Inhaltsverzeichnis

[1. Einführung: Bedeutung von Software in der heutigen Zeit 2](#_Toc32267167)

[1.1 Eigenschaften einer Software 2](#_Toc32267168)

[1.2 technische Anforderungen an Software 2](#_Toc32267169)

[1.3 unternehmensspezifische Anforderungen an Software 2](#_Toc32267170)

[1.4 Normen/Richtlinien/Standards 2](#_Toc32267171)

[2. Software-Entwicklungsprozesse 2](#_Toc32267172)

[2.2 Vorgehensweise 2](#_Toc32267173)

[2.3 Softwarewerkzeuge 2](#_Toc32267174)

[2.4 Wie wird Software bewertet 3](#_Toc32267175)

[2.4.1 Allgemeine Software Standards 3](#_Toc32267176)

[2.4.1.1 Das V-Modell XT 3](#_Toc32267177)

[2.4.1.2 Softwarelebenszyklus: ISO IEC 12207 5](#_Toc32267178)

[2.4.1.3 Reifegradmodelle: CMMI, ISO IEC 15504 (SPICE) und Automotive-SPICE, ISO IEC 3300x-Normenreihe 6](#_Toc32267179)

[3. Fallbeispiel 7](#_Toc32267180)

# Einführung: Bedeutung von Software in der heutigen Zeit

Durch die stetig wachsende Digitalisierung und Globalisierung ist es für Unternehmen von essenzieller Bedeutung ihre Geschäftsprozesse zu optimieren um zukunftsfähig zu bleiben. Doch die zunehmende Verschmelzung von virtueller und realer Fertigungswelt durch modernste IT und Software führt zu einem Paradigmenwechsel. Dabei spielt der Einsatz von leistungsstarker Software einen fundamentalen Baustein in der Entwicklung.

Der Fokus liegt nicht mehr nur in der Optimierung von Geschäftsprozessen sondern vielmehr im zugrundeliegendem System, das die einzelnen Geschäftsprozesse miteinander wie auch mit betriebswirtschaftlichen Ebenen und mit allen Wertschöpfungsstufen außerhalb des eigenen Unternehmens verknüpft. Diese zunehmende Verschmelzung der virtuellen und realen Welt durch industrielle Software birgt ein derart großes Produktivitätspotenzial, dass zukunftsorientierte Produktionsbetriebe diesem Thema oberste Priorität geben werden.

**Software für die Produktion der Zukunft**

## Eigenschaften einer Software

## technische Anforderungen an Software

## unternehmensspezifische Anforderungen an Software

## Normen/Richtlinien/Standards

* + ISO-Norm 15504
  + ISO/IEC-Norm 12207
  + ISO/IEC-Norm 15288

# Software-Entwicklungsprozesse

## 2.2 Vorgehensweise

* + arbeitsorganisatorisch
  + methodisch
  + benutzerbezogen

## 2.3 Wie wird Software bewertet

## 2.3.1 Allgemeine Software Standards

Die historische Entwicklung der in der Softwareentwicklung verwendeten Prozess- bzw. Vorgehensmodelle kann in 9 Phasen unterteilt werden:

1. erster Ansatz zur Entwicklung übersichtlicher Programme (Dijkstra, 1968)
2. Entwicklung von Software-Engineering-Prinzipien (1968-74)
3. Entwicklung von phasenspezifischen Software-Engineering-Methoden (1972-1975)
4. Entwicklung von phasenspezifischen Werkzeugen (1972-1975)
5. Entwicklung von phasenübergreifenden (integrierten) Software-Engineering-Methoden (ab 1980)
6. Entwicklung von phasenübergreifenden (integrierten) Werkzeugen (ab 1980)
7. Definition verschiedener, konkurrierender objektorientierter Methoden (OO) (ab 1990)
8. Integration der OO-Methoden zur UML (Unified Modeling Language) (ab 1995)
9. UML 2.0

Wird die Entwicklung über diese einzelnen Phasen betrachtet, zeigt sich, dass am Anfang die Fehlerarmut bei der Implementierung von Software im Fokus stand. Doch durch die Digitalisierung und dem Einfluss der Industrie 4.0 spielt der Aspekt „Umgang mit komplexen Systemen“ zunehmend eine größere Rolle. Die Verschmelzung von der virtuellen und realen Welt findet mit dem heutigen Stand der Technik als eine integrierte Prozess- und Werkzeugkette statt, die alle Phasen des Softwareentwicklungsprozesses abdeckt. Gängige Prozessmodelle werden in den folgenden Absätzen eingeführt.

## 2.3.1.1 Das V-Modell XT

Das V-Modell XT ist ein Vorgehensmodell zur Organisation und Durchführung von Software- und Systementwicklungsprojekten und löste seine beiden Vorgänger erfolgreich ab. Den Anfang machte das im Auftrag des Bundes konzipierte V-Modell[[1]](#footnote-1), das den typischen Entwicklungsprozess in definierte Phasen gliedert. Dadurch ermöglicht man ein methodisches, phasenorientiertes Vorgehen mit integrierter Qualitätssicherung und den dazugehörigen Testmaßnahmen. Die folgende Abbildung zeigt den grundlegenden Aufbau des V-Modells.

