but do offer goods and/or services in the specified region. Fines for breaking the GDPR are high, maxing out at either 20 million euros or 4% of a company's global revenue, whichever is highest. The GDPR is primarily concerned with how a business processes and stores data, emphasizing the need to minimize the amount of data being stored and to become more communicative with consumers in how their data is being used. It is an 88 paged document, and discusses in length numerous security and privacy issues (Wolford 2025).

Next, the ISO 27001 is an information security standard document published by the International Organization for Standardization (ISO) in partnership with the International Electrotechnical Commission (IEC). It is a part of a series of documents that outline best practices surrounding information security, and is a leading international standard in the software industry. The three principles of the ISO 27001 are confidentiality, integrity, and availability, and emphasizes the keeping of records related to company procedures and practices. While this is not a standard that is legally required of companies, becoming ISO 27001 compliant can lend credibility to a company and ensure the safety of the company's information (International Organization of Standardization, 2013).

Thirdly, the Health Insurance Portability and Accountability Act (HIPAA) was a security regulation enacted in 1996 for specifically medical information. The privacy rule outlines how people can use private health information (PHIs). It describes several security policies that must be in place to ensure safe data, and maintains that all access and use of information should be disclosed to the patient and limited to only a necessary amount. HIPAA applies to anybody interacting with or using sensitive medical information. Compliance with HIPAA is required by law in the United States (U.S. Department, 2025).

Next, the Payment Card Industry Data Security Standard (PCI DISS) was enacted to ensure the safety and integrity of payment card transactions. Compliance is not legally enforced by the US government, but companies like Visa and Mastercard will impose fines for non-compliance. Compliance is tested by three major tests: a self assessment questionnaire (SAQ), a qualified security assessor (QSA), and an internal security assessor (ISA). PCI DISS is primarily concerned with building strong and durable systems, protecting credit card data. and regularly monitoring systems. There have been a lot of mixed reviews about PCI DISS, some saying that it is incredibly expensive to implement and very confusing to comply with, while others praise the effects it has had on the payment industry (University of California, 2023).

The Federal Risk and Authorization Management Program (FedRAMP) is a government program that also seeks to ensure protection of data in the cloud. It attempts to standardize how one interacts with, monitors, and authorizes information online. All federal agencies are required to be compliant with FedRAMP. FedRAMP bases its compliance standards on many different documents, one of which is the OMB Circular A-130 (Federal Risk, 2025). Similarly, the Federal Information Security Management Act (FISMA) details regulation on how to prevent cyber threats. It focuses on how to best conduct risk assessments and how to make sure one's systems are secure enough (Centers for Medicare, 2025).

Research Question and Methods

As the relation between healthcare software and patient care is analyzed, one question will serve as the focal point for the paper: How do current testing regulations impact the quality of medical software? In order to answer this question, I will perform a content analysis on the seven major policy software compliance standards: the General Data Protection Regulation (GDPR), ISO 27001, HIPAA, the Payment Card Industry Data Security Standard (PCI DISS), FedRAMP, and the Federal Information Security Management Act (FISMA) (Ali 2024). Using SCOT, I will analyze the strictness and effectiveness of each compliance standard, analyzing how these policies and regulations influence testing systems.

In order to conduct a proper analysis of these policy documents, I will use a set of words that relate to software efficiency, consisting of both positive and negative connotations, and test to see the number of times each document contains the words. This set will be broken up into two parts, one group attempting to determine the social group each article is addressing and the other attempting what specifically each document is attempting to improve about software. The first group, attempting to ascertain the social groups addressed by the policy documents, consist of the words: companies/company, government, people/persons/humans, employee, attacker, hacker, country, state, citizen, consumer/buyer, vendor/seller, distributor, and developer. The second group is made up of several categories of words: safety, punishment, privacy, test driven development, and legality. The grouping of these words is shown in table 1. These lists of words are by no means exhaustive, and do not have an even number of words across categories. These are only meant to help show which words are being referenced in policy documents, helping to get a sense of the intentions behind these standards and helping to direct any research in the future. Through performing a content analysis with this set of words, one can begin to see the aims of each document, specifically what each code standard is attempting to improve. One will then be able to look at each document and discern its effectiveness at improving the quality of medical software, and whether these seven major compliance standards are sufficient code regulation for EHRs. Using SCOT, specifically the three analytical components relevant social groups, interpretive flexibility, and stabilization, we can analyze the impact and reach of these documents, and if needed, propose a solution for a more effective set of code standards.

