

Softwareentwicklung in der Praxis

1

Masi Aziz

Fakultät für Mathematik, Informatik und Naturwissenschaften Fachbereich
Informatik

Seminar „Softwareentwicklung in der Wissenschaft“

Dr. Hermann Lenhart

Gliederung

- Softwareentwicklung
- Top-Down- und Bottom-Up
- Teilgebiete
- Softwarefehler
- Entwicklungsplan
- Softwaretest
- Endgültige Software bzw. Softwareeinführung
- High performance Computing
- Quellenangabe

Softwareentwicklung

- Übersetzung des engl. Begriffs *software engineering*
- Beschäftigt sich mit der Herstellung und Entwicklung von Software

„Zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Softwaresystemen.“

Allgemeine Definition von Helmut Balzert (2001)

Top-Down und Bottom-Up

- Top-Down und Bottom-Up sind Ansätze zur Programmentwicklung
- Bei der Top-Down Methode liegt der Schwerpunkt auf Planung und Verständnis
- Code kann erst geschrieben werden, wenn System gut genug geplant ist

- Beim Bottom-Up sind Programmbestandteile klar definiert
- Aus diesen Teilen werden Elemente des Programms zusammengesetzt
- Programmieren beginnt schneller

- Beides wird in der Praxis miteinander verknüpft

Teilgebiete

- Planung
- Analyse
- Entwurf
- Programmierung
- Validierung und Verifikation

Softwarefehler

- ▶ Softwarefehler werden auch „Bug“ genannt
- ▶ Bezeichnet im Allgemeinen ein Fehlverhalten von Computerprogrammen
- ▶ Vor der Behebung von Programmfeldern wird die Projektphase Softwaretest durchlaufen
- ▶ Bug – Tracker
- ▶ Softwarepatch

Arten von Programmfehlern

- Lexikalische Fehler
- Syntaxfehler
- Semantische Fehler
- Logische Fehler
- Designfehler
- Laufzeitfehler

Wirtschaftliche Bedeutung

- ▶ Verursachen Kosten
- ▶ Laut IX / Studie aus dem Jahre 2006
- ▶ 84,4 Mrd€ betragen die jährlichen Verluste durch Softwarefehler
- ▶ 14,4 Mrd€ jährlich werden für die Beseitigung von Programmfehlern verwendet
- ▶ 70 Mrd€ betragen die Produktivitätsverluste durch Computerausfälle aufgrund fehlerhafter Software

Vermeidung von Programmfehlern

- Planung
- Analysephase
- Entwurfsphase
- Programmierung
- Beim Testen
- Im Betrieb

Fehlerfreiheit

- ▶ Völlige Fehlerfreiheit für Software, ist weder erreichbar noch nachweisbar
- ▶ Selbst teure oder auch vielfach getestete Software können Fehler enthalten
- ▶ Stabilität und Robustheit
- ▶ Eine Software gilt erst dann als stabil und robust , wenn Fehler nur sehr selten auftreten

Klassifizierung von Fehlern

- Aufgetretene Fehler werden im Fehlermanagement bearbeitet
- Hier durchläuft jeder Fehler einen Klassifizierungsprozess
- Bestehend aus vier Schritten:
 - Recognition
 - Investigation
 - Action
 - Disposition

