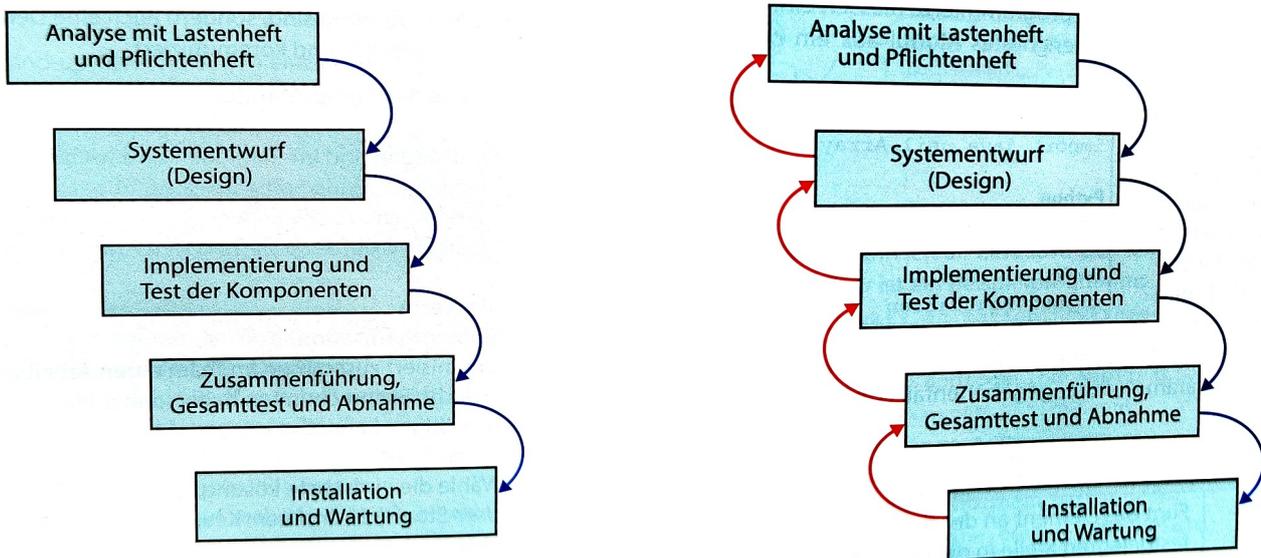


Vorgehensmodelle

Ein **Vorgehensmodell** ist ein organisatorisches Hilfsmittel um das Projektmanagement bei einer Softwareentwicklung in verschiedene Phasen in einer sinnfälligen logischen Ordnung zu strukturieren. **Ziel ist es den schrittweise erfolgenden Weg vom Problem zur Lösung systematisch vorzugeben.** Um das Risiko und die Kosten des Scheiterns zu minimieren werden phasenorientierte **Meilensteine** definiert. Meilensteine teilen den Projektverlauf in überprüfbare Etappen mit Zwischenzielen ein und erleichtern damit sowohl die Projektplanung als auch die Kontrolle des Projektfortschritts.