Table 1. Table of Words Sorted by Category

Safety	**Punishment**	**Privacy**	**Test Driven Development**	**Legality**
safety	penalty	privacy	code	freedom
health	fine	security	test	right
		data	quality	crime
		sensitive	development	offence
		personal		principle
		information		rule
		protection		legal/lawful
		distribution		standard
		sharing		
		transaction		
		payment		

Results

Through analyzing six major policy documents, the number of mentions to test driven development have been determined to be minimal (except for one document), while the references to privacy are relatively high. The documents tested included the 2016 GDPR official legislative act, the 2022 ISO version, the official HIPAA document from 1996, the 4.0 version of the PCI DSS document from 2022, the revised version of the circular A-130 document from 2016, and the FISMA annual report from 2016. These documents are all relatively recent (or as in HIPPA's case, still very relevant to software production), and show the general state of each software standard. Most of the documents had a high number of the word "person" or "persons", while most did not have a high number of any of the other "social group" words such as government, state, and employee. This would imply that each document is primarily geared towards the everyday user of software, rather than a more specific demographic. There are a couple of notable exceptions to this, mainly HIPAA and PCI DSS, that will be expanded on later in this analysis. As for the words "test" and "quality", both in the "test driven development" category, they are barely mentioned within these documents, aside from the PCI DSS which has 1283 instances of the word "test". This implies that the majority of policy documents are concerned with the impacts of software quality, rather than development and creation of code, supported also by the relatively low uses of the word "development".

