

# Einführung in die Softwareentwicklung

Thorsten Lemburg  
Universität Hamburg

Seminar: Softwareentwicklung in der  
Wissenschaft

# Gliederung

1. Einleitung
2. Definition von Software
3. Übersicht über Softwareentwicklung
4. Teilgebiete der Softwareentwicklung
5. Vorgehensmodelle
6. Zusammenfassung
7. Quellen

# 1. Einleitung

- Absturz der Ariane 5
- 4. Juni 1996
- Explosion 40 Sekunden nach Start
- Schaden:
  - ca. 290 Millionen Euro
  - 1 Jahr Verzögerung



# 2.1 Rolle der Software

- Software durchdringt alle Lebensbereiche
- Wirtschaft und Gesellschaft ist davon abhängig
- Erstellung wird nur ungenügend beherrscht
- Häufig scheitern auch Projekte:
  - Erstflug der Ariane 5
  - CONFIRM-Projekt

## 2.2 Definition Software

### Definition Software (IEEE 610.12)

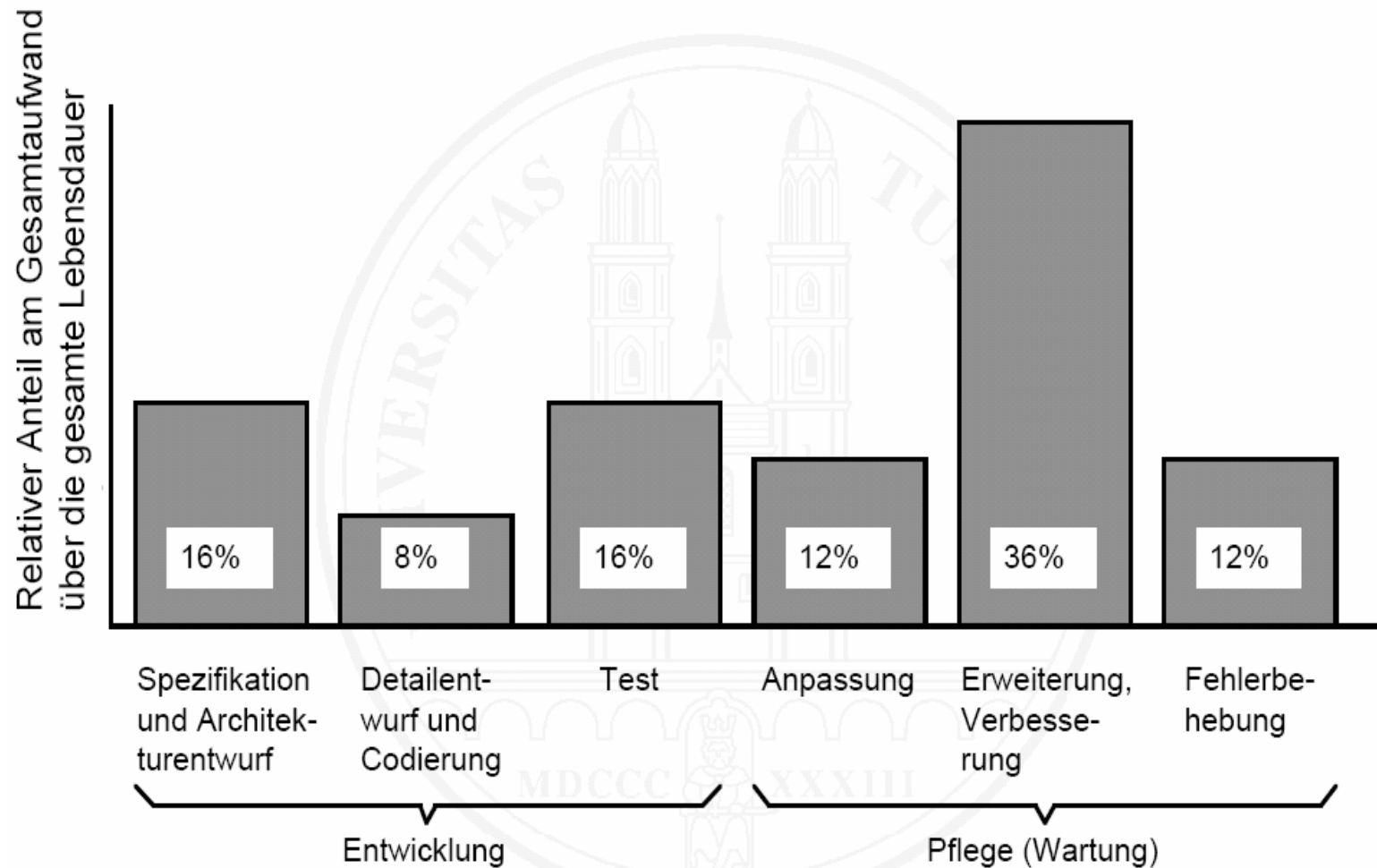
Software umfasst die Programme, Verfahren, zugehörige Dokumentation und Daten, die mit dem Betrieb eines Computersystems zu tun haben.

