

## Das W-Modell

### Vorteile der agilen Prozesse in einem konservativen Umfeld nutzen

Prof. Dr. Andreas Spillner  
Hochschule Bremen  
Zentrum für Informatik und Medientechnologien

## Übersicht

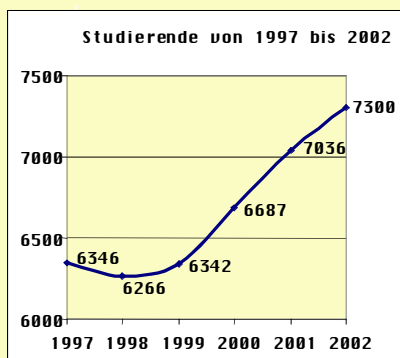
- Hochschule Bremen
- Motivation
- Prozessmodelle der Softwareentwicklung
- Vorteile der agilen Prozesse
- V-Modell
- W-Modell
  - Testvorbereitungen parallel zur
    - Erhebung der Anforderungen
    - Erstellung der Systemarchitektur
    - Spezifikation der Komponenten
  - Testdurchführung und Debugging
  - Vor- & Nachteile
- Certified Tester
- Zusammenfassung und Hinweise

## Hochschule Bremen - Geschichte

- 1799 Bremische Navigationsschule
- 1894 Technikum der freien Hansestadt Bremen
- 1919 Soziale Frauenschule
- 1968 Wirtschaftsakademie
- 1982 Hochschule Bremen



## Hochschule Bremen - heute



Jahr	Anzahl	
	Studiengänge	Studenten
1982	8	3600
2002	42	7300

### Auszeichnungen

- 2000 - Best-Practice Award
- 2001 - Reform-Fachhochschule
- 2001 - Marketing-Preis des DAAD

## Hochschule Bremen - heute

- über 7000 Studierende
- 9 Fachbereiche mit insgesamt 42 Studiengängen
  - davon 2/3 Internationale Studiengänge mit verpflichtendem Auslandsaufenthalt
  - 20 ingenieur- und naturwissenschaftliche Studiengänge
  - 11 wirtschaftswissenschaftliche Studiengänge
  - 5 geistes- und sozialwissenschaftliche Studiengänge
  - 6 postgraduale Master-Studiengänge
- mehr als 200 Hochschulkooperationen weltweit



## Fachbereich Elektrotechnik und Informatik

### Studiengänge

- Elektrotechnik (ET)
  - Studienrichtung Elektrische Energietechnik
  - Studienrichtung Informationstechnik
- Technische Informatik (TI)
  - Studienrichtung Angewandte Informatik
  - Studienrichtung Automatisierungstechnik
- Europäisches Studium Technische Informatik (ESTI)
- Internationaler Studiengang Microsystems Engineering (ISMS)
- Medieninformatik
- Digitale Medien (HSB, UNI, HfK)



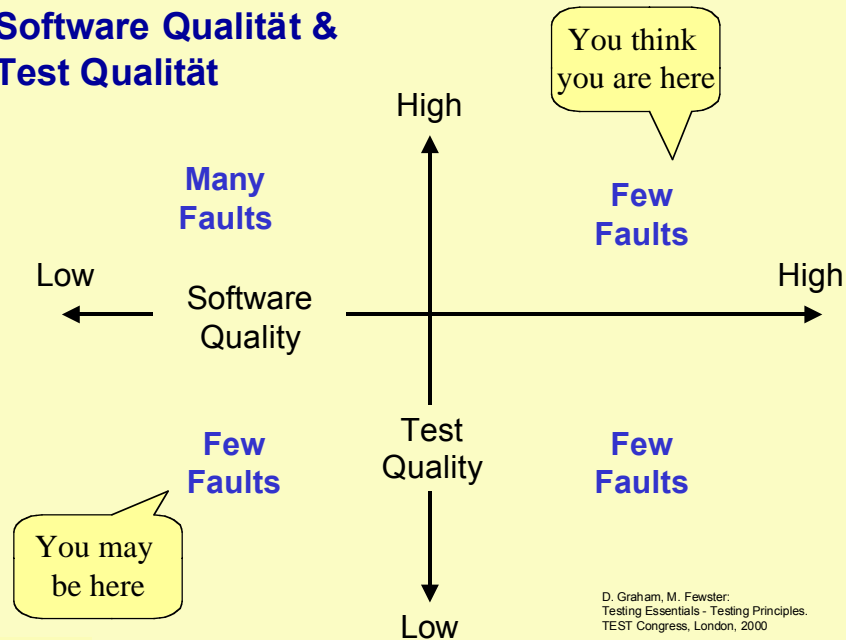
## Software-Kosten & -Qualität

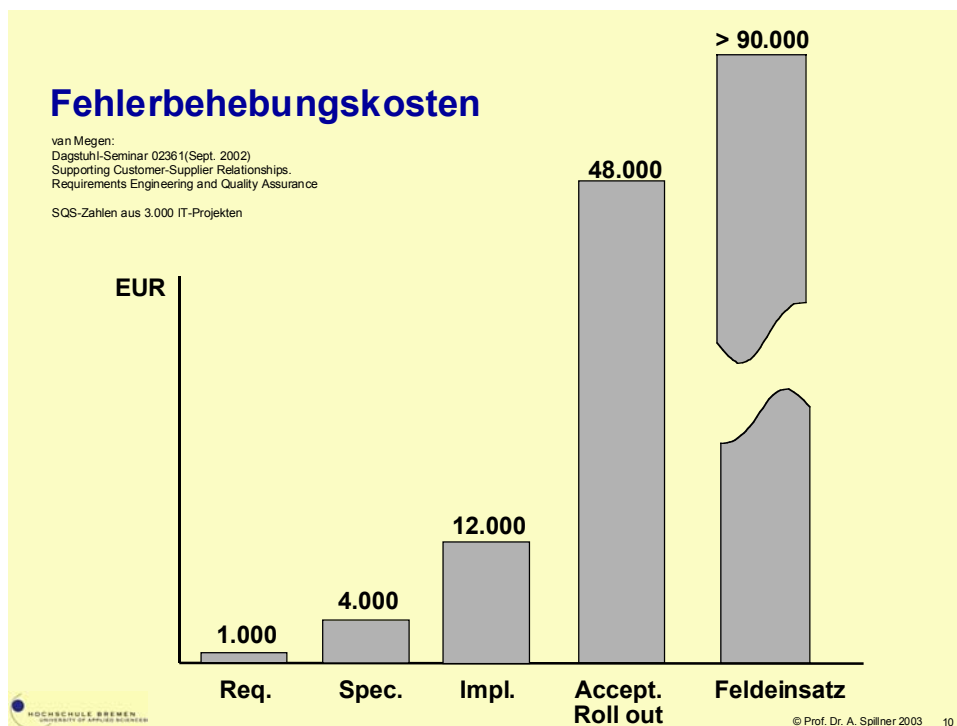
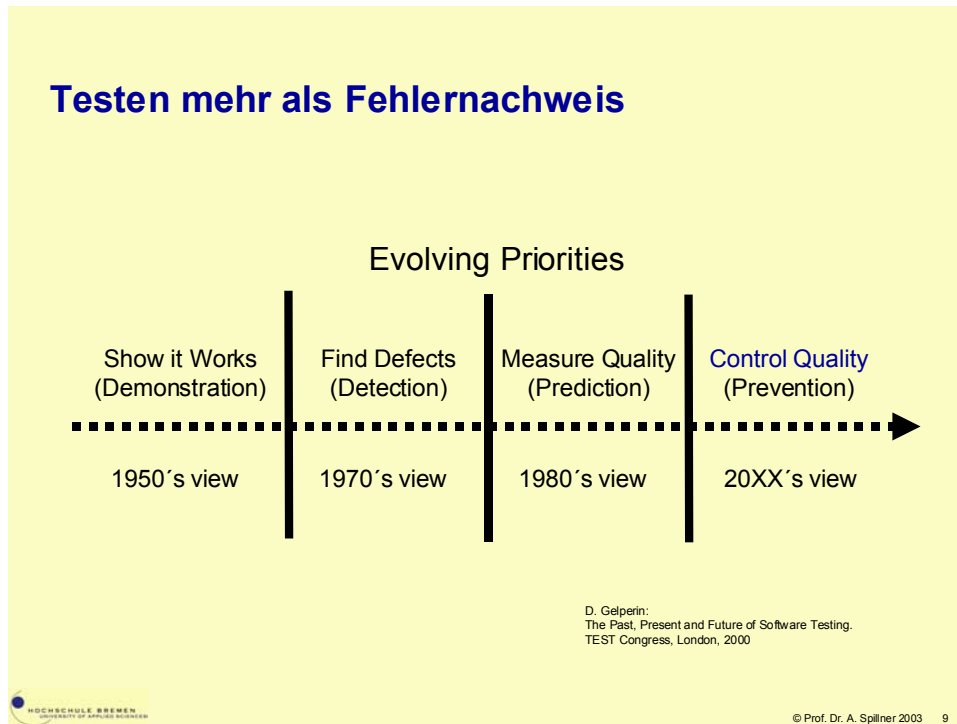
"Aktuelle Studien (z.B. von Mummert Consulting, veröffentlicht im März 2003) behaupten, dass allein in Deutschland jedes Jahr **190 Millionen Euro wegen schlechter Software verloren gehen**, z.B. wegen mangelnder Bedienbarkeit.

