

SECURE SOFTWARE DEVELOPMENT

WHITEPAPER

July 2025

© 2025 Software AG. All rights reserved.

softwareAG
A SOFTWARE GMBH BRAND

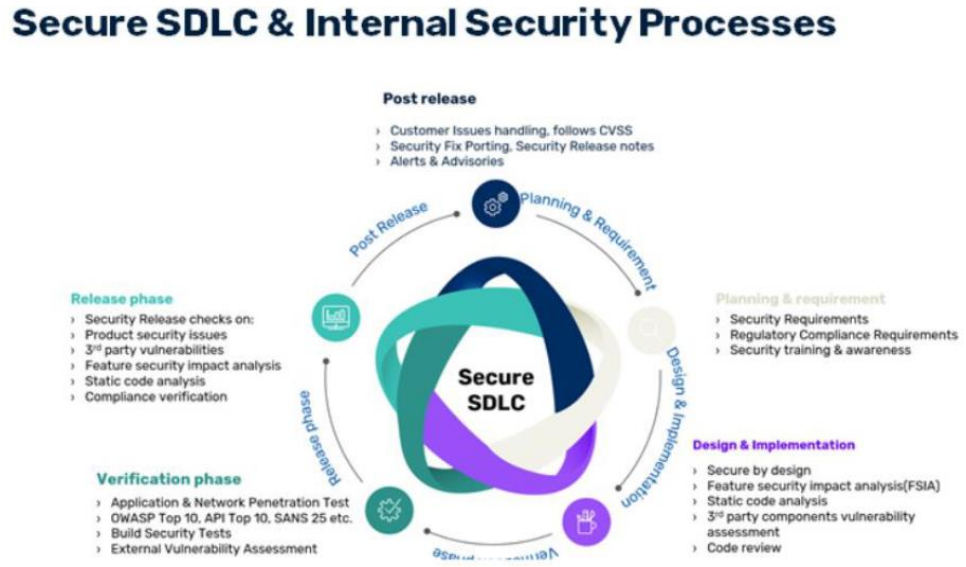
1 Introduction

The Software Development Life Cycle (SDLC) is a method for designing, developing, and testing high-quality software in the software industry. The SDLC is designed to deliver high-quality software that meets or exceeds customer expectations and is completed on time and within budget.

SDLC is a detailed strategy that explains how to build, maintain, replace, and change or improve certain software. The life cycle is a mechanism for enhancing software quality and the development process.

Any application that includes crucial functionality must have security. This might be as basic as protecting your database from malicious actors, or as complicated as applying fraud processing to qualified leads before importing them into your platform.

Previously, security-related activities were normally undertaken solely as part of testing—at the end of the SDLC. However, due to the rapid change in the pace of digital transformation, it is critical to construct secure products for customers. Applying security at every stage of the SDLC is the best method to create a secure solution. This gave rise to the concept of 'Secure SDLC'.



Secure SDLC is a set of best practices aimed at making the regular SDLC more secure. From requirement collection to deployment and maintenance, creating a secure SDLC process demands committed work at each stage of the SDLC.

Security vulnerabilities can be addressed in the SDLC pipeline well before deployment to production with committed effort. This lowers the chances of discovering security vulnerabilities in your application and tries to limit the impact if they are discovered.

© 2025 Software GmbH. All rights reserved.

2 Overview: Secure SDLC

As application security is critical, a secure SDLC is essential. The days of releasing a product and then patching it to fix the flaws are long gone. Developers must now be aware of potential security issues at every stage of the development process. This necessitates incorporating security into your SDLC in ways that were previously not required.

As someone could potentially obtain access to source code, developers must be aware of the vulnerabilities while writing the code. As a result, ensuring that your application is not vulnerable to attacks by hackers and other malicious users requires a comprehensive and secure SDLC procedure.

2.1 Planning & Requirements

The requirements for new features are gathered from many stakeholders during this early stage. It's critical to identify any security concerns when gathering functional requirements for the new release.

Software AG takes several steps to ensure that the product under development is secure. For instance, professionals must undergo security training to gain an understanding about concepts such as secure programming, and so on. Another strategy is having a regulatory compliance policy in place that all Software AG products must comply with.

2.2 Design

This phase translates in-scope requirements into a plan for how the application should be developed. We embrace one of the important security principles - security by design. This means that all software products have been designed to be foundationally secure.

Every product feature is analyzed for security impact during the design phase and recommendations are provided early in the SDLC to the developers. This is achieved by an internally developed scalable solution that helps the developers to analyze the security impact and associated threats of the new feature which is to be developed.