## 2.3 Feststellungen

### Feststellung 1:

Software umfasst erheblich mehr als nur  
Programme

## 2.3 Feststellungen



## 2.3 Feststellungen

### Feststellung 2:

Software ist ein immaterielles technisches Produkt. Man kann Software nicht anfassen.



## 2.3 Feststellungen

- Keine natürlichen Grenzen für Software
- Fehler häufig nur schwer zu finden
- Beurteilung des Entwicklungsstandes
- Scheinbar flexibel und leicht zu ändern

## 2.3 Feststellungen

### Feststellung 3:

Software verhält sich (im mathematischen Sinn)  
unstet.

# 3.1 Definition Softwareentwicklung

Definition:

Softwareentwicklung umfasst alle Tätigkeiten und Ressourcen, die zur Herstellung von Software notwendig sind. Es ist die Umsetzung von Bedürfnissen der Benutzer in Software.

# 3.1 Definition Softwareentwicklung

Konkret umfasst die Entwicklung:

- Spezifikation der Anforderungen
- Konzept der Lösung
- Entwurf und Programmierung der Komponenten
- Zusammensetzung der Komponenten
- Einbindung in vorhandene Software
- Inbetriebnahme der Software
- Überprüfung des Entwickelten nach jedem Schritt

## 3.2 Gesetzmäßigkeiten

Die Entwicklung von kleinen Programmen unterscheidet sich fundamental von der Entwicklung großer Programme.

# 3.2 Gesetzmäßigkeiten

<b>klein</b>	<b>groß</b>
Programme von bis zu ein paar tausend Zeilen oder auch alle Programme, wo der Benutzer die Übersicht im Kopf behalten kann	Längere Programme, auch bis zu Millionen Zeilen
Für den Eigengebrauch	Für den Gebrauch durch Dritte
Vage Zielsetzung genügt	Genaue Zielbestimmung, d.h. Spezifikation der Anforderungen erforderlich
Ein Schritt vom Problem zum Ziel	Mehrere Schritte: Konzept der Lösung, Entwurf der Teile, Programmieren der Teile, Zusammensetzen der Teile, Inbetriebnahme
Validierung und nötige Korrekturen finden am Endprodukt statt	Auf jedem Entwicklungsschritt muss ein Prüfschritt folgen, sonst wird das Risiko, dass das Endprodukt unbrauchbar ist, zu groß

# 3.2 Gesetzmäßigkeiten

<b>klein</b>	<b>groß</b>
Eine Person entwickelt! Keine Koordination und Kommunikation erforderlich	Mehrere Personen entwickeln: Koordination und Kommunikation notwendig
Komplexität der Software ist in der Regel klein. Das Strukturieren und Behalten der Übersicht ist nicht schwer	Komplexität sehr hoch: explizite Maßnahmen zur Strukturierung und Modularisierung erforderlich
Software besteht nur aus wenigen Komponenten	Software besteht aus vielen Komponenten, die eine Maßnahme zur Komponentenverwaltung erfordern
In der Regel wird keine Dokumentation erstellt	Dokumentation dringend erforderlich, damit Software wirtschaftlich betrieben und gepflegt werden kann
Keine Planung und Projektorganisation erforderlich	Planung und Organisation dringend erforderlich für zielgerichtetes und wirtschaftliches entwickeln

# 3.2 Gesetzmäßigkeiten

Anzahl beteiligte Personen

1

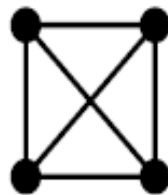
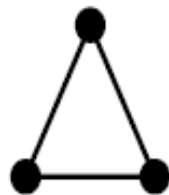
2