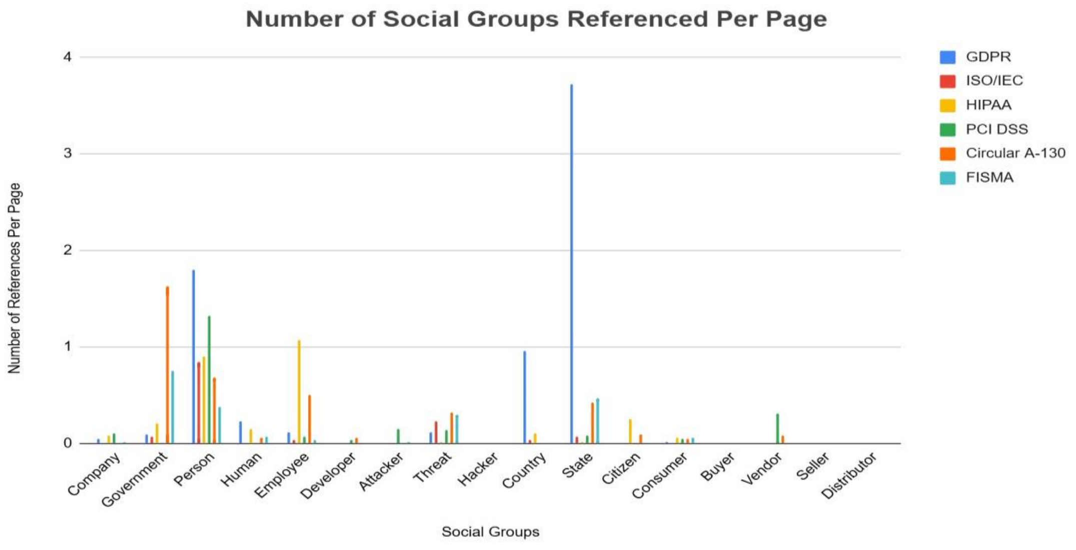

Figure 1. Number of Social Groups Referenced Per Page

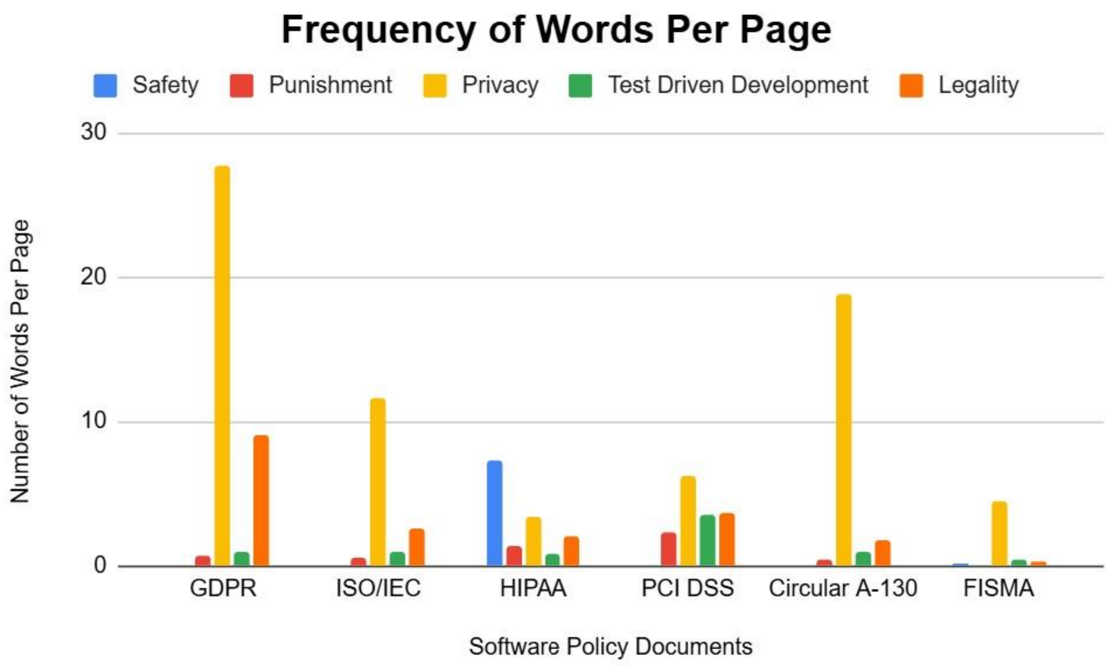

Figure 2. Frequency of Words Per Page

Specifically for the GDPR document, there were a large number of words in the category of security and legality. There were 27.75 words per page in the category of security, and 9.11 words per page in the category of legality. This is the third smallest document, at 88 pages, yet shows a significantly high percentage of the usage of these words compared with other documents. There were 0 mentions of the safety category, and low references for punishment and test driven development. As the GDPR is a document created by the EU to encourage the protection of data, it makes sense that privacy and legality are the main categories referenced. The social group most referenced by far is "state", emphasizing that this document's intent is directed towards the governing bodies under the EU, the relevant social group to this specific document. This also demonstrates stabilization, as this is the formalization of different coding standards, however, this movement of stabilization does include many references to test driven development. The GDPR is more focused on security and protection of the people, rather than the software development process.

Next, for the ISO 27001, there was a significant frequency of the categories privacy and legality, though not as much as the GDPR, which also makes sense considering ISO is an international standard for information security. The ISO is the smallest document with only 26 pages of text. The social group mentioned the most frequently was “people/person” with a number of 22 mentions (or 0.85 per page). No other group reached above 2 mentions. Thus, the relevant social group is the generic software consumer, not particularly companies, governments, or distributors. The small size of the document allows for more interpretive flexibility among companies, meaning that these regulations might not be as effective as other documents. There are very few mentions of test driven development, though still some, showing that this document is also not concerned with the development cycle.

Next, for HIPAA, the largest category referenced is safety with a frequency of 7.32 per page, with a slightly smaller frequency for privacy (3.41 per page). The HIPAA document is the second largest, with a page number of 169 pages. The highest social group referenced is “employee”, followed by “person”. As HIPAA is primarily concerned with health care, safety of patients and privacy of sensitive information is at the top of its priority. Thus, the relevant social groups are the employees handling information, and the patients involved in the software. This is the most relevant to healthcare software, but even this has very little mention of test driven development. This emphasis on outward metrics allows companies flexibility within their testing strategies (interpretive flexibility), to test or not test their software. If companies are presented with those options, they probably will choose the easier of the two, leading to a standardization of software procedures that does not properly enforce healthy software practices. Thus, there should be a higher frequency of software developers and test driven development mentions, to make sure developed software is healthy.

Next, for PCI DSS, the highest category used is “security” with 6.22 times per page, followed by test driven development and legality with 3.57 and 3.63 times per page respectively. This is by far the highest usage of test driven development words, more than 3 times the next closest ratio, which is the ISO 27001. The PCI DSS is the largest document, containing 397 pages. The highest social group referenced is “person”, the second and third being “attacker” and “company”. Notably, PCI DSS is the only document that references “attackers” (besides FISMA which only has one reference). The PCI DSS is concerned primarily with credit cards, thus the relevant social groups are people who have credit cards, companies that store them, and attackers that want people’s information. These standards take into account the software development process, encouraging testing more than any other document. This means that companies have less flexibility in how they should test their software, improving the odds of a healthy software testing strategy.

