

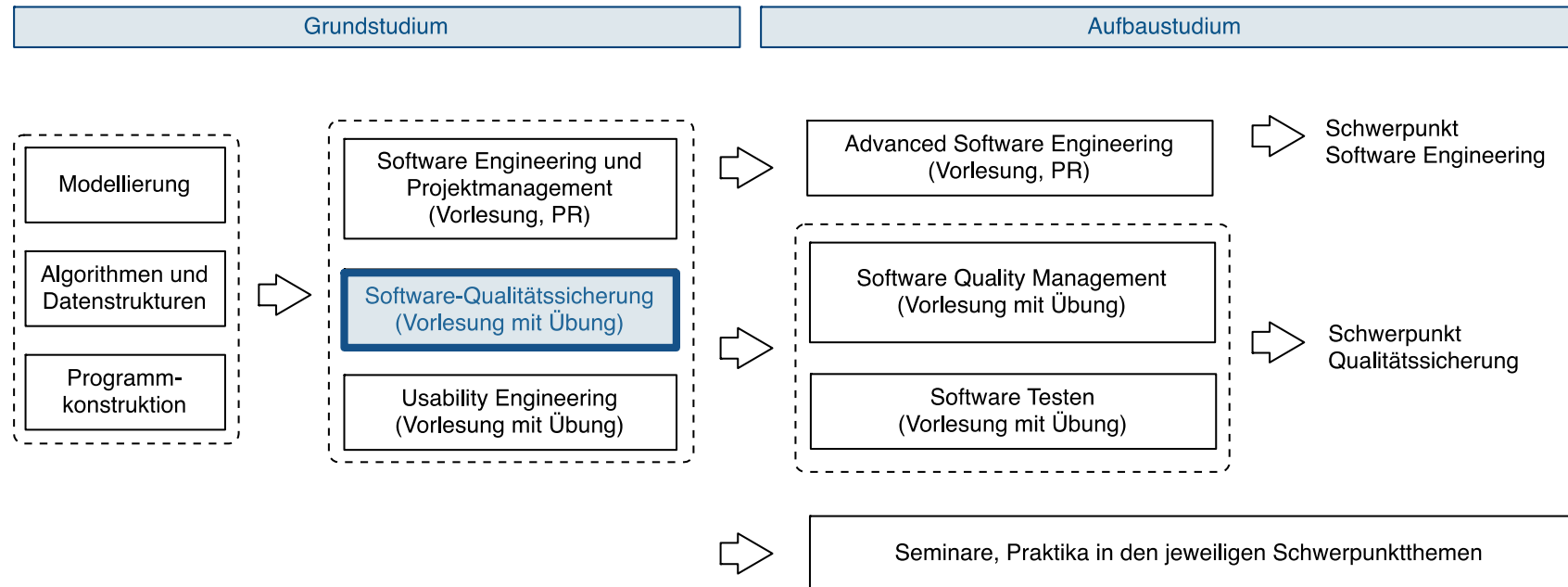
# **Software-Qualitätssicherung VU**

## **Block 1**

Stefan Biffl   Dietmar Winkler

Organisation  
Einführung in die QS  
Qualitätsplanung und -erhaltung

# Qualitätssicherung im Studium



Datum	Thema
<b>VO-Teil:</b> 13.10.	Einführung Qualität und deren Sicherung; Begriffsdefinitionen; Qualitätsplanung; Messen von Qualität
20.10.	Qualitätskontrolle und Fehlerreduktion; Reviews und Inspektionen
27.10.	Ableiten von Testfällen aus SE-Modellen: Methoden der Verifikation und Validierung
10.11.	Testen: Testprozess als Rahmen; Fokus auf Unit- und Modul-Test
17.11.	Test-Driven Development, Refactoring, Testautomatisierung
1.12.	Testen in agilen Prozessen

## ■ Skriptum zur Lehrveranstaltung

- „Qualitätssicherung, Qualitätsmanagement und Testen in der Softwareentwicklung“
- <http://qse.ifs.tuwien.ac.at/courses/skriptum/script.htm>
- Kapitel 1, 2, 4, 5, 6, 7, 9 sind für die Prüfung besonders relevant.

## ■ Best Practice Software-Engineering

- Eine praxiserprobte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen
- Spektrum Akademischer Verlag  
Auflage: 1st Edition. 2010  
ISBN: 978-3827424860
- Kapitel 1, 2, 3, 5



# Featured Project *Presentr*

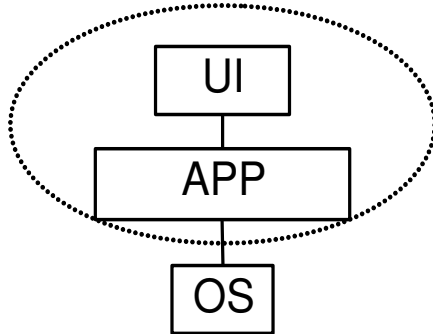
## Web-Plattform für effiziente interaktive Präsentationen

- *Presentr* ist ein aus einem erfolgreichen Projekt aus *Advanced Software Engineering* entstanden.
- Beiträge von *Presentr* (<http://presentr.at>)
  - Vortragende können via *Presentr* im Browser präsentieren.
  - Studierende können an Audience Response Tasks teilnehmen.
  - Studierende können die Präsentation mit den Ergebnissen des Audience Response Tasks als PDF herunterladen.
- Dadurch können Vorträge effizienter interaktiv gestaltet werden.
- *Presentr* wird anhand des Feedbacks kontinuierlich weiter entwickelt.
- *Presentr* wird in diesem Semester in der SEPM Vorlesung durchgehend verwendet.
- Ihre Rückmeldungen sollen in die Gestaltung von *Presentr* einfließen.

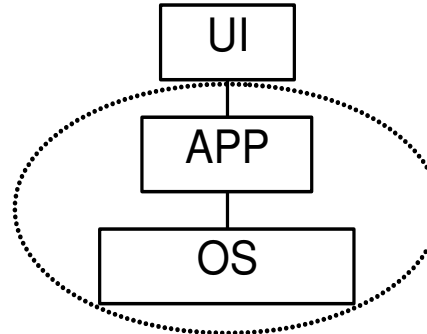
# Planung & Sicherung von Qualität

Definitionen, Leistungen, Prinzipien der  
Qualitätssicherung (QS)

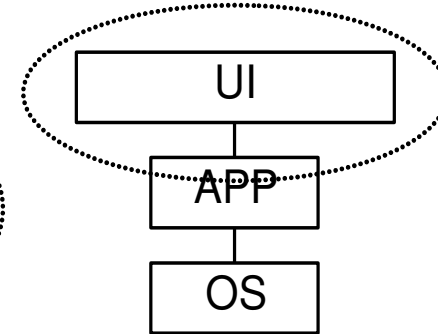
# Systemtypen im Überblick: Fokus Wert/Aufwand



A) Kommerzielles System

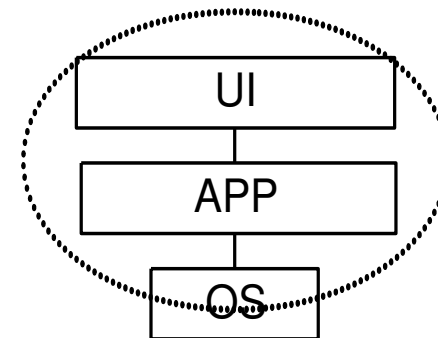


B) Echtzeit



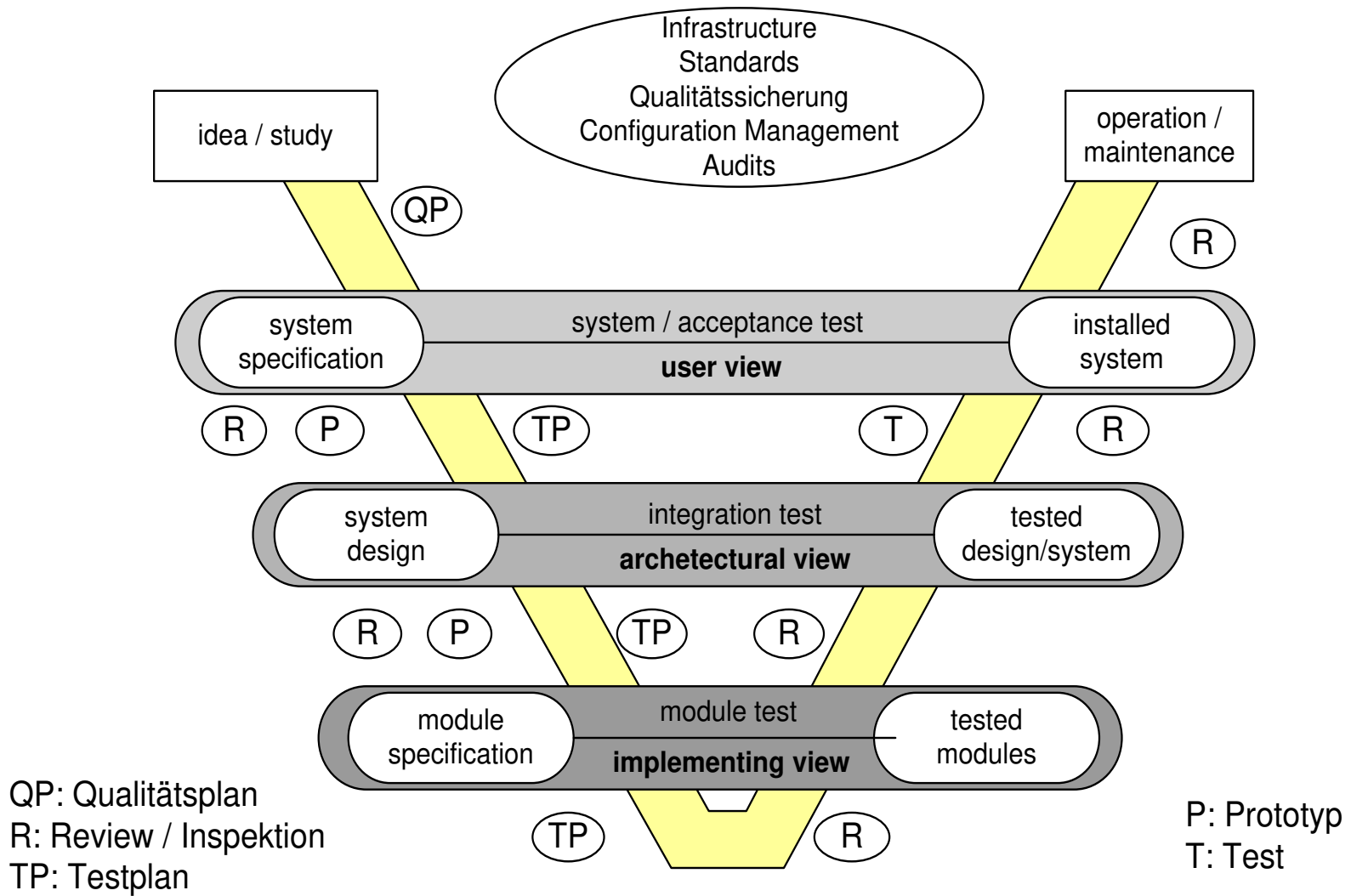
C) Webapplikation

UI ... User Interface (Benutzerschnittstelle)  
APP ... Application (Anwendung)  
OS ... Operating System (Betriebssystem)



