
Qualitätsmanagement im eCommerce

Mythen und Tatsachen

 **Technische
Universität
Dresden**

Professur für Wirtschaftsinformatik,
insbesondere Informationssysteme im
Dienstleistungsbereich (ProWi)



Prof. Dr. Georg Herzworm
<http://www.Herzwurm.de>
Email: Georg@Herzwurm.de

Was ist Qualitätsmanagement?



Das Qualitätsmanagement (QM) ist eine Wissenschaft verschwommener Annahmen und stützt sich auf anfechtbare Werte, die als Ergebnis erfolgloser Experimente mit Instrumenten problematischer Genauigkeit von Personen zweifelhafter Zuverlässigkeit und fragwürdiger Geisteshaltung ermittelt wurden.

(ein unbekannter Kritiker)

● Qualitätsmanagement im allgemeinen

- ◆ „Qualität ist die Abwesenheit von Fehlern“
- ◆ „Ziel jeder industriellen Software-Entwicklung muss es ... sein, ein möglichst fehlerfreies Software-Produkt zu erstellen.“
- ◆ Die „Zehnerregel der Fehlerkosten“

● Qualitätsmanagement im eCommerce

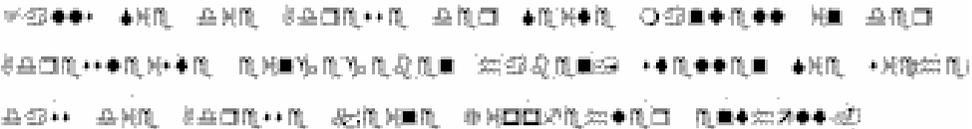
- ◆ Geschwindigkeit (im Sinne schneller Verfügbarkeit/kurzer Time-to-Market) ist der entscheidende Erfolgsfaktor und somit vorrangiges QM-Ziel
- ◆ eCommerce Projekte erfordern grundsätzlich andere QM-Maßnahmen als „normale“ Projekte
- ◆ Anforderungen sind vor Beginn der Implementierung bekannt oder können vorab ermittelt werden

Kein Qualitätsprodukt im Sinne von Fehlerfreiheit und trotzdem erfolgreich!?

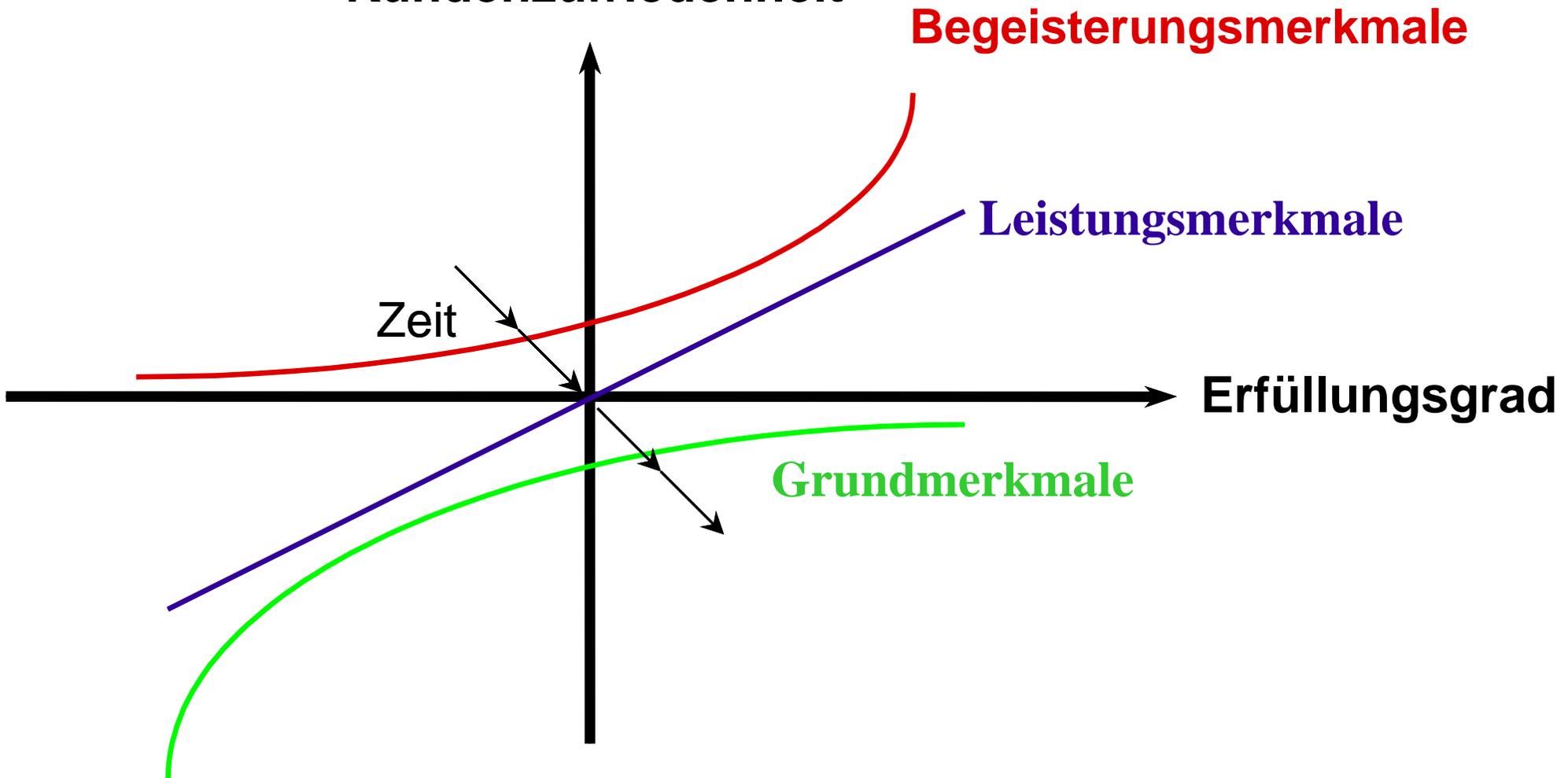
 Die Seite wurde nicht gefunden.