3

4

5

6



Anzahl Kommunikationspfade

0

1

3

6

10

15



Quantensprung:  
Kommunikation  
wird erforderlich



Quantensprung: Zahl der  
Kommunikationspfade über-  
steigt Zahl der Personen



## 3.2 Gesetzmäßigkeiten

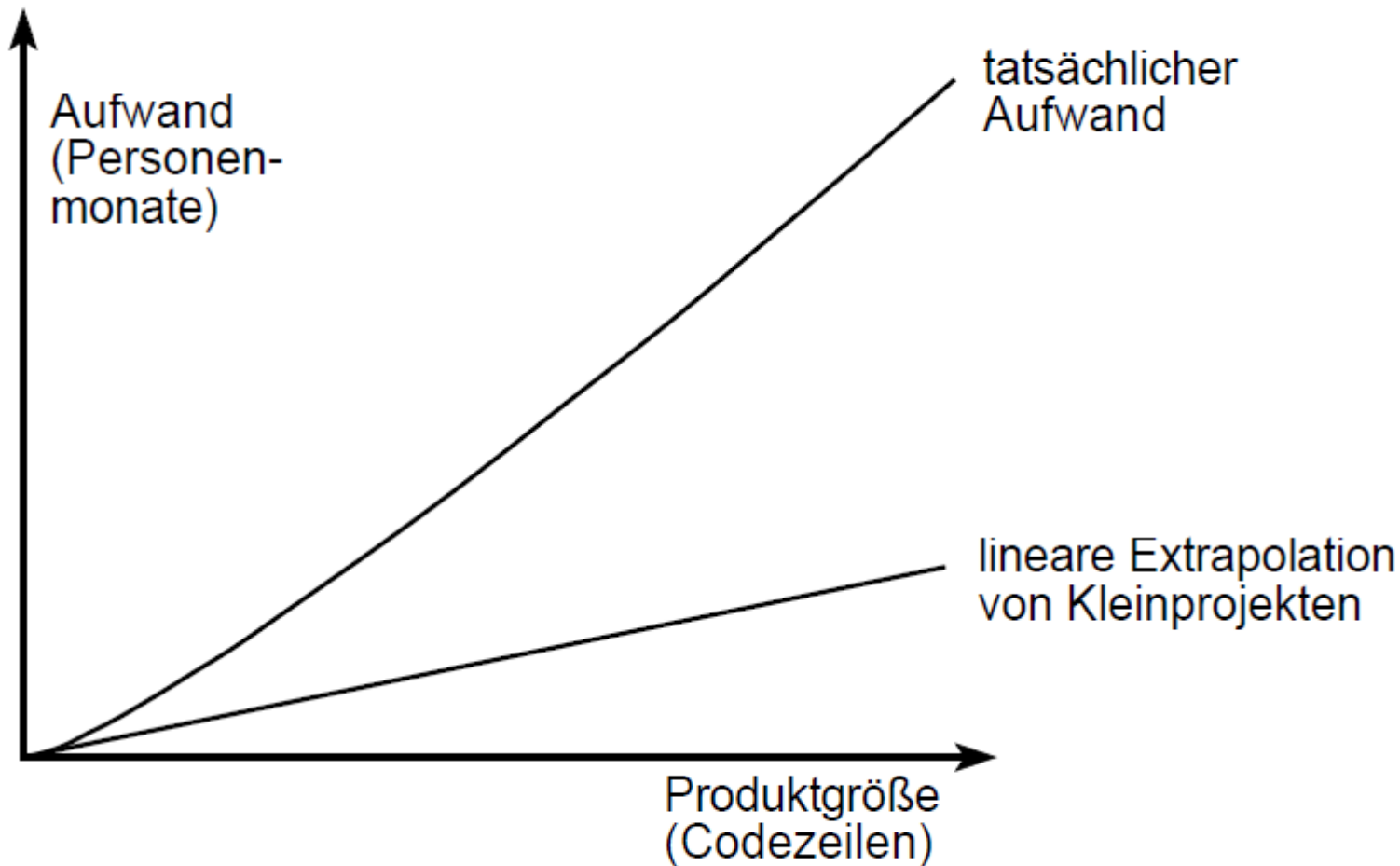
Software ist einer Evolution unterworfen.

Software wird von Menschen gemacht.

# 3.3 Warum ist Softwareentwicklung so schwierig?

- Die Größe der zu lösenden Probleme
- Software ist ein immaterielles Produkt
- Sich permanent verändernde Ziele
- Fehler infolge von Fehleinschätzungen

# 3.3 Warum ist Softwareentwicklung so schwierig?



# 3.4 Die 13 häufigsten Fehler in der Softwareentwicklung

1. Es wird mit der Codierung sofort angefangen
2. Es wird nicht systematisch bzw. unzureichend getestet
3. Festlegung der Anforderungen/Qualitätsmerkmale fehlt
4. Standards und Richtlinien werden nicht beachtet
5. Die Dokumentation fehlt, ist veraltet oder unzureichend
6. Ein Vorgehensmodell fehlt bzw. wird nicht verfolgt
7. Eine Abnahme der Phasenergebnisse erfolgt nicht

# 3.4 Die 13 häufigsten Fehler in der Softwareentwicklung

8. Schlechte Namensvergabe
9. Die Systemarchitektur ist nicht oder nur sehr umständlich erweiterbar
10. Die Schulung für die Software Ersteller und Anwender wird vernachlässigt oder als nicht notwendig erachtet
11. Die Terminvorgaben sind unrealistisch
12. Begriffe werden nicht definiert
13. Die Auswahl der Werkzeuge/Methoden ist unzureichend vorbereitet

Quelle: <http://www.theoinf.tu-ilmenau.de/~riebisch/swqs/fehler.html>

# 4.1 Projektmanagement

Definition (DIN 69901-5:2009-01)

Gesamtheit von Führungsaufgaben, -organisation, -techniken und -mitteln für die Initiierung, Definition, Planung, Steuerung und den Abschluss von Projekten

# 4.1 Projektmanagement

Definition (Project Management Institute (PMI))

Projektmanagement ist die Anwendung von Wissen, Werkzeugen und Techniken auf Projektaktivitäten, um Projektanforderungen zu erfüllen

