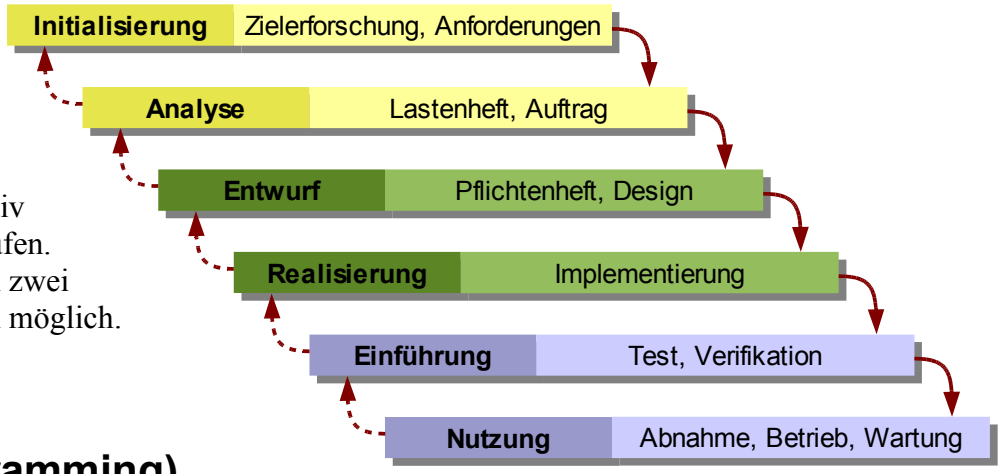
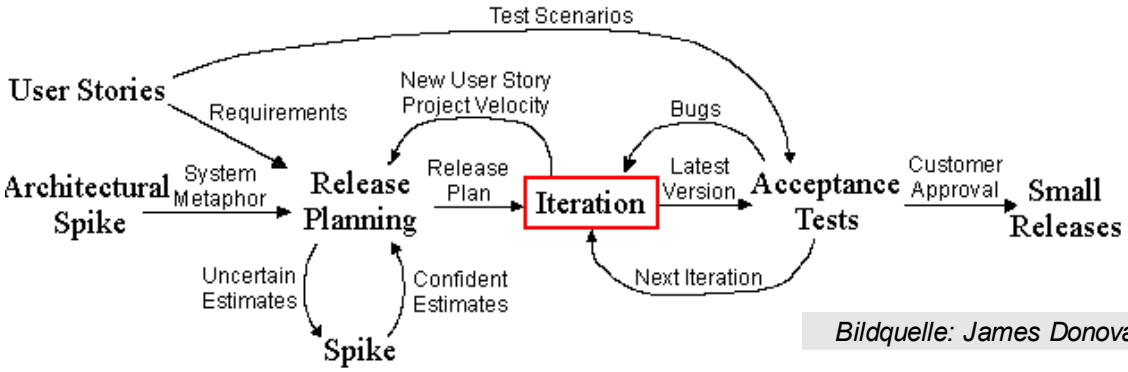
Wasserfallmodell

Beim klassischen Wasserfallmodell werden die Entwicklungsphasen relativ starr und sequentiell durchlaufen. Iterationen sind nur zwischen zwei aufeinanderfolgenden Phasen möglich.



XP (Extreme Programming)