Die gewünschte Seite wurde möglicherweise ent- oder umbenannt, oder sie ist vorübergehend nicht erreichbar.

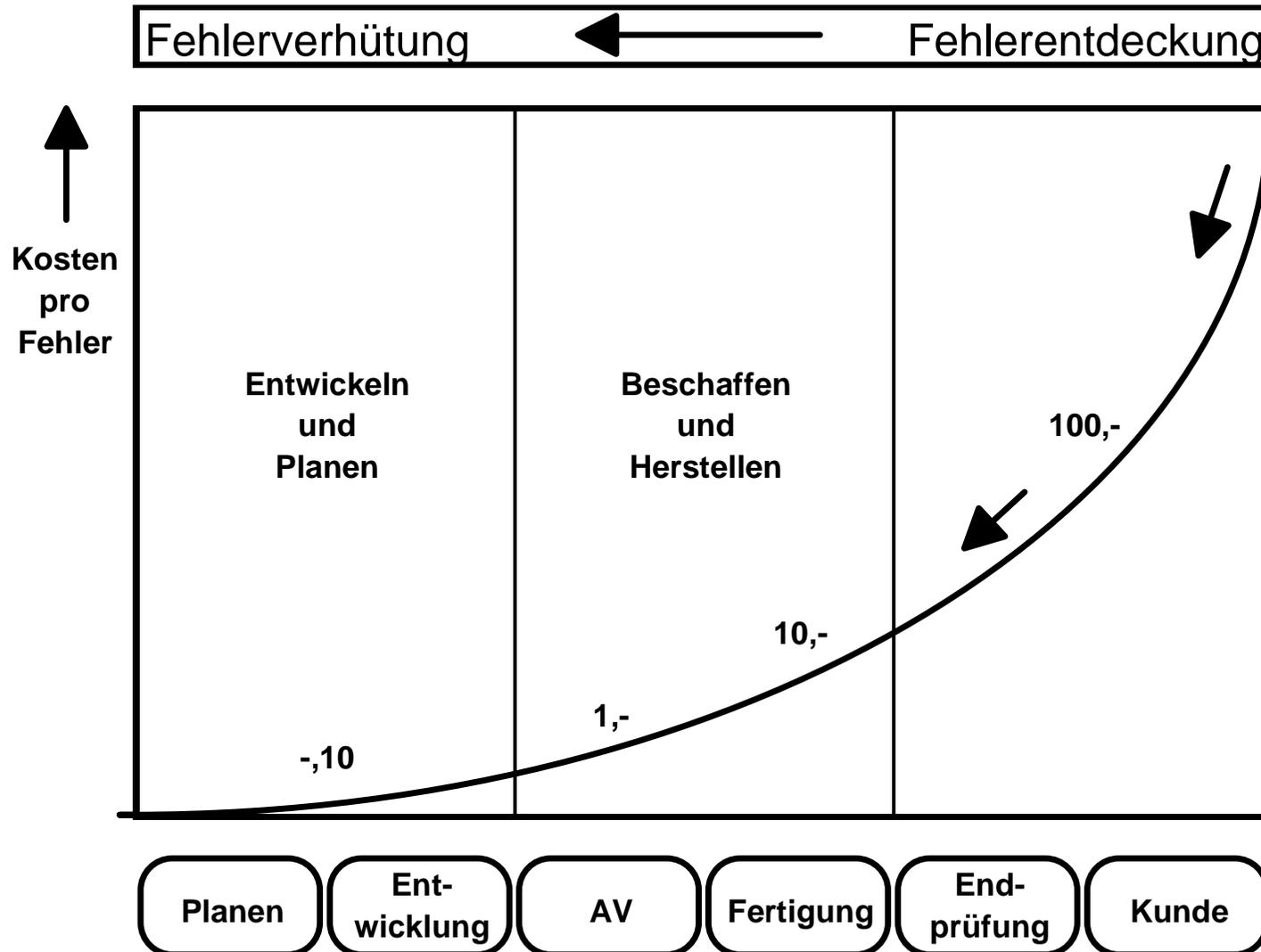
Versuchen Sie folgendes:

- 
- 
- 
- 

Kundenzufriedenheit



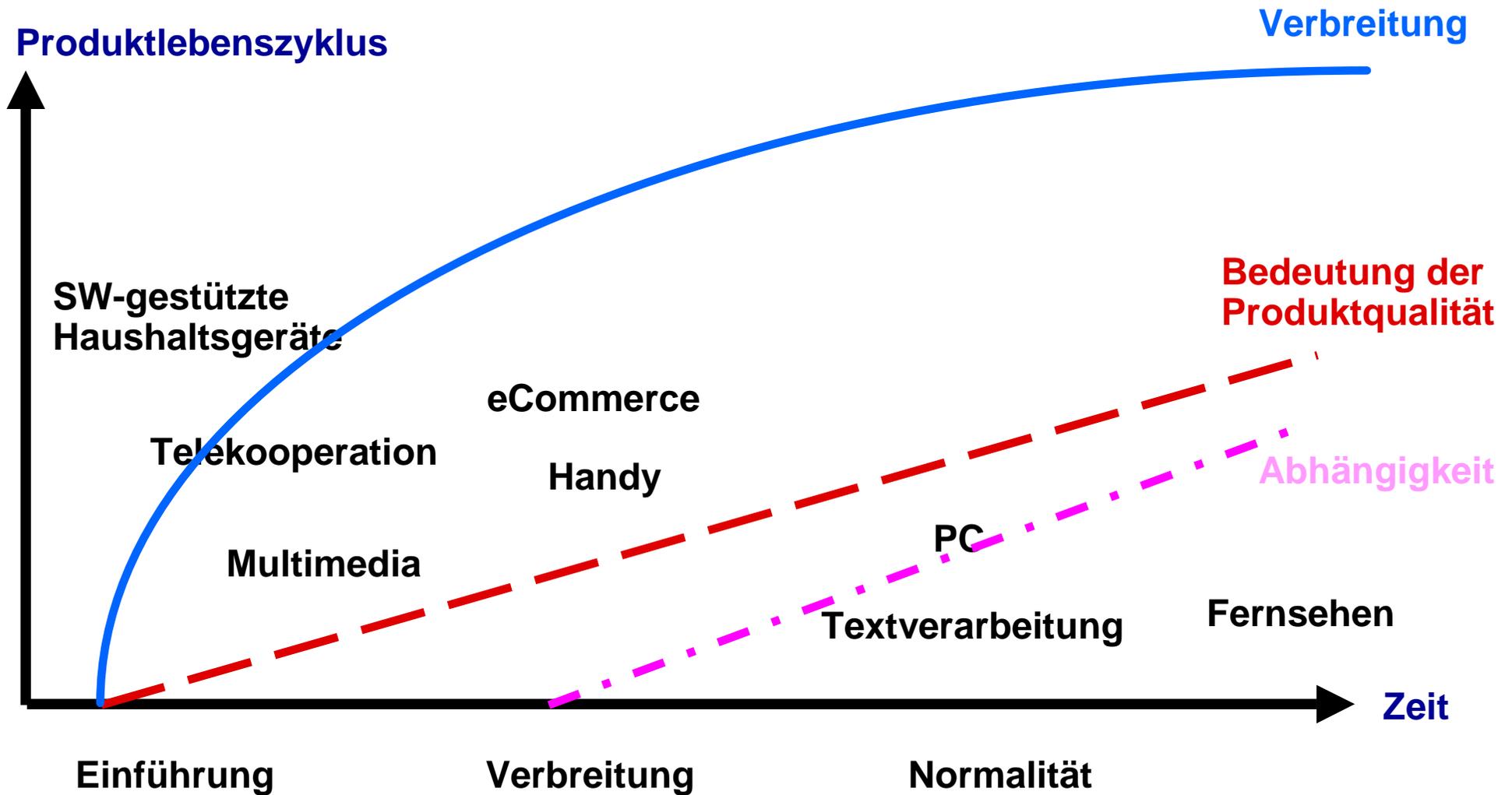
Irrtum „Zehnerregel der Fehlerkosten“



- **Wer zu früh kommt, den bestraft (oft) der Markt**
 - ◆ Nicht immer ist der Erste auch der Erfolgreichste!
 - ◆ Beispiel: PDA Newton von Apple
- **Erfahrungen beim Internet-Shopping (Zona Research)**
 - ◆ 33% aller potentiellen Kunden scheitert beim Versuch, Waren online zu bestellen
 - ◆ 62% der Kunden gaben an, mindestens einmal bei der Suche nach einem Produkt aufgeben zu haben



Irrtum „Frühe Verfügbarkeit auf Kosten der Qualität“ (2 von 2)



Irrtum „eCommerce QM grundsätzlich anders als normales QM“ (1 von 2)

SIZ INFORMATIKZENTRUM
DER SPARKASSEN -
ORGANISATION GMBH

Inhalte: [Architekturspezifikationen](#) | [Vorgehensmodell](#) | [UML-Modellierung und Webdesign](#) | [Basistechnologien](#) | [Java-Technologien](#) | [Komponenten](#) | [Multimedia-Technologien](#) | [Sicherheit](#) | [Zielsetzung](#) | [Anforderungsprozess](#) | [Literatur](#)

SIZ AE-Modell Internet/Intranet-Entwicklung

Das **Anwendungsentwicklungsmodell (AE-Modell)** ist der Standard für die Durchführung von Software-Projekten in der deutschen Sparkassenorganisation. Das **AE-Modell für Internet / Intranet-Anwendungen** deckt die Kernelemente der Anwendungsentwicklung auf der Basis von Web-Technologien ab.