This is a proactive control that helps in identifying the threats associated with the feature well in advance, even before the development activity starts. Any new code introduced can have a list of security consequences both locally and globally.

2.3 Implementation

When it comes to putting the idea into action and making it a reality, the focus frequently shifts to ensuring that the code is well-written from a security standpoint. Secure coding guidelines are

© 2025 Software GmbH. All rights reserved.

usually created, as are code reviews, to ensure that these guidelines have been followed appropriately.

At Software AG, commercial tools are used to conduct the process of static code analysis, which assists in ensuring that the code written by a developer complies with the coding guidelines in place. It also helps in discovering the flaws in the code, thus allowing for a faster turnaround time for fixes. Static code analysis combined with multiple code reviews ensures that the product under development has minimal loopholes by the time it moves to the next stage of the lifecycle.

The products of Software AG product suite use many 3rd party components. The vulnerabilities in the open-source third-party components are extremely common and could be used to compromise the security of the systems.

A continuous scan is performed on these components for detecting and monitoring newer vulnerabilities using commercial scanners. These scanners have been integrated into the CI/CD pipeline for faster feedback to the developer to regularly improvise on the products. This helps us to efficiently follow the 'shift-left' approach.

Internal security processes on vulnerability management of 3rd party components ensure continuous verification and monitoring across the lifecycle of the 3rd party component within the organization starting from the time it gets introduced into the products and till it gets retired/ removed. The software can evaluate the product artifacts for critical, high, medium, and low severity security issues that are categorized based on the CVSS 3.x base score.

If there are issues identified with any of the components used, they are upgraded or replaced before the release.

2.3 Verification

The Verification phase involves a thorough testing cycle to ensure that applications meet the original design and requirements. This is also an excellent opportunity to implement automated security testing, which can be performed using a variety of technologies.

Unless these tests pass, the application will not be deployed. To control verification and release, this phase frequently involves automated security scans as a part of the CI/CD pipeline.

In addition to this, penetration testing is performed where applicable on Adabas Natural products.

2.4 Release

All Software AG products must undergo a mandatory Quality Gate check before they can be released. The following are activities that are conducted as a part of this process:

- Penetration Testing: Product suite undergoes internal penetration testing activity as a mandatory quality gate check before the release. The findings of these assessments must

© 2025 Software GmbH. All rights reserved.

be addressed prior to releasing the product, and no product is released with outstanding security vulnerabilities.

• Compliance Verification: All products undergo assessments related to regulatory compliance. The findings of these assessments must be addressed prior to releasing the product. This data is tracked and maintained by an internally developed compliance assessment tool.

• Third-party Vulnerability Scanning: Products are regularly scanned to identify the vulnerable components, if any. The scan results are reported and followed up with the product development team to upgrade or use an alternate solution. The third-party vulnerability assessment is one of the security release criteria for product release. A product having outstanding issues cannot be released. Hence, all the outstanding issues must be taken care of before the release.

• Static Code Analysis and Code Reviews: All the products must also undergo these activities, as a part of the CI/CD pipeline, to guarantee that the code contains as fewer defects as possible.

2.5 Post Release

Once released, security flaws that slipped through the loopholes in the application may be discovered long after it has been deployed. These vulnerabilities can be detected in the code, but they're also becoming more common in the underlying open-source components that make up an application. As a result, the number of "zero-days"—vulnerabilities that were previously undiscovered but were discovered in production by the application's maintainers—increases.

During the maintenance phase, issues that arise within a product may also be reported by the customers. A customer issue handling process has been put in place to carefully handle the vulnerabilities reported by customers in our products.

Software AG's process for handling issues post release involves 'Security Fix Porting'. As per this, any issues reported on a particular version are fixed in the reported version and backward ported to all the standard maintenance lines.

Yet another method to keep our customers up to date on the latest available fixes is by making use of alerts & advisories. The internal security engineer may uncover the security vulnerability, or the client may report it. The customer is notified in the form of an early warning about vulnerability and the fix that must be installed using the Empower early warning channel.

© 2025 Software GmbH. All rights reserved.

3 Conclusion

Several KPIs have been put in place to measure and track the performance of all Software AG products. This helps to ensure that the products are constantly being improved as per the metrics provided by these KPIs. Software AG is making great strides in promoting secure software development by applying security best practices at every stage of the SDLC as described in this white paper. Customers have witnessed a significant improvement in security in the recent decade.

4 Feedback

If you have feedback or questions on this whitepaper, please send an email to devsecurity@softwareag.com

© 2025 Software GmbH. All rights reserved.