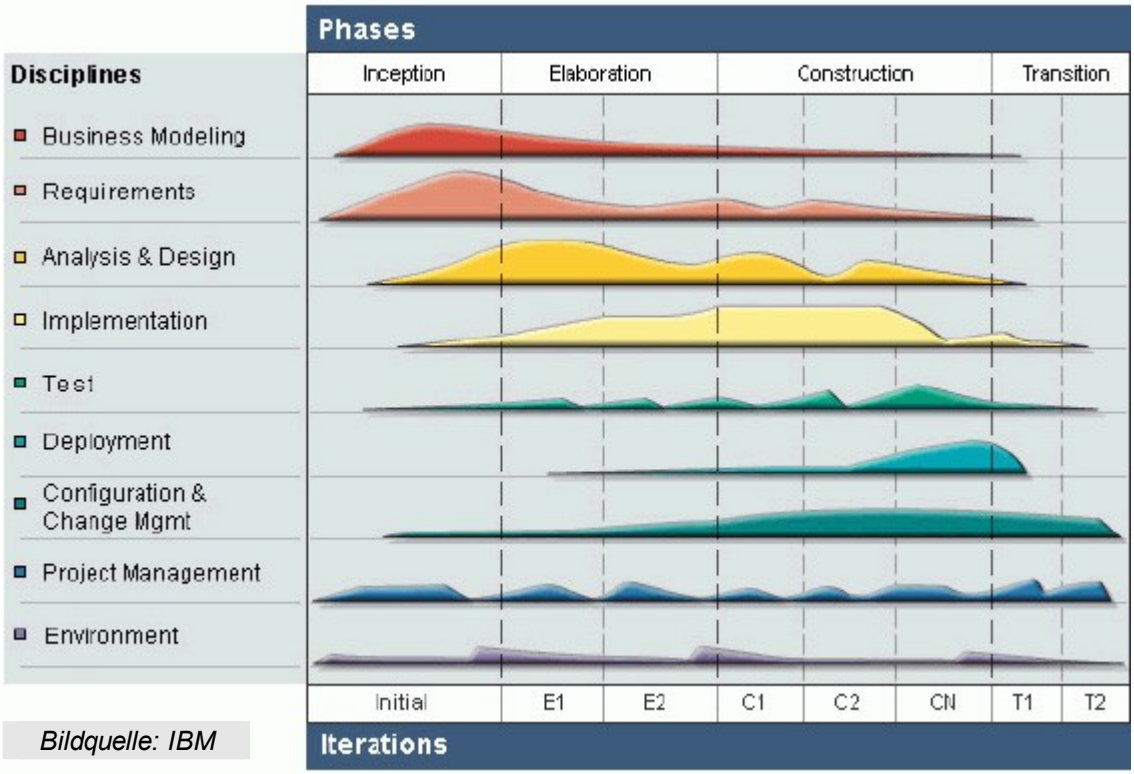
Bildquelle: James Donovan Wells

XP (Extreme Programming) beschreibt eine agile iterative Vorgehensweise beim Softwareentwicklungsprozess mit leichtgewichtiger Methodologie und wenig Dokumentation und Overhead.

Schwerpunkte sind:

- ***User Stories:*** Die Anforderungen an die zu erstellende Software werden nicht wie beim RUP in Use Cases, sondern in User Stories erfasst. User Stories beschreiben GUIs, Funktionalitäten und Testszenarien.
- ***On-site Customer:*** Ein kompetenter Vertreter des Kunden ist während der gesamten Entwicklungszeit bei den Entwicklern anwesend (dürfte nur selten realisierbar sein).
- ***Pair Programming:*** An den Entwicklungsrechnern sitzen jeweils zwei Entwickler und entwickeln gemeinsam.
- ***Testing:*** Vor der Entwicklung eines Moduls werden automatisierbare Testfälle (Unit Tests) programmiert.
- ***Simple Design:*** Es werden keine unnötigen Features implementiert.
- ***Small Releases:*** Es werden häufige Iterationen durchgeführt mit lauffähigen Programmen als Ergebnis, welche der Kunde begutachtet.
- ***Refactoring:*** Der Sourcecode wird (wenn notwendig) vergleichsweise früh restrukturiert.
- ***Continuous Integration:*** Von verschiedenen Teammitgliedern produzierter Code wird sehr häufig zusammengeführt.
- ***Collective Ownership:*** Der entwickelte Sourcecode gehört dem gesamten Team, jeder ist für jeden Code verantwortlich. Die Teams rotieren zyklisch.
- ***Coding Standards:*** Es werden Konventionen zum Aufbau des Codes erstellt, um Lesbarkeit zu erleichtern.

