

Software- Qualitätsmanagement

**Vorlesung im Modul 10-202-2319
Software-Management**

Sommersemester 2010

Prof. Dr. Hans-Gert Gräbe

<http://bis.informatik.uni-leipzig.de/HansGertGraebe>

Prinzip des Primats der Qualität

- Alle Prozesse müssen Qualitätsprozesse sein
- An die Prozesse gestellte Anforderungen müssen 100%ig erfüllt werden – keinerlei Kompromisse
- Jeder Mitarbeiter soll seine Arbeit sofort beim ersten Mal und jedes Mal erneut richtig tun
- Qualitätsverbesserung durch Verbesserung der Entwicklungsprozesse
- Vermeidung von Nacharbeit und Verschwendung

Problem:

- Praktisch hat der störungsfreie Ablauf der Entwicklung Vorrang vor grundlegenden Verbesserungsvorschlägen
- Gefahr der Verschleppung von Fehlern
- Fehler werden eher symptomatisch bekämpft

Prinzip der Zuständigkeit aller Mitarbeiter

- Alle an der Erstellung und Vermarktung eines Produkts beteiligte Mitarbeiter müssen für dessen Qualität sorgen
- Jede Führungskraft muss es ihren Mitarbeitern ermöglichen, keine oder weniger Fehler zu machen
- Alle Prozesse eines Unternehmens müssen unter Qualitätsgesichtspunkten „gemanagt“ werden
- unabhängige QS-Abteilung ist überflüssig

Prinzip der ständigen Verbesserung (Kaizen)

- Managementprinzip, das auf Verbesserungen durch kleine, aber kontinuierliche Schritte setzt statt auf große Innovationsschübe
- Motto: „Jeder Tag bringt eine konkrete Verbesserung im Unternehmen.“
- Führungsstil setzt auf langfristige Perspektiven und Verhaltensänderungen statt auf kurzfristige Ergebnisse (Leistung, Kontrolle)
- Gewachsene soziale Strukturen eines Unternehmens sollen genutzt und entwickelt, und nicht missachtet werden.
 - Betroffene einbeziehen, Teamarbeit, ständiges Lernen, offenes Klima

Prinzip der Kundenorientierung

Primäres Ziel: Erfüllung der Kundenanforderungen

- Kundennutzen und Kundenzufriedenheit

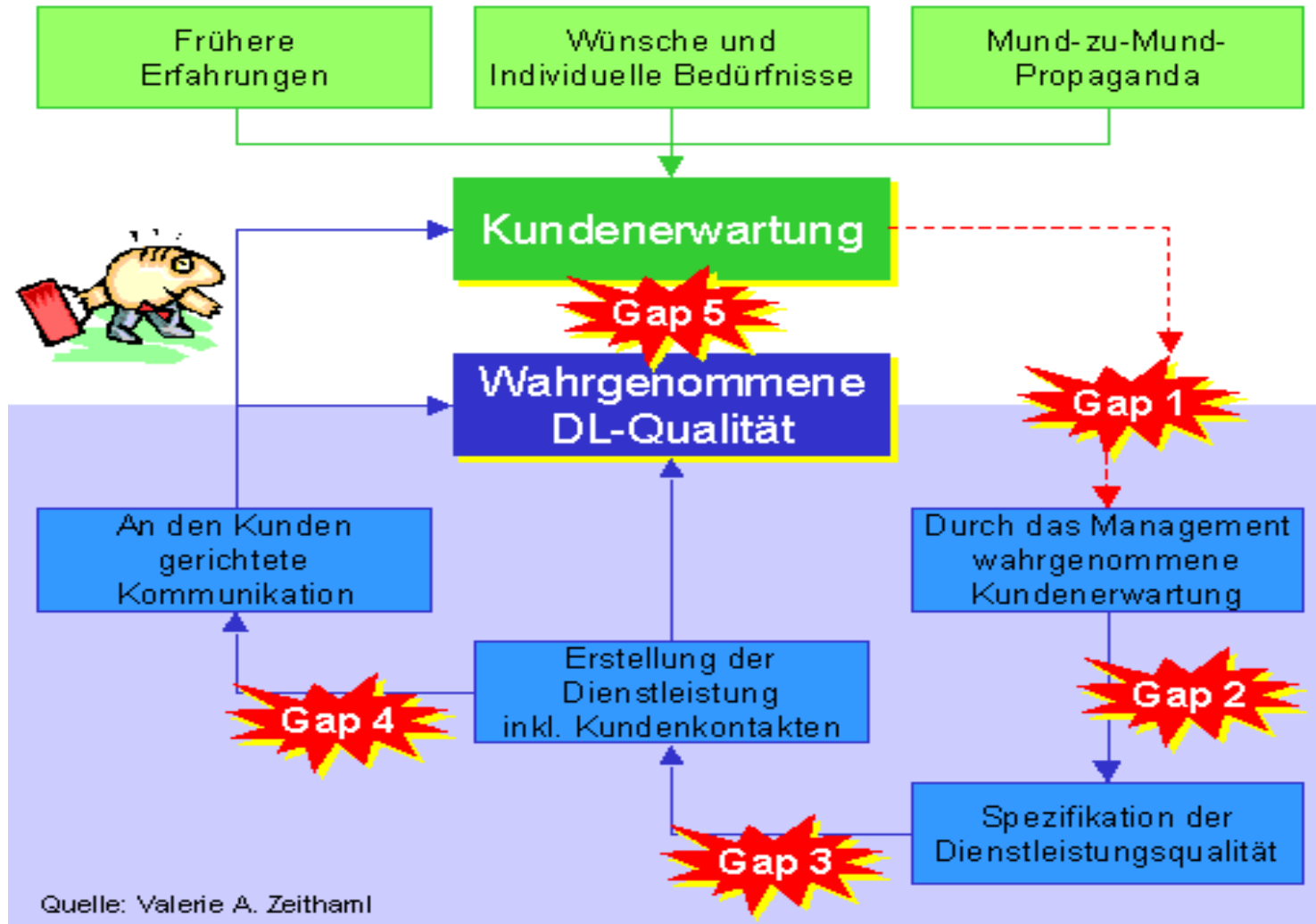
Enge Zusammenarbeit zwischen Entwicklung, Marketing und Kundendienst