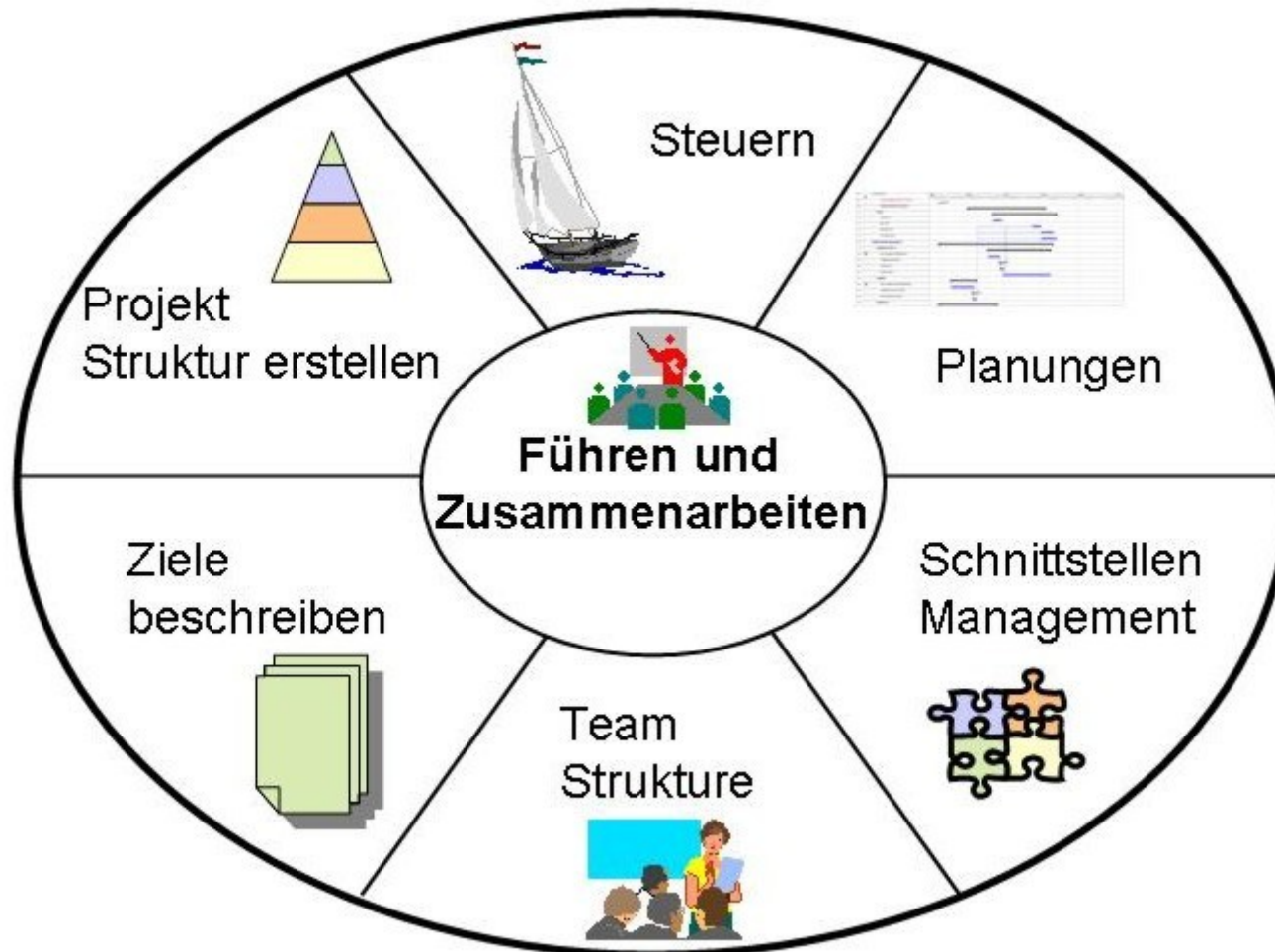
# 4.1 Projektmanagement

Definition (Gesellschaft für Informatik)