In einer ähnlichen Studie vom National Institute of Standards and Technology wird für die USA geschätzt, dass von den **60 Milliarden US-Dollar Kosten**, die durch schlechte Software entstehen, mindestens ein Drittel durch besseres Testen vermieden werden könnten."

F. Paulsh:  
Editorial, Schwerpunkt Testen  
Objektspektrum, Mai/Juni 2003, Nr. 3

## Software Qualität & Test Qualität







## Grundlegendes

Entwicklungsmodell beschreibt die Systematik einer geordneten Projektabwicklung:

- Phasen (Arbeitsabschnitte und einzusetzende Methoden)
- Aktivitäten (Rollen) und zu erzielende Ergebnisse
- Meilensteine (Phasenabschluss)
- Qualitätssicherungsmaßnahmen zur Feststellung des Erreichens von Meilensteinen

Entwicklungsmodell definiert die für alle Beteiligten gemeinsame und verbindliche Sicht der logischen und zeitlichen Struktur eines Projekts

## Verbesserungspotenzial

Technology	Payoff Range
Structured Techniques	No hard numbers
4GL	(-90%)-500%
CASE	9%-128%
Formal Methods	9%
Cleanroom	0%-70%
Process Models (CMM)	6%-570%
Object-Orientation	No hard numbers

Robert L. Glass „The Realities of Software Technology Payoffs“  
 in COMMUNICATIONS OF THE ACM February 1999/Vol. 42, No. 2

- Größte Verbesserung durch Prozessänderung
- Testprozess von Bedeutung

aus G. Keffer:  
 Extreme Programming Considered Harmful for Reliable  
 Software Development  
 Proceedings, SQM-Kongress 2003, 2.-4.4.2003, Köln

## Agile Prozesse

**Manifesto for Agile Software Development**

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

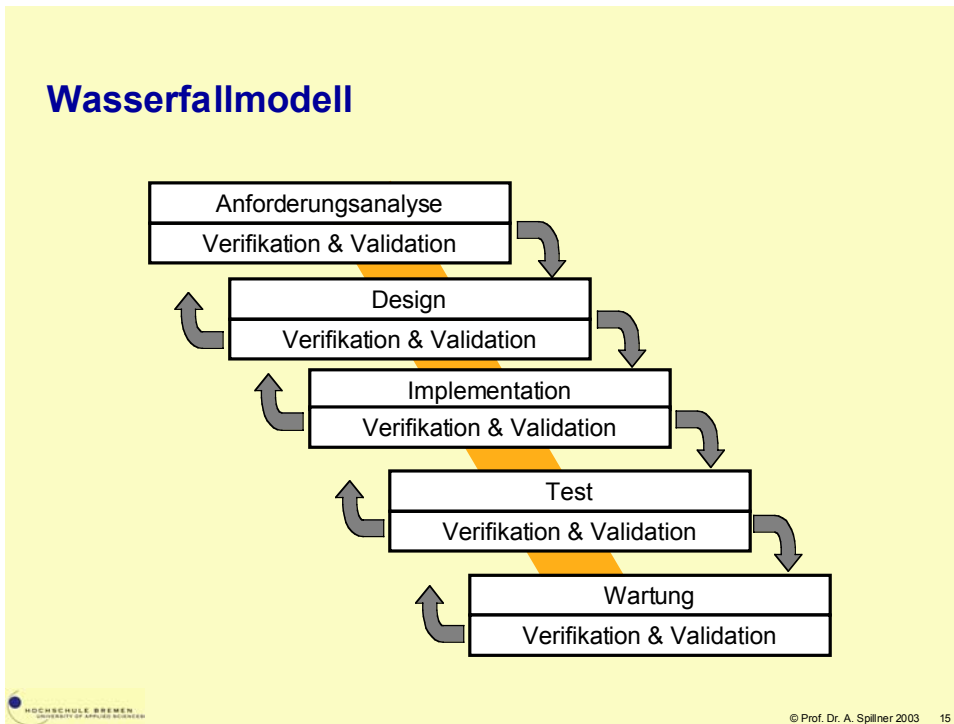
**Individuals and interactions** over processes and tools  
**Working software** over comprehensive documentation  
**Customer collaboration** over contract negotiation  
**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

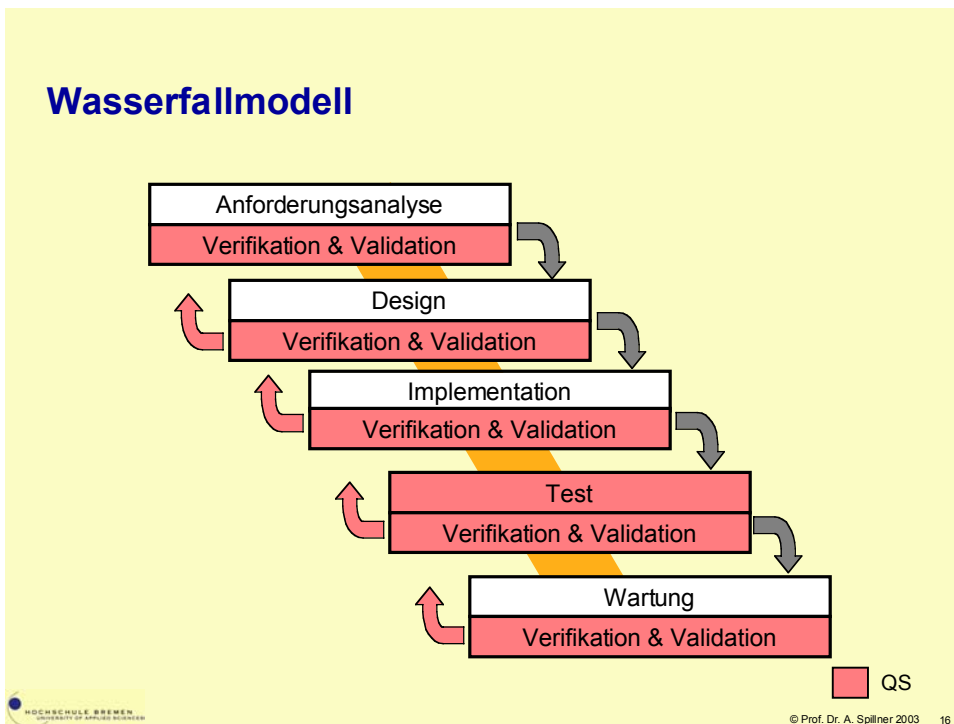
Kent Beck      James Grenning      Robert C. Martin  
 Mike Beedle    Jim Highsmith      Steve Mellor  
 Arie van Bennekum    Andrew Hunt      Ken Schwaber  
 Alistair Cockburn    Ron Jeffries      Jeff Sutherland  
 Ward Cunningham    Jon Kern          Dave Thomas  
 Martin Fowler      Brian Marick