- Individualsoftware: Kunde soll bei der Formulierung seiner Bedürfnisse unterstützt werden
- Standardsoftware: die Bedürfnisprofile der Hauptzielgruppen werden durch intensive Marktanalysen ermittelt

Problem:

- Adäquate Wahrnehmung der Kundenwünsche durch das Unternehmen (relevant. GAP-Modell)

Das Gap-Modell



http://www.im-group.ch/t_module/qs/gap-mod.htm#top

Prinzip des internen Kunden-Lieferanten-Verhältnis

- Mitarbeiter-Integration in der internen Prozesskette:
 - Mitarbeiter gilt als **Kunde** vom Vorgänger-Prozess
 - Mitarbeiter gilt als **Lieferant** für den Folge-Prozess
- Auch interne Leistungen werden formell abgenommen und übergeben (wie bei externen Leistungen).
- Erfolg des Teams wird gemessen an der Zufriedenheit seiner internen/externen Kunden
 - Qualität orientiert am Erfolg des nächsten Teams in der Wertschöpfungskette
 - lokale Verantwortung für Qualität

Prinzip der Prozessorientierung

- Fehler werden primär als Defizite des Entwicklungsprozesses angesehen
 - Fehlervermeidung vor Fehlerbehebung
 - Produktprüfung zur Prozessüberwachung
- Nicht Fehlersuche, sondern Fehlerursachensuche
- Software-Erstellung als reproduzierbarer und verbesserungsfähiger Prozess