Das Projekt führen, koordinieren, steuern und kontrollieren

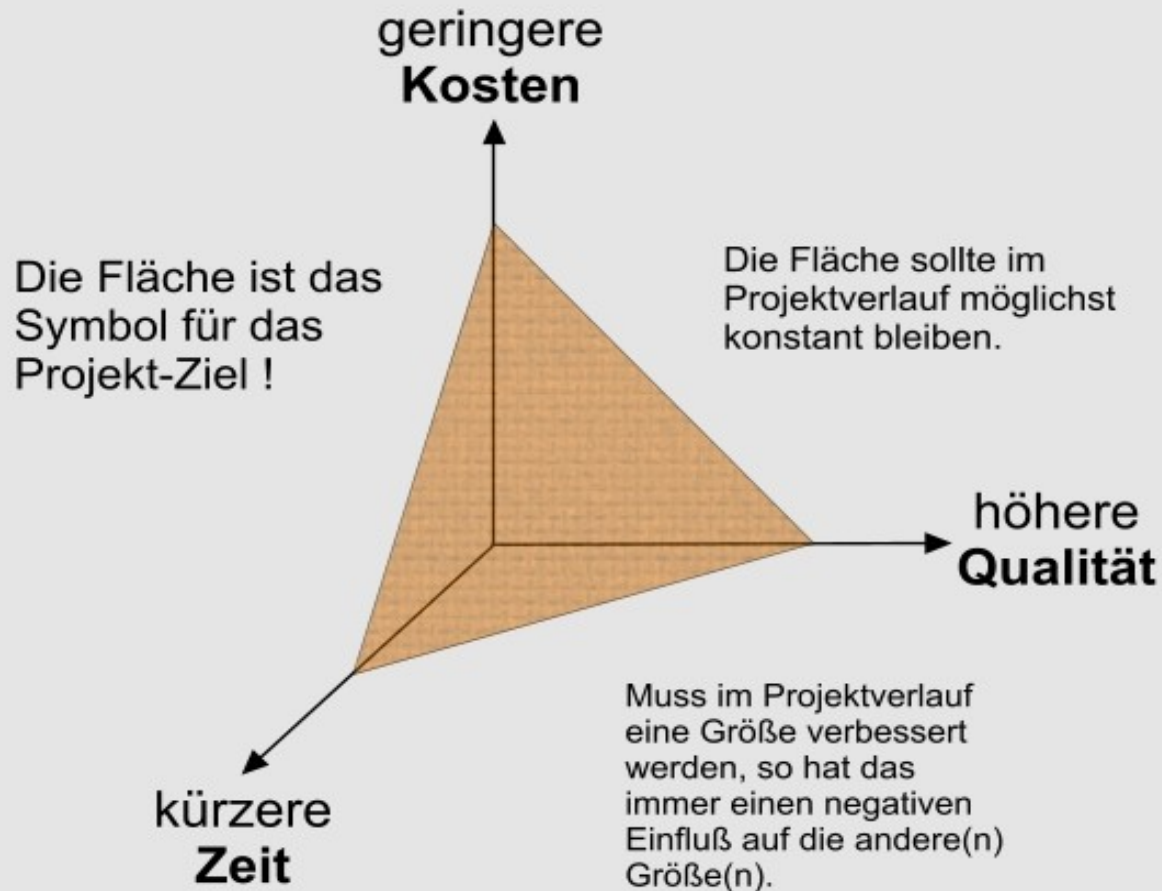


# 4.1 Projektmanagement



# 4.1 Projektmanagement

## MAGISCHES DREIECK IM PROJEKTMANAGEMENT



# 4.2 Qualitätsmanagement

Definition:

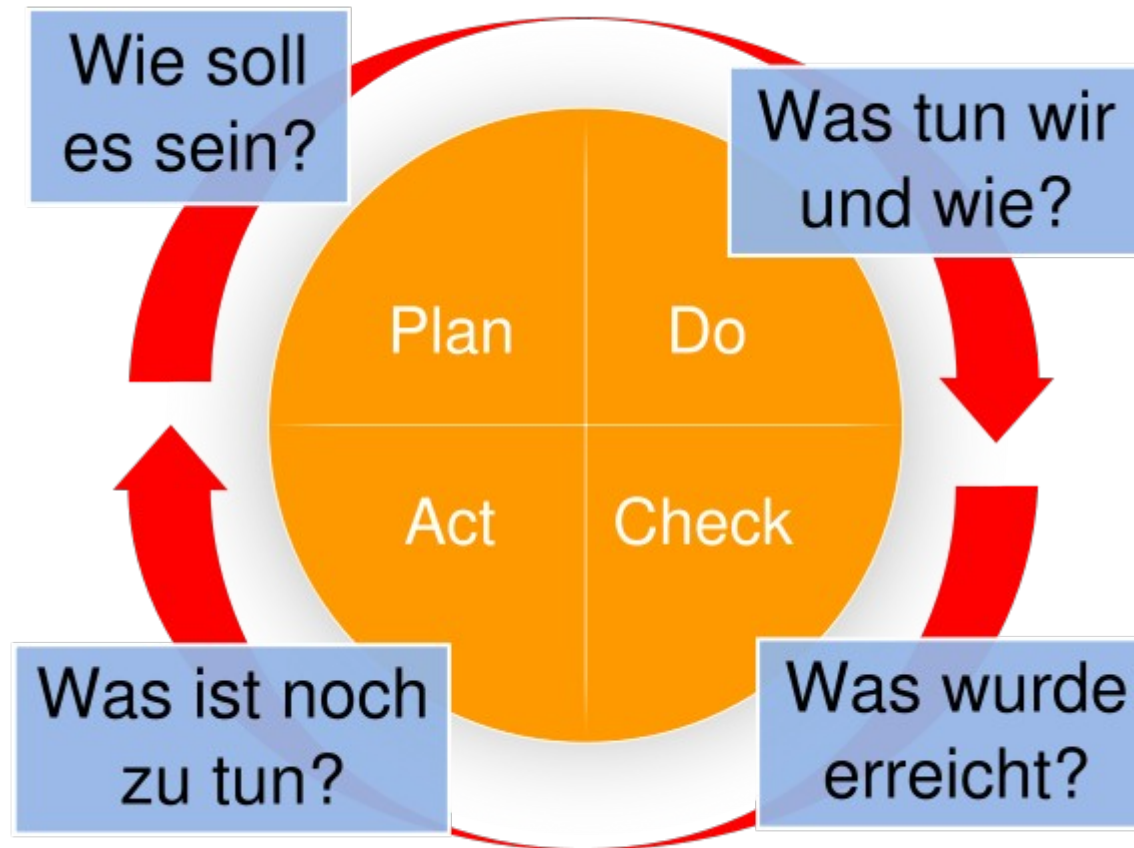
Qualitätsmanagement bezeichnet alle organisierten Maßnahmen, die der Verbesserung von Produkten, Prozessen oder Leistungen jeglicher Art dienen.