AE-Modell Internet/Intranet-Entwicklung

- 10 Muster-Architekturen für Web-Anwendungen, incl. Sicherheitscheck
- Anwendungsentwicklung für Network Computing
- UML-Modellierung und Webdesign
- Sicherheit: Schwachstellen und Maßnahmen
- Einsatz von JavaBeans, Active X, EJB
- Java 2-, JavaScript-, Perl-, VBScript-Entwicklung
- Werkzeugauswahl

AE-Modell
Strukturierte Entwicklung

AE-Modell
Objektorientierte Entwicklung

AE-Modell für Internet/Intranet-Entwicklung

- Home
- Guided Tour
- Data Sheet

Erfahrungsberichte

Arbeitskreis AE-Modell

Aktuelle Projekte

Copyright

SIZ Homepage

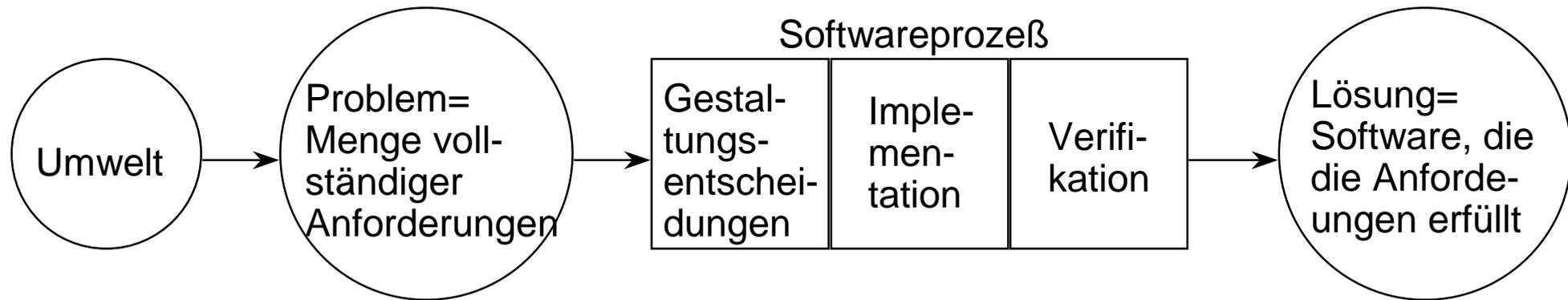
E-Mail

Irrtum „eCommerce QM grundsätzlich anders als normales QM“ (2 von 2)

Rahmenbedingungen	Wasserfall	Evolutionär	Inkrementell	Adaptiv
◆ Arbeitet gut mit schlecht verstandenen Anforderungen	☹	☺	☺	☺
◆ Arbeitet gut mit schlecht verstandenen Architekturen	☹	☺	☺	☹
◆ Produziert Systeme hoher Zuverlässigkeit	☺	☹	☺	☹
◆ Produziert Systeme mit langfristigem Horizont	☺	☺	☺	☺
◆ Berücksichtigt Risiken	☹	☺	☺	☺
◆ Erlaubt Einhaltung fixer Termine	☺	☺	☺	☺
◆ Erlaubt flexible Reaktionen	☹	☺	☺	☺
◆ Liefert Kunden/Auftraggeber sichtbare Fortschritte	☹	☺	☺	☺
◆ Liefert Management sichtbare Fortschritte	☺	☺	☺	☺
◆ Benötigt keine Begeisterung von Management/Entwicklern	☺	☺	☺	☺

Irrtum: „Anforderungen frühzeitig bestimmbar“ (1 von 4)

- **Annahme: Anforderungen frühzeitig bestimmbar**
- **Konsequenz: Auffassung der Softwareentwicklung als Problemlösungsprozess**
- **Idealvorstellung: Der Softwareprozeß ist so zu gestalten, dass er (im Prinzip) für jedes Problem, d. h. für jede Menge von Anforderungen, mit vorhersagbarem Aufwand und Termin eine Lösung liefert**



- **Interpretation der Probleme der Softwareentwicklung: Instabilität des Prozesses führt zu Qualitätsproblemen und schwankenden Vorhersagen von Aufwand, Termin**

● **Tatsächliche Rahmenbedingungen im eCommere am Beispiel UMTS**

◆ **Zeitwettbewerb**

- **MobilCom-Chef Gerhard Schmid:**
- **„MobilCom wird Mitte 2002 mit seinem Netz als Erster der deutschen UMTS-Lizenzinhaber auf dem Markt sein.“**

◆ **Unklare und instabile Anforderungen**

- **Schaffen eines „attraktiven“ Produkts statt Erfüllung spezifizierter Anforderungen**
- **Konzentration auf essentielle Anforderungen und einzelne Begeisterungsmerkmale**

◆ **Erhebliche Komplexität**

- **Innovationsfähigkeit**
- **Schnelleres Verständnis von Anforderungen neuer Applikationen**
- **Schnelleres Verständnis von Potentialen neuer Technologien**

