

Vorgehensmodelle in der Softwareentwicklung

Scrum

Scrum ist insbesondere für die agile¹ Entwicklung von Software geeignet.

Viele Entwicklungsprojekte zu sind zu komplex, um in einen vollumfänglichen Plan gefasst werden zu können.

Hier setzt Scrum an. Scrum arbeitet empirisch², inkrementell³ und iterativ⁴. Ein wesentlicher Teil der Anforderungen und der Lösungsansätze ist zu Beginn unklar. Diese Unklarheit lässt sich beseitigen, indem **Zwischenergebnisse** geschaffen und auf diesen aufgebaut wird.

Anhand dieser Zwischenergebnisse lassen sich die fehlenden Anforderungen und Lösungstechniken effizienter (schneller, genauer) finden als durch eine anfängliche, lange und meist sehr abstrakte Planungsphase.

Die empirische Verbesserung (also das Verarbeiten neu erlangter Erkenntnisse) baut auf drei Säulen auf:

- **Transparenz:** Fortschritt und Hindernisse eines Projektes werden regelmäßig und für alle sichtbar festgehalten.
- **Überprüfung:** In regelmäßigen Abständen werden Produktfunktionalitäten geliefert und sowohl das Produkt als auch das Vorgehen beurteilt und getestet.
- **Anpassung:** Anforderungen an das Produkt, Pläne und Vorgehen werden nicht initial festgelegt, sondern kontinuierlich verfeinert und angepasst. Scrum strukturiert die Komplexität einer Aufgabe in kleinere und weniger komplexe Bestandteile.

In Scrum werden die **Anforderungen aus der Anwendersicht** formuliert (z.B. durch User Stories). Die Liste dieser Anforderungen ist das **Backlog**. Diese Anforderungen werden Stück für Stück in zwei bis vier Wochen langen **Sprints** umgesetzt. **Am Ende eines Sprints steht die Lieferung eines fertigen Teilprodukts**, das in einem Zustand sein, der an den Kunden ausgeliefert werden kann (potentially shippable product).

Im Anschluss an den Zyklus werden Produkt, Anforderungen und Vorgehen überprüft und im nächsten Sprint weiterentwickelt.

→ **In Scrum wird nicht nur das Produkt, sondern auch die Planung iterativ und inkrementell entwickelt.**

1 Agil (lat. flink): Bei der agilen Software-Entwicklung wird versucht, mit geringem bürokratischem Aufwand und Regeln auszukommen und sich so schnell an Veränderungen anzupassen, ohne dabei das Risiko für Fehler zu erhöhen.

2 Empirisch (griech. Erfahrung, Erfahrungswissen): auf methodischem Wege gewonnene Erfahrung/Wissen im Projektverlauf.

3 Inkrementell: Prozess der kontinuierlichen Verbesserung im Projektverlauf aufgrund neuer Erfahrungen

4 Iterativ (lat. wiederholen): mehrfaches Wiederholen gleicher oder ähnlicher Handlungen zur Annäherung an eine Lösung

Vorgehensmodelle in der Softwareentwicklung

Wasserfall-Modell

Beim Wasserfall-Modell wird der Softwareentwicklungsprozess in Entwicklungsphasen organisiert. Das Wasserfall-Modell ist linear (nicht iterativ, wie z.B. Scrum). Im Wasserfallmodell hat jede Phase vordefinierte Start- und Endpunkte mit eindeutig definierten Ergebnissen.