<http://agilemanifesto.org/>

## Wasserfallmodell

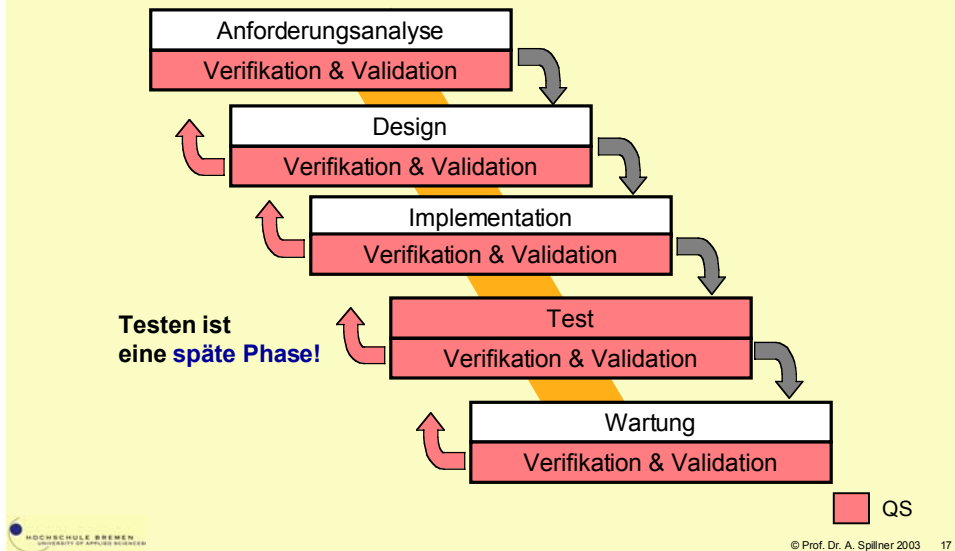


## Wasserfallmodell

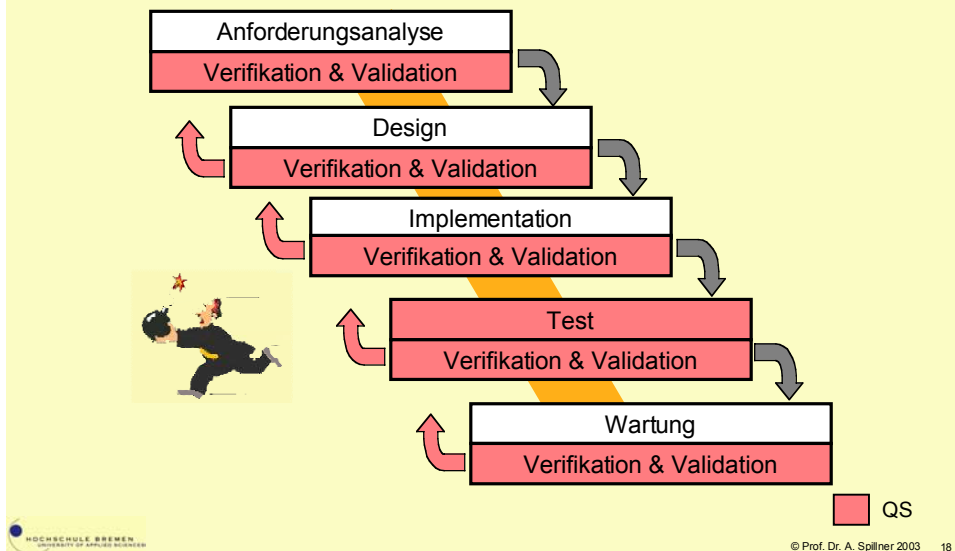


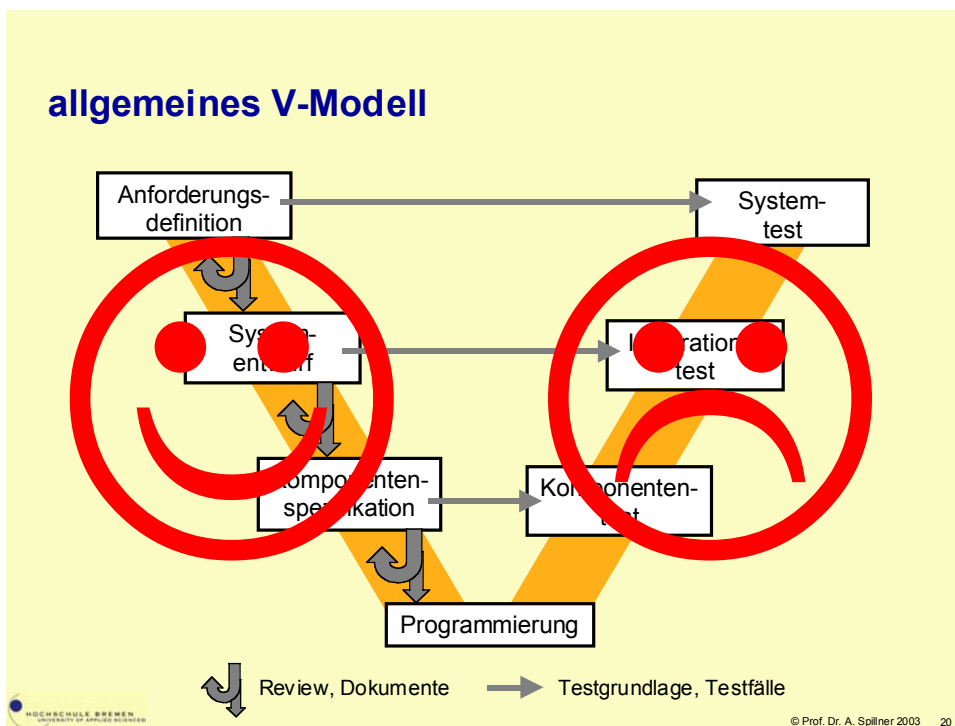
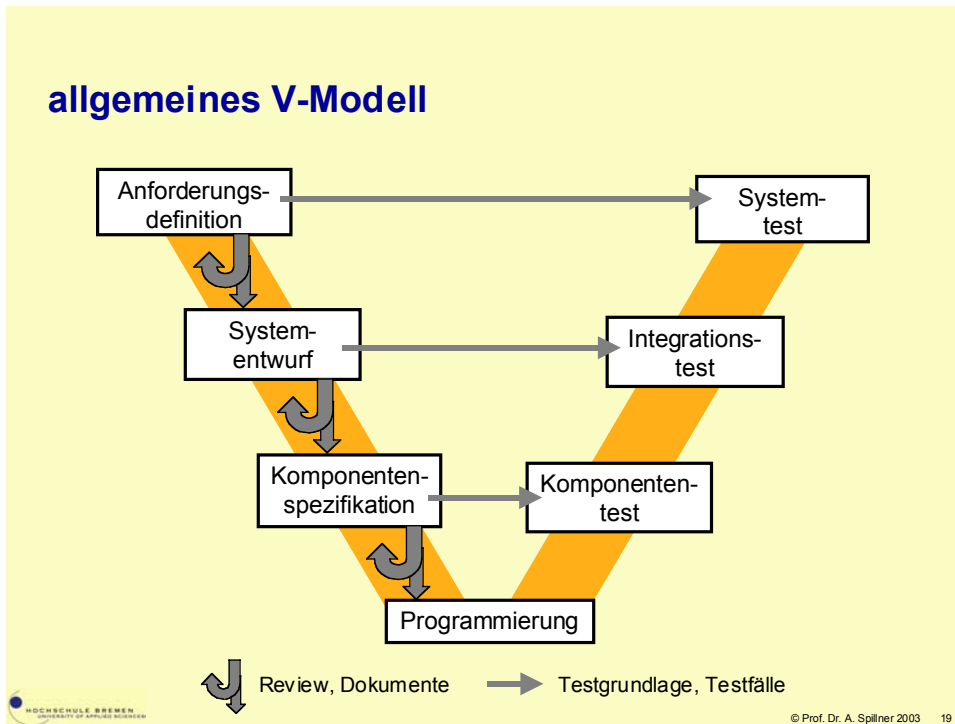