Maßnahmen und Konzepte

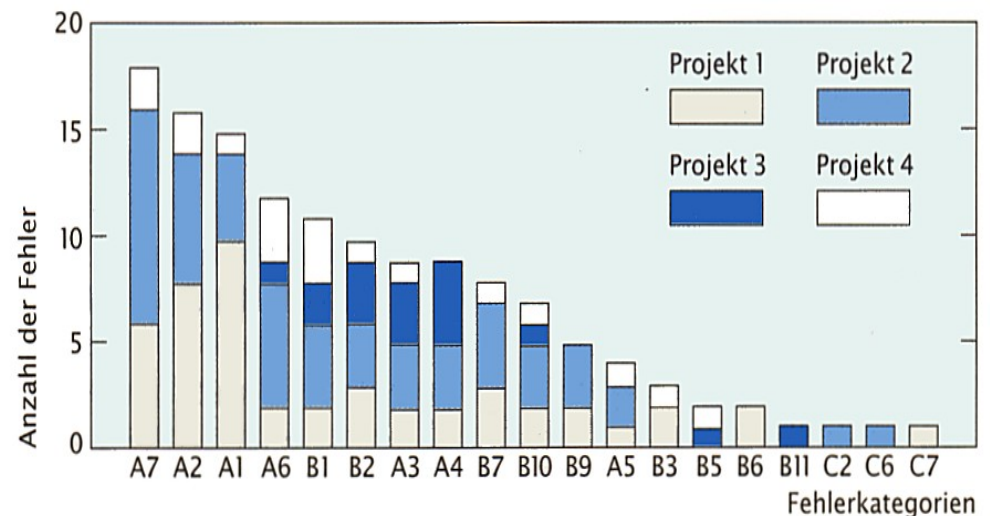
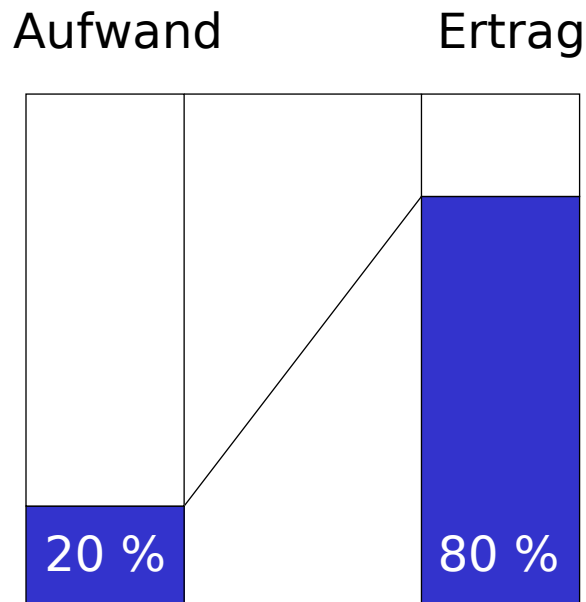
- Wichtige Maßnahmen zur Realisierung von TQM
 - Klar formulierte Qualitätspolitik und nachvollziehbare Q-Ziele
 - Festlegung und Bekanntgabe der Kompetenzen, Befugnisse und Verantwortungen zur Durchführung und Durchsetzung der Q-Politik
 - Einführung eines QM-Systems mit ausreichender analytischer, dokumentarischer und verändernder Reichweite
 - Konsequente Schulung aller Mitarbeiter in Sachen Qualität und Qualitätsmanagement
- Typische Konzepte des TQM
 - **Qualitätszirkel**
 - Entfaltung der Qualitätsfunktionen
(***Quality Function Deployment, QFD***)

Qualitätszirkel

- Regelmäßige Team-Treffen von Mitarbeitern im kleinen Kreis, um auftretende Qualitätsprobleme im gemeinsamen Arbeitsbereich zu lösen oder aktiv Verbesserungen zu ermitteln.
 - etwa 1h pro Woche innerhalb der Arbeitszeit
 - Verbesserungen: Einführung und Erfolgskontrolle vom Team selbst (nach Genehmigung)
 - wichtig ist Einbeziehung der und Unterstützung durch die Geschäftsführung
- Hilfsmittel zur Realisierung:
 - *Brainstorming*
 - Pareto-Analyse
 - Ursache-Wirkungs-Diagramme (*Fishbone Chart*, Ishikawa-Diagramm)

Pareto-Analyse/Prinzip (80:20-Regel)

