Donald Doss

Gabriel Hernandez

Michael Kohronas

Michael Vazquez

Xavier Rivera Maysonet

In 100 words or more, determine and describe which framework will work best for SnowBe, and why.

While reviewing the differences between Secure software Development Framework (SSDF) and the Microsoft Software Development Life Cycle (MSDL), We found it to be a choice for SnowBe to go with (SSDF). The overall reason that brought us to this decision was the fact on how (SSDF) focuses on security throughout the whole development process. Since we see SnowBe is focusing on a organized approach and they have shown history of security issues, SSDF would emphasize taking control of these risks by implementing the security measure from the very beginning. This is vital to protecting customer information, while also ensuring adherence to PCI standards when creating these software solutions that would on the flip side drastically improve SnowBe's operations. Overall, SSDF constantly focuses on regularly conducting security assessments and testing, which is important in discovering and patching any vulnerabilities as the software is developed.

In 200 words or more, provide the steps you have taken to decide on the desired framework. You will want to show your thought process that went into this decision. You will want to address some of the following items: programming practices from the framework, development processes, and methodologies that lead to secure software for SnowBe.

For us to determine the most adequate framework for SnowBe, we sought to compare the Secure Software Development Framework (SSDF) and Microsoft Software Development Life Cycle (MSDL). So, we took it upon ourselves to look over the practices and principles of both frameworks. We believe that the SSDF puts prioritization into secure coding standards, overall risk management, and well-versed security measure for every stage of the development process. This is of overall importance to address all potential vulnerabilities before they can be discovered. On the other hand, MSDL given it does provide a well-structured approach, it mainly focuses on the efficiency and project management side of this matter. Sadly, this would cause SnowBe to overlook a lot of the security vulnerabilities, which is a vital when it references to the threat in today's world.

Moving forward, we considered the specific needs of SnowBe. Based on the case study the overall negligence of control brought out our awareness regarding the supply chain security which would warrant security to be placed at the top of the priority list. When reviewing the methodologies within these frameworks, we acknowledged that the SSDF would support the overall agile practice, which would allow our overall team to be pretty much flexible and responsive to the ever-changing demands and security threats.

In conclusion, we discussed as a team and gathered insights from the many stakeholders to make sure that the framework we chose aligns with the overall business goals SnowBe holds. By comparing the standalone strength of SSDF it promotes safe software practice in the scope of project management orientation of the MSDL. Overall, SSDF would ensure best fit the current goal and practices for a secure and resilient software solution. This decision would not only support any and all immediate security needs but give SnowBe stability and growth for the future.

For the framework that was not selected, describe in 100 words or more why it was not chosen over the other.

The Microsoft Software Development Life Cycle was not picked because it had limited regard for security and more of reactive take for risk management. Even though, MSDL did support overall project efficiency, it did lack guidelines throughout the development process. This would have caused a lot of harm than good for SnowBe, as the company increases the more threats would as well and PCI compliance requirements. Microsoft Software Development Life Cycle overall stance was not flexible while the procedures would conflict with SnowBe's need for agility. While also bringing worry about the controls in placed for the security of the supply chain. The overall result was SnowBe needed the framework the would focus on security as core.

Document the roles and responsibilities of Karen and her three new developers. Each developer should have different roles and responsibilities. If you are unfamiliar with software development teams, feel free to use the roles and responsibilities in the SSDF.

1.Project Manager (Karen):

Responsibility: To ensure proper software development and security practices are followed, collaborating with the teams effort, managing schedules and timelines. Furthermore, communicating with vendors (i.e Stakeholders) to give updates, progress, and/or scope.

2.Software Developer A:

Responsibility: To focus on the overall design and implementation of software feature while ensuring guidelines and best practices for secure coding are being followed. This would also intel testing and reviewing the code of the other developers to resolve any vulnerabilities that may appear before launching.

3.Quality Assurance Tester:

Responsibility: Will develop and push out the test plan to analyze the overall performance function of the software and security compliances. They will conduct vulnerability testing by performing (PenTest) to get a good assessment to see if the overall software follows the security guidelines.

4.Security Specialist:

Responsibility: Will assses the overall foundation of the software's security and conduct threat modeling. They will be responsible for all security controls, and audit while ensuring compliance with all standards to include PCI.

In 100 words or more, decide and describe the programming methodology you feel would work best for the new SnowBe development team. You will want to list the methodology or a hybrid and explain the whys and why nots of your decision. Elaborate on your answer to demonstrate your depth of knowledge for this week's topics.

For the development team We would recommend a more modern yet agile approach while also being flexible in which it would allow us to develop the software at a micro level by making the quick adjustment that are needed. With an agile approach it would promote the overall collaboration of the team to deliver the feature at a faster rate. If we add DevSecOps, we can allow for security analysis at every stage of the development process, ensuring security protocols are followed the start. This would provide us more of an opportunity to work closely with the stakeholders, and modify or adapt to the feedback quickly, and maintaining security at the top of our list.

In 100 words or more, describe the importance of secure software pertaining to the two development frameworks, practices, and methodologies.

The importance of secure software development vital in both frameworks SSDF and MSDL to ensure the overall security of customer data and to avoid any potential breaches. We know that SSDF implements security at every possible level while keeping it the main priority, even during the development phase. This is overall a very motivated approach to reinforce software while also minimizing the risks. On the other hand, I would say that MSDL forgets to focus on security, which would cause a lot vulnerabilities to be left exposed even so risking the SnowBe and their reputation. It is best for both frameworks to ensure the security practices that are important to protect assets, stakeholder confidence, and overall of the success of the business for long-term effect.

References:

Mitigating the Risk of Software Vulnerabilities by Adopting a Secure Software Development Framework (SSDF) (fso-lms4-immortal-assets.s3.us-east-1.amazonaws.com)

Microsoft Security Development Lifecycle (SDL) - Microsoft Service Assurance | Microsoft Learn

Microsoft Security Development Lifecycle Practices

What is DevSecOps? - Developer Security Operations Explained - AWS (amazon.com)

What is DevSecOps? | IBM