Integrating Cybersecurity into Software Development Processes

CS4991 Capstone Report, 2025

Bryant Lisk
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
rhq6nu@virginia.edu

ABSTRACT

Cybersecurity incidents can be incredibly damaging to both a company's reputation and revenue, as well as potentially disastrous for the users of that company's services. To address this issue, companies should take a more proactive approach to Cybersecurity instead of leaving it as an afterthought. I outlined and reviewed various methods of integrating cybersecurity into existing development processes. I found that having dedicated teams for cybersecurity and managing a proper escalation pipeline for identification of cybersecurity threats are essential to improve cybersecurity in applications. More work needs to be done to study the methods covered by this paper being put into practice, both on the efficiency of integration into the workflow and in detection and handling of threats.

1. INTRODUCTION

Strong cybersecurity is incredibly important to our modern internet and the applications we use every day. With vast amounts of data being stored on servers, it is increasingly important for that data to be protected properly. Beyond just the data, it is also very important that accounts themselves are kept secure, as hackers who gain access to accounts can hijack popular platforms and use the trust of their audiences against them.

Agile development has been the main approach to developing most software since the Agile Manifesto was published in 2001. While Agile development is great for moving fast on building software, it is less suited for the cybersecurity needs of the software, which is more suited for plan-driven development. This means that there is conflict between focusing on moving fast in development and keeping things secure. I aim to examine which practices of integrating cybersecurity into existing workflows work best at both achieving security and avoiding friction during development.

2. RELATED WORKS

Salin & Lundgren (2022) outline five major challenges in agile development for cybersecurity. These include the conflict between the plan driven nature of tackling cybersecurity tasks with the incremental development of an agile project; making sure the cybersecurity risks are aligned with the incremental progression of an agile project; making sure security risks are adequately identified and addressed; formulating and establishing security requirements between developers and customers; and having the proper incentives to develop more secure software. Salin & Lundregen analyze how these challenges are addressed in different papers and propose a framework to integrate cybersecurity risk management into agile development, which they call *Risk Refinement*.

Lenhart et al. (2020) tackle the challenge of how to best integrate cybersecurity development into existing development methods. They note that one of the main

challenges is the lack of acceptance by teams. They also recognize that conflict often arises between security and usability, and generally a balanced approach must be taken. Lenhart et al. ultimately recommend that new roles specifically dedicated to cybersecurity be created, along with a process that involves feature definition; threat analysis and risk assessment; outlining the cybersecurity concept; formally defining the cybersecurity requirements; and outlining a cybersecurity testing and data protection concept.

3. PROPOSAL DESIGN

To properly assess the models presented by Lenhart et al. (2020) and Salin & Lundregen (2022), we must first understand their processes. Once the models are properly summarized, I will compare what they do and attempt to identify strengths and weaknesses in the proposed models.

The first step of *Risk Refinement* (Salin & Lundregen, 2022) is called *Risk Collection*. This step involves a daily walkthrough of new, current, and unmitigated risks along with a quick initial assessment and prioritization of the risks as a natural extension of the daily stand-up. *Risk Collection* aims to address the conflict between the nature of agile software development and cybersecurity risk management by making thread detection more in line with other agile ceremonies. The next step is called *Risk Refinement*. This step, to be done weekly or biweekly as an extension of the regular backlog refinement in an agile project, involves more deeply assessing the risks identified in *Risk Collection*. *Risk Mitigation*, a process that lasts for a whole sprint starting with planning, involves turning identified risks into actionable and testable user stories, then taking steps to mitigate those risks in the code. The step called *Knowledge Transfer*, done during sprint retrospectives, aims to draw conclusions from the work done in *Risk Mitigation* by inviting security experts and others to walk through risk solutions. This helps broaden the knowledge of a team and allow for the whole team to have a greater understanding of risks that may come up in future sprints and their solutions. Finally, the step called *Escalation* is a process that involves reporting activity and high risks to management with escalations in an iterative matter. It is important to make sure that management is aware of serious risks that could seriously impact budget and prioritization. It is important to note that these steps are not necessarily in sequential order. Salin & Lundregen define a chart (Figure 1) to explain the flow process between these steps.

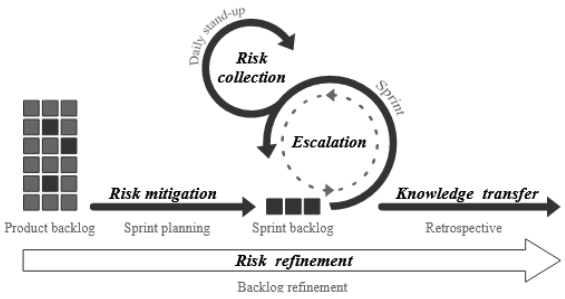

Figure 1: Risk Refinement Diagram

Basing their cybersecurity recommendations on their experience in the automotive industry, Lenhart et al. (2020) propose a process that also involves the creation of four new roles for a cybersecurity team. The proposed roles include a Security Manager, Security Architect, Security Engineer, and a Data Protection Officer. The role of the Security Manager is to interface the project management and the cybersecurity development team, coordinating any activities that arise due to cybersecurity needs. The Security Architect, analogous to a System Architect role in a traditional development environment, creates the architectural design of the cybersecurity features and works closely with the Security Manager and Engineer. If the System Architect possesses adequate cybersecurity knowledge, they can take on the responsibilities of this role instead of creating

an entirely new role. The Security Engineer focuses on implementing and testing security features and third-party components relevant to the system. The role should be taken on by an experienced developer who writes secure code and who is familiar with the system. Finally, the Data Protection Officer handles the protection of data according to laws and regulations. The authors note that while the role of Security Manager should be filled by someone who can dedicate their time solely to the role, the other positions can be filled by team members with similar roles.