# 4.2 Qualitätsmanagement

## Definition Softwarequalität (DIN ISO 9126)

Unter Softwarequalität versteht man die Gesamtheit der Merkmale und Merkmalswerte eines Softwareproduktes, die sich auf dessen Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen.

# 4.2 Qualitätsmanagement



# 4.3 Risikomanagement

Definition:

Risikomanagement ist die systematische Erfassung und Bewertung von Risiken sowie die Steuerung von Reaktionen auf die erkannten Risiken.

# 4.3 Risikomanagement

Phasen des Risikomanagements:

- Managementplanung
- Identifikation
- Qualitative Analyse
- Quantitative Analyse
- Planung der Bewältigung
- Überwachung und Verfolgung

# 4.3 Risikomanagement

## Risikosteuerungsstrategien:

- Vermeidung
- Verminderung
- Begrenzung
- Überwälzung
- Akzeptanz



# 4.4 Anforderungserhebung

Nach IEEE kann man die Anforderungserhebung in

- Aufnahme
- Analyse
- Spezifikation und
- Bewertung

unterteilen

# 4.4 Anforderungserhebung

Beim Sammeln der Anforderungen sollte auf folgende Kriterien geachtet werden:

- Vollständig
- Eindeutig definiert
- Verständlich beschrieben
- Atomar
- Identifizierbar
- Einheitlich dokumentiert
- Nachprüfbar
- Rück- und vorwärtsverfolgbar

# 4.4 Anforderungserhebung

Strukturierung und Klassifizierung der Anforderungen unter Berücksichtigung von:

- Abhängigkeit
- Zusammengehörigkeit
- Rollenbezogen

# 4.4 Anforderungserhebung

Prüfung und Bewertung der Anforderungen auf:

- Korrektheit
- Machbarkeit
- Notwendigkeit
- Priorisierung
- Nutzbar, nützlich
- Benutzerfreundlichkeit

# 4.5 Systemdesign / technische Konzeption

- Festlegen der Programmarchitektur
- Standardsoftware: Spezifikation der geplanten Produkteinbindung oder Anpassung
- Neuer Entwurf: Entwurf des Datenmodells, Funktionen, Algorithmen, Struktur
- Vorhandene Software: Welche Veränderungen und Anpassungen?

# 4.6 Implementierung

- Konzipieren der Anwendungslösung:
  - Software konfigurieren
  - Anpassen
  - Vollständige Neuentwicklung

# 4.7 Softwaretest

## Definition:

Ein Softwaretest ist ein Test, der im Rahmen der Softwareentwicklung durchgeführt wird. Er bewertet die Funktionalität der Software gemäß ihrer Anforderungen und misst ihre Qualität. Die aus dem Softwaretest gewonnenen Erkenntnisse werden zur Behebung und Vermeidung von Softwarefehlern herangezogen.

# 4.7 Softwaretest

## Testphasen:

- Planung
- Vorbereitung
- Spezifikation
- Durchführung
- Auswertung
- Abschluss



# 4.7 Softwaretest

Verschiedene Arten von Tests:

