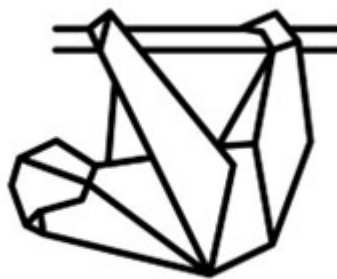




Fachbereich VI - Informatik und Medien  
Studiengang IT-Sicherheit Online / Medieninformatik

**Vorbereitung 8 - Bedrohungsmodellierung, Entwicklungsprozess**



**PARETO CONSULTING**

Modul: Sicherheitsmanagement

Dozent: Sven Zehl

Gruppe 1

Christine Kuczera

Dirk Drutschmann

vorgelegt von: Hicham Naoufal

Michael Schröter

Jan Zimmermann

Ivo Valls

# Aufgabe 1) Kennen Sie einige Vorgehensmodelle der Softwareentwicklung? Erläutern Sie sie!

## Antwort

### 1. Wasserfallmodell:

Das Wasserfallmodell ist ein sequenzielles und phasenbasiertes Vorgehensmodell. Es besteht aus verschiedenen aufeinanderfolgenden Phasen, die nacheinander abgeschlossen werden, wie Anforderungsanalyse, Design, Implementierung, Testen und Wartung. Jede Phase baut auf den Ergebnissen der vorherigen Phase auf. Das Wasserfallmodell eignet sich gut für Projekte mit klaren und stabilen Anforderungen, bei denen Änderungen während des Entwicklungsprozesses selten sind.

### 2. Agile Methoden:

Agile Methoden sind iterative und inkrementelle Vorgehensmodelle, die auf Flexibilität und Zusammenarbeit setzen. Scrum und Kanban sind zwei bekannte agile Methoden. Agile Ansätze betonen die kontinuierliche Lieferung von Software in kurzen Zeitrahmen, sogenannten Sprints oder Iterationen. Das Team arbeitet eng zusammen, um regelmäßig funktionsfähige Software zu liefern und Feedback zu erhalten. Agile Methoden passen gut zu Projekten mit sich ändernden Anforderungen und einem hohen Maß an Kundeninteraktion.

### 3. V-Modell:

Das V-Modell ist ein Modell, das den Softwareentwicklungsprozess mit dem Testprozess verbindet. Es stellt eine Verbindung zwischen den Phasen des Softwarelebenszyklus und den entsprechenden Testaktivitäten her. Das V-Modell betont die Verifizierung und Validierung der Software. Die Phasen des V-Modells umfassen Anforderungsanalyse, Systemdesign, Moduldesign, Implementierung, Modultests, Integrationstests, Systemtests und Abnahmeprüfung. Das V-Modell hilft sicherzustellen, dass die Anforderungen erfüllt werden und die Software ordnungsgemäß funktioniert.

### 4. Spiralmodell:

Das Spiralmodell ist ein iteratives und risikobasiertes Vorgehensmodell. Es kombiniert Elemente des Wasserfallmodells mit Prototyping und Risikomanagement. Der Entwicklungsprozess im Spiralmodell durchläuft mehrere aufeinanderfolgende Schleifen, die als Spiralen dargestellt werden. Jede Schleife besteht aus vier Hauptphasen: Planung, Risikoanalyse, Entwicklung und Bewertung. Während jeder Schleife werden die Risiken identifiziert und bewertet, Prototypen erstellt und getestet, um das Verständnis der Anforderungen zu verbessern. Das Spiralmodell ermöglicht eine schrittweise Entwicklung und Integration von Funktionen und bietet Flexibilität bei der Handhabung sich ändernder Anforderungen und Risiken.

### 5. DevOps:

DevOps ist ein Vorgehensmodell, das die Zusammenarbeit zwischen der Entwicklung (Dev) und dem Betrieb (Ops) betont. Es zielt darauf ab, die Kluft zwischen den beiden Bereichen zu überbrücken und eine nahtlose Integration und kontinuierliche Bereitstellung von Software zu ermöglichen. DevOps umfasst eine Reihe von Praktiken, Tools und kulturellen Aspekten, die darauf abzielen, die Zusammenarbeit, Automatisierung, kontinuierliche Integration und Bereitstellung zu verbessern. Durch den Einsatz von DevOps können Entwicklungs- und Betriebsteams effizienter zusammenarbeiten, Software schneller

bereitstellen, kontinuierliches Feedback erhalten und auf Probleme reagieren, um die Softwarequalität zu verbessern.

