

## **IT - Projektmanagement**

# **Vergleich: XP und V-Modell**

**Fabian Hässelbarth**  
**Florian Oeser**

# eXtremeProgramming vs. V-Modell

## Inhaltsverzeichnis

### Vorstellung von XP

#### 1.1 Kriterien/ Übersicht

#### 1.2 Vorteile

#### 1.3 Nachteile

### Vorstellung von V-Modell

#### 2.1 Kriterien/ Übersicht

#### 2.2 Vorteile

#### 2.3 Nachteile

### Vergleich der beiden Modelle

#### 3.1 Tabellarischer Vergleich

#### 3.2 Ist-Vergleich

### Fazit

#### 4.1 Fazit

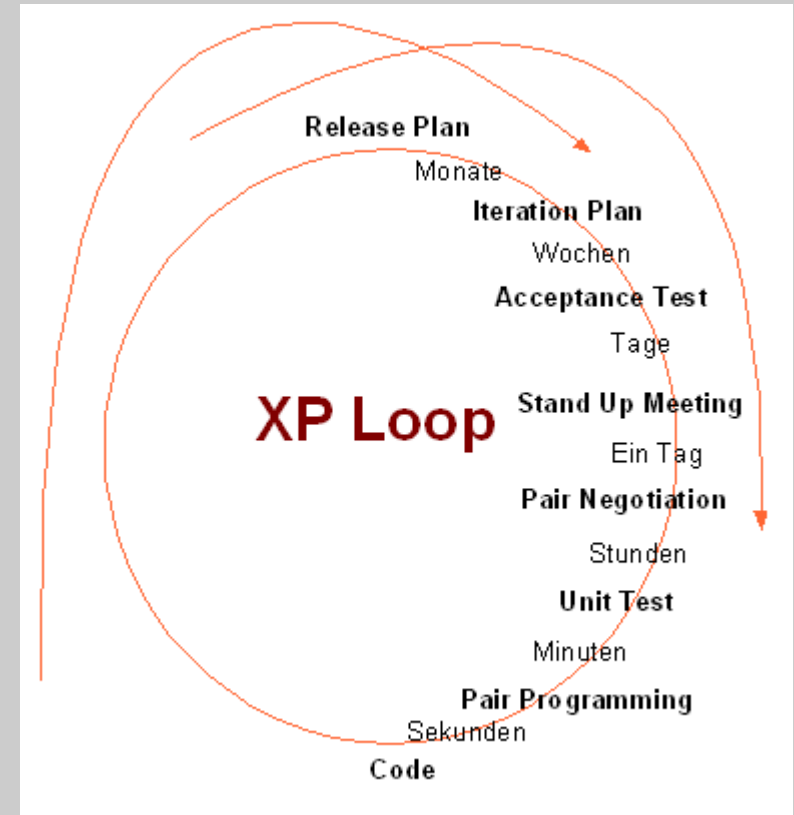
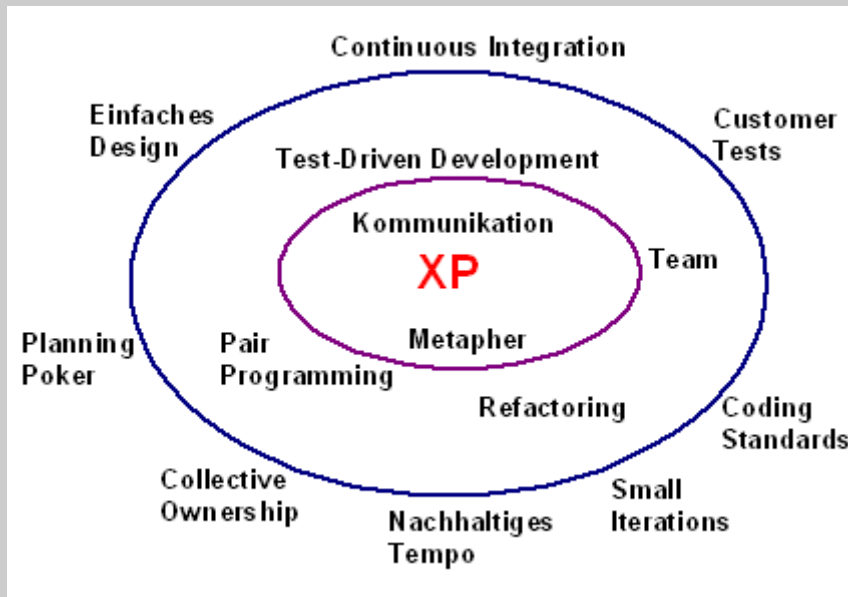
# eXtremeProgramming vs. V-Modell

