

Einführung in das Software Engineering

Dozent: Dr. Thomas Fischer

Niveau: Bachelor

Modul: SE-B-03

Kapitel 1: Vorgehensmodelle der Softwareentwicklung

1. Der Software-Lebenszyklus

Software Engineering ist die ingenieurmäßige Spezifikation, Entwicklung, Qualitätssicherung und Wartung von Softwaresystemen. Jede Software durchläuft dabei einen strukturierten Lebenszyklus, der typischerweise aus den Phasen Anforderungsanalyse (Requirements Engineering), Systemdesign, Implementierung (Codierung), Verifikation (Testen) und anschließender Wartung (Maintenance) besteht.

2. Das klassische Wasserfallmodell

Das Wasserfallmodell ist das Urgestein der Vorgehensmodelle. Es ist ein strikt lineares und sequentielles Modell. Das bedeutet, dass jede Phase exakt einmal durchlaufen wird und vollständig abgeschlossen sein muss (Meilenstein), bevor die nachfolgende Phase initiiert werden darf. Ein Rücksprung in vorherige Phasen ist im Standardmodell nicht vorgesehen.

- **Vorteile:** Extrem transparente Strukturierung, einfache Budgetierung und Terminplanung, exzellent dokumentierte Zwischenergebnisse.
- **Nachteile:** Hohe Unflexibilität gegenüber Anforderungsänderungen während der Laufzeit. Da lauffähige Software erst ganz am Ende entsteht, werden fundamentale Designfehler oder Missverständnisse mit dem Kunden oft erst extrem spät entdeckt, was zu immensen Korrekturkosten führt.

3. Agile Transformation: Das Scrum Framework

Um den Defiziten starrer Modelle zu begegnen, etablierte sich die agile Softwareentwicklung. Scrum ist hierbei das am weitesten verbreitete Framework. Es bricht den linearen Ablauf auf und setzt auf eine **iterativ-inkrementelle** Entwicklung in kurzen, zeitlich fixierten Zyklen, den sogenannten ****Sprints**** (Dauer: 1 bis 4 Wochen).

Der Scrum-Prozess wird maßgeblich durch drei klar definierte Rollen gesteuert:

Rolle	Verantwortlichkeiten im Entwicklungsprozess
Product Owner	Repräsentiert die Fachseite und die Endkunden. Er maximiert den Wert des Produkts, pflegt und priorisiert das Product Backlog (die Anforderungsliste) und formuliert klare fachliche Akzeptanzkriterien.
Scrum Master	Fungiert als Servant Leader und Coach für das Team. Er ist verantwortlich für die Einhaltung des Scrum-Prozesses, moderiert die Scrum-Events und räumt methodische oder organisatorische Hindernisse (Impediments) aus dem Weg.
Developers	Das interdisziplinäre und selbstorganisierte Entwicklungsteam. Es besitzt alle Kompetenzen (Architektur, Coding, Testing), um am Ende eines jeden Sprints ein potenziell auslieferbares Produktinkrement fertigzustellen.