RUP (Rational Unified Process)



Bildquelle: IBM

RUP (Rational Unified Process) ist eine Vorgehensweise beim Softwareentwicklungsprozess mit eher schwergewichtiger Methodologie, vielen formalen Definitionen und Dokumenten, iterativ, architekturzentriert, Use-Case-getrieben, wohldefiniert und sehr strukturiert.

RUP teilt das Projekt in vier ***Phasen***:

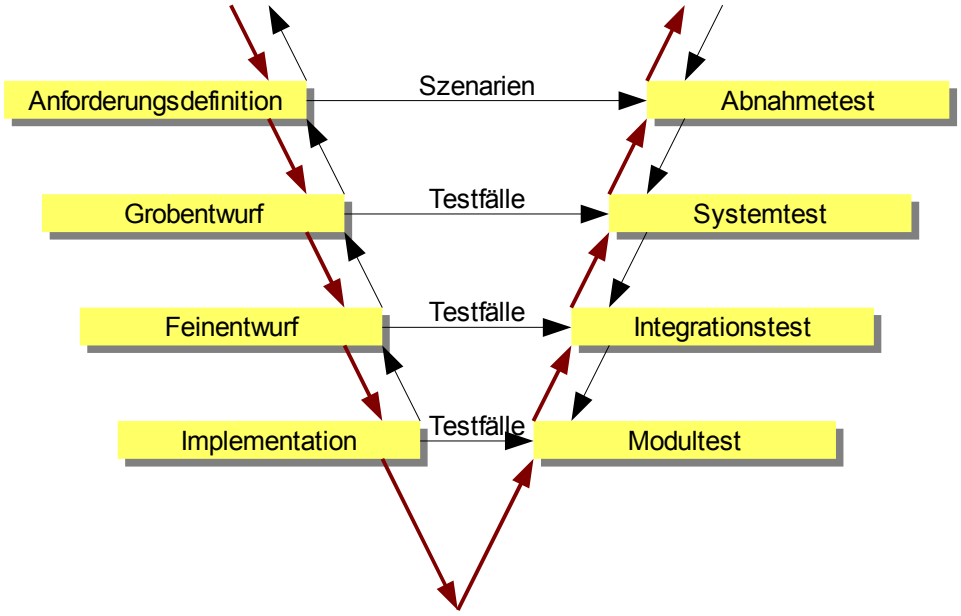
- ***Inception Phase*** (Projektsetup, Konzeptualisierung)
- ***Elaboration Phase*** (Ausarbeitung, Entwurf)
- ***Construction Phase*** (Implementierung)
- ***Transition Phase*** (Übertragung, Inbetriebnahme)

RUP definiert ***Workflows*** für neun Kernaufgaben (Disciplines):

- ***Business Modeling*** (Geschäftsprozessmodellierung)
- ***Requirements*** (Anforderungsanalyse)
- ***Analysis & Design*** (Analyse & Design)
- ***Implementation*** (Implementierung)
- ***Test***
- ***Deployment*** (Softwareverteilung)
- ***Configuration & Change Management*** (Konfigurations- und Änderungsmanagement)
- ***Project Management*** (Projektmanagement)
- ***Environment*** (Umgebung)

Innerhalb der Phasen gibt es inkrementelle Iterationen über die Workflows, die teilweise ähnlich dem Wasserfallmodell abgearbeitet werden. Zu jedem Zeitpunkt bietet RUP Planungshilfen, Leitlinien, Checklisten und Best Practices. Die konsequente und komplette Nutzung von RUP macht erst bei Teams mit über 10 Personen Sinn.