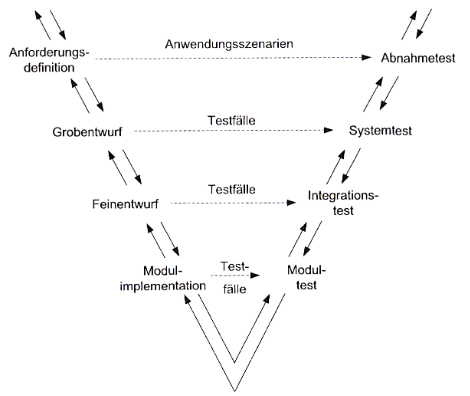


Abbildung : V-Modell nach (Diesterer et al., 2003) S.543

Auf der linken oberen Seite des V-Modells beginnt die Entwicklung mit den Anforderungen des zu entwickelnden Systems. Je tiefer man geht desto detaillierter werden die Anforderungen und desto näher kommt man zur Implementierung. An der Spitze des V-Modells ist das Ergebnis zu finden, also das fertige Produkt. Auf der rechten Seite wird diese Implementierung entsprechend den links vorgenommenen Spezifikationen getestet. Durch die Testfälle lassen sich phasenbezogene Ergebnisse für die darunter liegende Projektphase abheben.

Neue Softwareentwicklungsansätze machten eine Überarbeitung des V-Modells notwendig. Das V-Modell ist allgemeingültig verfasst und als Ergebnis wurde das V-Modell 97[[2]](#footnote-2) veröffentlicht. Im Zuge von neuen Erkenntnissen in der Softwareentwicklung wurde das V-Modell 97 durch das aktuelle V-Modell XT[[3]](#footnote-3) erfolgreich abgelöst.

Die in dem V-Modell enthaltenen vier Submodelle:

* Systemerstellung (SE)
* Qualitätssicherung (QS)
* Konfigurationsmanagement (KS)
* Projektmanagement (PM)

werden im Folgenden nach den Vorgaben des V-Modells XT auf die lösende Aufgabe bezogen betrachtet.

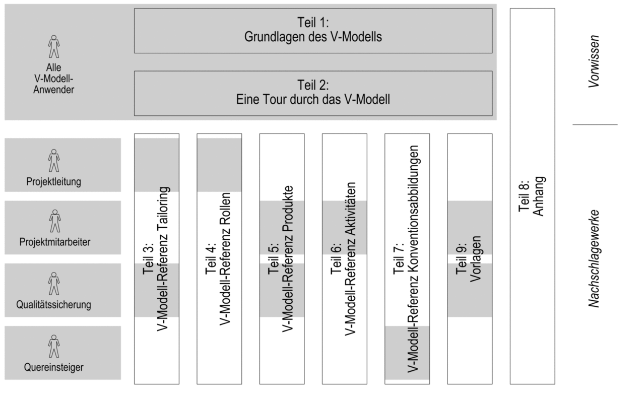


Abbildung : Aufbau des V-Modell XT 1.3 nach (IT-Beauftragte der Bundesregierung, 2006)

Das V-Modell hat folgende Zielsetzungen:

* Minimierung der Projektrisiken
* Verbesserung und Gewährleistung der Qualität
* Eindämmung der Gesamtkosten über den gesamten Projekt- und Systemlebenszyklus
* Verbesserung der Kommunikation zwischen allen Beteiligten

Das V-Modell definiert Rollen und Anforderungen an die Entwicklungsprozesse für komplexe Projekte. Für die Abwicklung öffentlicher Aufträge ist das V-Modell Standard in Deutschland und ist über das Bundesverwaltungsamt verfügbar. Die „Entwicklungsstandards für IT-Systeme des Bundes“ wurden auf Basis des V-Modells entwickelt. Das V-Modell hat sich zum Industriestandard entwickelt und zahlreiche Normen beeinflusst, die Entwicklungsprozesse adressieren. Somit ist das V-Modell XT kompatibel zu Standards wie ISO 9001 (Deutsche Norm, 2008b), CMMI, und SPICE.

## 2.3.1.2 Softwarelebenszyklus: ISO IEC 12207

S. 60 ~

## 2.3.1.3 Reifegradmodelle: CMMI, ISO IEC 15504 (SPICE) und Automotive-SPICE, ISO IEC 3300x-Normenreihe

S. 70 ~

## 2.4 Sichere Softwareentwicklungswerkzeuge und Werkzeugketten

S. 75 ~

* + Standards/Normen
  + Analyse
    - qualitative
    - quantitativ
      * eingeschränkt
      * erweitert
  + Benutzeroberfläche
  + Softwaredokumentation
  + Datensicherung
  + Datenschutz

# Fallbeispiel

1. Bröhl & Dröschel, 1995 [↑](#footnote-ref-1)
2. (Juni 1997) [↑](#footnote-ref-2)
3. (Februar 2012) [↑](#footnote-ref-3)