

Secure Development Lifecycle (SDL)

Gehört Sicherheit zum Software Development Prozess?

Heutige Softwarelösungen sehen sich ständig steigender Sicherheitsbedrohungen ausgesetzt. Der Erfolg zur Verbesserung der Sicherheit liegt darin, mögliche Bedrohungen bereits in der Entwicklungsphase zu identifizieren und zu beheben, zudem reduziert das die Kosten gegenüber einer nachträglichen Behebung.

Durch die Planung und Umsetzung eines Secure Development Lifecycle (SDL), ist die Sicherheit ein klarer Bestandteil jeder Software Lösung und dient dazu, die Anzahl der Sicherheitsschwachstellen bereits in der Ideen- und Designphase frühzeitig zu erkennen und während der Implementierung zu beheben.

Übersicht SDL

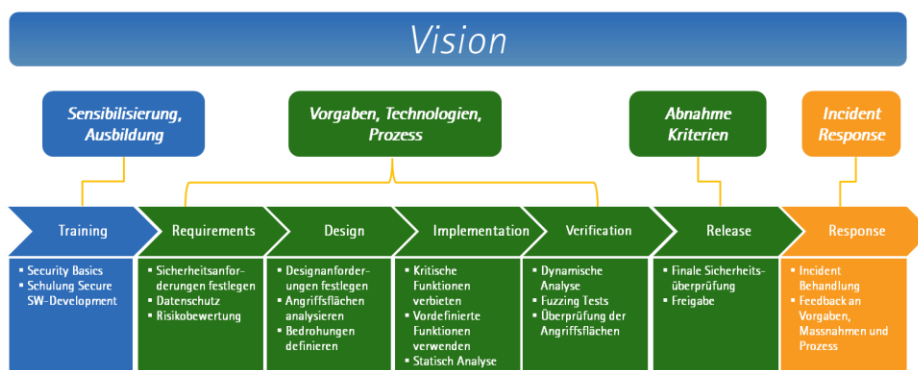


Abbildung 1 SDL- basierend auf dem Microsoft SDL

Dienstleistungen

Im Zusammenhang mit einem auf den Kunden angepassten Secure Development Lifecycle, können wir unsere Unterstützung in folgenden Bereichen anbieten:

- **Training:** Security Training für sicheres Design und Programmierung
- **Requirements:** Definieren von Sicherheitsanforderung, Risiko Analyse
- **Design:** Definieren von Design-Sicherheitsanforderung, Mögliche Bedrohungen definieren
- **Implementierung:** Erstellen von Vorgaben für den Einsatz von sicheren und unsicheren Funktionen, bereitstellen von Libraries & Tools
- **Verification:** Überprüfung der Vorgaben durch manuelle und automatisierte Sicherheitsanalysen (Fuzzing, Code Analyse, Scanning)
- **Release:** Definieren von Freigabeprozessen und Massnahmen wie Security Review, Penetration Testing o.ä

Sie erhalten

- Auf Ihr Bedürfnis individuell angepasster SDL
- Produktneutrale Unterstützung bei Ihrem Vorhaben
- Punktuelle oder vollumfängliche Umsetzung
- Nach Bedarf Entwicklung individueller Tools
- Flexibler und zukunftsorientierter Lösungsansatz

Ihr Vorteil

- Zeitreduktion für Fehlerkorrektur
- Reduzierung der Entwicklungskosten
- Reduzierung von Mängeln
- Sicherheit als integraler Bestandteil der Software Entwicklung
- Erhöhte Sicherheit
- Gewährleistung von Compliance Anforderungen

Security Software Testing

Automatisiertes Security Software Testing

Um den Sicherheitsanforderungen heutiger Applikationen gerecht zu werden, sind neben definierten Sicherheits-Vorgaben und Prozessen auch geeignete Tools zur Unterstützung nötig.

Das funktionale Testing ist im SW-Entwicklungsprozess verankert und stellt heute einen wichtigen Bestandteil dar, doch Security Tests sind dabei kein Bestandteil. In heutigen Applikationen gehört das Security Testing genauso zum Testumfang wie die funktionalen Tests

Heutige Problemstellungen

- Applikationen werden hinsichtlich ihrer Sicherheit, gar nicht oder in unregelmässigen Abständen getestet
- Mangelnde Qualität in Bezug auf Sicherheit der Applikationen
- Kürzere Releasezyklen verstärken das Problem noch zusätzlich
- Keine integrierten Lösungen vorhanden
- Erhöhte Kosten durch spätes aufdecken von Sicherheitsmängeln

Dienstleistungen

Im Zusammenhang mit Security Testing können wir Ihnen folgende Dienstleistungen anbieten:

- Gemeinsame Erarbeitung der zu erreichenden Zielen und Auswahl der geeigneten Mittel
- Integration von einem automatisierten Security Testing-Framework oder einzelnen Testingtools nach Kundenbedürfnissen
- Unterstützung bei der Integration des Frameworks in den Entwicklungsprozess resp. SDL

Der Nutzen

Durch den geeigneten Einsatz von Security Testing Tools in ihrem Entwicklungsprozess, erhöhen Sie die Sicherheit Ihrer Produkte, verkürzen die Releasezyklen und sparen durch frühzeitige Behebung von Mängeln Ressourcen.

Nehmen Sie mit uns Kontakt um mehr über den Einsatz und die Möglichkeiten von Security Testing Frameworks zu erfahren.

Protect7 – Ihr Partner für Security Software Testing

Sie erhalten

- Unser Know-how für die Auswahl geeigneter Tools
- Produktneutrales Security Testing-Framework inkl. Reporting
- Integration der Tools in das Framework
- Nach Bedarf Entwicklung individuell angepasster Erweiterungen
- Eine flexible und zukunftsorientierte Lösung

Ihr Vorteil

- Reduktion der Sicherheitslücken
- Sicherheitstests ohne personellen Ressourcen
- Zeitreduktion für Fehlerkorrektur
- Schnellere Produkteinführung (Time-to Market)
- Reduzierung der Entwicklungskosten
- Kosteneinsparungen für Compliance- und Penetrationstests