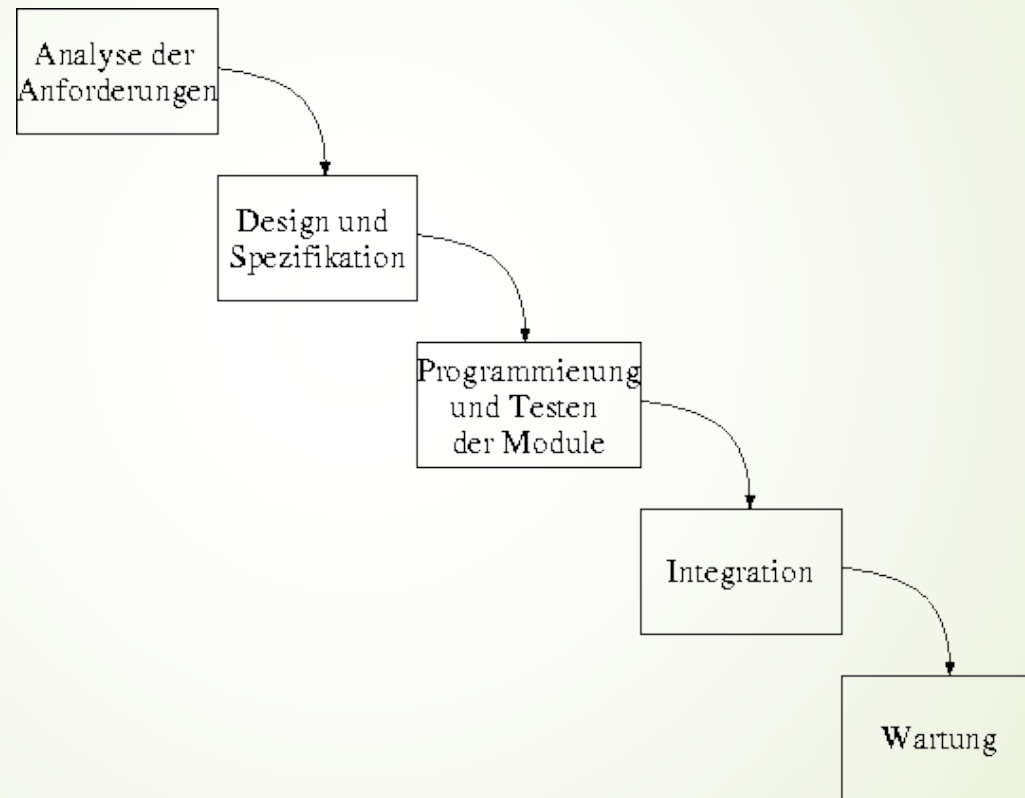
Folgen von Programmfehlern

- ▶ Luft – und Raumfahrt
- ▶ Medizin
- ▶ Verkehr
- ▶ Finanzwesen
- ▶ Kommunikation

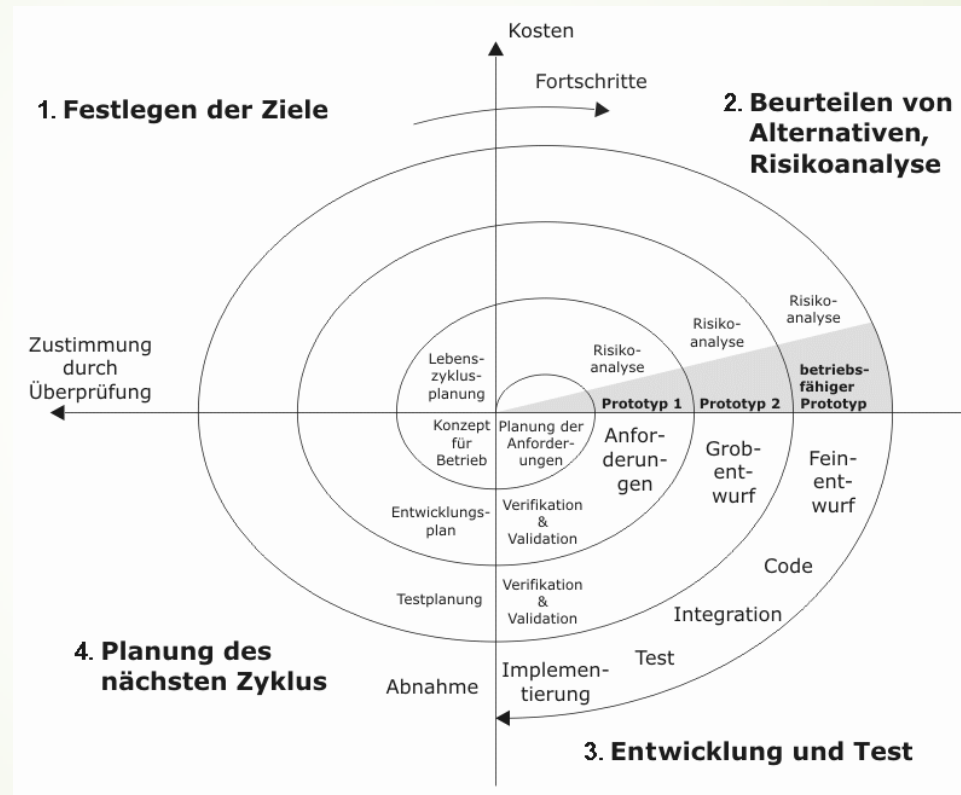
Entwicklungsplan

- ▶ Softwareentwickler entwickeln Software anhand eines Planes
- ▶ Vorgehensmodelle unterteilen des Entwicklungsprozess
- ▶ Entwicklungsprozesse haben zwei Tätigkeitsgruppen
- ▶ Analyse von Geschäftsprozessen
- ▶ EDV-technische Realisierung