D) Computerspiele

# QS im Kontext mit dem V-Modell





# Was ist „Software-Qualität“?

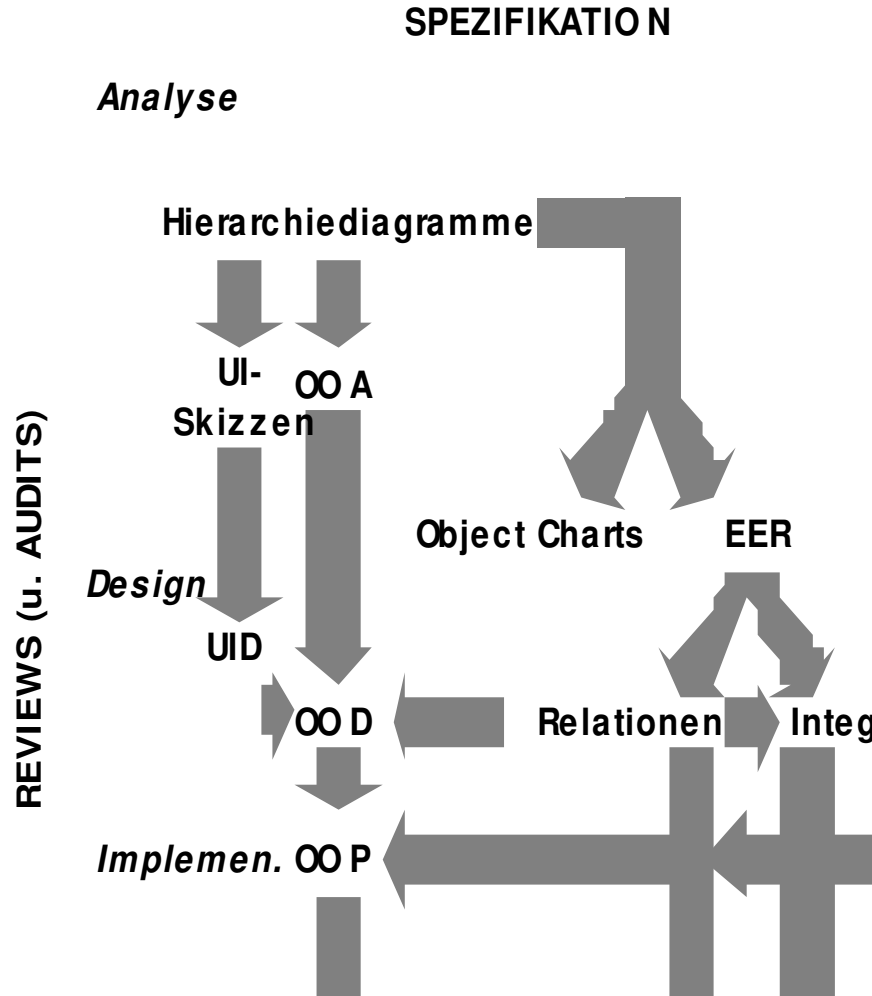
In der gängigen Literatur gibt es

- keine einheitliche Definition von „Qualität“.

**aber verschiedene Ansätze, um „Qualität“ begrifflich zu fassen:**

- Die Kundenzufriedenheit definiert die Qualität eines Produktes.
- Erfüllung von Anforderungen und Attributen bzw. Qualitätsfaktoren (z.B. nach dem Modell von McCall, o.ä.).
- Erfüllung von Vorgaben, Richtlinien, Normen, Standards und gesetzlichen Regelungen für Produkte, Projekte, Prozesse und Dienstleistungen.
- Qualität ist die Gesamtheit von Eigenschaften und Merkmalen eines Produkts oder einer Leistung, die sich auf die Eignung zur Erfüllung vordefinierter Anforderungen beziehen.

- **Eine formale Definition nach DIN8402:**  
„Die Beschaffenheit einer Einheit bezüglich ihrer Eignung, festgelegte und vorausgesetzte Erfordernisse zu erfüllen“.
- **Qualitative hochwertige Software-Produkte sind beispielsweise**
  - termingerecht und im Rahmen des Budgets erstellt.
  - für den **Anwender** verwendbar.
  - für den **Professionisten** verständlich und änderbar.
  - für den **Betreiber** effizient und administrierbar.
- **Beispiele:**
  - „Guter“ Projektablauf bezüglich Terminen/Kosten:  
Einhalten von Meilensteine gemäß Projektplan.
  - Einfache Wartung des Produktes möglich:  
Messbar über Wartungszeit, -dauer und -kosten.
  - Keine bzw. wenige aufgetretene Fehler während der Programmausführung.



**Qualitätsplanung**

**Reviews**

**Testvorbereitung**

**Testdurchführung**

**Fehlerbehebung**

# Qualitätssicherung im Software Engineering

- **Qualitätssicherung** (QS) besteht in der Durchführung von **Verifikation** und **Validierung** in jeder Phase der Software-Herstellung.
- Die Organisation der QS sollte **unabhängig** von einem bestimmten Projekt sein, und projektübergreifende Leistungen erbringen.
- Qualität ist keine Eigenschaft, die später hinzugefügt werden kann. Qualität muss **während der Entwicklung** gesichert werden.
- **Qualitätsmanagement** ist die Menge aller Aktivitäten, Vorgehensweisen, Techniken und Hilfsmittel, die sicherstellen, daß ein Software-Produkt **vordefinierte Standards** erreicht oder übertrifft.

# Qualitätsfaktoren nach IEEE

Die IEEE beschreibt einen allgemeinen Ansatz zur Definition von **Qualitätsfaktoren**:

- Korrektheit
- Zuverlässigkeit
- Effizienz
- Integrität
- Verwendbarkeit
- Wiederverwendbarkeit
- Wartbarkeit
- Flexibilität
- Testbarkeit
- Portabilität
- Verknüpfbarkeit

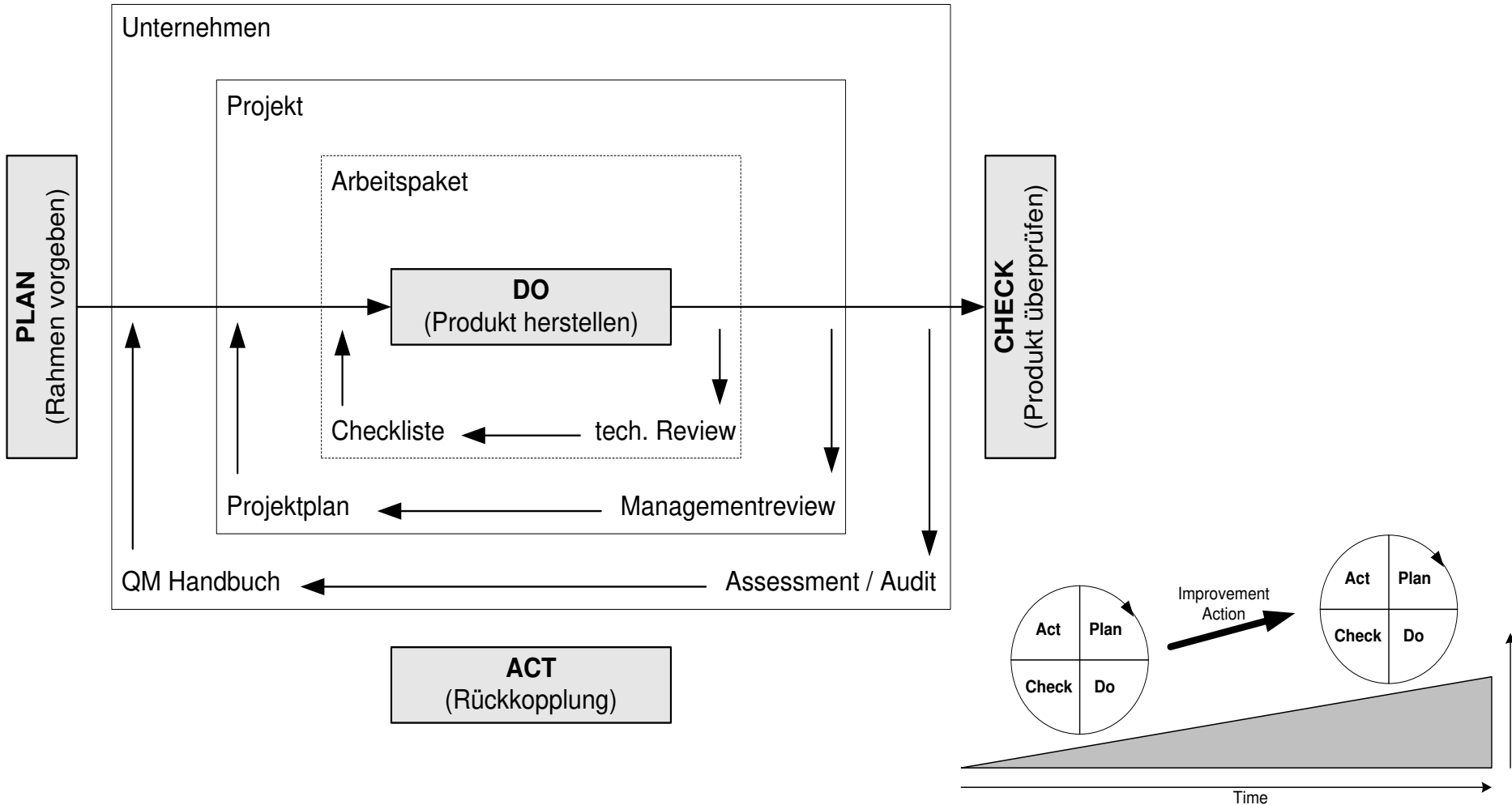
# Erstellung qualitativ hochwertiger Software

- **Die Erstellung qualitativ hochwertiger Produkte erfordert, ...**
  - Einen angemessenen **Softwareprozess** (z.B. Life-Cycle Modell) und adäquate **Vorgehensmodelle** zur kontrollierten Projektabwicklung (z.B. V-Modell XT, RUP, Spiralmodell, Agile Ansätze).
  - Wirkungsvolle **Methoden zur Produkterstellung**.
  - Wirkungsvolle **Methoden zur Produkt- und Prozessverbesserung** (z.B. Reviews, Inspektionen, Testen)

**Die Erzeugung eines qualitativ hochwertigen Produktes  
wird dadurch NICHT garantiert!**

**Die Wahrscheinlichkeit dafür steigt jedoch.**

# Produkt- und Prozessverbesserung (PDCA-Zyklus)



# Software Process Improvement (SPI) mit dem PDCA-Zyklus

Kernelemente des SPI Zyklus sind [nach W.E. Deming]

- **Plan**: Planen der nächsten Iteration des SPI-Zyklus basierende auf aktuellen Erfahrungswerten (inkl. Zielsetzung)
- **Do**: Durchführen der geplanten Aktivität (Task, Projekt usw).
- **Check**: Überprüfung der Produkt- und Prozessergebnisse (z.B. durch Messung)
- **Act**: Analyse der Ergebnisse als Feedback für den nächsten SPI Zyklus. Verbesserung der Erfolgsfaktoren, Vermeidung von Wiederholungsfehlern, Beseitigen von Schwachstellen usw.