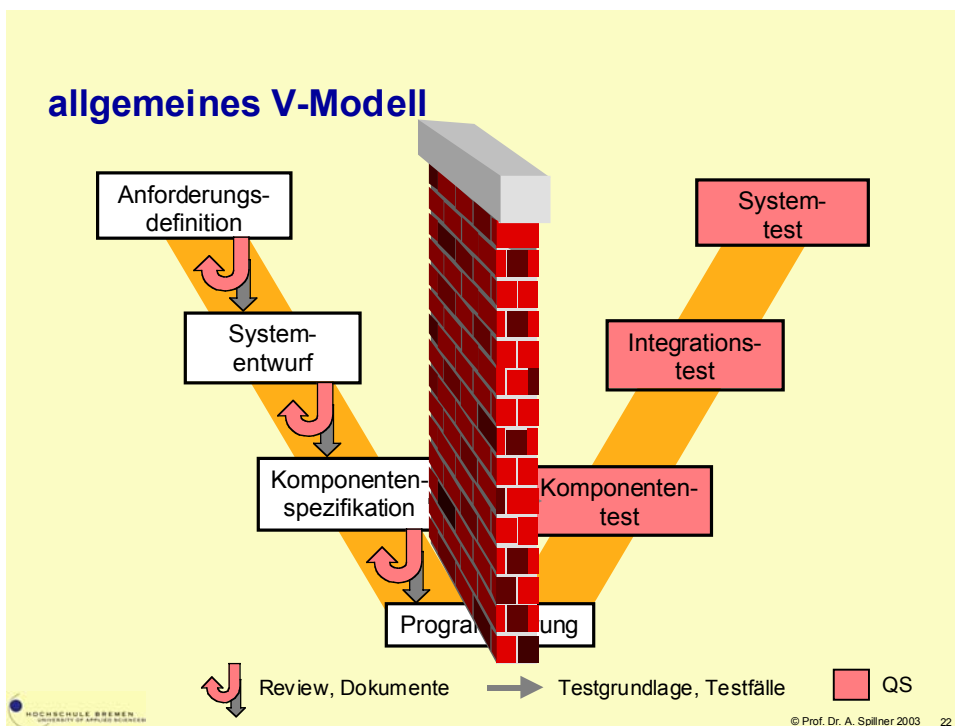
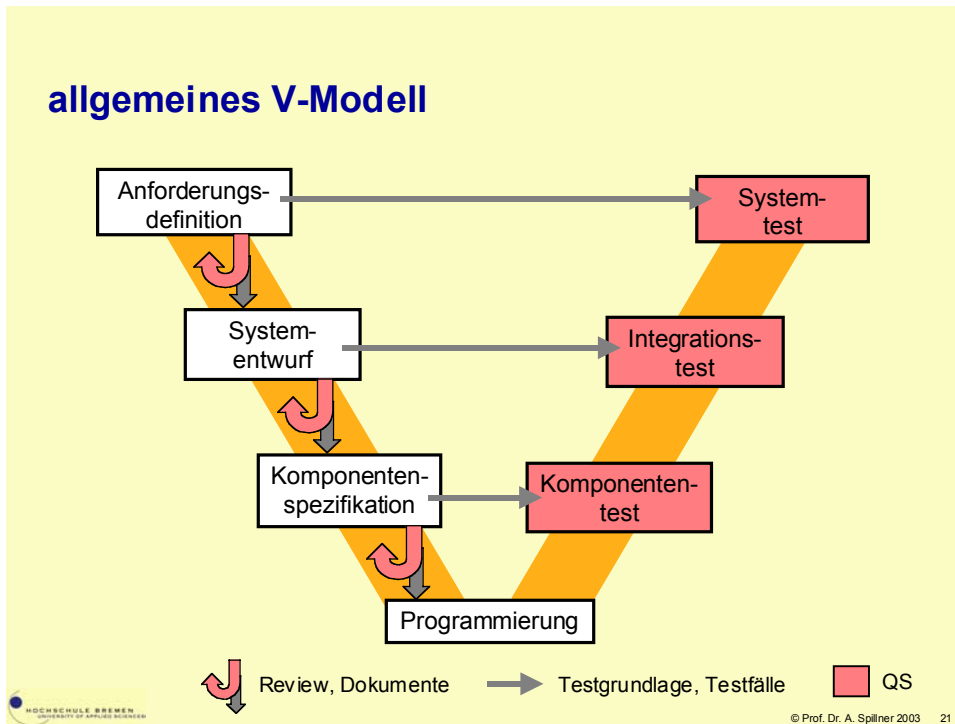
## Wasserfallmodell



## Wasserfallmodell





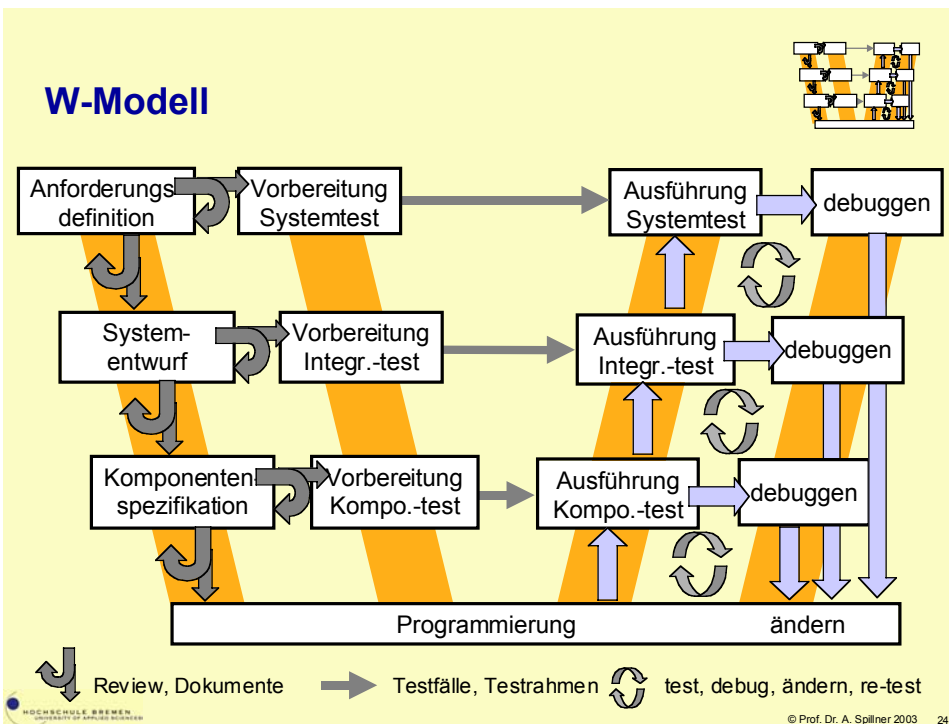


## Vom V-Modell zum W-Modell

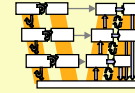
- Nachteile des V-Modells
  - Testaktivitäten beginnen zu spät
  - Verbindung von Testaktivität und Testbasis ist unklar
  - Trennung in Test-, Debug- und Änderungsaktivitäten werden nicht deutlich
- W-Modell soll diese Nachteile beheben und (einige) Vorteile der agilen Prozesse übernehmen

**Testprozess  
parallel zur  
Entwicklung!**

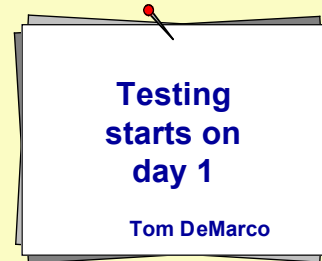
## W-Modell



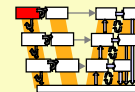
## Start der Testaktivität beim Projektstart



- Festlegung von **Teststrategie** und **Testkonzept**
  - Risiko-Analyse
  - Kritikalität feststellen
  - Testaufwand abschätzen
  - Testintensität festlegen
- **Testplan** erstellen
- **Organisation des Testteams**
- ggf. **Schulung des Testteams**
- **Berichtswesen** festlegen
- **Organisation der Hardware-Ressourcen**  
(Rechner, ...)
- **Organisation der Software-Ressourcen**  
(Softwareversion, Datenbank, Testwerkzeuge, ...)
- ...

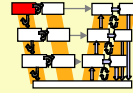


## W-Modell



Anforderungs-  
definition

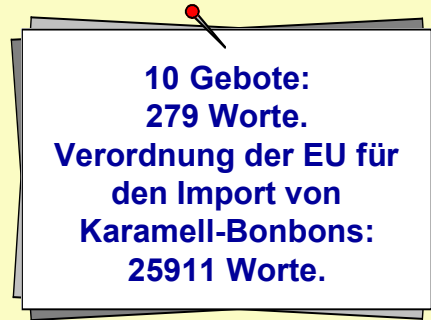
## Anforderungsdefinition



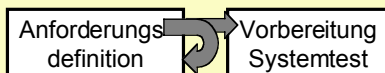
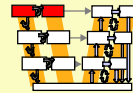
Anforderungen einem **Review** unterziehen

*Blick zurück:*

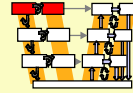
- Kundenwünsche erfüllt?
- Anforderungen vollständig?
- widerspruchsfrei?
- konsistent?
- kurz und präzise?
- Anforderungen realisierbar?