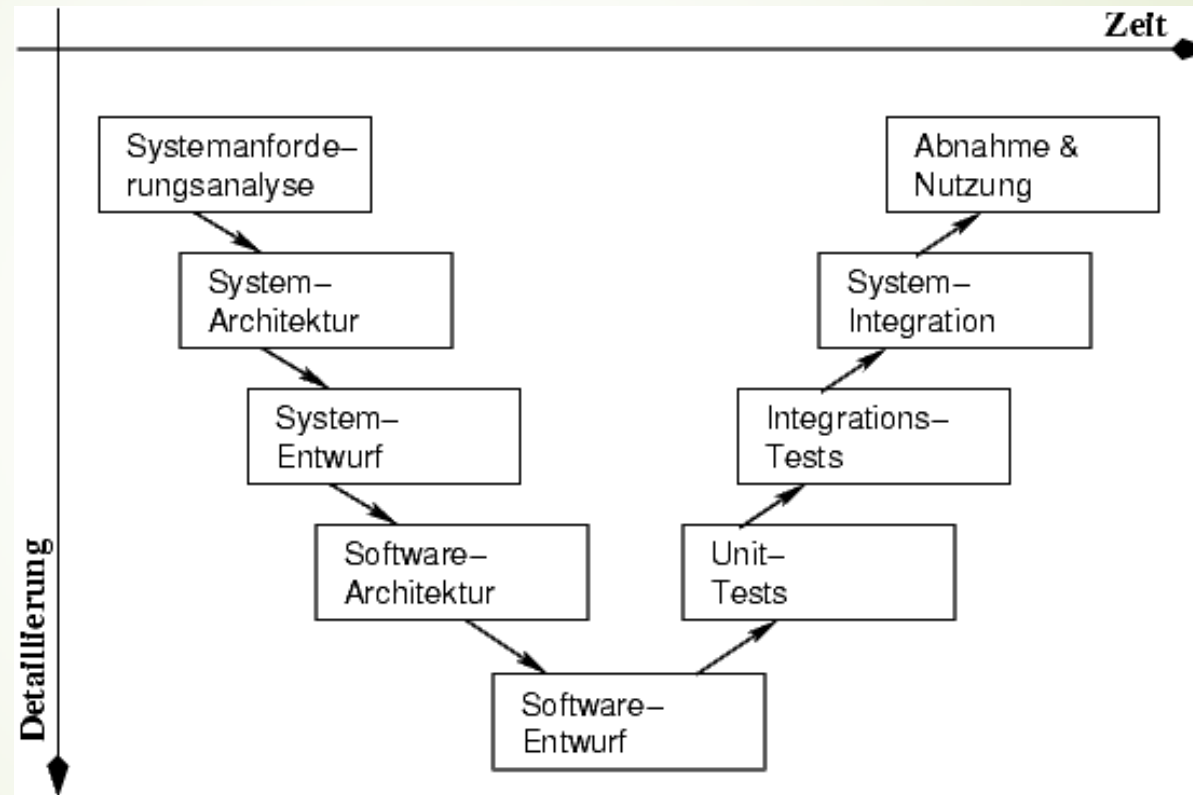
Wasserfallmodell



Spiralmodell



V - Modell



Softwaretest

- Ein Softwaretest prüft und bewertet Software
- Die Überprüfung der Software ist wichtiger Bestandteil
- Standardisierung
- Quantitativ
- Qualitativ
- Risikominimierung

