Decide on an SDLC Framework, Development Practices, & Methodology

Daphnie Bruno
Professor: Bill Moran
CYB359-0 | C202503-01

Case Study Company:

Company: SnowBe

SnowBe is a lifestyle brand for those who love the beach and snow. The owners started the company with a laid-back culture. Their customers instantly connected with their brand taking them to $100 million in sales in three years. After being so successful, the management team decided to take the company public.

Technical Information:

The majority of their sales are processed online through their website, housed on the AWS platform.
All credit cards are accepted and stored on the company's website database.
All customer information and purchase history are stored on the website indefinitely.
They have multiple storefronts in the U.S. and Europe, which accept checks, cash, or credit cards. The credit card transactions are processed using bank-provided credit card terminals in each store.
There are twenty desktops and thirty laptops in the main office in Los Angeles.
The desktops are used to run the business and customer support.
The thirty laptops are used for sales (retail and wholesale). The laptops use a VPN to log into the office to access company applications.
There are six servers (on-premises and AWS) for access management, storage, customer relations management, order management, accounting, and vendor applications.
As a result of SnowBe's laid-back culture, they neglected to implement technical controls and processes. As a result, they recently hired a technical consultant to get their neglected system and processes under control. The consultant started with implementing controls using the NIST 800-53 r5 framework.

Additional Information:

The technical consultant was impressed to find a well-run company with no reported technical issues or breaches despite SnowBe's laid-back culture. Although, there had been a few attempts that did not cause any harm or alerts to worry anyone. The technical consultant analyzed the risk of the company using the NIST Risk Management Framework.

Here are some initial steps he suggested:

- The need to update the firmware of all network devices.
- The need to update the patches for all PCs and Windows servers to ensure they are using the latest Windows version.
- The need to update their Anti-Virus and backup software.
- The need to implement more processes into the access management system since most employees had access to almost all the data on each server.
- The need to lock the servers in a secured area of the office.
- The need to update the company's WordPress shopping cart.
- The need to implement the required PCI compliance items.
- Login audit records need to be saved, and records older than 3 months should be archived to a cloud storage facility.
- Mobile devices need to be reviewed and approved to have access to the company data.

SnowBe does not have a development process, and Karen is leaning towards using the Secure Software Development Framework (SSDF) or the Microsoft Software Development Life Cycle (MSDL). Both frameworks are listed in the resources section below.

You will be Karen for this assignment. Elaborate on your answers to demonstrate your depth of knowledge for this assignment. Your feedback will need to be free of spelling, grammar, and punctuation errors.

1) You will review both frameworks and the practices of each and document the following:

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

For SnowBe, the most appropriate is NIST Secure Software Development Framework (SSDF). For SnowBe's fast growth, increasing security risks, and no prior organized development processes, SSDF presents a comprehensive, flexible, and security-focused approach that is best for them. SSDF includes security activities throughout the whole software lifecycle and not as an add-on. The framework will help SnowBe identify vulnerabilities early, secure their software supply chain, and achieve compliance requirements like PCI DSS for credit card transactions. Unlike Microsoft's Secure Development Lifecycle (MSDL), which is intended for mature development teams, SSDF has a flexible methodology, so it is ideal for a new team that is establishing a secure foundation from scratch.

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.

Alignment with SnowBe’s Security Needs

- SnowBe lacks existing secure development processes and stores sensitive customer data indefinitely, making security a top priority.
- SSDF provides a structured approach that ensures security at every phase while being adaptable to a company just starting its development journey.

Software Development Practices Comparison

- Microsoft’s SDL: Focuses on prevention-first security, ideal for companies with existing development teams and a mature security culture. However, it assumes developers already follow strict security processes, which SnowBe lacks.

- SSDF: Designed to integrate security into any SDLC, making it more practical for a company like SnowBe that is starting from scratch.

Security Methodologies & Compliance

- Since SnowBe processes payments and stores personal data, regulatory compliance (e.g., PCI DSS) is mandatory.
- SSDF aligns with NIST 800-53 r5, which SnowBe is already adopting for broader security controls.
- It also supports supply chain security, addressing the CEO's concerns about supply chain risks.