## W-Modell



## Anforderungsdefinition



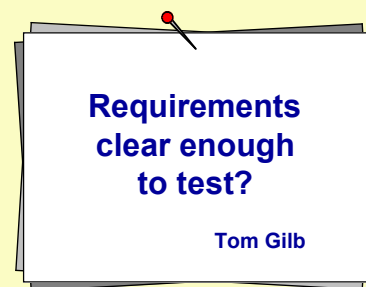
Beim **Review** unter **Mitwirkung der Tester**

*Blick voraus:*

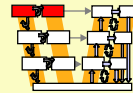
- Anforderungen präzise genug um Testfälle ableiten zu können?
- Alle Randbedingungen für den Systemeinsatz (und damit für den Systemtest) definiert?
- Testaufwand vertretbar?
- Nur testbare Anforderungen sind realisierbar!

Kent Beck:

„If you don't know how to test it  
you don't know how to implement it!“

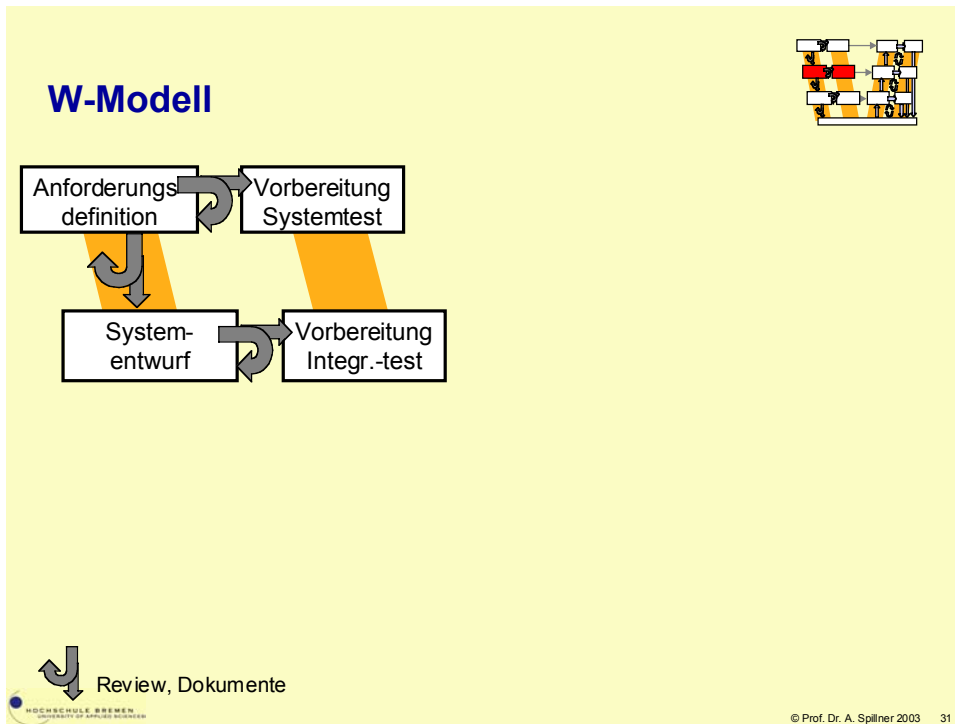


## Planung und Vorbereitung Systemtest



Testaktivitäten für den Systemtest:

- **Planung und Vorbereitung**
- Spezifikation der (funktionalen) **Systemtestfälle**
- **Priorisierung** der Tests nach Kritikalität
- Bereitstellung der **Infrastruktur**
- ...
  
- Benötigtes „**Systemtest-Wissen**“ hier vorhanden!
  
- **alle vorbereitenden Arbeiten für den Systemtest sind abgeschlossen!**



## Systementwurf

Systementwurf einem **Review** unter **Mitwirkung der Tester** unterziehen

*Blick zurück:*

- Systemaufteilung sinnvoll?
- ...

*Blick voraus:*

- Teilsysteme und Schnittstellen präzise spezifiziert?
- Integrationstestfälle ableitbar?
- Teilsysteme separat testbar?
- mit vertretbarem Aufwand?

**Klare und einfache Systemarchitektur ist einfach zu testen!**

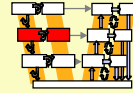
System-Architektur, die einen hohen Testaufwand bedingt, muss überarbeitet werden!

HOCHSCHULE BREMEN  
UNIVERSITY OF APPLIED SCIENCES

© Prof. Dr. A. Spillner 2003 32



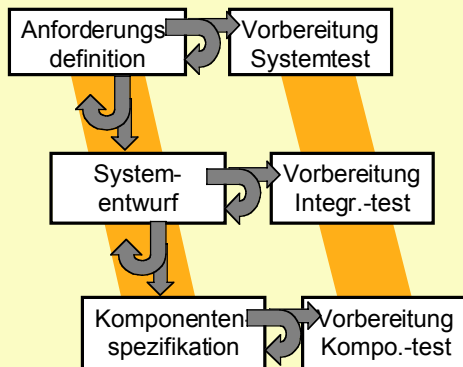
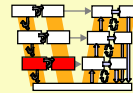
## Planung und Vorbereitung Integrationstest



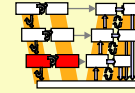
Testaktivitäten für den Integrationstest:

- Planung und Vorbereitung
- Auswahl der Testmethoden  
(funktional, fehlerorientiert, ...)
- Spezifikation der Testfälle
- Priorisierung der Tests nach Kritikalität
- Bereitstellung der Infrastruktur und der Testrahmen
- ...
  
- Benötigtes „Teilsystem-Wissen“ hier vorhanden!
  
- alle vorbereitenden Arbeiten für den Integrationstest sind abgeschlossen!

## W-Modell



## Komponentenspezifikation



Komponentenspezifikation einem **Review** unter **Mitwirkung der Tester** unterziehen

*Blick zurück:*

- Komponentenaufteilung sinnvoll?
- ...

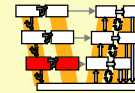
*Blick voraus:*

- Komponente und Schnittstellen präzise spezifiziert?
- Komponententestfälle ableitbar?
- Komponenten separat testbar?
- mit vertretbarem Aufwand?

**Schmale  
Komponenten-  
schnittstellen sind  
mit wenig Aufwand  
zu testen!**

Komponenten, die einen hohen Testaufwand bedingen, müssen überarbeitet werden!

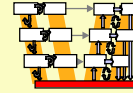
## Planung und Vorbereitung Komponententest



Testaktivitäten für den Komponententest:

- Planung und Vorbereitung
- Auswahl der Komponententestmethoden (kontrollfluss-, datenflussorientiert, ...)
- Spezifikation der Testfälle
- Priorisierung der Tests nach Kritikalität
- Bereitstellung der Infrastruktur und der Testrahmen
- ...
- Benötigtes „Komponenten-Wissen“ hier vorhanden!
- alle vorbereitenden Arbeiten für den Komponententest sind abgeschlossen!

## bis zur Programmierung



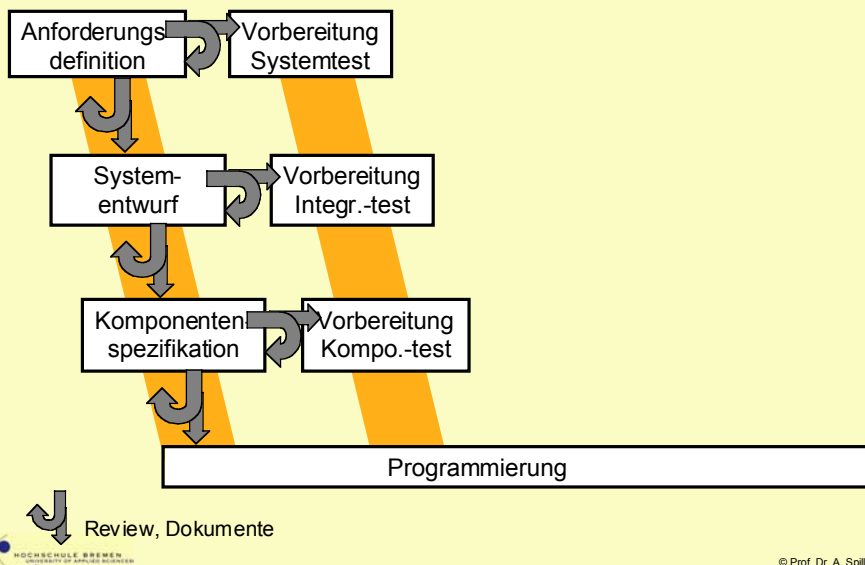
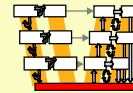
sind