Wasserfallmodell

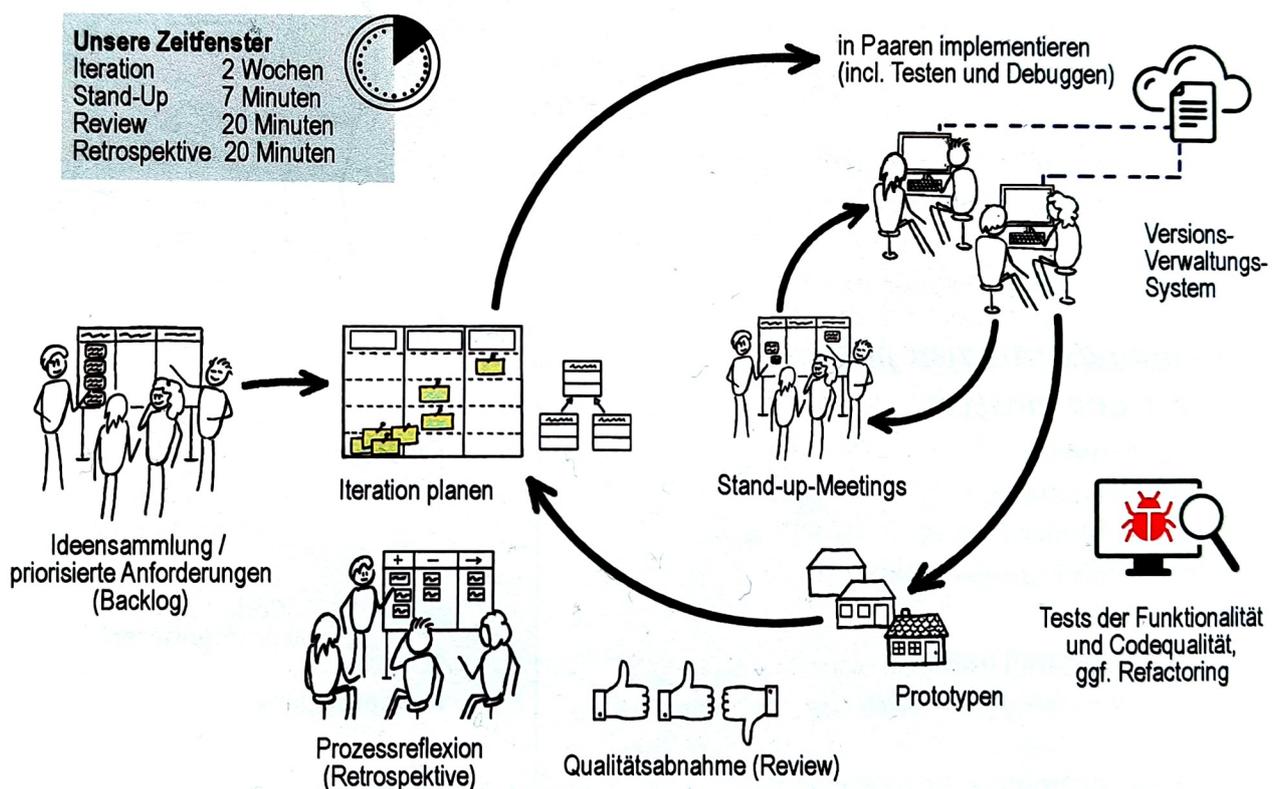


- **Analyse (Anforderungen):** Der Auftraggeber beschreibt im **Lastenheft** möglichst präzise die Gesamtheit der Anforderungen (**Was?**). Das **Pflichtenheft** beschreibt in konkreter Form, wie der Auftragnehmer die Anforderungen des Auftraggebers zu lösen gedenkt. (**Wie? Womit?**)
- **Systementwurf (Design):** Die Struktur des geplanten Softwarepakets wird erstellt. Die einzelnen Teile werden in Leistungsumfang und Schnittstellen genau beschrieben (spezifiziert).
- **Implementierung:** Die einzelnen Teile werden in parallel arbeitenden Gruppen erstellt; jedes Teil wird vollständig auf seinen geforderten Leistungsumfang hin getestet.
- **Test, Bewertung und Abnahme:** Die einzelnen Teile werden zusammengeführt; das Gesamtprodukt wird auf Fehlerfreiheit getestet. Dabei wird auch festgestellt, ob es die vereinbarten Pflichten in vollen Umfang erfüllt.
- **Installation und Wartung:** Das Produkt wird beim Auftraggeber installiert. Von hier aus kommt es oft zu Weiterentwicklungen aufgrund von auftretenden Mängeln, neuen Anforderungen oder Änderung der Rahmenbedingungen.

In der Praxis hat sich gezeigt, dass das Wasserfallmodell **Schwächen** aufweist. Zum Beispiel kann sich beim Entwurf oder in der Implementierung herausstellen, dass die Anforderungen des Pflichtenhefts so nicht umsetzbar sind. Meist ist es dann viel zu teuer mit dem Projekt komplett neu zu beginnen. Daher hat man das Wasserfallmodell durch eine **Rückkopplung** erweitert.

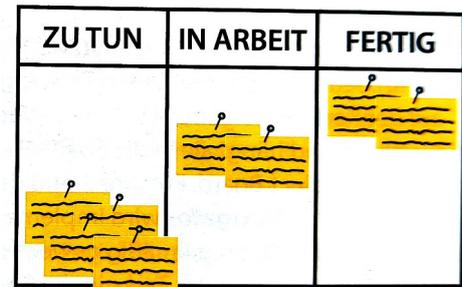
Agile Vorgehensmodelle

Die **agile Softwareentwicklung** ist eine Gegenbewegung zu den oft als schwergewichtig und bürokratisch angesehenen traditionellen Softwareentwicklungsprozessen wie dem Wasserfallmodell. Agile Softwareentwicklung zeichnet sich durch selbstorganisierende Teams sowie eine iterative und inkrementelle Vorgehensweise (als Alternative zu kaskadenartigen Vorgehen) aus. Es wird versucht, mit geringem bürokratischem Aufwand und Regeln auszukommen und sich schnell an Veränderungen anzupassen, ohne dabei das Risiko für Fehler zu erhöhen.