Next, the Circular A-130 talks about privacy more than any other category, with a frequency of 18.93 per page. This is the second smallest document, containing 85 pages. The largest social group mentioned is “government”. This is a document created by the United States government, and thus shows the primary interests of the government in relation to software. This is consistent with GDPR, created by the EU, which showed a high proportion of uses for the words “data” and “personal”, demonstrating an interest of the government to protect sensitive information of their respective citizens and governments/states. This shows again that this policy document is primarily concerned with the impact and effect of code, rather than proper development strategies.

Finally, for FISMA, the highest usage of words was for the security category, with a frequency of 4.52. This document had 95 pages, the third largest of the group. The largest social group mentioned is “government”. Again, this is consistent with the previous US government policy document, showing the emphasis on protecting the government’s and citizen’s private information. Using SCOT, one can see how the most common relevant social group, the government, is pushing for stabilization of guidelines centered on security rather than the development life cycle.

Thus, specifically for healthcare software, these policy documents do not amply encourage test driven development. The only document that had a large number of references to testing was the PCI DSS, a document primarily concerned with credit card transactions. While this is important, and still relevant in healthcare software as billing is a huge aspect of EHRs, this shows that the only policies that attempt to promote healthy software development are concerned with payment, rather than the user’s experience. Using SCOT, one can see how the national concern of data privacy has positively influenced policy documents to protect patient information. While this is the main concern of HIPAA, many other documents also had several regulations involving data privacy. However, the lack of an interest from social groups for proper testing led to a significant reduction in standardization centered around test driven development, and too much interpretive flexibility in how companies can test.

Discussion

This research shows how companies are able to get away with poor development strategies. The policy documents are primarily concerned with results, not the methods in getting those results, which leads businesses to focus on getting their metrics “up to par” with current regulations, not their internal structure. The lack of direction in regards to testing leads to a wide range of testing practices, what Bjiker calls “interpretive flexibility”, in that software developers are having differing interpretations of the policies surrounding testing created by invested social groups. This shows how only 8% of developers actually practice TDD. If most policy documents surrounding software are primarily concerned with the external results of businesses, then one can expect software developers to have the same blindspot for development.

The limitations of this research are that these policy documents, while significant in the software community, are a small number of the actual amount of regulations that exist. While one could expect for there to be similar results when testing other policy documents, one would need to test them to make sure. Smaller, more niche documents are also more likely to focus on the development process, thus might slightly differ from these six documents. Also, the list of words was not a comprehensive list and might be somewhat lacking in understanding the motive and impact of these policies. In the future, I would test more documents to get a broader view of the role of policy in software development, and I would poll different companies to see how often they interact with these regulations. It would be interesting to see the opinions and views of different software developers when it comes to the effectiveness of software policy documents.

I will use this research to better advocate for proper testing practices. While most developers may not see the first step to dissuading potential software defects as changing policy, through this research, one can now see how the language and politics of organizations impacts the view developers have on testing. As code becomes increasingly more complex, and AI poses even more inconsistencies and uncertainties in regards to output, remembering this battle can be fought at the legal level is incredibly important (Ali, 2024). As I join the workforce, I now have an in-depth understanding of the policies and regulation documents enacted to help developers, so I can best use them to encourage proper development practices in my work. This research helped me see clearly the advantages and shortcomings of software policies, and how I can best work with them moving forward.

Conclusion

Thus, when it comes to healthcare software, proper development practices are crucial in ensuring the welfare of patients. We as a society need to be doing everything we can to encourage testing, specifically test driven development. This can only happen by being informed citizens who engage in dialogue surrounding these issues, advocating for more policy surrounding the development process, rather than external metrics. We need many different social actors at each level. We need developers seeking to make quality code, policy makers who understand the complexities of software development, and citizens who can support both of these groups, all to ensure the health of patients dependent on EHRs. The solution is not "stop having bugs", rather a complex intersection of policy and software development, a solution of which anyone can contribute, not just software engineers.

