DEON SSDLC – Richtlinie für sicheren Softwareentwicklungszyklus

SDLC Übersicht

Das DEON-Entwicklungsteam verwendet einen agilen Softwareentwicklungsprozess. Unsere Ingenieure arbeiten in kurzen iterativen Sprints, die aus Entdeckungs-, Design-, Entwicklungs-, Test- und Release-Phasen bestehen. Dies ermöglicht es uns, Funktionen schnell und mit Zuversicht zu veröffentlichen und schnell zu iterieren, um Funktionen im Laufe der Zeit zu verbessern.

Entdeckungsphase

Die Entdeckungsphase umfasst die Produktmachbarkeit und Anforderungserhebung und wird typischerweise von Engineering-Managern und Tech-Leads durchgeführt.

Designphase

Während der Planungs- und Designphase werden High-Level-Designs, User Stories und UI-Mockups erstellt. Stories werden in Aufgaben unterteilt und hinsichtlich Komplexität und Aufwand geschätzt, der in Zeit (Tagen) gemessen wird. Alle Planungen und Verfolgungen werden auf Azure Boards und in DEON-Projekten durchgeführt.

Entwicklungsphase

Während der Entwicklungsphase wählen und implementieren Ingenieure Aufgaben aus dem Backlog. Ingenieure verlassen sich auf Peer-Code-Reviews und Pull-Requests, um sicherzustellen, dass die Codeänderungen korrekt sind und die Codequalität hoch ist. Pull-Requests werden erst in den Hauptzweig gemerged, wenn der Code-Reviewer die Änderung ausdrücklich genehmigt hat. Wir pflegen unseren gesamten Code in Azure Repos und führen die Integration mit Azure Pipelines durch.

Testphase

Ingenieure schreiben in dieser Phase Unit-Tests und Integrationstests. Änderungen werden lokal getestet und verifiziert, um in Entwicklungsumgebungen zu funktionieren, bevor sie in die Produktion freigegeben werden. Wir verwenden verschiedene Testwerkzeuge für die Entwicklung von Testfällen.

Bereitstellungsphase

Ingenieure pushen ihre Änderungen mit Unterstützung der Engineering-Manager bei Bedarf in die Produktion. Produktions-Pushes werden über Azure DevOps-Tools koordiniert und verwaltet. Alle Bereitstellungen müssen in Übereinstimmung mit der DEON-Change-Management-Richtlinie erfolgen.

Post-Release-Phase

Wir überwachen alle Bereitstellungen nach der Veröffentlichung, um Stabilität und Leistung sicherzustellen. Unser Engineering-Team rotiert im Bereitschaftsdienst und ist verantwortlich für die Behebung oder das Zurücksetzen von Problemen, falls diese jemals auftreten.
Sicherheitsentwickler müssen während des gesamten Entwicklungszyklus die DEON-Codierungsstandards einhalten, einschließlich Standards für Qualität, Kommentierung und Sicherheit. Zumindest wird von Entwicklern erwartet, dass sie die gängigen Sicherheitsprobleme in den OWASP Top-10 (www.owasp.org) im Laufe ihrer Design-, Entwicklungs-, Überprüfungs- und Testbemühungen angehen.
Entwickler, die Peer-Code-Reviews durchführen, müssen die erforderliche Sicherheitsschulung absolviert haben und den neuen oder überarbeiteten Code untersuchen und Feedback geben. Die Überprüfung muss bestätigen, dass der Code keine Sicherheitsprinzipien oder Designziele verletzt.
In-Scope-Software muss standardisierten Tests unterzogen werden, die sowohl Funktions- als auch Sicherheitstests umfassen. Alle während des Tests festgestellten Sicherheitsprobleme müssen vor der Veröffentlichung behoben werden.
Maßnahmen, die von Entwicklern auf Produktionssystemen ergriffen werden, müssen immer protokolliert und geprüft werden.

Der Zugriff auf das private Quellcode-Repository (in Azure Repos und anderswo) ist auf autorisiertes Personal beschränkt, und Zugriffskontrollen werden durchgesetzt, um die Sicherheit des Repositorys zu gewährleisten. Der Zugriff von Entwicklern auf die Quellcode-Repositories muss vor der Gewährung neuer oder zusätzlicher Zugriffsrechte vom entsprechenden Engineering-Management genehmigt werden. Jeder Zugriff auf Quellcode oder Änderungen der Zugriffsrechte auf die Quellcode-Repositories muss protokolliert werden und unterliegt der Prüfung.
Um Sicherheitsanforderungen zu erfüllen, gelten folgende Richtlinien für Funktionen und den Anforderungsrozess:

Erstellen Sie Richtlinien zur Bewertung der Anwendungssicherheit unter Verwendung von

Kriterien wie:

- Datenhandhabung, -offenlegung und -verhalten
- Externe Sicherheits- und Compliance-Anforderungen
- Interaktionen mit anderen Systemen
- Benutzerkonto- und Berechtigungsverwaltung
- Kundensicherheitsanforderungen
- Bekannte oder erwartete Sicherheitsschwächen oder -risiken
- Sicherheitsherausforderungen aufgrund einer einzigartigen oder nicht standardmäßigen Architektur