Rollen und Funktionen im Software-Test

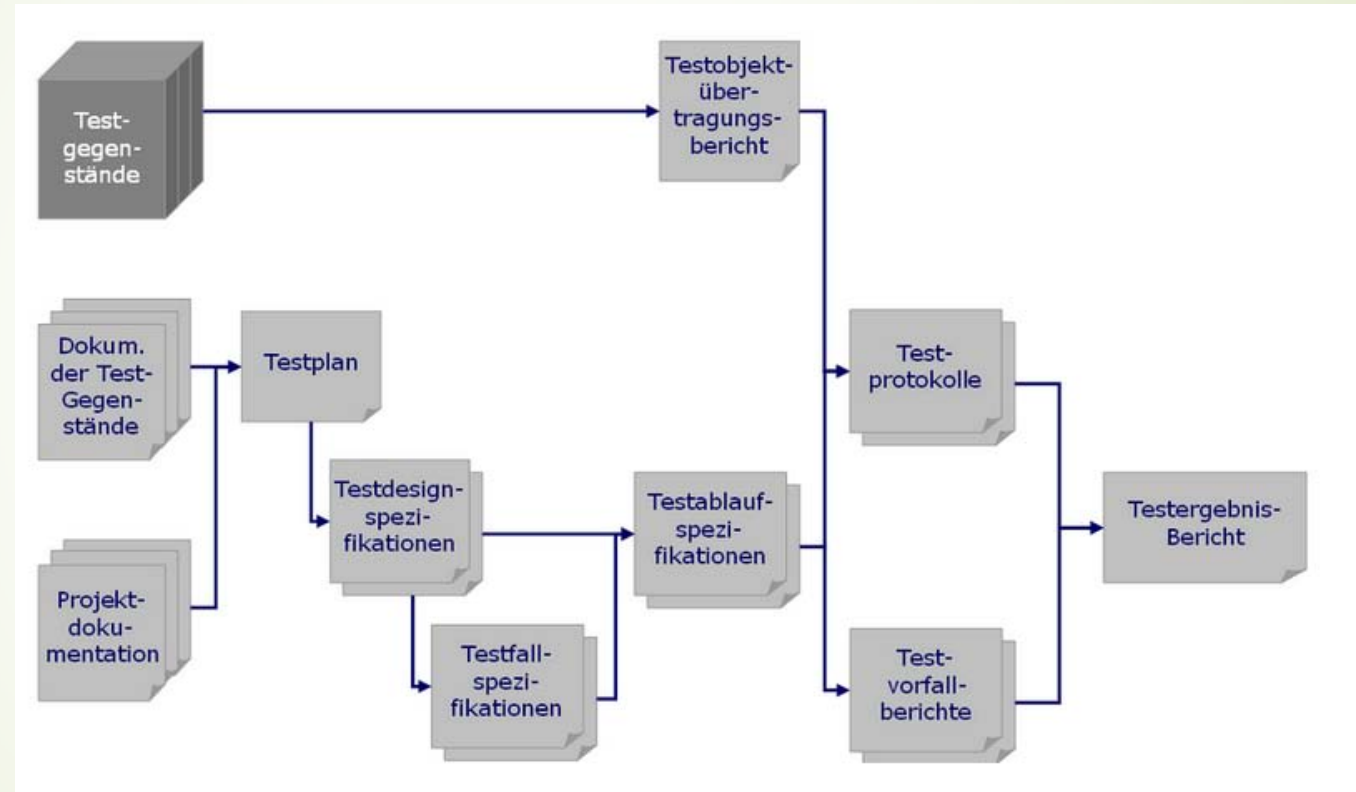


[<http://www.software-saxony.de/wp-content/uploads/2008/02/professioneller-software-test.pdf>]

Testprozess

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

Dokumentation



[http://de.wikipedia.org/wiki/Softwaretest#mediaviewer/Datei:IEEE829_Uebersicht_Deutsch.png]

Softwareeinführung

- Projektartige Vorgang der Etablierung einer Software in größeren Organisationen
- Datenmigration kann zustande kommen
- Altsystem wird abgelöst
- Strategien
- Big Bang
- Iterative Einführung

Zusammenarbeit

- Wissenschaftler haben keine IT – Kenntnisse
- Sie fangen einfach an zu Programmieren
- Haben keine Erfahrung
- Lasten und Pflichtenheft ist nicht vorgegeben
- Softwareentwickler sollen helfen
- Pair Programming wäre ein Lösung

High performance Computing

- ▶ Geowissenschaftler entwickeln HPC Softwaresysteme und stehen vor Problemen
- ▶ Softwareingenieure beobachten die Entwicklung
- ▶ Viele Wissenschaftler haben ihre Grundausbildung von anderen Wissenschaftlern
- ▶ Obwohl die Wissenschaftler öfters Software geschrieben, setzen Sie sich an die Ausbildung der Softwareentwickler fest

High performance Computing

- ▶ Programmierer sollen Kommunikation angeben
- ▶ Codes müssen geschrieben werden, um HCP Systeme anzuspannen
- ▶ Oft vereinigt ein einziges System vielfache Programmiersprachen

Zusammenfassung

- Ansatz zur Softwareentwicklung muss hergestellt werden
- Softwareentwickler entwickeln Software anhand eines Planes
- Es gibt verschiedene Vorgehensweisen
- Testprozesse verfeinern Vorgehensweisen
- Verschiedene Varianten um Software endgültig "einzuführen"

Quellenverzeichnis

- ▶ Enzyklopädie Wikipedia
- ▶ <http://www.software-saxony.de>
- ▶ Understanding the High-Performance-Computing Community / Paper