◆ **Hohe Anforderungen an die Fehlerfreiheit**

- **Schnelligkeit ohne Qualität ist wertlos**

● **Tatsache:**

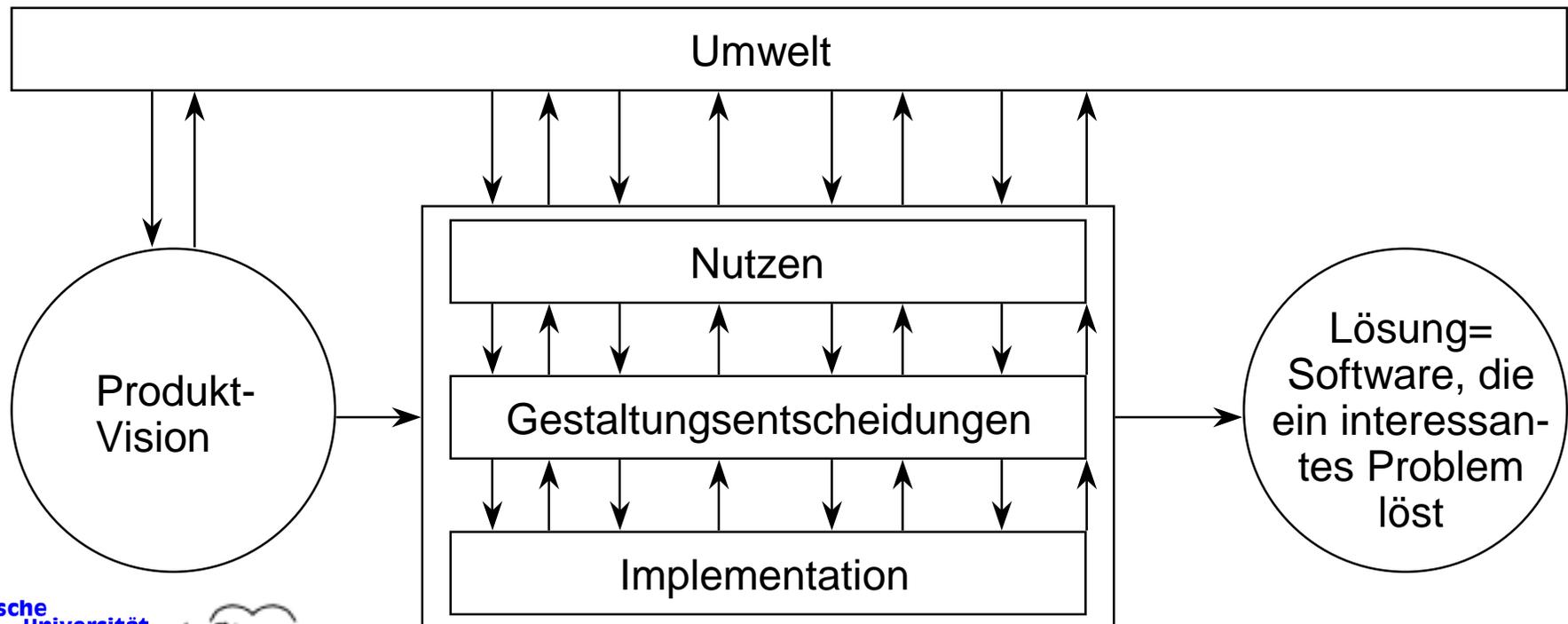
- ◆ **Anforderungen verändern sich während der Entwicklungszeit**
 - **Grund: turbulentes Umfeld**
- ◆ **Anforderungen ändern sich als Reaktion auf die Lösung**
 - **Grund: Wettbewerb und unbestimmte Kundenanforderungen**
- ◆ **Lösungsraum ist nicht vollständig bekannt, sondern vergrößert sich als Reaktion auf die erreichte Lösung**
 - **Grund: besser verstandene Technologie**

● **Konsequenz:**

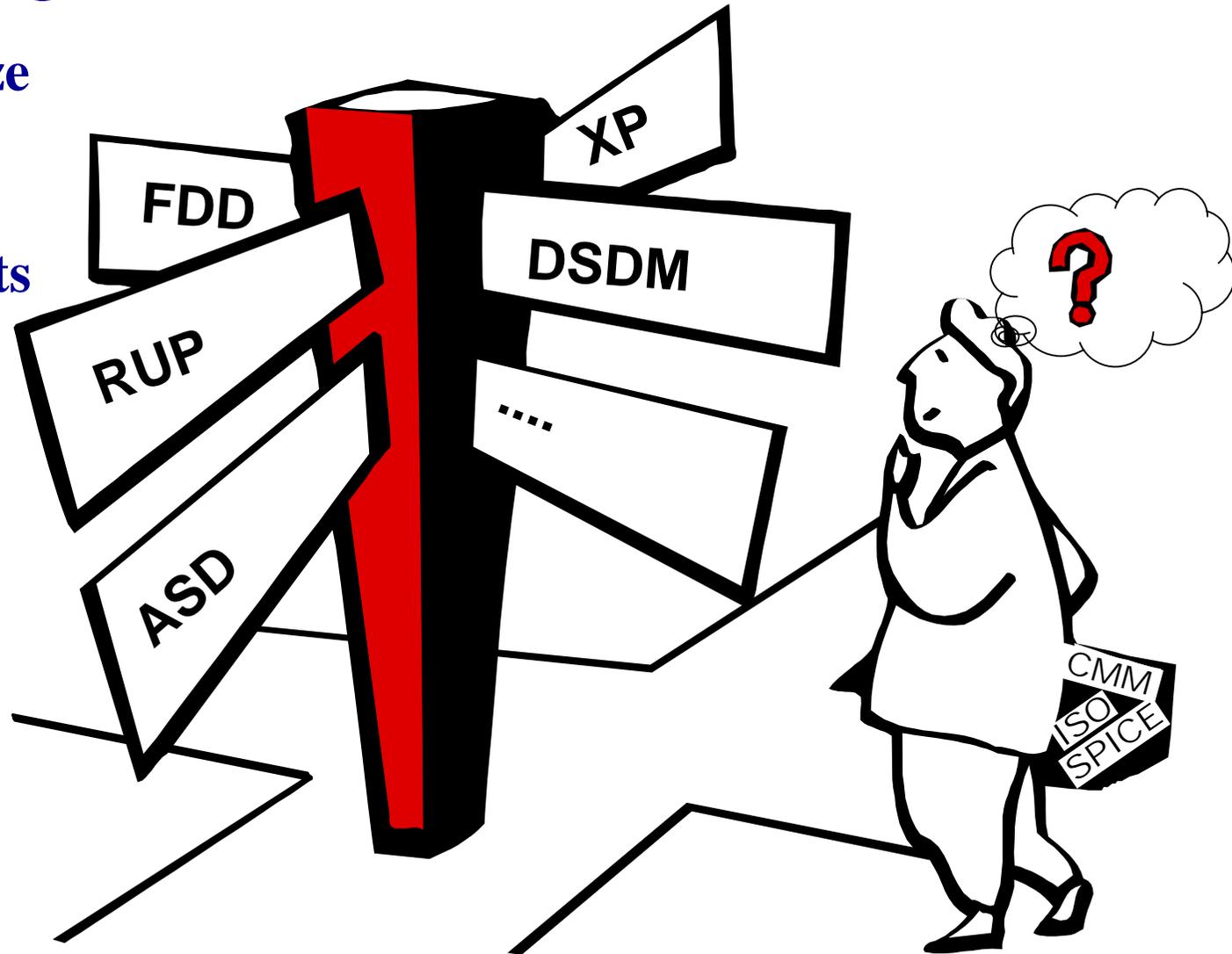
- ◆ **Softwareentwicklung ist ein adaptiver Lösungsprozess, bei dem das Problem und die Lösung gleichzeitig entwickelt werden**
- ◆ **konventionelle Auffassung führt zu Lösungen, die veraltete Anforderungen erfüllen bzw. dazu zwingen frühzeitig unbestimmte Anforderungen unter hohem Risiko zu bestimmen**