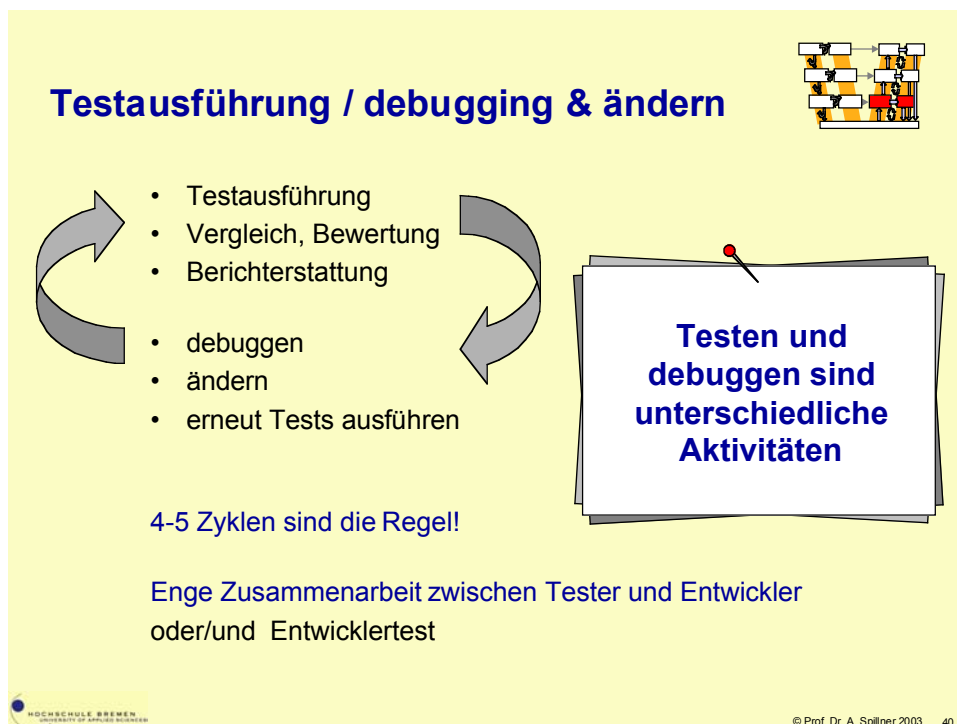
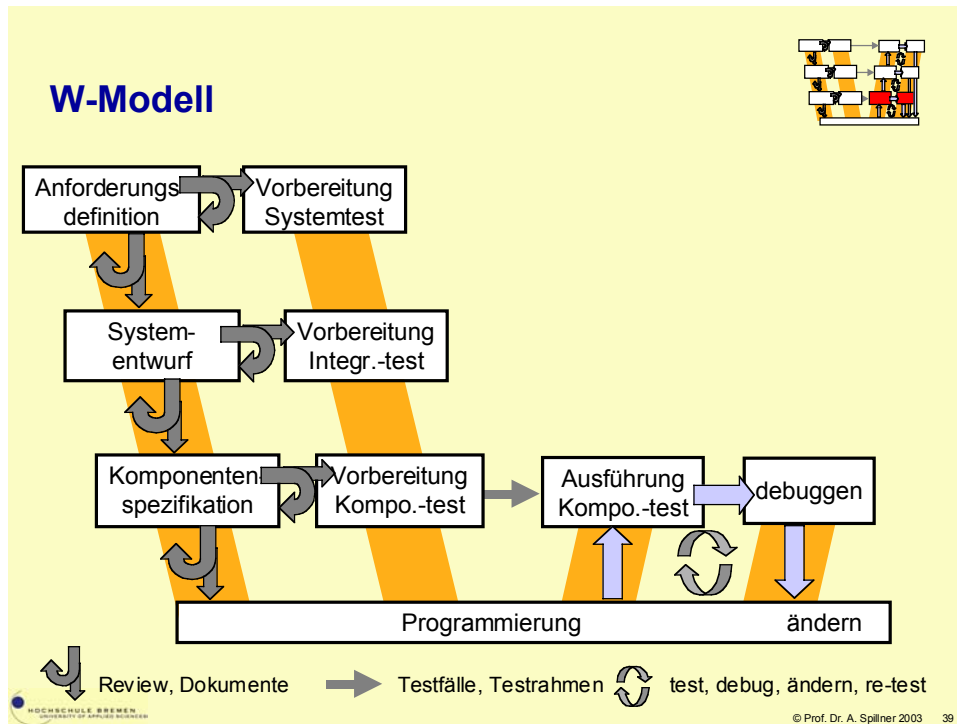
- 75% aller Testaktivitäten bereits abgeschlossen
- alle Testfälle spezifiziert
- alle benötigten Testrahmen vorhanden
- alle benötigten Testwerkzeuge zum Einsatz bereit
- ...
- die Tester nicht arbeitslos

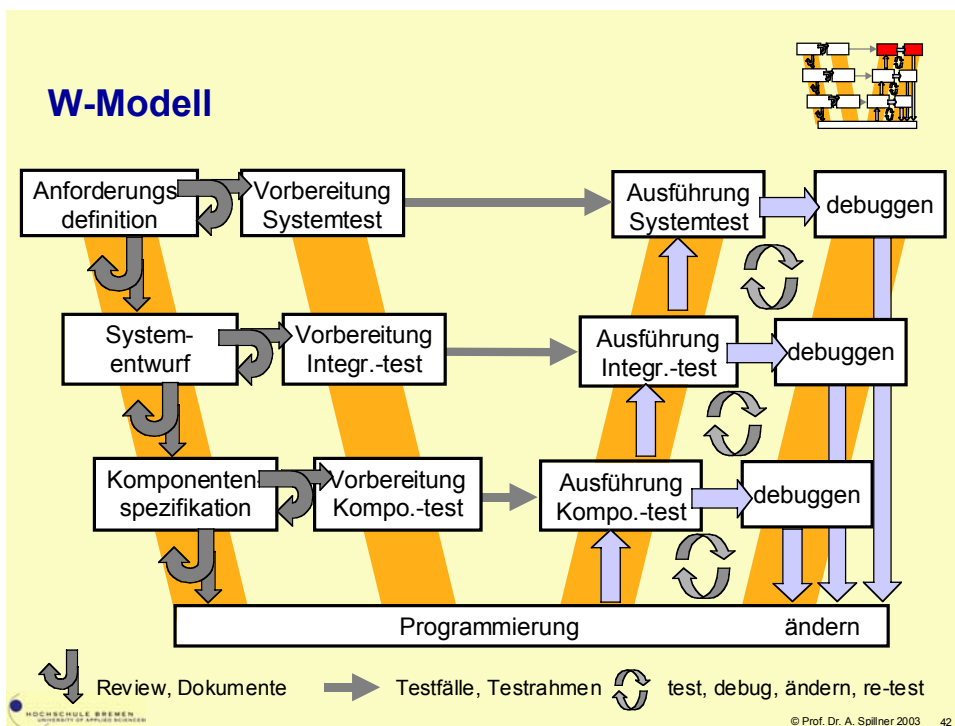
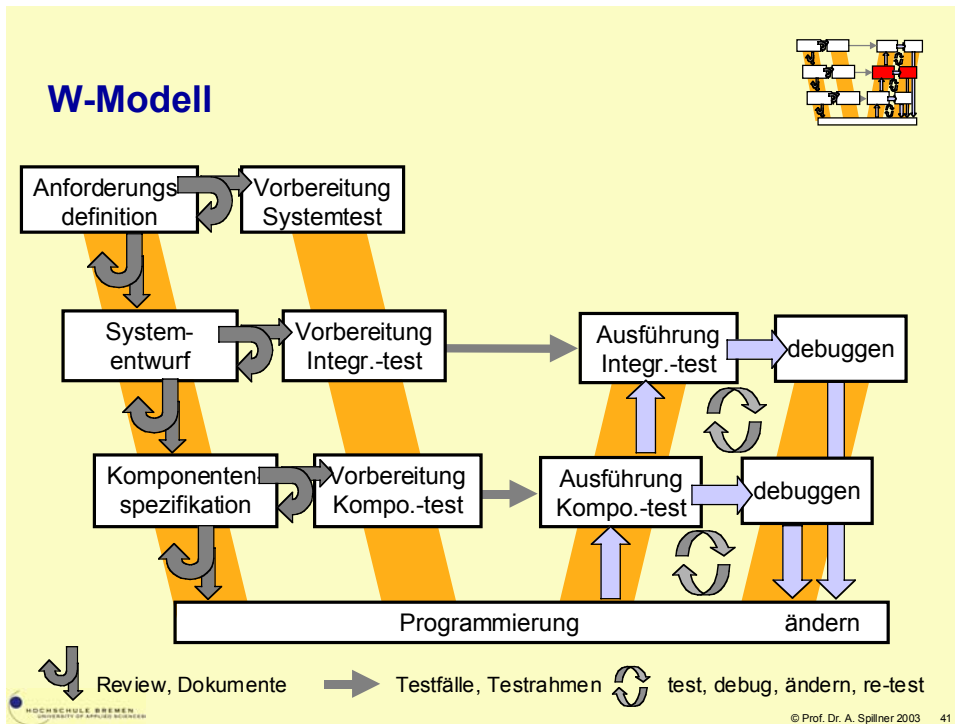
Durch die Beteiligung der Tester an den Reviews