Die verschiedenen Vorgehensmodelle haben jeweils ihre Vor- und Nachteile und sind für unterschiedliche Projektsituationen geeignet. Das Wasserfallmodell bietet eine strukturierte und vorhersehbare Herangehensweise, während agile Methoden Flexibilität und Kundenorientierung bieten. Das V-Modell legt den Schwerpunkt auf die Qualitätssicherung, während das Spiralmodell Risikomanagement und Prototyping betont. DevOps konzentriert sich auf die Zusammenarbeit zwischen Entwicklung und Betrieb für eine schnellere und kontinuierliche Bereitstellung. Die Wahl des geeigneten Modells hängt von den Projektanforderungen, der Komplexität und dem Kontext ab.

## Aufgabe 2) Teamentwicklung: Sollten Sie Ihren Codeabschnitt akribisch zu Ende testen oder ist es nicht besser, den halbwegs fertigen Code einem unabhängigen Dritten zu übergeben?

### Antwort

In der Teamentwicklung ist es sowohl wichtig, den Codeabschnitt akribisch zu testen als auch ihn einem unabhängigen Dritten zur Überprüfung zu übergeben. Beide Ansätze haben ihre Vorteile und ergänzen sich gegenseitig.

Das akribische Testen des Codeabschnitts durch das Entwicklungsteam ist von großer Bedeutung. Durch gründliche Tests können Fehler und Schwachstellen frühzeitig identifiziert und behoben werden. Dies hilft, die Qualität des Codes zu verbessern und potenzielle Probleme zu vermeiden. Die Entwickler haben das beste Verständnis für ihren eigenen Code und können spezifische Testszenarien erstellen, um sicherzustellen, dass der Code richtig funktioniert. Durch das Testen während des Entwicklungsprozesses können Probleme frühzeitig behoben werden, was letztendlich Zeit und Ressourcen spart.

Jedoch kann es trotz sorgfältiger Tests vorkommen, dass Entwickler "blind" für ihre eigenen Fehler werden. Sie könnten sich so sehr auf ihren Code konzentrieren, dass sie potenzielle Probleme übersehen. In solchen Fällen ist es äußerst wertvoll, den halbwegs fertigen Code einem unabhängigen Dritten zur Überprüfung zu übergeben. Ein unabhängiger Reviewer kann den Code aus einer anderen Perspektive betrachten und potenzielle Probleme oder Optimierungsmöglichkeiten identifizieren, die den Entwicklern entgangen sind. Dies erhöht die Qualität des Codes und hilft dabei, potenzielle Risiken zu minimieren.

Es ist wichtig zu beachten, dass der Zeitpunkt für die Übergabe des Codes an einen unabhängigen Dritten von Projekt zu Projekt unterschiedlich sein kann. Es kann hilfreich sein, bestimmte Meilensteine im Entwicklungsprozess zu definieren, an denen der Code für eine externe Überprüfung bereit ist. Dies stellt sicher, dass die eigenen Tests abgeschlossen sind und der Code in einem ausreichend stabilen Zustand ist, um von einem unabhängigen Reviewer überprüft zu werden.

Zusammenfassend lässt sich sagen, dass sowohl gründliche Tests durch das Entwicklungsteam als auch eine Überprüfung durch einen unabhängigen Dritten wichtig sind. Die Kombination dieser Ansätze ermöglicht es, die Qualität des Codes zu maximieren, Fehler zu minimieren und eine solide Basis für die Weiterentwicklung zu schaffen.

Aufgabe 3) Verdeutlichen Sie sich im Rollenspiel Kosten der Sicherheit die unterschiedliche Risikoposition von Entwicklern und Entwicklungsleitung. Wer wird meist optimistisch sein und warum?

Siehe: <https://hetzlefetz.github.io/uni-sicherheitsmanagement/chat/dist/index.html>