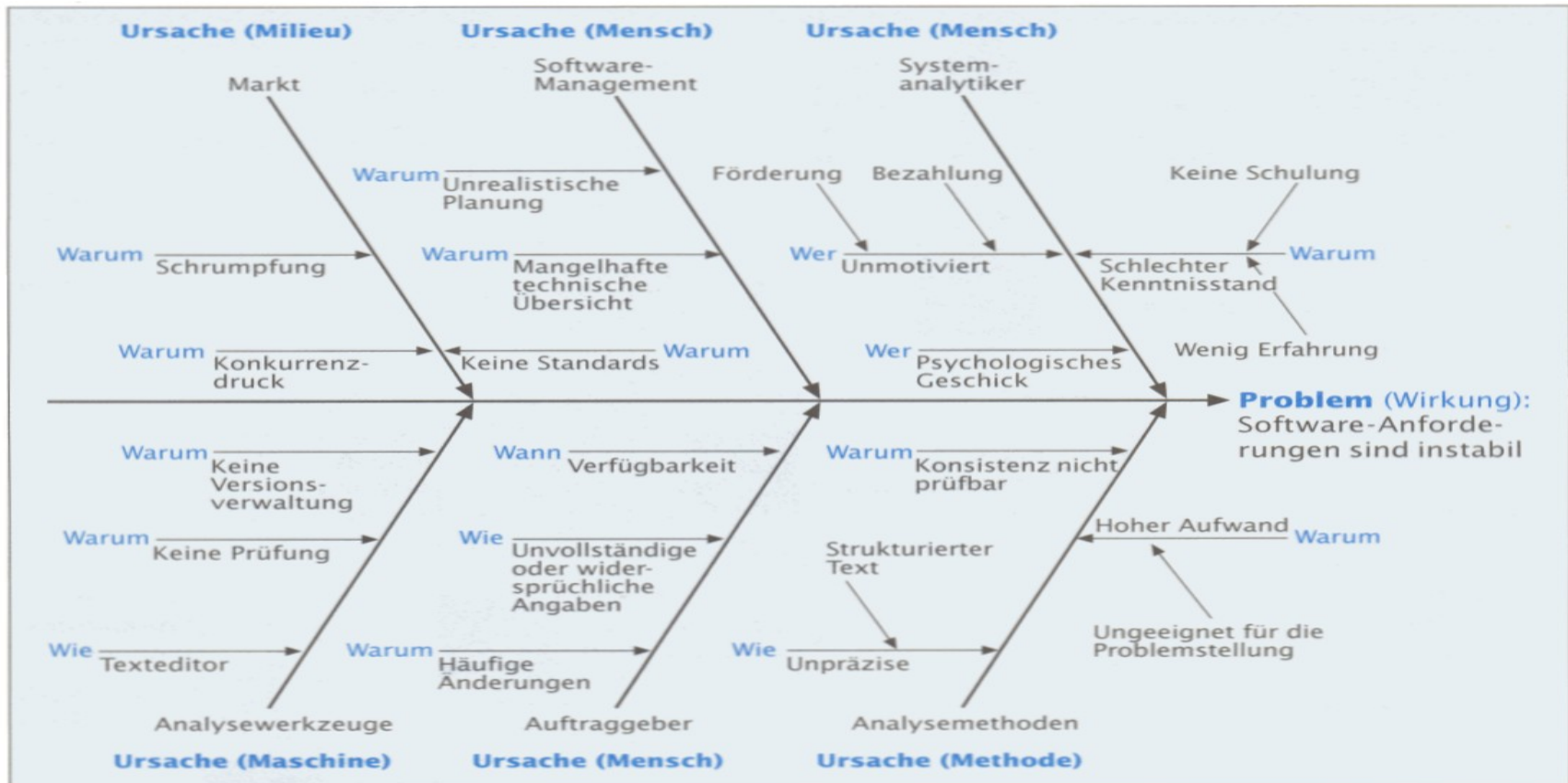
- 80 % des Aufwands geht in die Lösung von 20 % der Probleme oder anders:
- 80 % der Probleme werden mit 20 % des Aufwands gelöst



Pareto-Analyse von SW-Fehlern der Firma HP [Grady, Caswell 87]

Ursache-Wirkungs-Diagramm

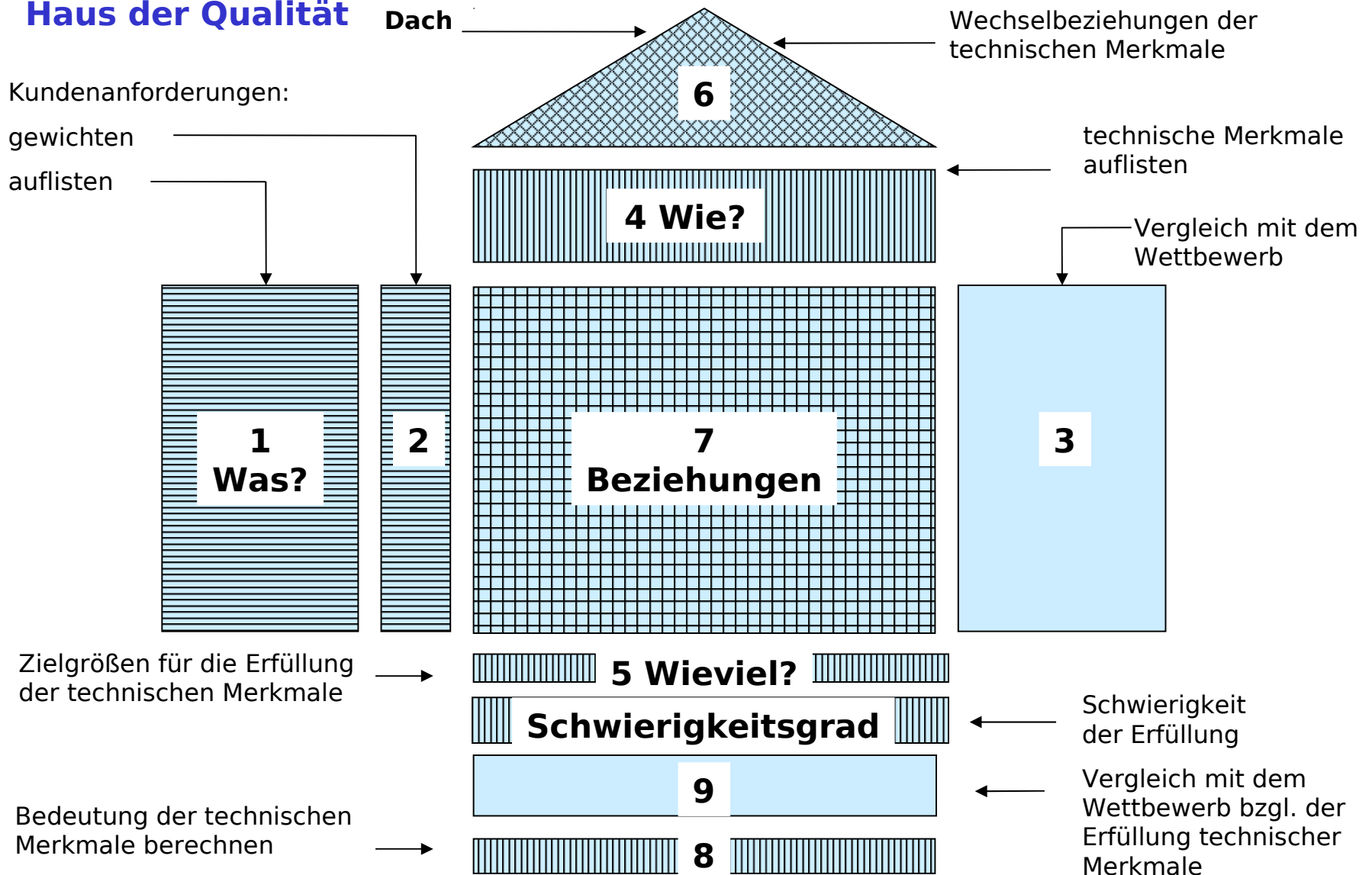
- Zu einem Problem werden die Hauptursachen gesucht und diese werden immer feiner aufgeteilt.