- **Gefordert wird ein adaptiver Prozeß,**

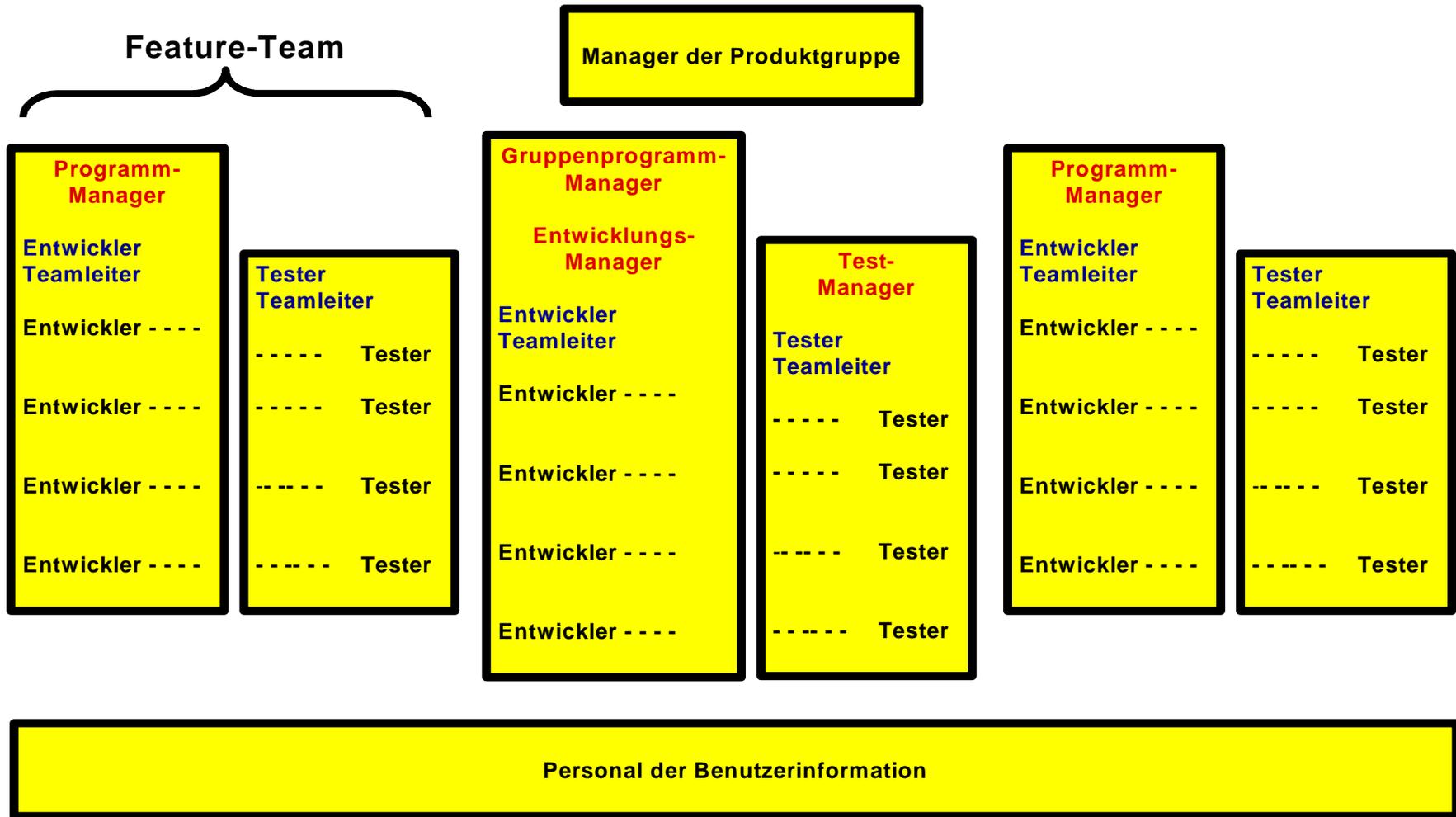
- **der mit unvollständigen Anforderungen beginnen kann,**
- **Teillösungen liefert, die zur Bestimmung von Anforderungen und zu einer verbesserten Kenntnis des Lösungsraumes führen,**
- **bei Änderungen von Anforderungen schnell reagieren kann**
- **und dennoch mit hoher Produktivität, schnell zu einem Produkt guter Qualität führt**