In addition to outlining roles, Lenhart et al. (2020) also outline artifacts the team should create for cybersecurity development. They start with *Feature Definition*, which simply outlines the scope of the security features that need to be developed for a system. A *Threat Analysis and Risk Assessment*, created from the *Feature Definition*, builds the foundation to select security measures for the final product, describing relevant security risks and any unacceptable risks that absolutely need to be handled. The *Cybersecurity Concept* outlines how these cybersecurity needs are addressed by the planned implementation. The *Cybersecurity Requirements* are a formal definition of the needs that must be met on a system level, and the procedures that will be used to verify them. Finally, the *Data Protection Concept* outlines data processed by the system and concerns that must be addressed.

While Salin & Lundregen (2022) focus on integrating cybersecurity tasks and risk management into the agile development cycle, Lenhart et al. (2020) concentrate on improving security focus in teams while sticking to a plan-driven approach to cybersecurity development. Lenhart and coauthors draw on their experience in the automotive industry, which generally releases finished products rather than incrementally updated ones as many agile teams do, to inform their cybersecurity recommendations. This contrasts with Salin & Lundregen, who focus on integrating the cybersecurity risk assessment closely with the agile process. Additionally, Salin & Lundregen do not explicitly call for the creation of new roles as Lenhart and coauthors do.

While testing these two methods was outside the scope of this project, I will outline a possible way to conduct a study on the effectiveness of the proposed methods. Since measuring quantitative results based on development processes is extremely difficult due to the number of other factors that could influence the results, the most reasonable way to measure would be through surveys. What's important is testing these methods against existing practices and comparing team sentiment on performance and organization before and after the methods are implemented.

4. ANTICIPATED RESULTS

While some teams will find better results by sticking to one side or the other of a more plan-driven or agile approach, the best way to get results would be to combine the two. While integrating with agile processes is important, it is also incredibly important that the risks are understood as early as possible, so any major and not easily reversible design decisions are made with risks in mind. While this can be done with an agile approach, important risks should be understood well before they come up in a sprint cycle. The roles outlined by Lenhart et al. (2020) help define clear duties necessary in developing secure software. While it may be harder to convince a team to create entirely new roles to hire for, Lenhart and coauthors make clear that most of the roles can come as an additional duty rather than an entirely separate role. This allows for a duty to be clearly defined while not requiring a company to hire a whole new staff.

While having a plan can be great for finding and addressing possible issues early, it is also not sufficient, which is why integrating cyber security development with agile is so important. The agile method proposed by Salin & Lundregen (2022) does a much better job at keeping up the understanding of security with the team's current non-security-related goals. Integrating every security related thing into sprint ceremonies that are likely already being done minimizes friction by keeping additional meetings to a minimum. Keeping up-to-date on significant risks that are important and being dealt with is just as important as having a big picture plan from the start.

5. CONCLUSION

Both the methods provided by Salin & Lundregen (2022) and Lenhart et al. (2020) provide a framework that can help developers build and orient their teams around keeping software development secure. Lenhart and coauthors' method focuses on defining clear cybersecurity roles and documentation, while Salin & Lundregen's method, called Risk Refinement focuses on brining cybersecurity development in line with agile development processes. While it is possible to combine the methods, some adjustments would be required due to Lenhart and coauthors' approach still assuming a plan-driven framework.

6. FUTURE WORK

One issue raised by Jøsang et al. (2022) in the trappings of certain frameworks around cybersecurity development is the lack of cybersecurity education for developers. Changing the frameworks around development does not matter if developers lack the knowledge necessary to write secure code. Ultimately, the only way to properly write secure software is to write secure software. Any framework intended towards secure development should be implemented alongside cybersecurity training for developers, even if they are not in a cybersecurity focused role.

Further research into the actual efficacy of the proposed frameworks is required to understand what methods are effective and what further refinements can be made. Additionally, it could be helpful to look at other proposed frameworks for further comparison and to find more common ground. Comparisons (on both a qualitative and quantitative basis) with current methods and formalized frameworks used by companies today would help researchers understand how necessary it is to switch frameworks.

REFERENCES

Jøsang, A., Ødegaard, M., & Oftedal, E. (2015). Cybersecurity Through Secure Software Development. In M. Bishop, N. Miloslavskaya, & M. Theocharidou (Eds.), Information Security Education Across the Curriculum (pp. 53–63). Springer International Publishing. https://doi.org/10.1007/978-3-319-18500-2_5

Lenhart, P., Arndt, P., Wedel, J. von, Beul, C., Weldert, J., Lenhart, P., Arndt, P., Wedel, J. von, Beul, C., & Weldert, J. (2020, April 14). Challenges in Integrating Cybersecurity into Existing Development Processes. WCX SAE World Congress Experience. https://doi.org/10.4271/2020-01-0144

Salin, H., & Lundgren, M. (2022). Towards Agile Cybersecurity Risk Management for Autonomous Software Engineering Teams. Journal of Cybersecurity and Privacy, 2(2), Article 2. https://doi.org/10.3390/jcp2020015