Ebenen des PDCA-Zyklus:

- Arbeitspaketebene (work space, task level)
- Projektebene
- Unternehmensebene (Management, Qualitätsmanagementsysteme)

**Auch gute Prozesses können bzw. müssen optimiert werden !**



# Leistungen der QS-Stelle für ein Projekt

- Qualitätsplanung als Teil der Projektplanung.
- Herstellen lokaler Standards auf Projekt- und Organisationsebene.
- Review zentraler Projektdokumente.
- Organisieren von Reviews: Ausbildung, Planung, Durchführung, Verbesserungsvorschläge.
- Unterstützung bei Personalauswahl und Software-Zukauf.
- Vorbereitung und Auswertung von Produkttests.

# Prinzipien des Qualitätsmanagements

- Konkrete **operationalisierbare** Qualitätsmerkmale.
- Produkt- und projekt**abhängige** Qualitätsplanung.
- **Unabhängigkeit** bei Qualitätsprüfungen.
- **Mehraugenkontrolle** bei Qualitätsprüfungen.
- **Frühzeitige Entdeckung und Behebung** von Fehlern und Mängeln.
- **Bewertungen** der eingesetzten Qualitätsmaßnahmen.
- Organisation in Form von **Qualitätsmanagementsystemen**, z.B. ISO 9001, CMM(I), SPICE (ISO 15504) u.ä.
- **Rückkopplung** der Ergebnisse der Qualitätsprüfungen.
- Prinzip der **ständigen Verbesserung von Produkten und Prozessen**.

- Qualität
- Qualitätssicherung/Qualitätsmanagement
- Einbindung in SE-Projekte
- Rollen
- Leistungen

# Erzeugung und Erhaltung von Qualität

Produkte, Prozesse, Personen  
Qualitätsplanung: Faktoren, Maßnahmen  
Prozessmodelle

# Beispiel WWW-Bank

Umfeld. Bank: Kontodisposition, Wertpapierkonten, Risikoanalyse für Wertpapierkonten.

Ziel. Fernbewirtschaftung der Konten durch mündige Kunden (schneller, billiger, flexibler).

Technologie. Internet-Anbindung mit Client/Server-Software.

Aufgaben in den Rollen Kunde, Entwickler und Betreiber.

1. Begriffe klären, **Qualitäten mit hoher Priorität** für eine Rolle.
2. Wichtigste **Qualitätskriterien meßbar** bzw. testbar machen (Meßgrößen, Vorgehen).
3. **Wie viel ist “gut genug”** von einer Qualität ?

**QS-Maßnahmen** für Qualitäten überlegen, die für verschiedene Zielgruppen unterschiedlich wichtig sind (z.B. Kunden und Entwickler).

- *Mehrwert oder Hygiene*
- *Sicht der beteiligten Rollen*
  - *Betreiber*
  - *Kunde/Interessent*
  - *Entwickler*
- *Feedback zu Qualitätserreichung in Phase*
  - *Entwicklung*
  - *Test*
  - *Betrieb*
  - *Wartung*
- *Feedback an Entwickler: direkt oder indirekt*

- Mehrwert oder Hygienefaktor
  - Kunde zahlt gerne für Mehrwert.
  - Kunde kauft nicht bei Mängeln, die Hygienefaktoren betreffen, egal wie gut das Produkt sonst ist.
- Wie wichtig für welche Personengruppe
  - Geldgeber
  - Betreiber
  - Kunden
  - Entwickler
- Feedback an Entwickler
  - In welcher Phase (Entwicklung, Test, Betrieb, Wartung)
  - Direktes oder indirektes
- Achtung im QM auf ...
  - Hygienefaktoren (sind bei Entwicklern eher ungeliebt und werden daher oft ausgeblendet)
  - Faktoren, die für die Entwickler nicht besonders wichtig sind bzw. die erst in späten Phasen auftauchen oder nur indirekte Auswirkungen auf die Entwickler selbst haben.

- *Wirksam: vorbeugend oder ‚im Nachhinein‘*
- *Rolle führt Maßnahme durch*

- *Betreiber*
- *Kunde/Interessent*
- *Entwickler*

- *Wirkung*

- *sofort*
- *Entwicklung*
- *Test*
- *Betrieb*
- *Wartung*

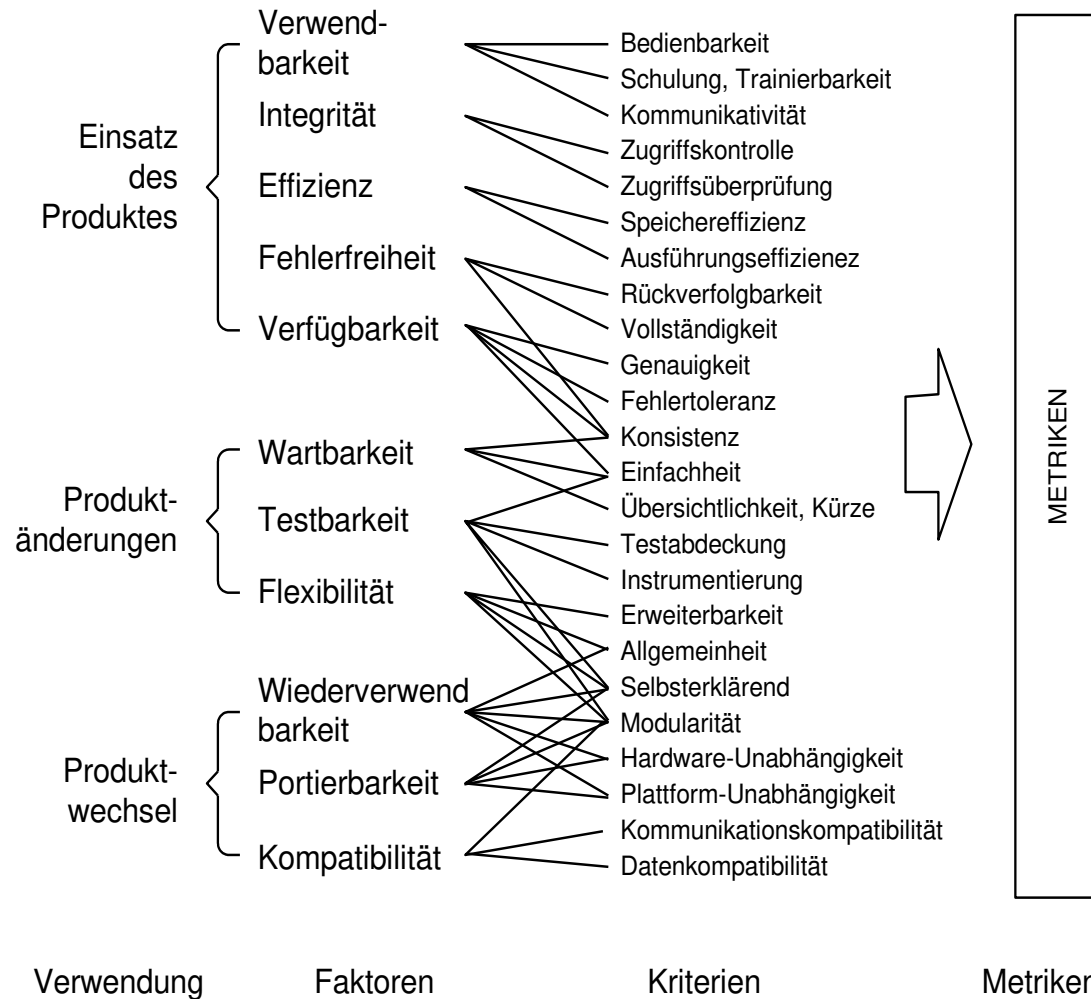
- *Feedback an Entwickler: direkt oder indirekt*

vorbeugend	im nachhinein	Betreiber	Kunde/interessent	Entwickler	sofort	Entwicklung	Test	Betrieb	Wartung	direkt/indirekt
------------	---------------	-----------	-------------------	------------	--------	-------------	------	---------	---------	-----------------



- Vorbeugend oder im Nachhinein
- Betrifft welche Personengruppe(n)
  - Geldgeber
  - Betreiber
  - Kunden
  - Entwickler
- Feedback an Entwickler
  - In welcher Phase (sofort, Entwicklung, Test, Betrieb, Wartung)
  - Direktes oder indirektes Feedback
- Maßnahmen sind besonders zu unterstützen, die ...
  - ... Hygienefaktoren betreffen.
  - ... vorbeugend wirken sollen.
  - ... Bedürfnisse anderer Personengruppen an die Entwickler kommunizieren.
- Maßnahmen sind besonders wirksam, die ...
  - ... das Richtige einfach machen und das Falsche erschweren.
  - ... möglichst rasches und direktes Feedback an die Entwickler liefern.

# Qualitätsmodell nach McCall (1980)



**Qualitätsfaktoren** beschreiben das Verhalten des Systems

**Qualitätskriterien** sind Eigenschaften von Qualitätsfaktoren in Bezug zur Softwareproduktion und Entwicklung

**Qualitätsmetriken** sind Messungen, die definierte Aspekte der Qualitätsfaktoren beschreiben

**„To measure is to know“  
(J.C. Maxwell)**

- “You can’t manage what you can’t measure”, Tom DeMarco
- “What is not measurable make measurable”, Galileo Galilei

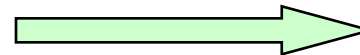
## ■ Definition

- Messen heißt “Ermitteln eines Wertes durch quantitativen Vergleich der Messgröße mit einer Einheit (Normal)” DIN 1319 Teil 1.