- Produktorientierung
- Synch and Stabilize
- Timeboxing
- Joint Requirements Engineering



Produktorientierung (Beispiel Microsoft)



Synch and Stabilize (Beispiel Microsoft)

- **Planungsphase**

- ◆ Vision
- ◆ Spezifikation und
- ◆ Zeitplan und Zusammensetzung des Feature-Teams für das Produkt definieren

- **Entwicklungsphase**

- ◆ Entwicklung von Features in 3 oder 4 sequentiellen Unterprojekten, (zuerst wichtigsten Features und gemeinsame genutzte Komponenten, dann zweites Drittel der Features, dann unwichtige Features), die jeweils mit einer Meilensteinabfrage abgeschlossen werden
- ◆ Parallele Entwicklung und Implementierung mit täglicher Integration („daily build“)

- **Stabilisierungsphase**

- ◆ Abstimmung mit OEMs und IVS; Analyse Kundenfeedback
- ◆ Intensive interne Tests (gesamtes Produkt innerhalb der Firma) und
- ◆ externe Tests (Beta-Stellen wie OEMs, IVS, und Endbenutzer),
- ◆ letzte Produktspezialisierung und
- ◆ Auslieferung

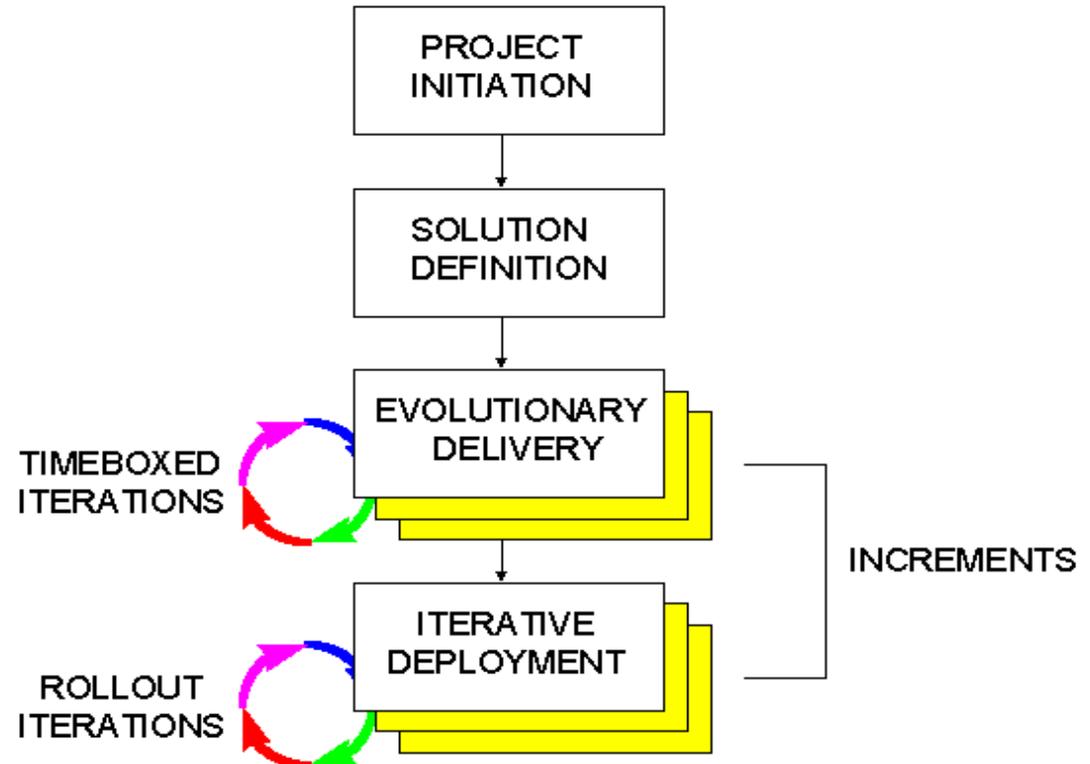
- **12-24 Monate von Meilenstein 0 bis zur Produktion (mit Pufferzeiten)**

● Timebox

- ◆ Iterativ ausgeführter Teil-Arbeitsschritt
- ◆ Anfangs- und Endtermin sind fest, ggfs. werden Funktion bzw. Qualität ‚angepasst‘
- ◆ Typischerweise 1 bis 4 Wochen
- ◆ Teilergebnis wird vollständig definiert, entwickelt und getestet bzw. integriert
- ◆ end-date is immovable and the resources to be used are probably fixed, but what is delivered is flexible