- **Backlog** → Grobplanung: Jede Anforderung an das Produkt wird als **User Story** (enthält Titel und knappe Beschreibung aus Nutzersicht) notiert. Die User Storys müssen priorisiert werden.

- Ein **Project Board** gibt einen Überblick über den aktuellen Bearbeitungsstand des Projekts. Wichtige Aufgabenpakete können z. B. weiter oben platziert werden. → Kanban-Board (mebis)
- Die User Storys werden in **Tasks**, also in Aufgabenpakete aus Entwicklersicht, aufgeteilt. Oft entstehen aus einer User Story mehrere Tasks. Achte auf kleine Aufgabenpakete. Für die Umsetzung muss man sich ggf. in neue Bereiche einarbeiten. Dies muss berücksichtigt werden.
- **Review** (Qualitätsabnahme): Vorstellen der Zwischenergebnisse. Ziel ist es möglichst schnell einen **Prototypen** zu implementieren und diesen dann entsprechend in seiner Funktionalität zu erweitern.
- **Retrospektive** (Reflexionsmeeting): Reflexion des Arbeitsprozesses in dieser Iteration (Was war gut/schlecht? → Verbesserungen für die nächste Iteration schriftlich festhalten)



Agile Leitsätze

Individuen und Interaktionen

sind wichtiger als Prozesse und Werkzeuge

Funktionierende Software

ist wichtiger als umfassende Dokumentation

Zusammenarbeit mit dem Kunden

ist wichtiger als Vertragsverhandlung

Reagieren auf Veränderung

ist wichtiger als das Befolgen eines Plans

Agile Werte

- **Selbstorganisation** (= aktiv Aufgaben angehen, Organisation, Zeitmanagement)
- **Einfachheit** (=einfachste Lösung wählen: Keep it small and simple (KISS))
- **Mut** (= Gute Fehlerkultur, Entscheidungen treffen, Ehrlichkeit bei (Miss-)Erfolgen)
- **Commitment** (= Verantwortung für das Team und den Lernfortschritt übernehmen)
- **Zielstrebigkeit und Fokussierung** (= Erreichen der gesteckten Ziele)

Wasserfallmodell versus agile Methoden

Vorteile agiler Methoden:

- Änderungen an Anforderungen sind jederzeit möglich.
- Iterative Entwicklung ermöglicht schnelles Reagieren auf neue Erkenntnisse oder Marktveränderungen.

- Durch kurze Entwicklungszyklen (Sprints) werden regelmäßig funktionierende Teile des Produkts geliefert.
- Kundenfeedback kann frühzeitig berücksichtigt werden.
- Kunden sind aktiv in den Entwicklungsprozess eingebunden und können Feedback geben.
- Höhere Wahrscheinlichkeit, dass das Endprodukt den Erwartungen entspricht.
- Probleme und Fehlentwicklungen werden frühzeitig erkannt.
- Durch regelmäßige Tests und Reviews werden Fehler schneller behoben.
- Ständiger Austausch zwischen Entwicklern, Stakeholdern und Kunden.
- Klare Priorisierung durch Backlogs und tägliche Stand-up-Meetings.
- Selbstorganisierte Teams haben mehr Verantwortung und Entscheidungsfreiheit.
- Fördert Kreativität und kontinuierliche Verbesserung.

Nachteile des Wasserfallmodells im Vergleich

- Starres Vorgehen, Änderungen sind teuer und schwer umzusetzen.
- Ergebnisse werden erst am Ende geliefert, wodurch Fehlschläge spät erkannt werden.
- Weniger Kundenbeteiligung während der Entwicklung.
- Höheres Risiko, dass das Endprodukt nicht den Erwartungen entspricht.

Fazit: Agile Methoden sind besonders geeignet für dynamische, komplexe Projekt wie die Softwareentwicklung mit unklaren Anforderungen, während das Wasserfallmodell besser für klar definierte, vorhersehbare Projekte funktioniert.