## ■ Beispiele



Aufwand



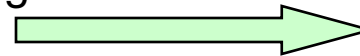
10 PM

Aufwand für eine  
Aufgabe

- Prozess



Größe



700 (LoCs)

Lines of Code

- Produkt



Erfahrung



“high”

>10 Projekte

- Ressourcen

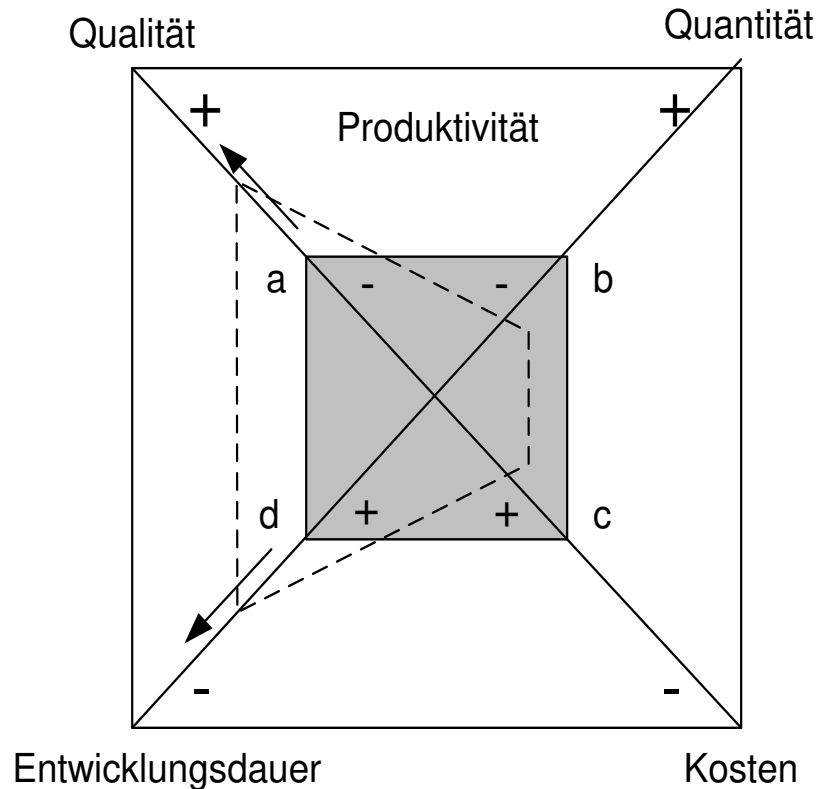
# Ausgewählte Typen von Messungen und Daten

- Direkte vs. indirekte Messungen:
  - **Direkte Messung**: Wertermittlung direkt beim zu untersuchenden Objekt (z.B. Dauer, Aufwand einer Aufgabe)
  - **Indirekte Messung**: Ermittlung von Messwerten aus direkten Messungen (z.B. Effizienz einer Fehlererkennungsmethode = Anzahl der gefundenen Fehler pro Zeiteinheit)
- Objektive vs. Subjektive Messungen
  - **Objektive Messung**: LoC, Auslieferungsdatum, Aufwand usw.
  - **Subjektive Messung**: Messergebnisse basierend auf der individuellen Sichtweise des Betrachters, z.B. Fragebögen zur Erfassung der Kundenzufriedenheit.
- Quantitative vs. Qualitative Daten:
  - **Quantitativ**: Daten als konkrete Zahlenwerte (z.B. für statistische Auswertungen)
  - **Qualitativ**: Visualisierte Informationen (Text, Bilder), z.B. durch Interviews, Interpretationen.

# Einflussfaktoren bei der SW-Entwicklung

- **In einem Entwicklungsprojekt sind 4 wesentliche Einflussfaktoren bzw. Zielsetzungen, die sich wechselseitig beeinflussen, zu finden:**
  - **Qualität**  
Beispiele: Anzahl der Fehler in einem Produkt, Termin- und Kostentreue.
  - **Quantität**  
Beispiele: die Anzahl der implementierten Funktionen, Function Points, Lines Of Code (LOC).
  - **Entwicklungsdauer (z.B. in Monaten) bzw. Aufwand (z.B. in Personenmonaten)**  
Beispiele: kleine“ vs. „große“ Projekte
  - **Kosten**
- **Diese Einflussfaktoren wirken wechselseitig auf die Produktivität, die in Abhängigkeit der verfügbaren Ressourcen als konstant betrachtet wird.**
- **Symbolisierung durch das „Teufelsquadrat nach Sneed“**

# Teufelsquadrat nach Sneed



- Spannungsfeld von Qualität, Quantität (Umfang), Entwicklungsdauer und Kosten.
- Messung und Skalierung der Werte (von den Ecken zur Mitte).
- Die Fläche des Quadrats ändert sich nicht (= Produktivität).
- Erhöhung der Qualität und Senkung der Entwicklungsdauer funktioniert nur durch Reduktion des Umfangs und/oder Erhöhung der Entwicklungskosten.

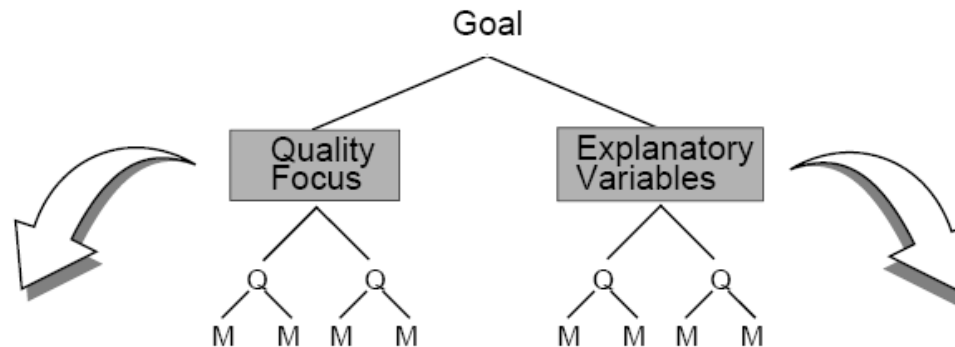
# Goal Question Metrics Paradigm nach Basili

Das Entwicklungsziel des **Goal-Question-Metrik** Modells (GQM) nach Basili und Rombach ist die *Entwicklung eines Systems zur Messung von Softwarequalität.*

Das GQM-Modell besteht aus folgenden drei Stufen :

- Konzeptuelle Stufe (Goals)
- Operative Stufe (Question)
- Quantitative Stufen (Metric )

## Question Categories



### Quality Focus

- specifies the quality focus: a specific interpretation of “Reliability”

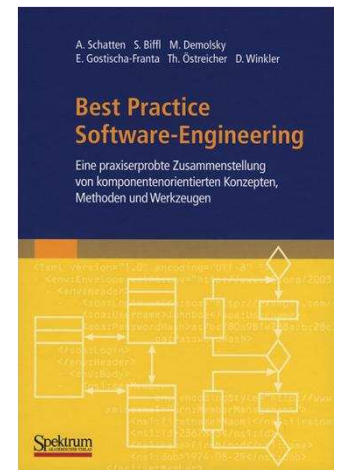
### Explanatory Variables

- specifies the explanatory variables that may have an impact on the quality focus: e.g., degree of experience of the developer, used detection mechanism

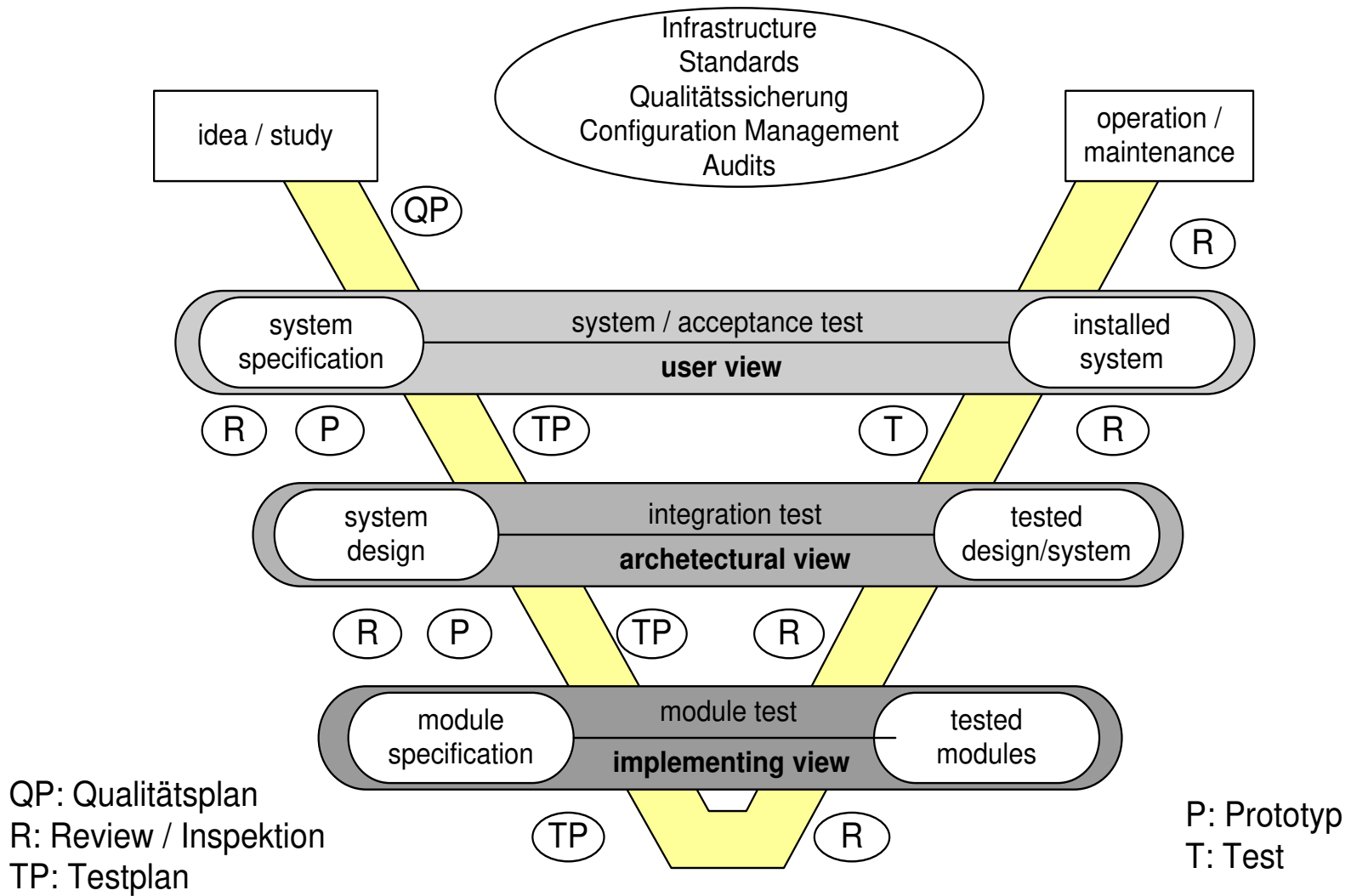


# SE-Prozessmodelle und QS

## Integrierte Organisation des SE und der QS als Bauplan für ein Projekt



# QS im Kontext mit dem V-Modell



# Konstruktive vs. Analytische Qualitätssicherung

## Konstruktive Qualitätsmaßnahmen

- **Technische** Maßnahmen  
z.B. Verwendung von Prinzipien, Methoden und Werkzeugen der Softwareentwicklung .
- **Organisatorische** Maßnahmen  
z.B. Verwendung von Vorgehensmodellen
- **Menschliche** Maßnahmen  
z.B. Schulung der Projektmitarbeiter.