## Vorstellung von XP: Kriterien / Übersicht

- agiles, iteratives Vorgehensmodell
- wenig Dokumentation, einfaches Code - Design
- ständig lauffähiges Programm (small releases)
- Kunde ist nah am Entwicklungsprozess (On - site Customer)
- Zwei-Mann-Programmierung (Pair - Programming)
- häufige Codezusammenführung (continues integration)
- User - Stories
- UnitTests vor Modullentwicklung
- Teams rotieren zyklisch
- keine unnötige Features (small design)
- Kommunikation, Offenheit und Teamarbeit stehen im Vordergrund

# eXtremeProgramming vs. V-Modell

## Vorstellung von XP: Kriterien / Übersicht



# eXtremeProgramming vs. V-Modell

## Vorstellung von XP: Vorteile/ Nachteile

- Vorteile
  - viele Zyklen/Iterationen
  - schnelle Reaktionszeit
  - hohe Qualitätssicherung
  - ständig lauffähiges Programm
  - unklare Anforderungen mgl.
  - starke Teamsymbiose
- Nachteile
  - schlechtes, globales Design
  - oft re-engineering notwendig
  - hohes Maß an Disziplinen notwendig
  - hohe Anforderung an
    - Teamleitung
    - Kunde
    - Team
  - nur für kleine (überschaubare) Projekte geeignet; keine Festpreisprojekte etc.

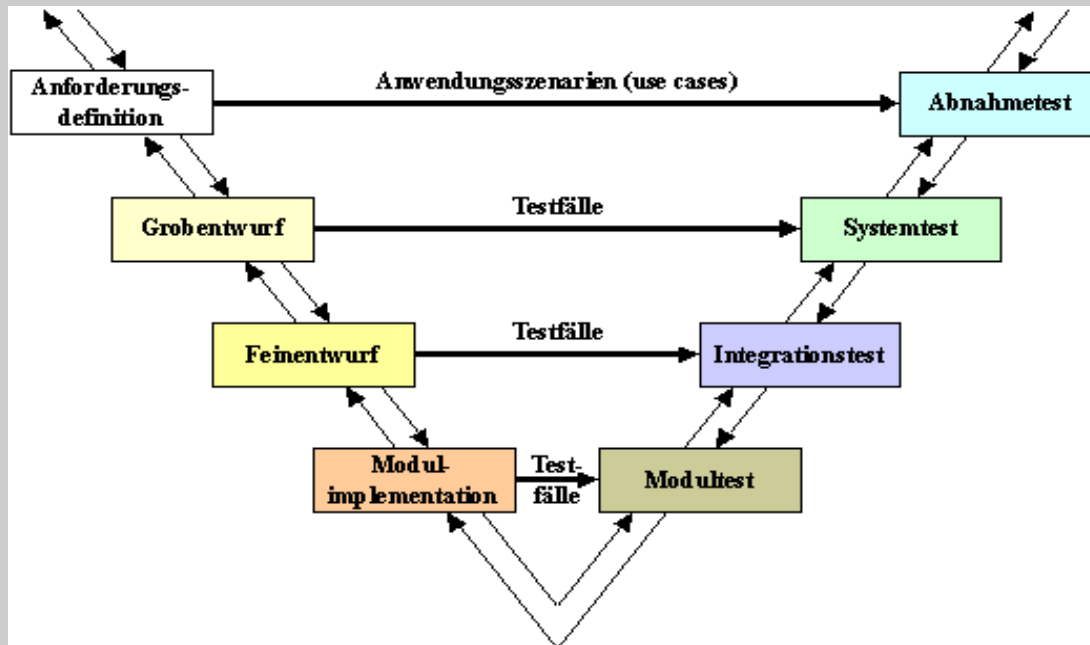
# eXtremeProgramming vs. V-Modell

## Vorstellung von V - Modell: Kriterien / Übersicht

- klassisches Vorgehensmodell
- hohe zivile- und militärische Nutzung (militärischer Hintergrund)
- vergleichbar mit Wasserfallmodell (mehr Testfälle)
- iterativ anwendbar
- Phasen und zeitliche Abfolgen stehen nicht im Vordergrund
- stark formalisiert, dokumentenorientiert
- vier Submodelle
  - Projektmanagement
  - Qualitätssicherung
  - Softwareentwicklung
  - Konfigurationsmanagement