● Prinzipieller Ablauf des TimeBoxing

- ◆ Kick-off meeting
- ◆ Review der letzten Ergebnisse
- ◆ Vereinbarung von
 - Priorisierung
 - Ressourcen
 - Abnahmekriterien
 - Meilensteinen
- ◆ Einrichtung eines Logbuches
- ◆ Aufzeichnung der Aktivitäten
- ◆ Abschluss der Timebox



- **Teamarbeit**

- ◆ alle Personengruppen, die Knowhow bezüglich Kundenanforderungen und/oder Lösungen besitzen, arbeiten in einem Team zusammen (moderierte Gruppensitzungen als wesentliches Element)
- ◆ Kunden und Entwickler sind Partner, der Prozess muss beiden Seiten nutzen

- **Trennung von Anforderungen und Lösungen**

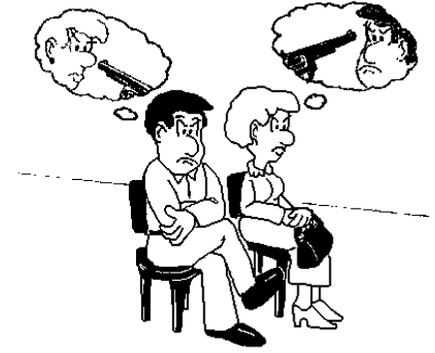
- ◆ Anforderungen sind stabiler als Lösungen
- ◆ getrennte Analyse erhöht Flexibilität auf beiden Seiten

- **Kundenorientierte Priorisierung und Fokussierung**

- ◆ Das 80/20 Pareto Prinzip aus Sicht der Qualität:
Durch Beschränkung auf die 20% wichtigsten Kundenanforderungen werden bis zu 80% der Gesamtzufriedenheit erreicht
- ◆ Das 80/20 Pareto Prinzip aus Sicht der Geschwindigkeit:
20% der Aktivitäten konsumieren 80% der Gesamtzeit
- ◆ Fazit: Fokussierung auf das Wesentliche beschleunigt Prozesse und/oder erhöht Qualität

● Irrtümer der Kritiker (Pessimisten)

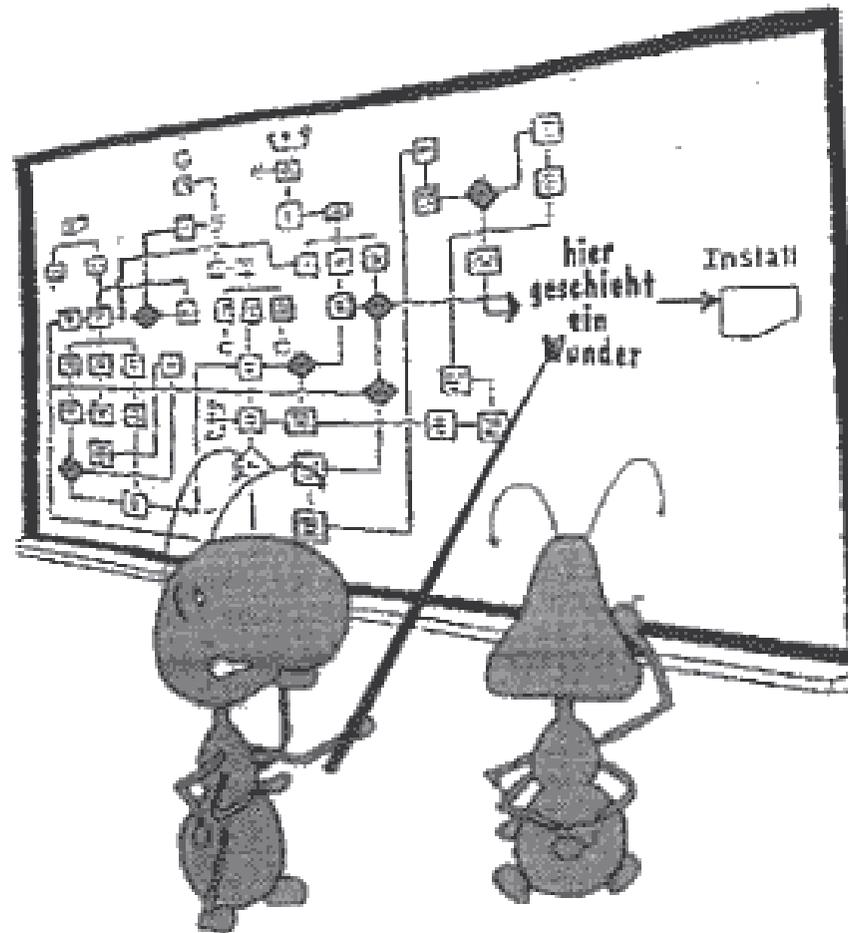
- ◆ Qualitätsmanagement behindert die „eigentliche“ Arbeit der Entwickler
- ◆ Qualitätsmanagement führt zu Bürokratie und Papierflut



● Irrtümer der Euphoriker (Optimisten)

- ◆ Gute Qualitätsmanagement produziert automatisch erfolgreiche Produkte
- ◆ Qualitätssicherung funktioniert im eCommerce genau so wie in der Fertigungsindustrie

"Der Optimist glaubt, wir leben in der besten aller möglichen Welten; der Pessimist glaubt, daß das stimmt."



Sehr gute Arbeit!
Aber sollten wir hier vielleicht nicht
noch ein wenig detaillierter werden...?