## Analytische Qualitätsmaßnahmen (Prüfung und Bewertung der Qualität eines Produktes)

- **Statische** Prüfungen  
z.B. Reviews, Inspektionen, Audits
- **Dynamische** Prüfungen  
z.B. Software Tests

# Integration von QS in Vorgehensmodellen

## V-Modell XT

- QS ist als **eigenständiger Vorgehensbaustein** realisiert.
- Dieser Vorgehensbaustein ist ein **verpflichtender** Bestandteil aller Projekttypen.
- Der Projekttyp „Einführung und Wartung eines organisationsspezifischen Vorgehensmodells“ beinhaltet den Gedanken der **ständigen Verbesserung**.

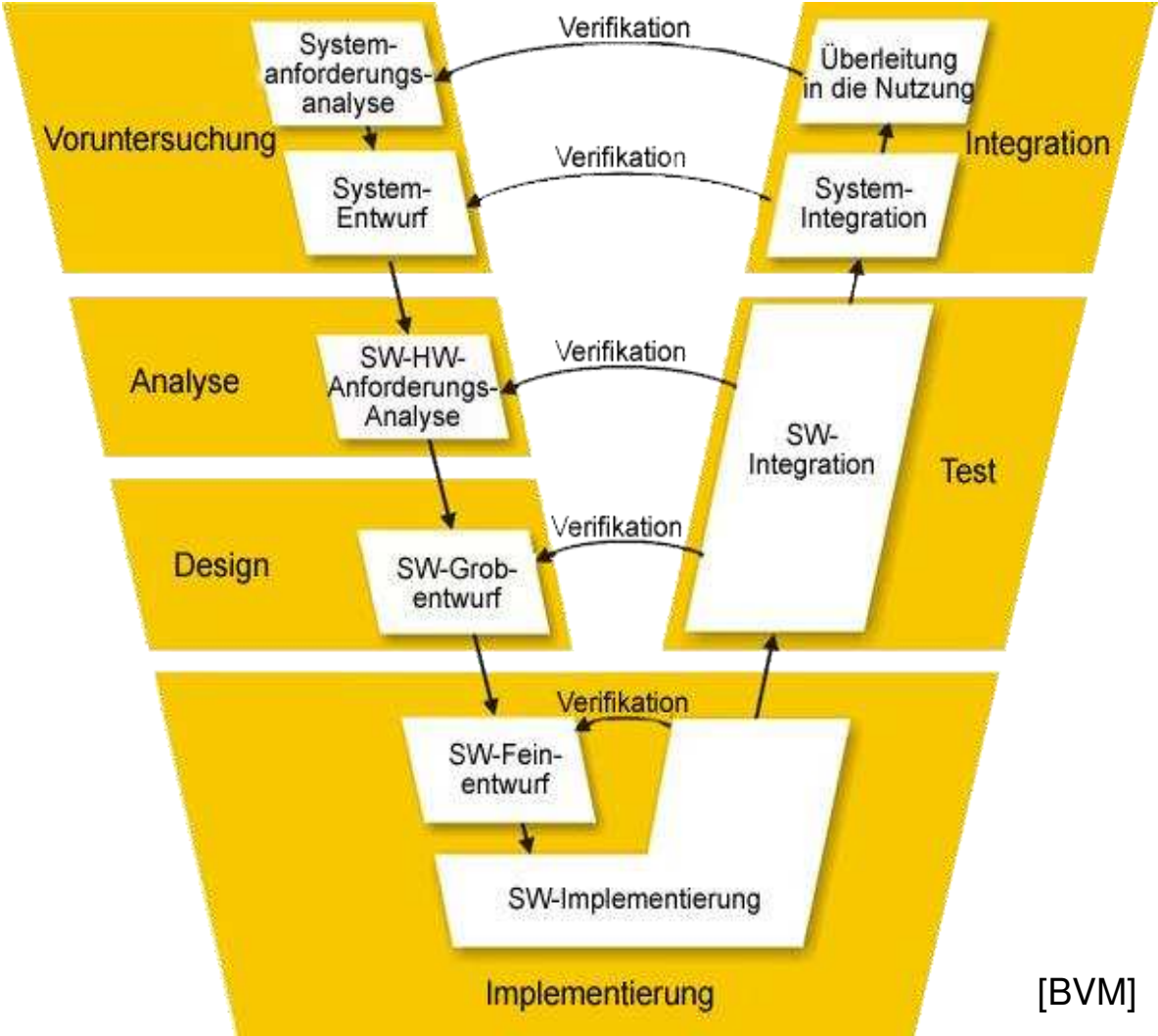
## Rational Unified Process (RUP)

- QS mit zugeordneten Rollen ist ständiger Begleiter während des **gesamten Projektverlaufs**.
- **Review** bei allen Meilensteinen.
- Eigene „**Test Discipline**“

## Agile Entwicklungsprozesse (z.B. SCRUM)

- Reviews und Tests als fixer Bestandteil der Sprints.
- Test-Driven Development.

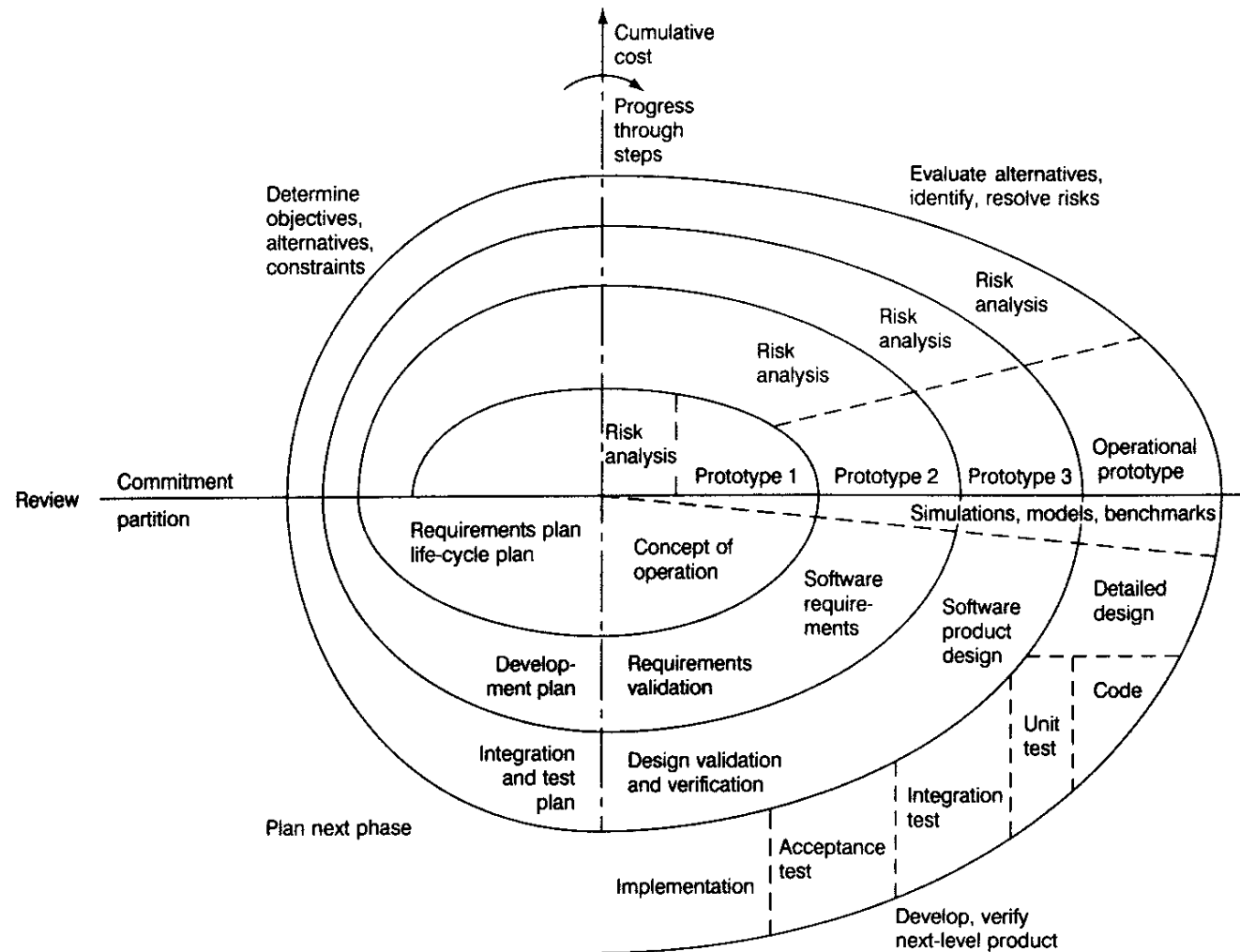
# Sequentielles Prozessmodell: V-Modell97



[BVM]

- Spezifikation vs. Umsetzen and Testen
- Verschiedene Sichten (Rollen, Abstraktion): Applikation, Architektur und Implementierung
- Kontext von Produkten and Tests
- Problem: Fehler aus frühen Phasen der Entwicklung
- Voraussetzungen und Annahmen (z.B. Funktionen, Kernqualitäten) sollten stabil und gut verstanden sein

# Iteratives Prozess Modell: Spiral Modell



[Boehm2000]

- Orientiert an **Investitionen und Risikomanagement**
- Sieht jede Phase als Subprojekt
  - Entscheidung über Projektumfang pro Phase
- Vier Schritte in jedem Zyklus
  1. Kick-off: Bestimme Ziele und Alternativen
  2. Einschätzung der Risiken der Alternativen
  3. Entwicklung und Test
  4. Plan der nächsten Phase; weiter bei 1.