Resources

Ali, N. (2024, June 3). *Navigating the maze: What you need to know about software compliance standards*. Beagle Security. Retrieved from https://beaglesecurity.com/blog/article/software-compliance-standards.html

Baresi, L., & Pezzè, M. (2006). An introduction to software testing. *Electronic Notes in Theoretical Computer Science*, 148(1), 89-111. https://doi.org/10.1016/j.entcs.2005.12.014

Bijker, W. E., Pinch, T. J., & Hughes, T. P. (Eds.). (1987). *The social construction of technological systems: New directions in the sociology and history of technology*. MIT Press.

Bowman S. (2013). Impact of electronic health record systems on information integrity: quality and safety implications. *Perspectives in health information management*, *10*(Fall), 1c.

Centers for Medicare & Medicaid Services. (2025). *Federal Information Security Modernization Act (FISMA)*. U.S. Department of Health and Human Services. https://security.cms.gov/learn/federal-information-security-modernization-act-fisma

Cohen, J. (2024, July 23). CrowdStrike's faulty Windows update causes BSOD issues. The Verge. https://www.theverge.com/2024/7/23/24204196/crowdstrike-windows-bsodfaulty-update-microsoft-responses

Davis, S. (2024, February 12). Explaining the largest IT outage in history and what's next. TechTarget. https://www.techtarget.com/WhatIs/feature/Explaining-the-largest-IToutage-in-history-and-whats-next

Diffblue. (2020, May 21). *2020 DevOps and testing report*. Diffblue. https://www.diffblue.com/resources/2020-devops-and-testing-report/

Federal Risk and Authorization Management Program. (2025). *FedRAMP: Federal Risk and Authorization Management Program*. U.S. General Services Administration. https://www.fedramp.gov/

General Data Protection Regulation (GDPR). (2018, May 25). *GDPR info*. https://gdpr-info.eu/

George, B., & Williams, L. (2003). An initial investigation of test driven development in industry. *Proceedings of the 2003 ACM Symposium on Applied Computing* (SAC '03), 1135–1139. https://doi.org/10.1145/952532.952753

Howe, J. L., Lammers, E. J., & Baker, S. (2018). Electronic health record usability issues and potential contribution to patient harm. *Journal of the American Medical Association, 319*(12), 1276–1278. https://doi.org/10.1001/jama.2018.1171

International Organization for Standardization. (2013). *ISO/IEC 27001:2013: Information technology—Security techniques—Information security management systems—Requirements*. https://www.iso.org/standard/27001

Joseph, Sb & Ness, Pedro. (2023). The Relationship Between Computers and Society, Impacts, Challenges, and Opportunities. https://www.researchgate.net/publication/376642881_The_Relationship_Between_Computers_and_Society_Impacts_Challenges_and_Opportunities

Makinen, Simo & Münch, Jürgen. (2014). Effects of Test-Driven Development: A Comparative Analysis of Empirical Studies. Lecture Notes in Business Information Processing. 166. 10.1007/978-3-319-03602-1_10.

National Institute of Standards and Technology. (2022). *CVSS severity distribution over time*. National Vulnerability Database. https://nvd.nist.gov/general/visualizations/vulnerability-visualizations/cvss-severity-distribution-over-time#CVSSSeverityOverTime

Parsa, S., Zakeri-Nasrabadi, M., & Turhan, B. (2025). Testability-driven development: An improvement to the TDD efficiency. *Computer Standards & Interfaces, 91*, 103877. https://doi.org/10.1016/j.csi.2024.103877

University of California, San Francisco. (2023). *Understanding Payment Card Industry Data Security Standard (PCI DSS)*. UCSF Controller's Office. Retrieved May 2, 2025, from https://controller.ucsf.edu/how-to-guides/accounting-reporting/understanding-payment-card-industry-data-security-standard-pci

U.S. Department of Health and Human Services. (2013). *Health Information Privacy (HIPAA)*. https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/combined-regulation-text/index.html

U.S. Department of Health and Human Services. (2025, March 14). *Summary of the HIPAA Privacy Rule*. https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/index.html

Wolford, B. (2025). *What is GDPR, the EU's new data protection law?* GDPR.eu. https://gdpr.eu/what-is-gdpr/

Zippia. (2024, April 5). *Software tester demographics and statistics in the U.S.* Zippia.
https://www.zippia.com/software-tester-jobs/demographics/