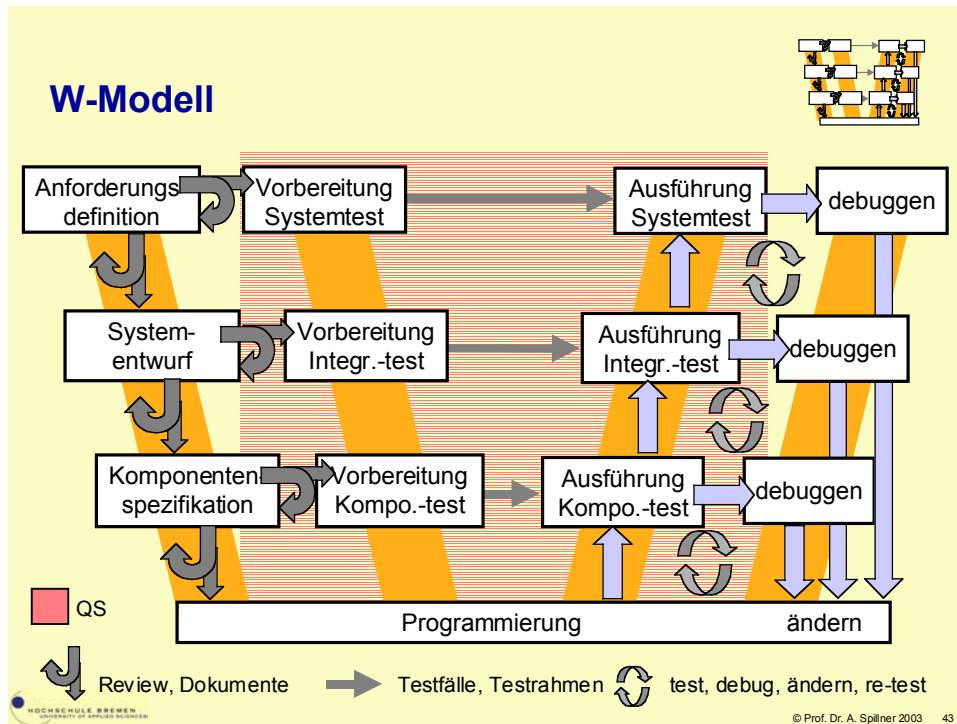
- wurden weitere Fehler durch eine andere (Test-)Sicht erkannt
- wurde eine aufwändig zu testende Systemarchitektur geändert
- wurden aufwändig zu testende Komponenten und deren Schnittstellen vereinfacht

## W-Modell









### Projektplan nach V- oder Wasserfall-Modell

Aktivität	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19
Anforderungen	█	█																		
Abnahme			█																	
Systementwurf				█	█	█														
Abnahme						█														
Komponentenspez.							█	█	█											
Abnahme									█											
Programmierung										█	█	█	█	█						
Komponententest												█	█	█	█	█				
Integrationstest															█	█	█	█		
Systemtest																			█	█

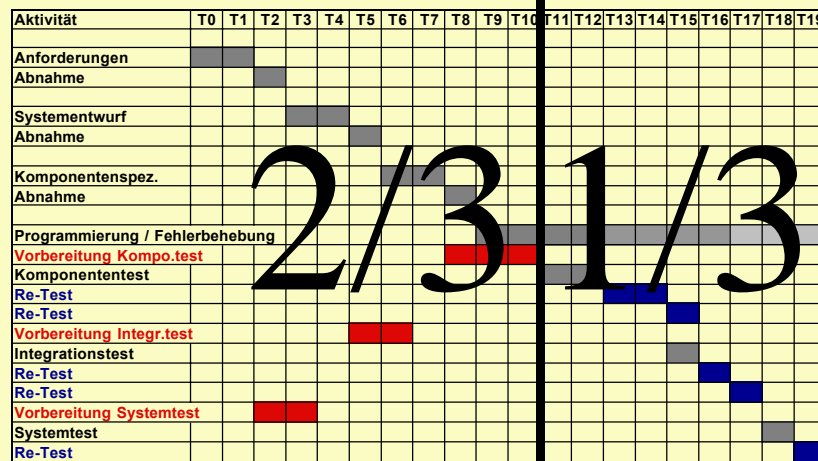
## Projektplan nach W-Modell ohne Test-Zyklen

Aktivität	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	
Anforderungen	█	█																			
Abnahme			█																		
Systementwurf				█	█																
Abnahme						█															
Komponentenspez.							█	█													
Abnahme									█												
Programmierung										█	█	█	█	█	█	█	█	█	█	█	█
Vorbereitung Kompo.test									█	█											
Komponententest												█	█	█	█						
Vorbereitung Integr.test							█	█													
Integrationstest																█					
Vorbereitung Systemtest				█	█																
Systemtest																	█				

## Projektplan nach W-Modell mit Test-Zyklen

Aktivität	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	
Anforderungen	█	█																			
Abnahme			█																		
Systementwurf				█	█																
Abnahme						█															
Komponentenspez.							█	█													
Abnahme									█												
Programmierung / Fehlerbehebung										█	█	█	█	█	█	█	█	█	█	█	█
Vorbereitung Kompo.test									█	█											
Komponententest												█	█	█	█						
Re-Test															█	█					
Vorbereitung Integr.test							█	█													
Integrationstest																	█				
Re-Test																		█			
Re-Test																			█		
Vorbereitung Systemtest				█	█																
Systemtest																					█
Re-Test																					█

## Projektplan nach W-Modell mit Test-Zyklen



## W-Modell – Vorteile

- Testprozess parallel zum Softwareentwicklungsprozess
- Bedeutung des Tests und der Zeitaufwand für Test und Debugging werden klar
- Testaktivitäten beginnen mit Projektstart
- Abfolge der einzelnen Testaktivitäten
- Teststufen und zugehörige Testbasis
- Zyklus von testen, debuggen, ändern, erneut testen
- Test und Debugging eng beisammen aber getrennte Aufgaben und es ist auch etwas anderes!
- Debugging ist wirklich eine späte Aktivität!
- Intensive Zusammenarbeit zwischen Entwickler und Tester von Beginn an!