Entfaltung der Qualitätsfunktionen (Quality Function Deployment, QFD)

- Ausgehend von den Kundenwünschen werden mit Hilfe von Matrizen systematisch Produkteigenschaften abgeleitet, die dann zu einer Komponenten-, Prozess- und Produktionsplanung führen („Haus der Qualität“)
- Matrix dient der Korrelationsanalyse zwischen Kundenwünschen (Was) und deren funktionaler Realisierung (Wie)
 - Kundenanforderungen auflisten und gewichten
 - technische Merkmale mit Zielgrößen und Schwierigkeiten definieren
 - Abhängigkeiten zwischen technischen Merkmalen bestimmen
 - Beziehungsmatrix Kundenanforderungen -- Merkmale aufstellen (Beziehungsfaktoren)
 - Bewertung der technischen Merkmale nach lokaler Priorität

Haus der Qualität



QFD: Bewertung

- Entwicklung erfolgt auf der Basis von Kundenanforderungen
- Übersicht über kritische Punkte und Zielkonflikte bei der Entwicklung
- Bereitstellung rationaler und transparenter Entscheidungsgrundlagen
- Entwicklung klarer Vorgaben für die Software-Prozessgestaltung
- Verfolgung der Umsetzung von Kundenanforderungen über alle Entwicklungsphasen
- Ableitung von Zielgrößen für die Entwicklung und Qualitätssicherung aus den Kundenanforderungen.

Voraussetzungen:

- TQM-Ansatz im Unternehmen realisiert
- Vorhandensein von Messdaten über Kundenanforderungen

TQM-Ansatz: Vor- und Nachteile

Vorteile

- Umfassender Ansatz, der das ganze Unternehmen auf Qualität ausrichtet
 - Sicht auf das Unternehmen als sozio-technisches System
- Umfassende, nicht nur funktionale Sicht auf Qualität (Beratung, Service, Einhaltung von Normen und Auflagen)
 - Kundenzufriedenheit im Mittelpunkt
 - erfordert ganzheitliches Denken und Handeln
- Qualitätsverbesserung ist Unternehmensziel

Nachteile

- nicht so konkret fassbar wie ISO 9001 oder ISO 15507
- setzt spezielle Unternehmenskultur voraus, nicht für externe Bewertung geeignet
- keine Trennung zwischen Management- und Qualitätsbegriff

Einführung

- Bisherige Ansätze: Schrittweise Verbesserung der Qualität bestehender Geschäftsprozesse
 - Stufenmodell von CMM oder ISO 15504
 - Für bestehende Organisationen mit geringem Prozessinnovationsbedarf geeignet
- Anderer Ansatz: Ingenieurmäßige Modellierung von Geschäftsprozessen ähnlich der Modellierung von Software
 - Für neu zu gründende oder bestehende Organisationen mit hohem Prozessinnovationsbedarf
 - Nicht nur die SW-Unterstützung, sondern auch die Geschäftsprozesse werden grundlegend neu überdacht und *gestaltet*
 - Auch als *Business Engineering* oder *Business (Process) Reengineering* bezeichnet
 - Erlaubt die integrale Berücksichtigung der Möglichkeiten moderner IuK-Technik und -Infrastruktur.

