

# Software Engineering – Zusammenfassung

## 1. Einführung in Software Engineering

Software Engineering ist die systematische Entwicklung, Betrieb und Wartung von Software.

Ziele:

- Zuverlässigkeit
- Wartbarkeit
- Effizienz
- Benutzerfreundlichkeit

## 2. Softwareentwicklungsprozesse

Wasserfallmodell:

- Anforderungsanalyse
- Entwurf
- Implementierung
- Test
- Wartung

Agile Methoden (z.B. Scrum):

- Iterativ und inkrementell
- Fokus auf Zusammenarbeit
- Anpassungsfähig

## 3. Requirements Engineering

Ziel: Erfassung, Analyse, Spezifikation und Validierung von Anforderungen.

Arten:

- Funktionale Anforderungen (Was?)
- Nicht-funktionale Anforderungen (Wie?)

Techniken:

- Interviews
- Use Cases
- User Stories

## 4. Softwarearchitektur

Definition: Strukturierung von Software in Komponenten und deren Beziehungen.

Beispiele für Architekturstile:

- Schichtenarchitektur
- Client-Server
- Microservices

## 5. UML – Unified Modeling Language

Visualisierung und Dokumentation von Softwareentwürfen.

Diagrammart:

- Anwendungsfalldiagramme
- Klassendiagramme
- Aktivitätsdiagramme

- Sequenzdiagramme

## **6. Testen und Qualitätssicherung**

Testarten:

- Unit Tests
- Integrationstests
- Systemtests
- Abnahmetests

Testmethoden:

- White-Box-Test
- Black-Box-Test

Weitere Maßnahmen:

- Code Reviews
- CI/CD (Continuous Integration/Delivery)

## **7. Wartung und Weiterentwicklung**

Arten von Wartung:

- Korrektive Wartung
- Adaptive Wartung
- Perfektionierende Wartung
- Präventive Wartung

## **8. Werkzeuge und Best Practices**

- Versionskontrolle mit Git
- Issue Tracking (z.■B. Jira)
- Automatisiertes Testing
- Dokumentation mit Markdown/Wikis