# eXtremeProgramming vs. V-Modell

## Vorstellung von V - Modell: Kriterien / Übersicht



# eXtremeProgramming vs. V-Modell

## Vorstellung von V - Modell: Vorteile/ Nachteile

### • Vorteile

- standardisiert, öffentlich zugänglich
- sehr detaillierte Darstellung
- Integration vieler Aspekte des Entwicklungsprozesses
- gute Kostenabschätzung
- organisationsneutral

### • Nachteile

- für kleine Projekte kaum geeignet
- sehr bürokratisch/ dokumentenlastig
- zeitintensive Planung/Preproduction
- Case - Tool abhängig
- kaum flexibel



# eXtremeProgramming vs. V-Modell

## Vergleich der beiden Modelle: Tabellarischer Vergleich

	XP	V-Modell
Fokus	Entwicklungsprozess	Unternehmensprozess
Projektumfang	klein	mittel bis groß
wichtige Rollen	alle Beteiligten	Teamleiter und Projektmanagement
Testplanung	Entwickler	Qualitätssicherung (Submodell)
Änderungen	alle Beteiligten	Konfigurationsmanagement (Submodell)
Flexibilität	leicht umsetzbar	schwer umsetzbar, mit hohen Kosten verbunden
Dokumentation	wenig	viel, sehr formalisiert, dokumentenzentriert
Planung	geringer Einführungsaufwand	hoher Einführungsaufwand

# eXtremeProgramming vs. V-Modell

## Vergleich der beiden Modelle: Ist-Vergleich

- XP ist:
  - agil und interaktiv
  - wenig Dokumentation
  - geringer Einführungsaufwand
  - kleine – mittlere Projekte
  - unklare Anforderungen mgl.
  - hoher Organisationsaufwand
  - Kostenaufwand kaum abschätzbar
- V-Modell ist:
  - statisch und "klassisch" (iterativ anwendbar)
  - hoher Dokumentationsaufwand
  - hoher Einführungsaufwand
  - mittlere – große Projekte
  - Anforderungen fest definiert
  - organisationsneutral
  - Kostenaufwand gut abschätzbar

## Fazit: XP

Das XP-Modell eignet sich für kleine bis mittlere, jedoch nur bedingt für große Projekte.

XP als Vorgehensmodell eignet sich besonders gut wenn man zu jeder Zeit ein lauffähiges Ergebnis haben möchte. Der Kunde ist permanent am Entwicklungsprozess beteiligt, sodass Anforderungen nicht komplett zum Projektbeginn feststehen müssen.

Es ist ein hohes Maß an Disziplinen, Know-How und Mut notwendig, um XP wirklich anwenden zu können (aus Entwickler- und Kundensicht). Jeder im Team trägt die gesamte Verantwortung für den Erfolg des Projekts.

## Fazit: V - Modell

V-Modell eignet sich für mittel- große Projekte. Beim V-Modell steht die Planung im Vordergrund. Die Umsetzung und Realisierung der Projekte werden strikt nach Planung durchgeführt. Änderungen während der Entwicklung sind daher nur schwer umsetzbar (in-time/in-budget). Ein Ergebnis ist meist erst am Ende zu sehen. Die Verantwortung liegt beim V-Modell bei den jeweiligen Teamleitern und Projektmanagern. Vorteilhaft ist die zu meist fixe Aufwands- und Kostenabschätzung.

# eXtremeProgramming vs. V-Modell

## Quellen

- [http://de.wikipedia.org/wiki/Extreme\\_Programming](http://de.wikipedia.org/wiki/Extreme_Programming)
- <http://de.wikipedia.org/wiki/V-Modell>
- <http://www.torsten-horn.de/techdocs/sw-dev-process.htm>
- <http://www.extremeprogramming.org/>
- <http://www.v-modell.iabg.de/>
- <http://www.wi.wu-wien.ac.at/~bernroid/lehre/gzmod/ws03/e4-vorgehensmodelle.pdf>
- <http://www3.informatik.uni-erlangen.de/Lehre/UML-Seminar/SS2002/StateOfTheArt.pdf>
- iX 01/2002 S. 94ff
- [http://www.swe.uni-linz.ac.at/teaching/lva/ws03-04/se\\_uebung/05\\_gruppen/g1\\_ploesch/VModell\\_01.pdf](http://www.swe.uni-linz.ac.at/teaching/lva/ws03-04/se_uebung/05_gruppen/g1_ploesch/VModell_01.pdf)

## **IT - Projektmanagement**

# **Vergleich: XP und V-Modell**

**Vielen Dank für Ihre Aufmerksamkeit!**