- Komponententest
- Integrationstest
- Systemtest
- Abnahmetest

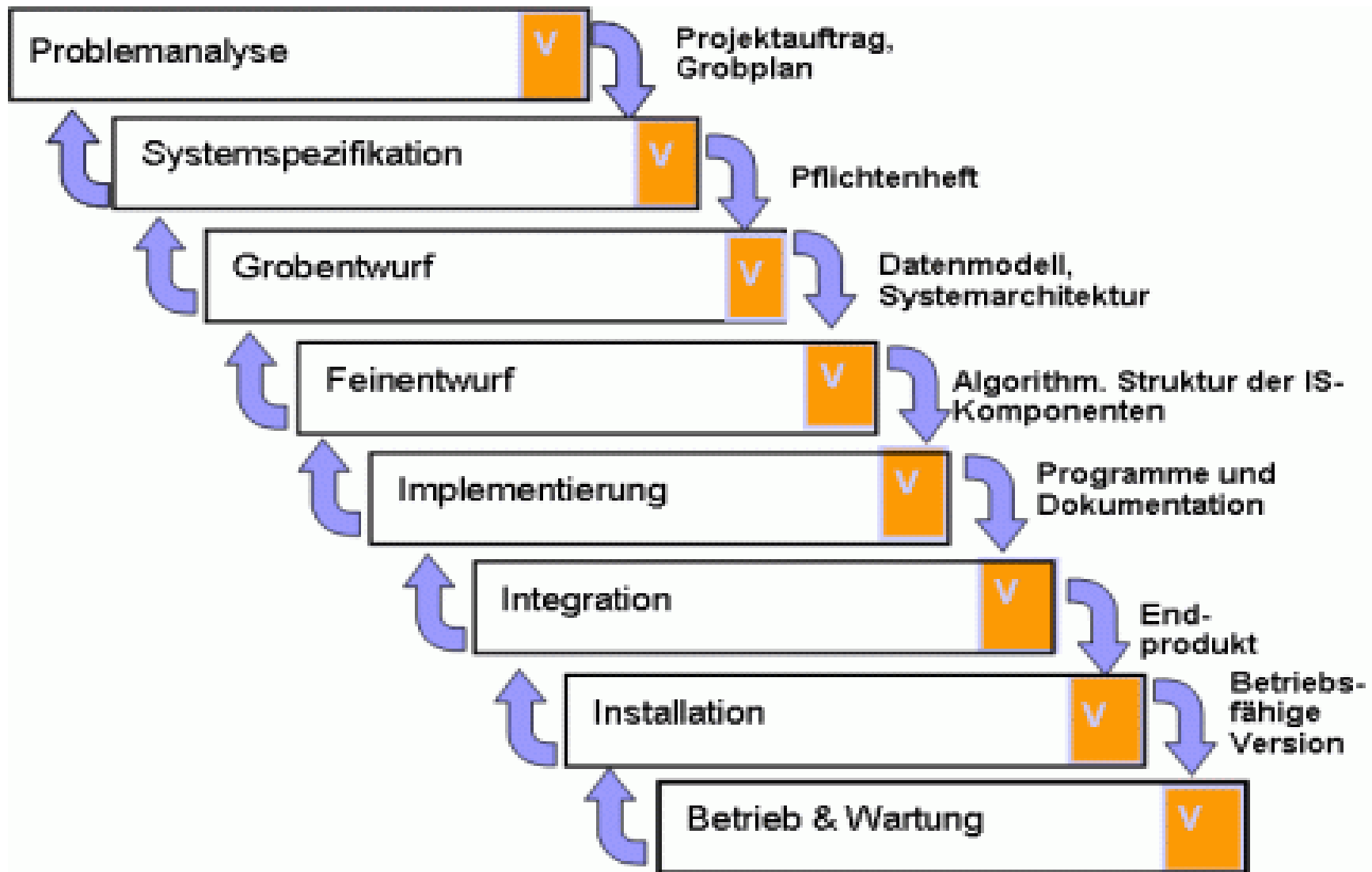
# 4.8 Softwareeinführung

- Installation der Software
- Bei größeren Projekten erst auf wenigen Testsystemen
- Einführung der Benutzer in das System mittels Schulungsmaßnahmen

## 4.9 Wartung / Pflege

- Kontinuierliche Weiterbetreuung durch Hotline oder Vor-Ort-Support
- Laufende Anpassung an sich verändernde Anforderungen
- Neue Versionen verwendeter Standardsoftware

# 5.1 Wasserfallmodell



# 5.1 Wasserfallmodell

Eigenschaften:

- Phasen in vorgegebener Reihenfolge zu bearbeiten
- Am Ende jeder Phase steht ein fertiges Dokument
- Ablauf ist sequentiell
- Top-Down-Verfahren
- Einfach, verständlich, wenig Aufwand
- Benutzerbeteiligung nur am Anfang