	Evolutionär	Revolutionär
Änderungen	Inkrementell	Radikal
Ausgangspunkt	Vorhandene Prozesse	Neuanfang
Änderungshäufigkeit	Einmalig/kontinuierlich	Einmalig
Benötigte Zeit	Kurzfristig	Langfristig
Beteiligte	Mitarbeiter	Geschäftsführung, Mitarbeiter
Typischer Geltungsbereich	Eng, innerhalb von Funktionen	Breit, über Funktionen hinweg
Risiko	Gering	Hoch
Ermöglicht durch ...	Statistische Kontrolle	Informationstechnik
Art der Änderung	Kulturell	Kulturell/strukturell

Business Engineering und SW-Engineering

- Bezug zum Thema **Unternehmensmodellierung**
 - Dort Schwerpunkt auf Einsatz von IuK-Mitteln zur Modellierung von Geschäftsprozessen in „klassischen“ produzierenden Unternehmen
 - Enthält SW-Modellierung als integrales Element; Software **unterstützt** Unternehmensprozesse
- Komplexere Frage der Qualitätssicherung
 - QS auf der Ebene der Geschäftsprozessmodellierung
 - QS auf der Ebene der Geschäftsprozess-Laufzeit
 - QS auf der Ebene der IT-Unterstützung
- Frage der Verzahnung von Entwicklungsmodellen auf GP-Ebene und IT-Ebene
 - GP-Modelle sind in der Regel inkrementell und iterativ

Business Engineering in SW-Organisationen

Weiteres Thema: Anwendung des Ansatzes der Unternehmensmodellierung auf SW-Organisationen

- Wird typischerweise eingesetzt, wenn höhere Stufen (3 bis 5) von CMMI oder ISO 15504 erreicht werden sollen
- Anwendung auf SW-Organisationen steht vor demselben Problem wie die Qualitätssicherung: Abläufe sind keine typischen Produktionsabläufe, sondern eher mit Projektierungsarbeiten in Ingenieurbüros zu vergleichen.
- **Frage:** Wie weit sind klassische Prinzipien der Unternehmensmodellierung im SW-Bereich überhaupt anwendbar?

Allgemeine Grundsätze für die Unternehmensmodellierung und deren Voraussetzungen in SW-Organisationen

Zusammenfassung mehrerer Positionen

- **Ziel:** Geschäftsprozesse sollten in einer oder wenigen Einheiten konzentriert werden, um Übernahmeprobleme zu vermeiden und Verwaltungsgemeinkosten zu senken
- **SW:** Meist erfüllt, da Mitarbeiter oft von der Systemanalyse bis zur Produktabnahme im Projekt arbeiten.

Mitarbeiter treffen selbstständig Entscheidungen

- **Ziel:** Entscheidungen werden „vor Ort“ gefällt (vertikal komprimierte Hierarchien), wenn entsprechende Kompetenz vorhanden ist
- **SW:** Durch hohes Qualifizierungsniveau der Ingenieure erfüllt.

Prozessschritte in natürlicher Reihenfolge bringen

- **Ziel:** Entlinearisierung und Reduktion auf tatsächlich bestehende Abhängigkeiten, Flexibilisierung, ermöglicht paralleles Ausführen von Prozessen
- **SW:** Wird in verschiedenen Vorgehensmodellen der SWT berücksichtigt.

Mehrere Prozessvarianten

- **Ziel:** Skalierbarkeit je nach Anforderungssituation
 - Je nach Anforderung kann ein Prozess in verschiedenen Varianten zur Ausführung kommen
 - Bsp.: 100 Euro-Bestellungen anders als 10000 Euro-Bestellungen
- **SW:** Wird im Reifegrad 5 vorausgesetzt. Spiralmodell als Metamodell zur Auswahl geeigneter Vorgehensmodelle

Arbeit wird dort erledigt, wo es am sinnvollsten ist.

- **Ziel:** Neuverteilung der Arbeit längs Prozessgrenzen. Aufbrechen organisatorischer Grenzen. Einbeziehung des Prozesskunden.
- **SW:** Widerspricht teilweise dem Prinzip der unabhängigen Qualitätssicherung (Prinzip war aber selbst widersprüchlich.)

Weniger Überwachungs- und Kontrollbedarf.