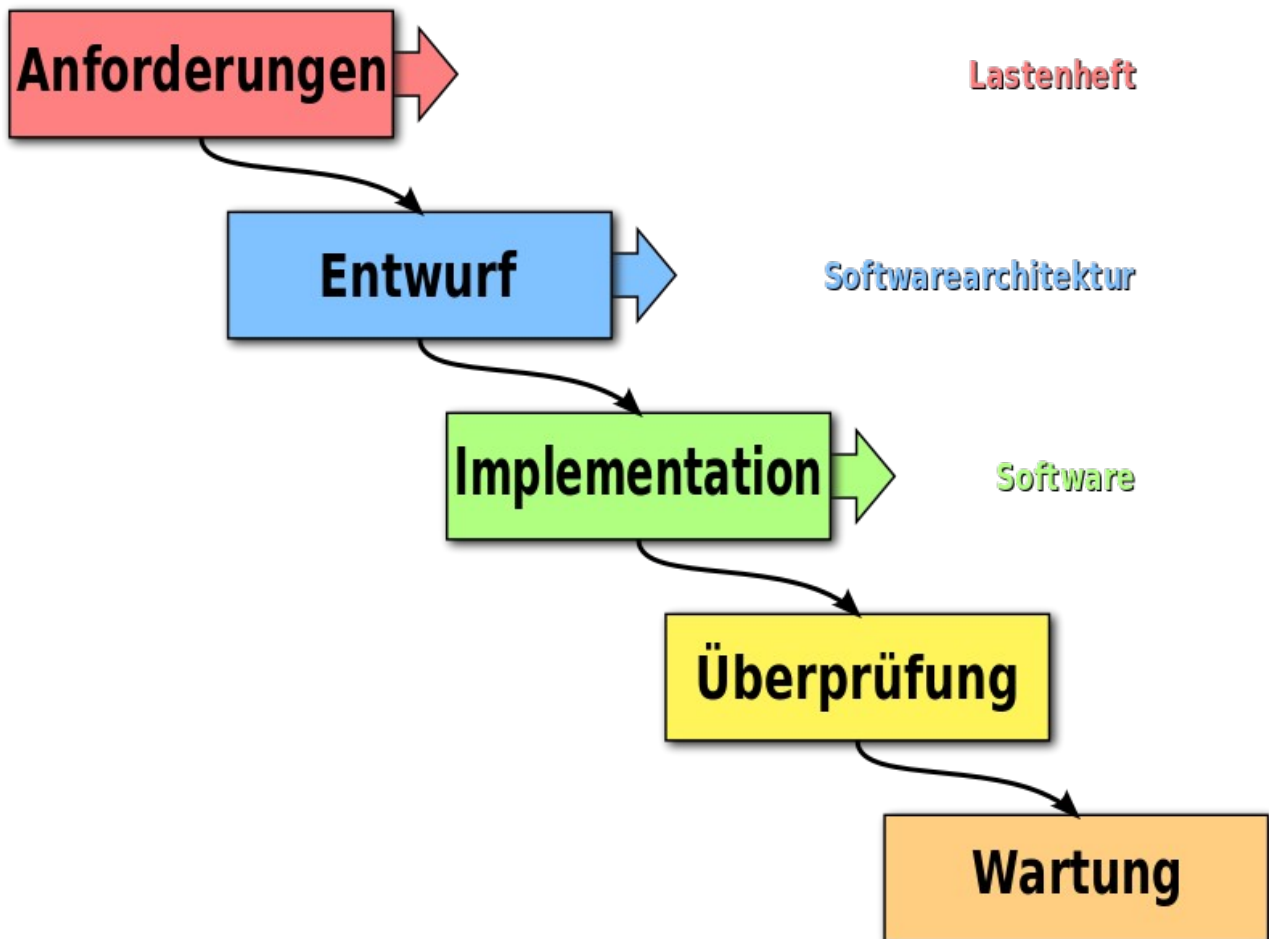


Abbildung 1: Ablauf des Wasserfall-Modells

Das Wasserfall-Modell definiert fünf Phasen:

1. **Anforderungsanalyse und -spezifikation** (engl. Requirement analysis and specification) resultiert im Lastenheft
2. **Systemdesign und -spezifikation** (engl. System design and specification) resultiert in der Softwarearchitektur
3. **Programmierung und Modultests** (engl. Coding and module testing) resultiert in der eigentlichen Software
4. **Integrations- und Systemtest** (engl. Integration and system testing)
5. **Auslieferung, Einsatz und Wartung** (engl. Delivery, deployment and maintenance)

Vorgehensmodelle in der Softwareentwicklung

V-Modell

Beim V-Modell wird der Softwareentwicklungsprozess wie beim Wasserfallmodell in Phasen organisiert. Es ist eine Erweiterung des Wasserfall-Modells und definiert neben den Entwicklungsphasen (linke Seite) auch das Vorgehen zur Qualitätssicherung (rechte Seite) phasenweise.