# Example: System with support systems



Counter

Passenger and Luggage Check In



Terminals



System



Parking



Fire Brigade



Service

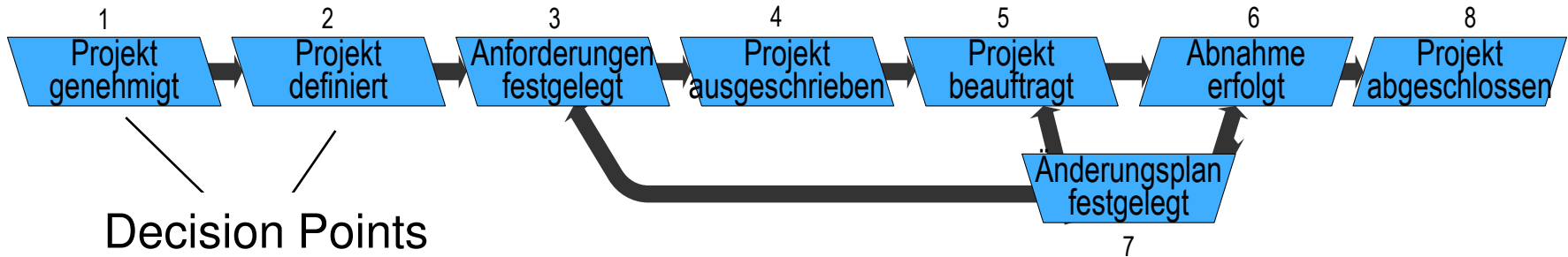


Tower

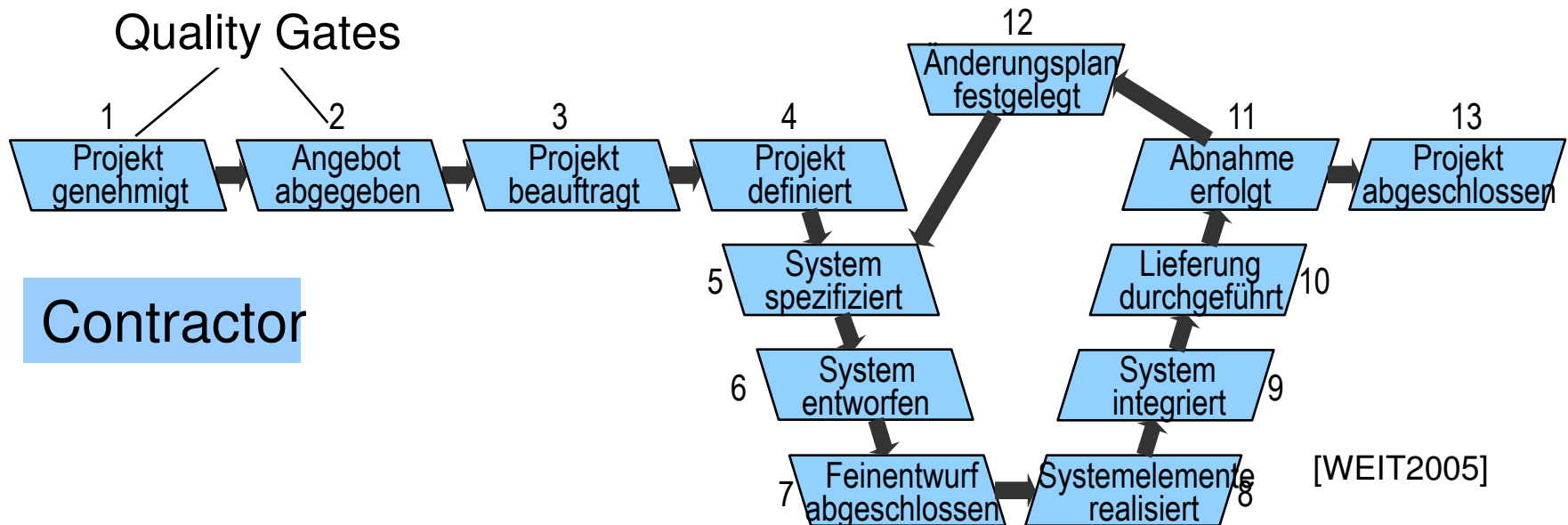
# V-Modell XT 2005:

## Projektdurchführungsstrategien

### Auftraggeber



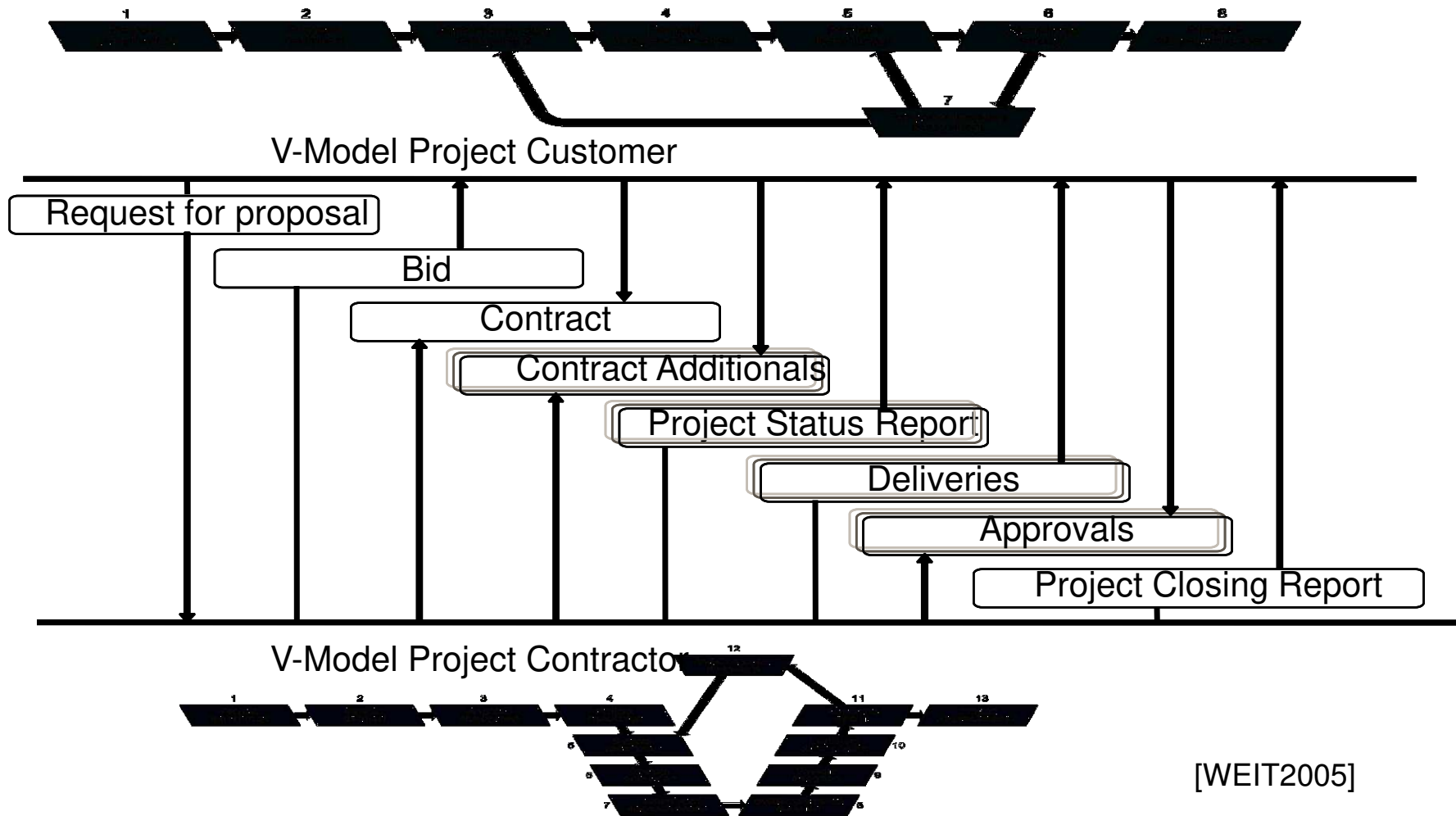
Decision Points  
Quality Gates



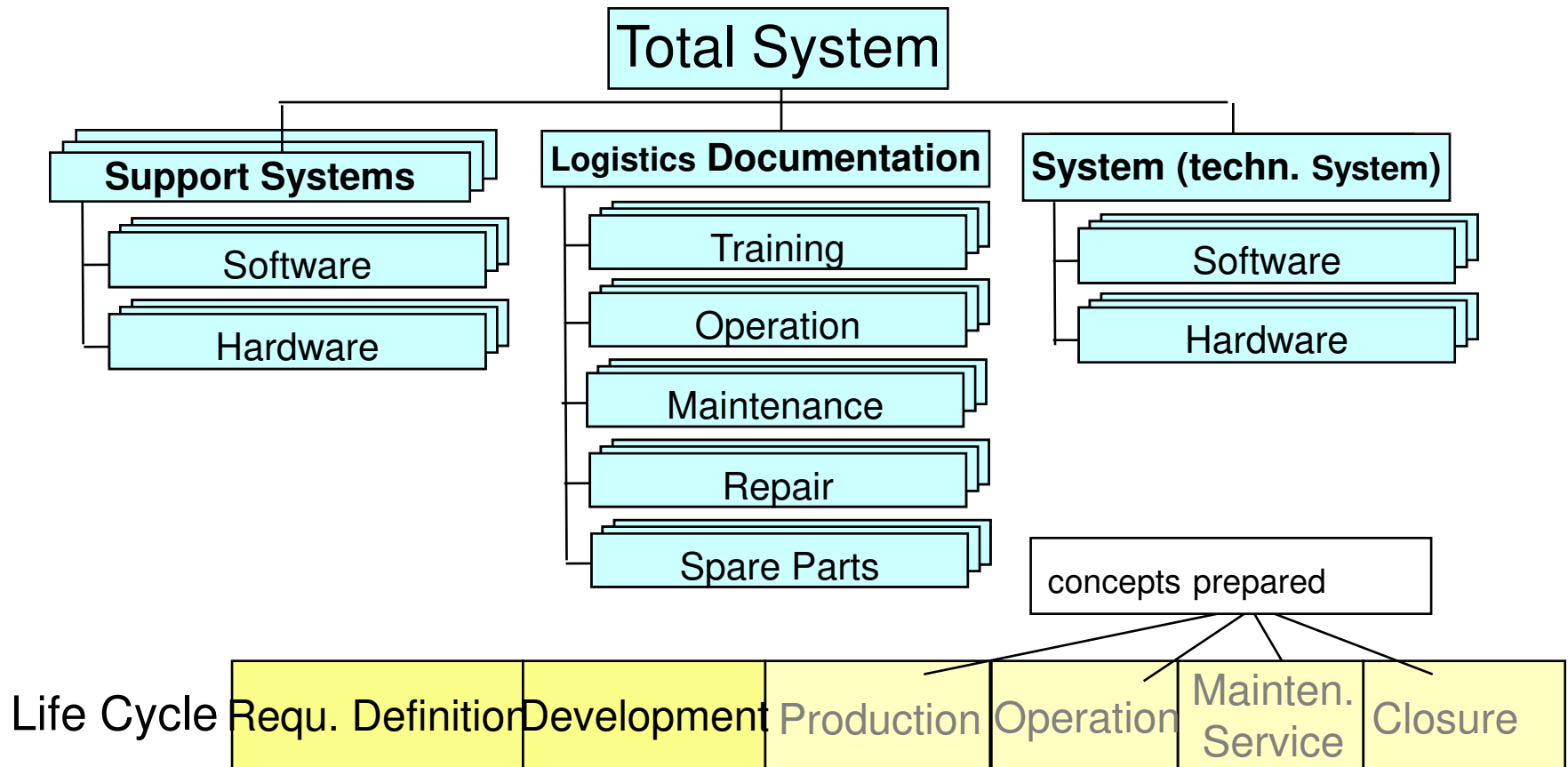
### Contractor

[WEIT2005]