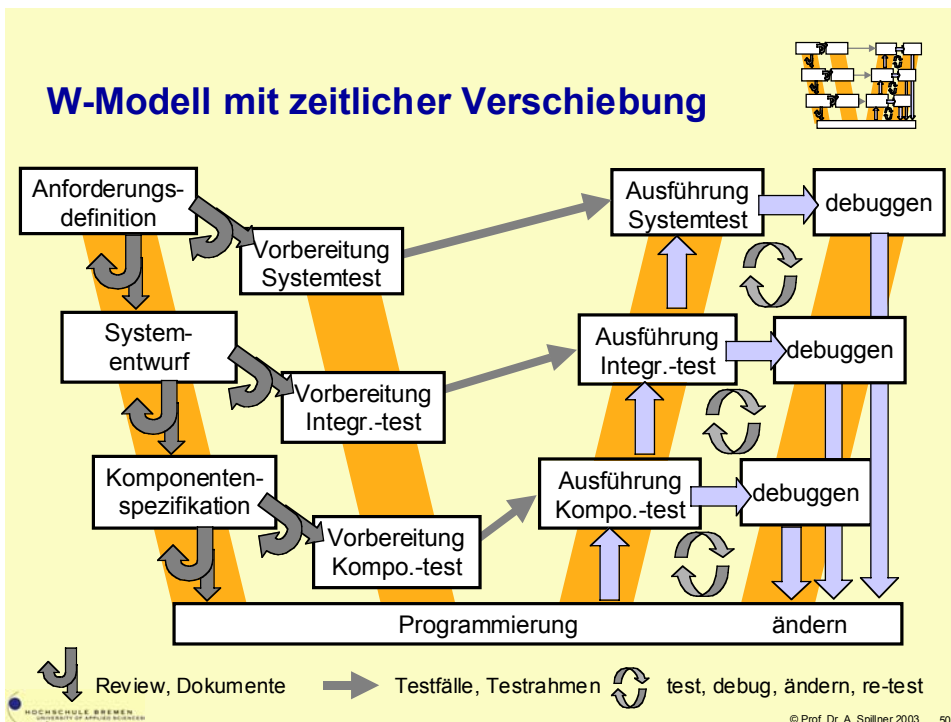
**Testen ist  
KEINE späte  
Phase der  
SW-Entwicklung!**



## W-Modell – Nachteile

- vereinfachte Darstellung - wie bei allen Modellen
- in der Praxis **weitere Beziehungen** notwendig und vorhanden
- **Iterationen** nicht dargestellt
- Integrationstest für jedes Systemteil  
Komponententest für jede Komponente durchzuführen
- Aufteilung der **Ressourcen** wird nicht deutlich
- Ressourcenverteilung kann in verschiedenen Projekten sehr unterschiedlich sein
- Änderungen der Anforderungen / Systemarchitektur verursachen Änderungen der entsprechenden Testfälle (Milderung des Problems: zeitliche Verschiebung)

## W-Modell mit zeitlicher Verschiebung



## Agile Prozesse

### Manifest:

- **Menschen und Zusammenarbeit** vor Prozessen und Werkzeugen
- **Funktionierende Software** vor umfassender Dokumentation
- **Zusammenarbeit mit dem Kunden** vor vertraglicher Verhandlung
- **Reaktion auf Veränderung** vor Einhaltung eines Plans

## Agile Prozesse

### Vorteile

- **Enge Zusammenarbeit** - Pair Programming  
Kent Beck: zentraler Punkt bei XP: *sit together* (4-5 h pro Tag!)
- **Testen frühzeitig** berücksichtigen, vor der Implementierung
- Kurze Reaktionszeiten auf veränderte Anforderungen
- Kurze Entwicklungszyklen (Iterationen)

### Nachteile

- **Refactoring**
- **Keine systematische Testfallerstellung**  
(ein bisschen testen, ein bisschen implementieren)
- Zusammenarbeit zwischen **Personen mit gleicher Sichtweise**

## Vergleich XP & W-Modell

### Gemeinsamkeiten von XP und W-Modell

- Testaktivitäten vor der Programmierung
- Testrahmen als ersten Schritt realisieren
- Testüberlegungen dienen der genaueren Spezifikation

### Vorteile des W-Modells

- Zusammenarbeit von Entwicklern und Testern (unterschiedliches Wissen)
- Entwurf weiterhin wichtig
- „Mehr Struktur“ vorhanden, für umfangreiche Projekte geeignet
- Kleiner Schritt vom V-Modell zum W-Modell

### Nachteile des W-Modells

- Iterationen wenig unterstützt
- Enge Kundenintegration nicht zwingend

## Testwissen erforderlich - Certified Tester

- Internationale Aus- und Weiterbildungsinitiative
- Ausgangspunkt  
BCS - British Computer Society  
ISEB - Information Systems Examinations Board
- 3-stufiges Qualifizierungsprogramm
- Ausbildung durch akkreditierte Weiterbildungsanbieter
- Prüfung durch unabhängige Instanz  
(Kursteilnahme nicht erforderlich)



## Information Systems Examinations Board (ISEB)



England bzw. englische Kurse

- seit 1998 ca. **8.000 Prüfungen** abgenommen  
derzeit ca. 500 pro Monat
- **Erfolgsquote zwischen 80 und 85%**
- Dorothy Graham  
*“Das Programm hat den Status des Software-Testens und die Anerkennung, die man Software-Testern entgegenbringt, deutlich gehoben. Auf dem britischen Arbeitsmarkt wird man inzwischen ohne das Zertifikat in der Regel gar nicht erst zu einem Vorstellungsgespräch als Software-Tester eingeladen.”*

## Certified Tester in Deutschland



- seit 2001  
vom **ASQF** (Arbeitskreis Software-Qualität Franken e.V.) und  
der **GI-Fachgruppe TAV** (Test, Analyse und Verifikation von Software)  
aufgegriffen
- ca. **300 zertifizierte Tester** (*Foundation Certificate*).
- aktuell sind **sieben Trainingsanbieter** akkreditiert bzw. haben einen  
Antrag auf Akkreditierung gestellt

## Lehrplan Foundation Level



### Kursdauer

- 3-4 Tage

### Kursinhalt

- Grundlagen des Software-Testens
- Testen während des Lebenszyklus
- Statischer Test
- Dynamischer Test
- Test Management
- Testunterstützungswerkzeuge

## Zusammenfassung

- Vorteile der agilen Prozesse nutzen
- W-Modell
  - kein neues Modell
  - zur Diskussionsanregung
  - veranschaulicht die Bedeutung des Tests
  - Test als paralleler Prozess
  - kleiner Schritt vom V- zum W-Modell
- Einsatz in der Praxis
  - Ausbildung bei einem großen Software-Haus
  - empfohlenes Vorgehensmodell eines großen SQ-Consulting-Unternehmen
  - ...
- Qualifizierte Ausbildung im Testbereich - Certified Tester

## Literatur zum W-Modell & Certified Tester

### From V-Model to W-Model - Establishing the Whole Test Process

Conquest 2000, Workshop on *Testing Non-Functional Software Requirements*,  
Invited Speaker, 13.-15.9.2000, Nürnberg, Proceedings, S. 221-231

### The W-Model - Strengthening the Bond Between Development and Test

STAReast - Conference 2002, 15.-17.5.2002, Orlando, Florida, USA, Proceedings  
oder <http://www.stickyminds.com/> -> Search „Spillner“

### Management des Testprozesses von Anfang an - Das W-Modell

Software Management Tagung 2002, eingeladener Hauptvortrag, 7.-8.11.2002,  
Hamburg, Tagungsband: Th. Spitta, J. Borchers, H. Sneed (Eds.)  
Software Management 2002 - Progress through Constancy, ISBN 3-88579-352-0

Andreas Spillner, Tilo Linz:

### Certified Tester - eine Erfolgsstory

Softwaretechnik-Trends, 23:1, Februar 2003, S. 50-52

Informationen zum Certified Tester

<http://www.certified-tester.de>

## Bücher zum Test und Testprozess

Andreas Spillner, Tilo Linz

### Basiswissen Softwaretest

Aus- und Weiterbildung zum Certified Tester

- Foundation Level

- nach ASQF- und ISEB-Standard

2003, ISBN 3-89864-178-3

dpunkt

Martin Pol, Tim Koomen, Andreas Spillner

### Management und Optimierung des Testprozesses

2002, 2. Auflage, ISBN 3-89864-156-2

dpunkt



## Bücher zur Qualitätssicherung

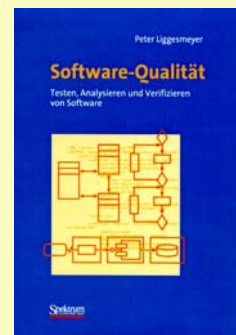
Helmut Balzert

Lehrbuch der Software-Technik,  
Band 2: Software-Management,  
Software-Qualitätssicherung und  
Unternehmensmodellierung  
1997, ISBN 3-8274-0065-1  
Spektrum



Peter Liggesmeyer

Software-Qualität  
2002, ISBN 3-8274-1118-1  
Spektrum



## Ihre Fragen, bitte

Prof. Dr. A. Spillner  
Hochschule Bremen  
Zentrum für Informatik und  
Medientechnologien  
Flughafenallee 10  
D - 28199 Bremen

++49 421 5905-5467 -5484 (FAX)  
spillner@informatik.hs-bremen.de  
<http://www.fbe.hs-bremen.de/spillner>