- **Ziel:** Reduzierung von Prüfung auf wirtschaftlich sinnvolles Maß.
 - Pauschale und nachträgliche Prüfungen statt starrem Kontrollsystem.
- **SW:** Teilweise umgekehrter Trend: V-Modell und ISO 9000

Reduzierung der Abstimmungsarbeiten auf ein Minimum.

- **Ziel:** Weniger Kontaktpunkte im Unternehmensprozess
 - Voraussetzung: Reduktion der Variabilität, welche durch externe Kontaktpunkte erzeugt wird.
- **SW:** Eher gegenläufiger Trend durch starke Einbeziehung des Kunden.

„Casemanager“ als einzige Anlaufstelle

- **Ziel:** Puffer zwischen dem Geschäftsprozess und dem Kunden.
 - Prozessadäquate Aufbereitung der Kundenanforderungen.
- **SW:** Durch Einbeziehung des Kunden in Produktentwicklung eher nicht einzuhalten.

Mischung aus Zentralisierung und Dezentralisierung

- **Ziel:** Individuelle Geschäftseinheiten arbeiten autonom, nutzen aber zentralisierte Dienste.
 - Deutliche Unterscheidung zwischen (lokalen) Geschäftseinheiten und (globaler, unternehmensweiter) Infrastruktur.
- **SW:** Typischer IuK-getriggelter Zugang, der in SWO bereits intensiv eingesetzt wird.
 - Unternehmensmanagement vs. Projektmanagement.
 - Referenzmodell von ISO 15504.

Business Engineering am Beispiel des Rational Unified Process (RUP)

Objektorientiertes Vorgehensmodell zur SW-Entwicklung, das die Ebene der GP-Modellierung einschließt.

- UML basiertes Framework, um Entwicklungsprozesse zu beschreiben
- Unterstützt modellzentrierten, komponentenbasierten, konfigurierbaren iterativen Software-Entwicklungsprozess
- Breite Sammlung von Beispielen (best practises), regelmäßige Nutzertreffen und Konferenzen

Entsprechende Werkzeuge, Erfahrungen, Handreichungen, Schulungen etc. werden als kommerzielles Produkt von der **Firma Rational Software** (<http://www.rational.com>) angeboten, die seit 2003 zum IBM-Konzern gehört.

- Interessantes Geschäftsmodell: Kommerzielle Firma kümmert sich um Reproduktion der Kompetenz
- Teil der IBM-Strategie „Technologieführerschaft“

Übersicht über wichtige RUP Werkzeuge und deren Anwendungsbereiche:

- *Rational ClearCase* und *Rational ClearQuest*: Produkte zum Konfigurationsmanagement (SCM) und zur Fehlerverfolgung
- *Rational Rose*, ein grafisches, auf UML basierendes Software Design Werkzeug (CASE)
- *Rational Application Developer* (RAD), Java IDE, ehemals Websphere Studio Application Developer (WSAD)
- *Rational Portfolio Manager* (RPM), Projektmanagement Tool
- *Jazz*, Tool zur Zusammenarbeit in Teams zur Softwareentwicklung
- *Rational Team Concert*
- *Rational Software Architect*, UML-Modellierungstool
- *Rational Business Developer*, EGL Entwicklungsumgebung

IBM bietet eine Reihe von Anpassungen des RUP an Entwicklungsaufgaben in verschiedenen Architekturkonzepten.

Im Mittelpunkt von RUP als Vorgehensmodell stehen 6 zentrale Erfahrungen, die von erfolgreichen Organisationen in der industriellen SW-Entwicklung eingesetzt werden:

- Software iterativ entwickeln
 - Erlaubt wachsendes Verständnis der Probleme durch erfolgreiche Verfeinerungen bis hin zu einer effizienten Lösung mit Fokus auf die höchsten Risiken in der jeweiligen Entwicklungsstufe
 - Häufige ausführbare Versionen, um den Endnutzer kontinuierlich einzubeziehen und Rückkopplungen zu initiieren
 - Hohe Agilität und Flexibilität im Bereich taktischer Änderungen
 - Fokus auf Ergebnissen und Soll/Ist-Vergleichen des Fahrplans
- Anforderungsmanagement
 - Klare Regeln, nach denen Anforderungen und Randbedingungen erhoben, organisiert und dokumentiert werden
 - Anwendungsfälle und -szenarien treiben Entwurf, Implementierung und Test der Software