# Interface Customer - Contractor

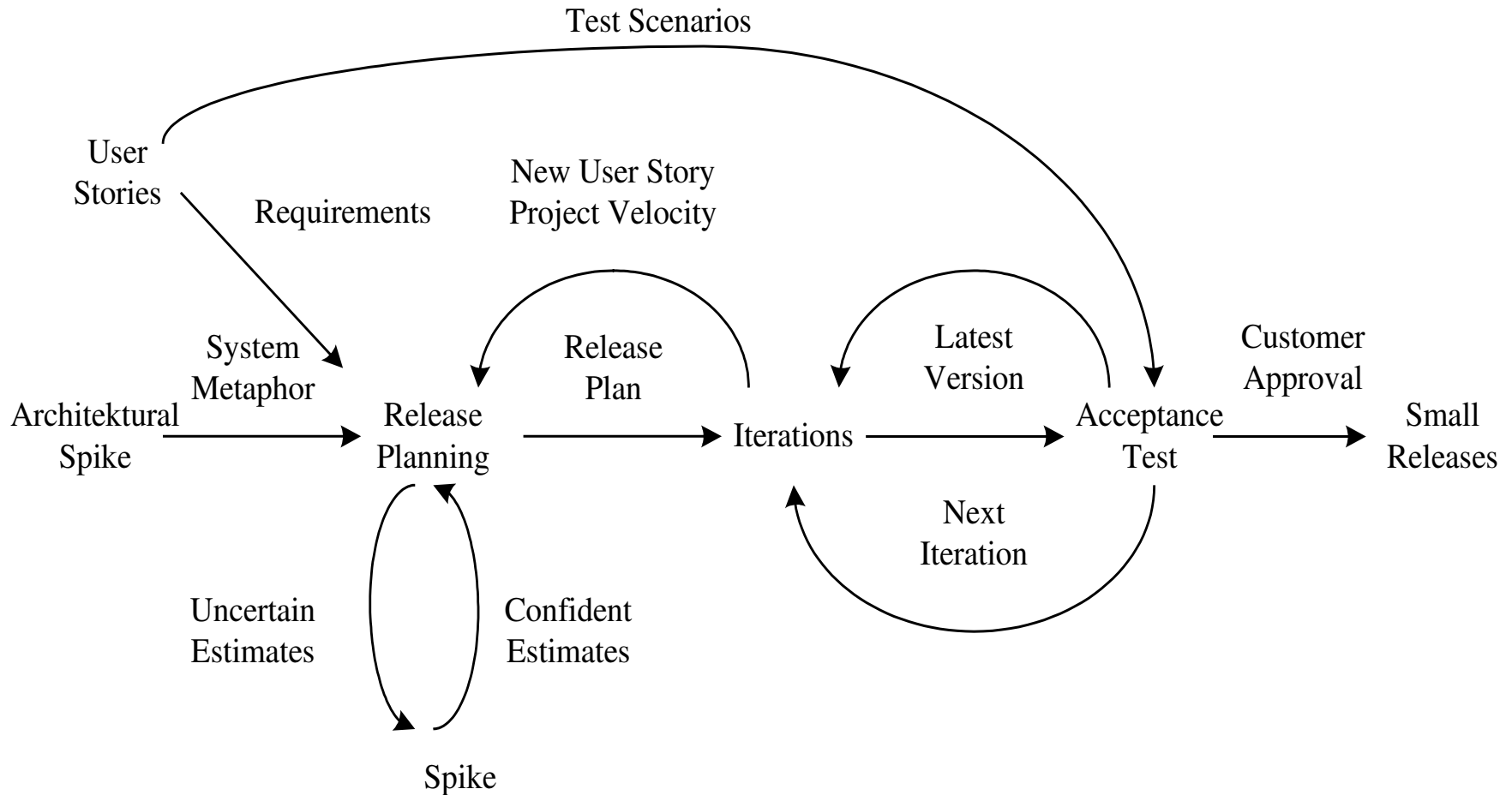


# „Systems of Systems“ in V-Model XT



**V-Model XT supports „Systems of Systems“** [WEIT2005]

# Agile Ansätze: XP-Projektablauf



## Zusammenfassung

- Es gibt keine einheitliche Definition für "Qualität" aber verschiedene Ansätze, um "Qualität" begrifflich zu fassen.
- Überprüfung, Beurteilung und Verbesserung der Produkt- und Prozessqualität erfordert messbare Attribute.
- Qualität, Quantität, Kosten und Entwicklungsdauer bilden ein Spannungsfeld in der Softwareentwicklung.
- QS ist integraler Bestandteil von Vorgehensmodellen (z.B. V-Modell (XT), Rational Unified Process, Agile Ansätze usw.)
- Das Prinzip der ständigen Verbesserung von Produkten und Prozessen ist ein integraler Bestandteil von Qualitätsmanagementsystemen und ist auf allen Ebenen (Arbeitspaket, Projekt, Unternehmen) anwendbar.
- Produkte und Prozesse können immer verbessert werden!

## Ausblick:

- Ausgewählte statischen Methoden der Qualitätssicherung (Reviews und Inspektionen).

- Basili, Victor R.; Caldiera, Gianluigi; Rombach, H. Dieter: "The Goal Question Metric Approach"; Encyclopedia of Software Engineering - 2 Volume Set, pp 528-532; John Wiley & Sons, Inc.; 1994
- Boehm, B. W.: A spiral model of software development and enhancement. Computer, 21(5):61–72, May 1988.
- Biffl Stefan, Winkler Dietmar, Frast Denis: „Qualitätssicherung, Qualitätsmanagement und Testen in der Softwareentwicklung“, Skriptum zur Lehrveranstaltung, 2004.  
<http://qse.ifs.tuwien.ac.at/courses/skriptum/script.htm>  
Kapitel 1, 2, 4, 5, 6, 7, 9 sind für die Prüfung besonders relevant.
- IEEE, 1990: IEEE Standard Glossary of Software Engineering Terminology. IEEE STD 610.12-1990
- IEEE, Guide to the Software Engineering Body of Knowledge, IEEE, 2004.
- IEEE-STD 610.12-1990 - Standard Glossary of Software Engineering Terminology, 1990
- Schatten, Biffl, Winkler, et al. Best Practice Software-Engineering - Eine praxiserprobte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen; Spektrum Akademischer Verlag, 2010.
- Sommerville Ian: „Software Engineering“, 9th Edition, Addison-Wesley, 2010.
- Thaller Georg Erwin: „Software Qualität“, VDE Verlag, 2000, ISBN: 3-8007-2494-4.
- Wallmüller Ernest: „Software Quality Engineering“, 3. Auflage, Hanser, 2011, ISBN 978-3-446-40405-2.

# Web-Ressourcen

[QSE] <http://qse.ifs.tuwien.ac.at>

[BPSE] <http://bpse.ifs.tuwien.ac.at/>

<http://best-practice-software-engineering.blogspot.com/>

<http://bpse.ifs.tuwien.ac.at/podcast.html>

[CMMI] <http://www.sei.cmu.edu/cmmi/>

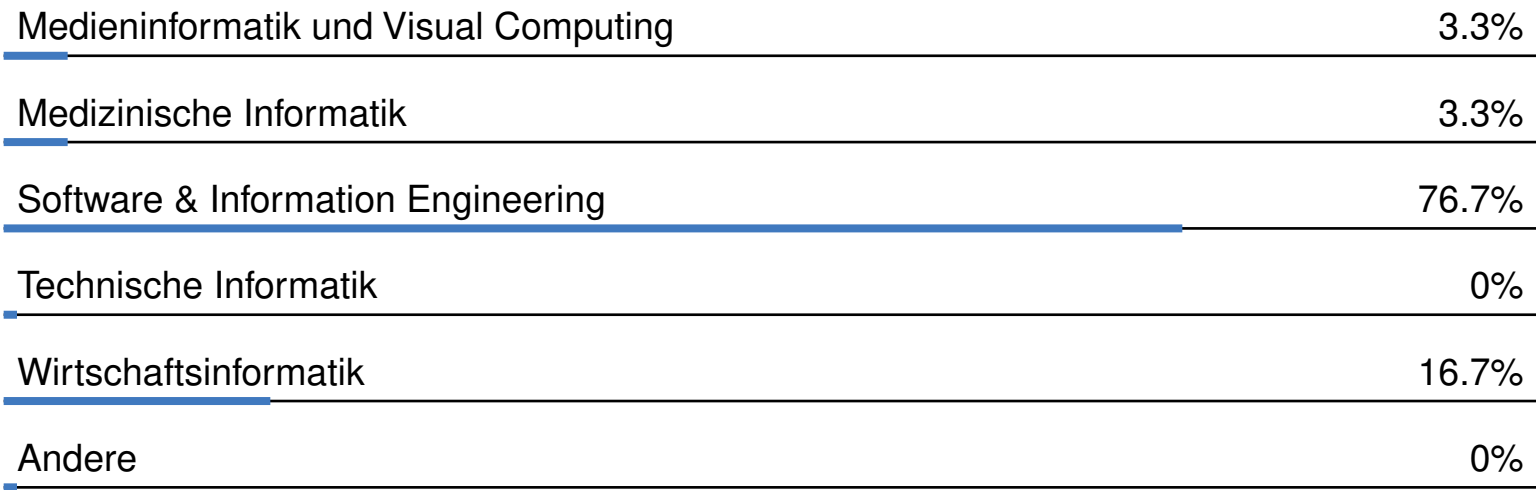
[IEEE] IEEE Software Engineering Body of Knowledge (SWEBOK) [www.swebok.org](http://www.swebok.org)

[PMBok] <http://www.pmi.org/PMBOK-Guide-and-Standards.aspx>

V-Modell: Entwicklungsstandard für IT-Systeme des Bundes, 1997, <http://www.iese.fhg.de/VModell/>



# Welche Studienrichtung studieren Sie?



# Nennen Sie ein Beispiel für eine "Software Suprise"

Was haben Sie erwartet, dass die Software tun sollte? Was ist tatsächlich passiert? Was wäre eine wahrscheinliche Ursache?

Anton Höbl

Votes: + 16 / - 0

Skype, wenn der schließen Button gedrückt wurde ist das Fenster minimiert anstatt geschlossen worden.

Stefan Reinsperger

Votes: + 14 / - 1

Werbung auf Homepages: Bei Klick auf X schließt sich die Werbung nicht sondern öffnet sich

Matthias Steinbrecher

Votes: + 12 / - 0

Presentr stellt die Frage gleich am Anfang der Slide

David Stemberger

Votes: + 9 / - 0

Bei Amazon kann man sicher mehrmals mit derselben E-Mail Adresse registrieren. Unterscheidung findet anhand des PWs statt.

Michael Pürmayr

Votes: + 6 / - 0

Microsoft Office: Das automatische Speichern des Dokuments versagte, wenn es auf einem USB-Stick abgelegt wurde.

Christoph Hafner

Votes: + 4 / - 0

Samsung Kies, Ja und Nein wurde vertauscht

Judith Pitzer

Votes: + 5 / - 1

In TISS kann man sich bei wissenschaftlichen Arbeiten zu 2 Gruppen gleichzeitig anmelden

Michal Palczynski

Votes: + 4 / - 0

---

Nach dem anmelden für QS im Tiss wird man nicht automatisch für den Tuwelkurs angemeldet.