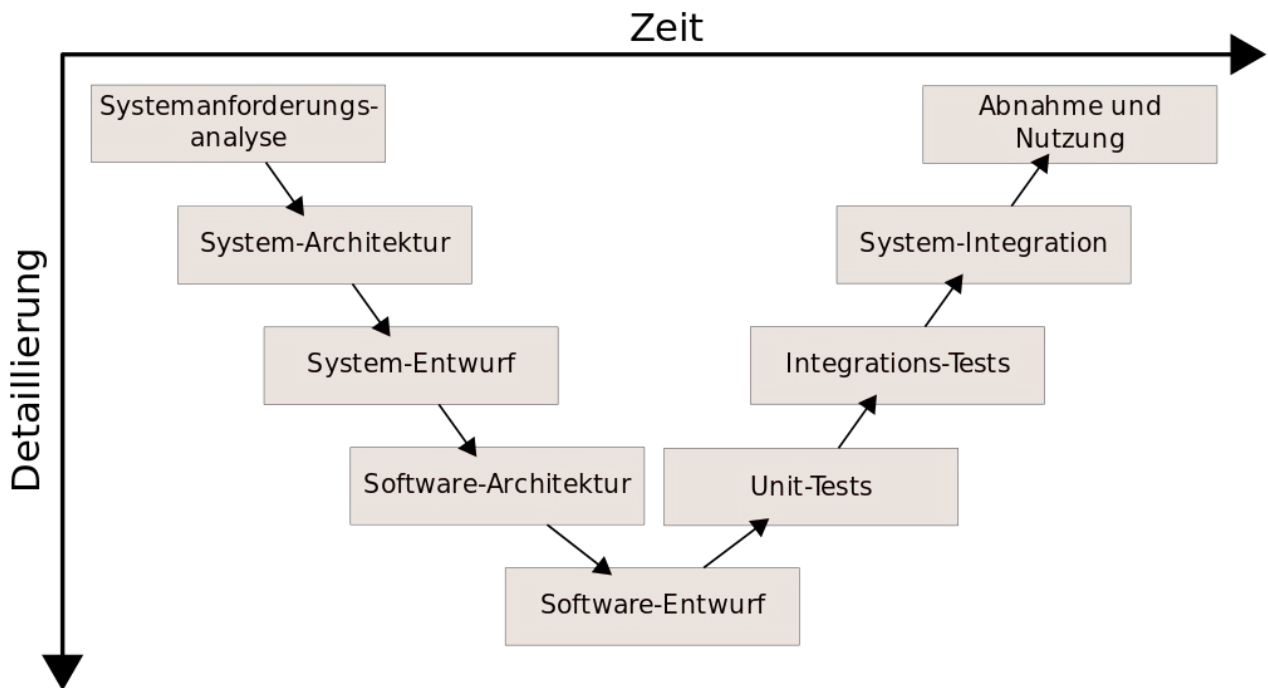


Abbildung 2: Ablauf des V-Modells

Auf der linken Seite wird mit einer funktionalen/fachlichen Spezifikation begonnen, die immer tiefer detailliert zu einer technischen Spezifikation und Implementierungsgrundlage ausgebaut wird. In der Spitze erfolgt die Implementierung (der Entwurf), die anschließend auf der rechten Seite gegen die entsprechenden Spezifikationen der linken Seite getestet wird.