# 5.1 Wasserfallmodell

## Vorteile:

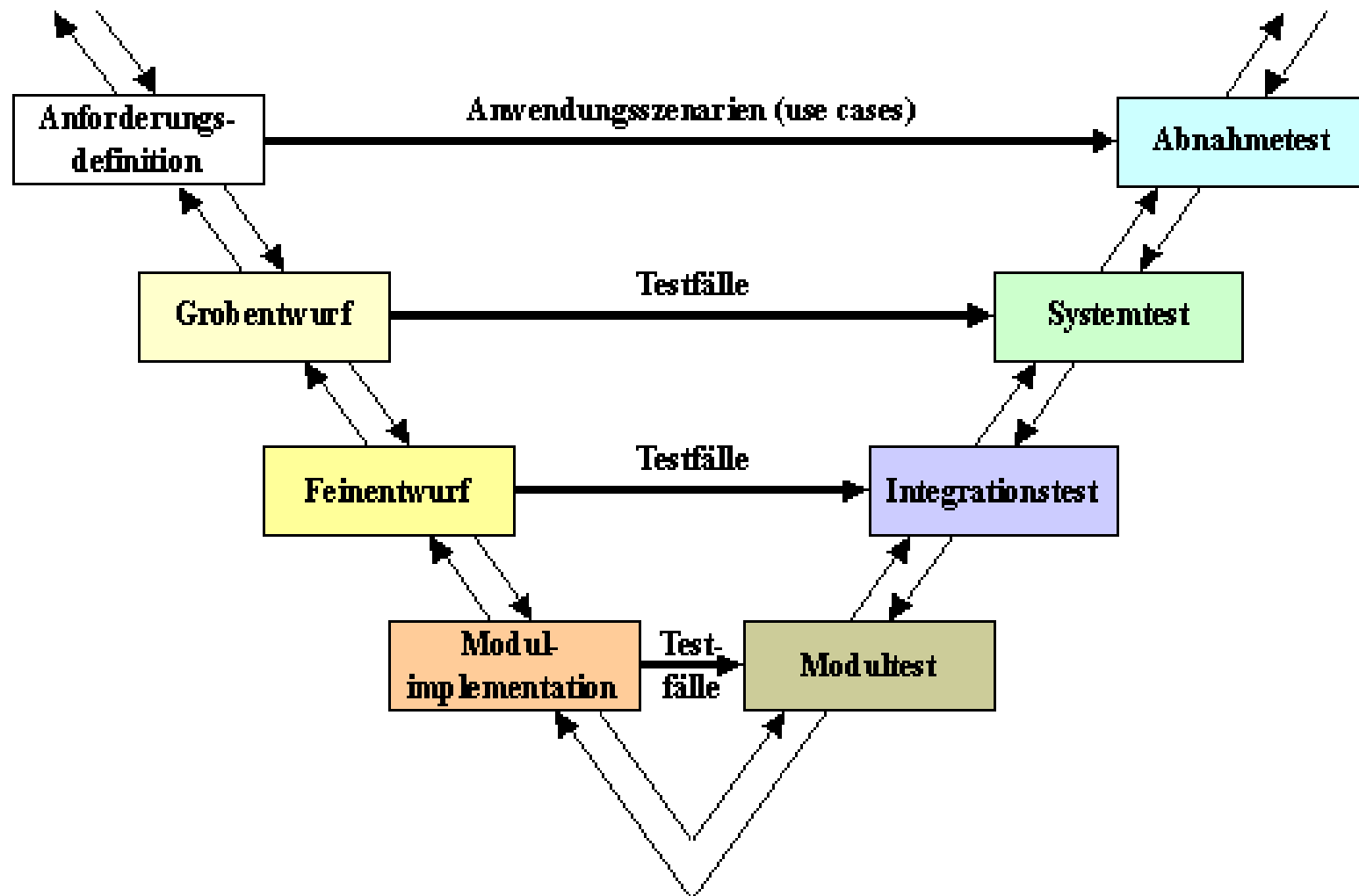
- Klare Abgrenzung der Phasen
- Einfache Möglichkeit der Planung und Kontrolle
- Bei stabilen Anforderungen und klaren Zielen ist es ein effektives Modell

# 5.1 Wasserfallmodell

Nachteile:

- Nur für einfache Projekte
- Unflexibel
- Frühes Festlegen der Anforderungen problematisch
- Fehler nur mit erheblichen Aufwand zu beseitigen

# 5.2 V-Modell





# 5.2 V-Modell

## Vorteile:

- Leichte Anwendbarkeit
- Umfassendes Modell mit hoher Sicherheit
- Gut geeignet für große Projekte

# 5.2 V-Modell

## Nachteile:

- Für kleine und mittlere Projekte ungeeignet
- Die definierten Rollen (bis zu 25) sind unrealistisch
- Ohne Werkzeugunterstützung schwer handhabbar

## 5.3 Weitere Modelle

- Spiralmodell
- Prototypen-Modell
- Objektorientiertes Modell
- Evolutionäres Modell

Nähere Informationen dazu siehe Ausarbeitung!

# 6. Zusammenfassung

- Software ist überall!
- Software umfasst mehr als Programme
- Softwareentwicklung umfasst alle Tätigkeiten und Ressourcen zur Erstellung von Software
- Man muss unterscheiden zwischen kleinen und großen Programmen!
- Es gibt mehrere Teilgebiete bei der Entwicklung
- Es sollte immer ein Vorgehensmodell verfolgt werden
- Die typischen Fehler sollten vermieden werden

# 7. Quellen

[http://www.it-infothek.de/fhtw/semester\\_3/se\\_3\\_01.html](http://www.it-infothek.de/fhtw/semester_3/se_3_01.html)  
<http://www8.informatik.uni-erlangen.de/IMMD8/Lectures/S>  
[http://archiv.rhein-zeitung.de/on/96/07/24/topnews/fehler\\_](http://archiv.rhein-zeitung.de/on/96/07/24/topnews/fehler_)  
<http://www.realtime-solutions.de/softwareentwicklung/proj>  
<http://www.innovationen-machen.de/index.php?id=5778>  
<http://ti.uni-due.de/ti/de/education/teaching/ss06/pet/folien>  
<http://www.theoinf.tu-ilmenau.de/~riebisch/swqs/fehler.htm>  
<http://de.wikipedia.org>  
<http://www.ifi.uzh.ch>