V-Modell [XT] (eXtreme Tailoring)



V-Modell ist ein Vorgehensmodell zum Softwareentwicklungsprozess, also eine Richtschnur für die Organisation und Durchführung von IT-Vorhaben. Das ***V-Modell 97***, auch ***EstdIT*** (Entwicklungsstandard für IT-Systeme des Bundes) genannt, bzw. der Nachfolger ***V-Modell XT*** ist bei vielen zivilen und militärischen Vorhaben des Bundes verbindlich vorgeschrieben. Es ist ein wenig mit dem Wasserfallmodell vergleichbar, aber frühe Phasen werden mit späten über Testdaten (V-förmig) verbunden. Es ist auch iterativ anwendbar und kann mit OOAD und UML eingesetzt werden. Phasen und zeitliche Abläufe stehen nicht im Vordergrund. Es ist sehr stark formalisiert und dokumentenzentriert und muss vor der Anwendung angepasst werden („Tailoring“).

Zum ***V-Modell 97*** sind in drei Standardisierungsebenen beschrieben:

- ***Vorgehensmodell***
  beschreibt durchzuführende Aktivitäten (Tätigkeiten) und zu erstellende Produkte (Ergebnisse).
- ***Methodenzuordnung***
  legt fest, mit welchen Methoden die Aktivitäten des Vorgehensmodells durchzuführen und welche Darstellungsmittel bei den Ergebnissen zu verwenden sind.
- ***Werkzeuganforderungen***
  legen fest, welche funktionalen Eigenschaften die Software-Tools aufweisen müssen.

Das V-Modell ist in vier Submodelle gegliedert:

- ***PM:*** Projektmanagement
- ***QS:*** Qualitätssicherung
- ***SE:*** Softwareentwicklung/Systemerstellung
- ***KM:*** Konfigurationsmanagement

Seit 2005 gibt es mit ***V-Modell XT*** (eXtreme Tailoring) einen Nachfolger des ***V-Modells 97***.
Es ist organisationsneutral, flexibel nach dem Baukastenprinzip anpassbar und wird durch Dokumentvorlagen wie beispielsweise Plan- oder Angebotsbausteine unterstützt. Es unterliegt keiner Einschränkung durch Nutzungsrechte und kann lizenzfrei adaptiert und eingesetzt werden.

Vergleich von Softwareentwicklungsmodelle

Das Wasserfallmodell hat sich in der Praxis als zu unflexibel erwiesen und kann nur bei sehr kleinen und überschaubaren Softwareprojekten sinnvoll eingesetzt werden.

Die folgende Tabelle vergleicht in stark vereinfachender Weise XP, RUP und das V-Modell (XT)

	XP	RUP	V-Modell (XT)
Abkürzung für	• Extrem Programming	• Rational Unified Process	• Vorgehens-Modell (Extreme Tayloring)
Fokus des Modells	• Entwicklungsprozess	• Projektprozess	• Unternehmensprozess
Wichtige Rollen	• Kunde • Softwareentwickler	• Softwarearchitekt	• Systemdesigner • Softwareentwickler • technischer Autor
Anforderungen und Änderungen	• nur die wichtigsten Anforderung müssen bekannt sein • sehr flexibel für Änderungen	• die meisten Anforderung sollten bekannt sein • flexibel für Änderungen	• Alle Anforderungen müssen bekannt sein. • unflexibel für Änderungen • KM (Konfigurations-management)
Projekt-komplexität	• bis mittlere Komplexität (Alle Entwickler im Team verstehen die Software)	• mittlere bis • hohe Komplexität	• hohe Komplexität
Entwicklungs-teamgröße	• kleine Teams bis 10 Personen	• mittlere bis • große Teams	• mittlere bis • große Teams
Bewertung	• leichtgewichtige Methodologie • wenig Dokumentation / Overhead • agil • iterativ • Architektur wird von allen definiert • geringer Einführungsaufwand • ***funktioniert nur mit homogenem (kleinem) Team.***	• schwergewichtige Methodologie • formalisierter strukturierter Prozess • Use-Case-getrieben • iterativ • architekturzentriert • Rolle des Architekten klar definiert. • hoher Einführungsaufwand	• sehr formalisiert und • dokumentenzentriert • Architektenaufgaben auf mehrere Rollen verteilt • hoher Einführungsaufwand