Lukas Pfeifhofer

Votes: + 4 / - 1

---

Zwangs-Update bei Windows. (8, 10) Nach Ablauf der Zeit wird ohne weitere Frage neu gestartet.  
(Speichern nicht mehr möglich.)

Gregor Steiner

Votes: + 4 / - 1

---

Man wird auf TISS nach kürzester inaktiver Zeit wieder ausgeloggt.

Markus Goedl

Votes: + 2 / - 0

---

Ariane V88 - es wurde die Software der Vorgängerrakete genommen. Diese passte nicht für das neue Modell und die Rakete zerstörte sich selbst.

David Schröder

Votes: + 2 / - 0

---

Win10 update erstellt neue Backup Partition, Grub kann nicht mehr booten da die Reihenfolge der partitionen nicht mehr stimmt

Niklas Natter

Votes: + 1 / - 0

---

Löschen von Kontakten auf synchronisierten Iphones, welche auf die selbe AppleID angemeldet sind.

Maximilian Irlinger

Votes: + 1 / - 0

---

Nvidia Treiber: Nach Update muss man sich auf einmal registrieren und einloggen Oo

Matthias Schrabauer

Votes: + 1 / - 0

---

CarToGo app löscht Username, wenn man das Passwort falsch eingibt

Stephan Goldschmidt

Votes: + 2 / - 1

Wenn installierte Software ungefragt im Autostart landet

Christoph Wasylewski

Votes: + 0 / - 0

nach Update von Firefox sind die Add-On Buttons verschwunden .. sie tauchten auch nicht mehr auf

Martin Kamleithner

Votes: + 0 / - 0

MySQL: Nur die ersten 64 (?) Bit des Passworts wurden berücksichtigt

Muhammad Mujahed Hainoun

Votes: + 0 / - 0

Activating skype dial assistant will identity dates as phone numbers.

Christoph Salat

Votes: + 0 / - 0

Software sagt sie startet sich neu. Bei Klick auf neu starten startet sich plötzlich der PC neu...

Bernadette Maria Obermair

Votes: + 0 / - 1

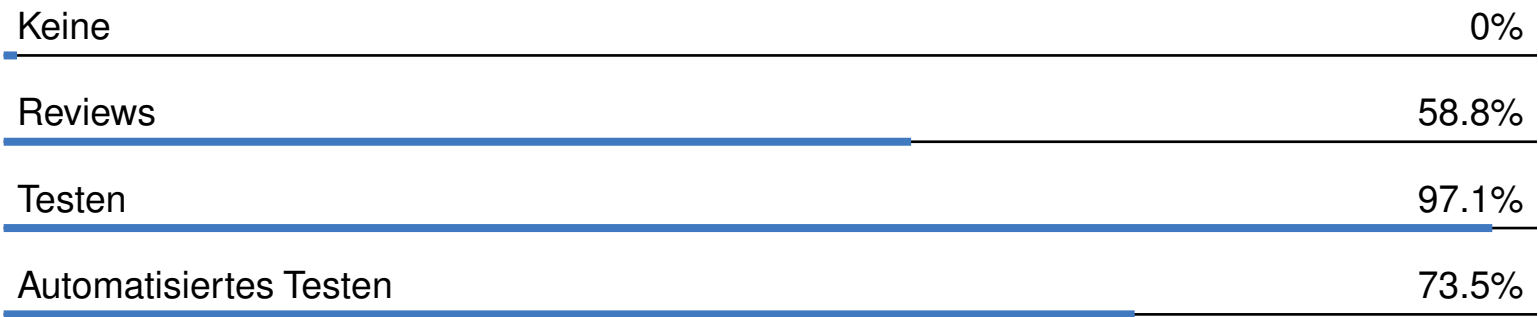
Fenster schließt nicht beim Klicken auf das X

Sebastian Luzian

Votes: + 0 / - 2

presentr lädt dauerhaft rechts oben

# Welche QS Maßnahmen haben Sie selbst in einem Software Projekt bisher durchgeführt?



# Welche Qualitäten sind für die von Ihnen gewählte Rolle (Kunde, Entwickler, Betreiber) besonders wesentlich?

Die Antwort soll bestehen aus [Rolle: Qualität] z.B. Kunde: Zuverlässigkeit

Stefan Reinsperger	Votes: + 20 / - 0
--------------------	-------------------

Kunde: Sicherheit

Tobias Kain	Votes: + 18 / - 0
-------------	-------------------

Kunde: Usability

Judith Pitzer	Votes: + 11 / - 2
---------------	-------------------

Entwickler: Wiederverwendbarkeit

Bernadette Maria Obermair	Votes: + 7 / - 0
---------------------------	------------------

Betreiber: Wartbarkeit

Markus Goedl	Votes: + 7 / - 0
--------------	------------------

Kunde: Plattformunabhängigkeit

Michael Pürmayr	Votes: + 7 / - 0
-----------------	------------------

Kunde: Aktualität (~Echtzeit), Verfügbarkeit 24/7

Julian Waibel	Votes: + 8 / - 1
---------------	------------------

Kunde: Stabilität (kein Absturz)

Stephan Goldschmidt	Votes: + 6 / - 0
---------------------	------------------

Kunde: Support/Hilfe

David Stemberger	Votes: + 5 / - 0
Betreiber: Skalierbarkeit	
Anton Hößl	Votes: + 7 / - 2
Entwickler Betreiber: Verfügbarkeit	
David Schröder	Votes: + 4 / - 0
Betreiber: Korrektheit	
Muhammad Mujahed Hainoun	Votes: + 4 / - 0
Entwickler: Testbarkeit	
Matthias Steinbrecher	Votes: + 8 / - 4
Betreiber: Sicherheit	
Radomir Djurdjevic	Votes: + 3 / - 0
Kunde: Effizienz	
Christoph Hafner	Votes: + 5 / - 3
Kunde: Verfügbarkeit	
Christoph Michael Weber	Votes: + 1 / - 0
Kunde: Support	
Maximilian Irlinger	Votes: + 1 / - 0
Kunde: Funktionalität	
Nikolina Blagojevic	Votes: + 1 / - 0

Entwickler: Lesbarkeit

Lukas Pfeifhofer

Votes: + 1 / - 0

Entwickler: strukturierte Codebasis

Christoph Michael Weber

Votes: + 1 / - 0

Betreiber: Gute Konfigurierbarkeit

Niklas Natter

Votes: + 0 / - 0

Betreiber: Effizienz

Christina Greil

Votes: + 1 / - 1

Entwickler/Betreiber: Hintergrundwissen

Martin Kamleithner

Votes: + 3 / - 3

Enwickler: Wartbarkeit, Wiedervewendbarkeit

Christoph Wasylewski

Votes: + 0 / - 0

[Kunde: Sicherheit]

[Betreiber: effizienter]

Stefan Reinsperger

Votes: + 0 / - 0

Kunde: Aktuelle Daten

Muhammad Mujahed Hainoun

Votes: + 0 / - 0

Betreiber ressourcen verwaltung

Matthias Schrabauer

Votes: + 0 / - 0



## Kunde Entwickler Betreiber Umrechnungskurs

Gregor Steiner	Votes: + 0 / - 0
Kunde: Eindeutige Bedienung	
Michael Koppmann	Votes: + 0 / - 0
Entwickler: Möglichst klare Spezifikation	
Stefan Reinsperger	Votes: + 0 / - 0
Kun:Keine Durchmischung zwischen Online/Stationär	
Christoph Hafner	Votes: + 2 / - 2
Entwickler: Wartbarkeit/Erweiterbarkeit	
Judith Pitzer	Votes: + 0 / - 0
Betreiber: Leistbarkeit	
Paul Stelzhammer	Votes: + 0 / - 0
Entwickler: Berufserfahrung	
David Stemberger	Votes: + 0 / - 1
Betreiber: Erreichbarkeit	
Bernadette Maria Obermair	Votes: + 0 / - 1
Kunde: Echtzeitkurse	
David Schröder	Votes: + 2 / - 6
Betreiber: Verfügbarkeit	

Patrick Fritz Johann Raser

Votes: + 0 / - 4

---

Entwickler: Wartbarkeit

Niklas Natter

Votes: + 0 / - 7

---

Betreiber: Verfügbarkeit

Christoph Hafner

Votes: + 0 / - 8

---

Betreiber: Sicherheit

Michal Palczynski

Votes: + 0 / - 9

---

Kunde: Verfügbarkeit

Christoph Hafner

Votes: + 0 / - 15

---

Kunde: Sicherheit

# Welche QS Maßnahmen wären geeignet, um die wichtigen Qualitäten im Beispiel Web-Bank abzusichern?

Sehen Sie die bereits gegebenen Antworten durch und fügen Sie bei Bedarf neue QS-Maßnahmen hinzu. Bewerten Sie Maßnahmen welche sie für geeignet halten positiv.

Christoph Wasylewski	Votes: + 15 / - 0
----------------------	-------------------

Sicherheitsstandards umsetzen (Sicherheit)	
--	--

Lukas Pfeifhofer	Votes: + 13 / - 1
------------------	-------------------

Penetration Tests	
-------------------	--

Christoph Wasylewski	Votes: + 11 / - 0
----------------------	-------------------

Usability-Tests	
-----------------	--

David Stemberger	Votes: + 5 / - 0
------------------	------------------

Sicherheit: 2-Factor-Auth, Session Duration	
---	--

Sebastian Luzian	Votes: + 5 / - 0
------------------	------------------

Stresstest (Belastbarkeit des Systems testen)	
---	--

Martin Kamleithner	Votes: + 4 / - 0
--------------------	------------------

Automatisierte	
----------------	--

Julian Waibel	Votes: + 4 / - 0
---------------	------------------

Uptime-Statistik des Service	
------------------------------	--

Michael Koppmann	Votes: + 4 / - 0
------------------	------------------

Externer Code Audit	
---------------------	--

David Stemberger	Votes: + 3 / - 0
Backups	
David Stemberger	Votes: + 2 / - 0
Code-Reviews	
Daniel Tramberger	Votes: + 2 / - 0
Usability-Tests	
David Schröder	Votes: + 1 / - 0
Korrektheit - > Integration-Tests	
Gregor Steiner	Votes: + 1 / - 0
Software am aktellen Stand halten um 0-Day Exploits vorzubeugen	
David Stemberger	Votes: + 1 / - 0
Schulung Mitarbeiter	
Michael Pürmayr	Votes: + 1 / - 0
Performance test	
Christoph Hafner	Votes: + 1 / - 0
Usability: Erarbeiten von Anwendungsszenarien anhand von Personas	
Michal Palczynski	Votes: + 1 / - 0
Konkurrenz beobachten	
Muhammad Mujahed Hainoun	Votes: + 0 / - 0

## Resources usage

# Welche Vorgehensmodelle haben Sie selbst schon in einem Software Projekt erlebt?

V-Modell	50%
Spiralmodell	12.5%
V-Modell XT	18.8%
Agiles Modell	100%
Andere	6.3%