Flexibility & Scalability

- SSDF is vendor-neutral and adaptable to different software development methodologies (Agile, DevOps, Waterfall, etc.), making it a better fit for SnowBe's evolving needs.
- It allows SnowBe to start small with security best practices and scale as the company grows.

Ease of Implementation for a New Team

- SnowBe is hiring a small development team of three, so they need a clear but flexible security framework.
- SSDF provides clear security goals without being overly prescriptive, making it easier for Karen and her team to adopt without prior experience.

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

While Microsoft's Secure Development Lifecycle (MSDL) is an industry-standard framework that has successfully secured many Microsoft products, it's not the best fit for SnowBe at this stage.

1. MSDL works best when developers already follow security best practices. Since SnowBe is building a team from scratch, they need a framework that focuses on establishing secure foundations rather than refining existing practices.
2. MSDL is designed for large-scale, enterprise-level development with strict guidelines for each development phase. SnowBe needs more flexibility as they develop their processes.
3. Focuses on Windows Ecosystems – While MSDL is effective for Windows-based environments, SnowBe uses AWS, WordPress, and various web technologies, which SSDF better accommodates.
4. Given the CEO's concerns about supply chain risks, SSDF is the better fit since it includes robust supply chain security recommendations.

Ultimately, SSDF is better suited for SnowBe's current level of maturity, offering flexibility, regulatory alignment, and strong security foundations.

2) Document the roles and responsibilities of Karen and her three new developers. Each developer should have different roles and responsibilities.

Project Manager & Security Lead - Karen

- Oversees software development processes
- Ensures SSDF security best practices are followed
- Works with executives & stakeholders to support improvement goals with the business and security needs
- Confirms compliance with NIST 800-53, PCI DSS, and all other industry standards

Frontend Engineer - Developer 1

- Builds secure user interfaces for the new software
- Ensures proper input validation to prevent injection attacks (e.g., XSS, CSRF)
- Implements secure authentication and session management practices

Backend Engineer - Developer 2

- Will develop server-side logic, database interactions, and API security
- Uses secure coding to prevent vulnerabilities like SQL injection or a denial-of-service attack
- Will implement access control tools to restrict data exposure

DevOps & Security Automation - Developer 3

- Manages CI/CD pipeline security and automated security testing
- Implements code scanning, dependency checks, and runtime security monitoring
- Ensures secure deployment practices and cloud security configuration

3) 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 list the why and why not of your decision. Elaborate on your answer to demonstrate your depth of knowledge for this week's topics.

For SnowBe, I feel that a hybrid Agile/DevSecOps methodology is the best fit. A hybrid Agile/DevSecOps gives the best of both with secure and effective development. Agile will allow SnowBe to remain agile when they are responding to the needs of the business, and DevSecOps will ensure security is executed and in place throughout all phases. Using DevSecOps, we will be able to make security automated within the CI/CD pipeline. This will lessen the risk of misconfigurations and coding flaws will be significantly reduced. Agile promotes cross-functional collaboration, while DevSecOps combines security and compliance into development. In my opinion, Waterfall is too strict for a company

that is just starting out, and it does not allow for quick modifications. Fully DevSecOps might be too complex for a small team, and they need a balance between speed and security.

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

With SnowBe handling credit card transactions, customer data, and online sales, secure software isn't optional anymore, it is a necessity. PCI DSS compliance requires strong security measures for handling payments. If there is a security breach SnowBe could be looking at legal penalties and loss of customer trust. Since SnowBe is using in-house software, securing third-party dependencies is critical. Following SSDF's supply chain security practices will help SnowBe reduce the possibility of software supply chain attacks. A security breach could stop operations, which would lead to loss of income and reputational damage. Secure software guarantees long-term stability. By inserting security into every stage of development, SnowBe ensures that security becomes part of its culture, not just a one-time effort.