- Komponentenbasierte Architekturen verwenden
 - Frühe Entwicklung und Fundierung einer robusten ausführbaren Architektur, ohne bereits Ressourcen für einen vollständigen Entwicklungsprozess bereitzustellen
 - Architekturentwicklung als systematischer Prozess der Erstellung neuer und Verwendung vorhandener Komponenten, die nach einem wohldefinierten Verfahren komponiert werden oder in eine Komponenten-Infrastruktur (CORBA, Webserver, .NET, J2EE, EJB usw.) eingebaut werden.
 - Komponente = nicht trivialer Modul oder Subsystem, das eine klar abgrenzbare Funktion erfüllt
 - Entscheidung über weitere Entwicklungsstufen oder Abbruch des Projekts auf der Basis verschieden detaillierter Prototypen
- Software visuell modellieren
 - Visuelle Modellierung ist gut geeignet, um Komplexität zu reduzieren und die erforderlichen Abstraktionen (Modellierung im Großen) zu spezifizieren. Einsatz von UML als zentraler Visualisierungsstandard.
 - Verfeinerungen werden erst vorgenommen, wenn sich die Konzepte „im Großen“ bewährt haben. Umsetzung des Prinzip der frühen Entdeckung von Entwurfsfehlern

- Softwarequalität sichern
 - Fokus auf
 - Funktionalität entsprechend der Anforderungen,
 - Zuverlässigkeit,
 - Leistungsfähigkeit
- als Grundlagen für die Akzeptanz von Software.
- RUP bietet Unterstützung bei Planung, Entwurf, Umsetzung, Ausführung, Auswertung solcher Tests
- Qualitätsbewertung ist im ganzen Prozess, in allen Aktivitäten und bei allen Rollen integriert, verwendet objektivierte Maße und Kriterien (Prinzip der entwicklungs-integrierten Qualitätssicherung)
- Änderungsmanagement
 - Änderungen sind ein zentrales Element von Software, die sich ändernde Geschäftsprozesse unterstützen soll
 - Änderungen müssen konsistent, verfolgbar, in ihren Auswirkungen überschaubar und akzeptiert sein
 - RUP legt fest, wie Änderungen kontrolliert, verfolgt, beobachtet werden, um so ein erfolgreiches interaktives Entwicklungsszenario umzusetzen.

RUP - Dimensionen

Horizontale Dimension „Zeit“: Der Gesamtlebenszyklus der Software wird in Iterationszyklen aufgebrochen, in denen jeweils eine neue Generation des Produkts entsteht

- Ein *Zyklus* wird in vier aufeinander folgende *Phasen* unterteilt: **Konzeptionsphase** (inception), **Entwurfsphase** (elaboration), **Konstruktionsphase** (construction), **Übergabephase** (transition)
- Jede Phase endet mit einem *Meilenstein*, zu dem gewisse *missionskritische Entscheidungen* getroffen werden. Dazu müssen grundlegende Ziele umgesetzt sein.

Dazu werden

- **Beteiligte** (auf der Ebene von Rollen) - „Wer“
- **Aktivitäten** - „Wie“
- **Artefakte** - „Was“ und
- **Arbeitsschritte** (Workflows) - „Wann“

unterschieden.

Vertikale Dimension “Abstraktion”:

Unterscheidung in Kernarbeitsschritte

- Geschäftsprozessmodellierung
- Anforderungsanalyse
- Analyse und Design
- Implementierung
- Test
- Auslieferung

und unterstützende Arbeitsschritte

- Konfigurations- und Änderungsmanagement
- Projektmanagement
- Infrastruktur

Konzeptionsphase

Ziel: Geschäftsfall abgrenzen, Erfolgskriterien festlegen und alle externen Größen und Akteure sowie deren Interaktionsbedürfnisse identifizieren. Identifizierung aller Anwendungsfälle und prototypische Beschreibung einiger wichtiger.

Ergebnisse:

- Visionäres Dokument zu den allgemeinen Verständnis über die grundlegenden Anforderungen, Eigenschaften und Randbedingungen
- Initiales Anwendungsfälle und Projekt-Glossar (Anwendungsfall-Ebene)
- Initialer Geschäftsfall mit Kontext, Erfolgskriterien und finanzieller Vorhersage, evtl. Geschäftsmodell (Geschäftsprozess-Ebene)
- Initiale Risikobewertung
- Projektplan mit Phasen und Iterationen (Projektmanagement-Ebene)
- Ein oder mehrere Prototypen (Fokus auf ausführbaren Ergebnissen)

Entwurfsphase

Ziel: Analyse des Problembereichs, Fundierung der Architektur und Elimination der höchsten Risiken für das Projekt. Entwicklung einer „Meilen weiten und Zoll tiefen“ Sicht auf das System. Auf der Basis wird entschieden, ob der Übergang zur nächsten Phase, die vollen Ressourceneinsatz erfordert, sinnvoll ist

Ergebnisse:

- Software-Architektur-Beschreibung, die Anwendungsbereich, grundlegende Funktionalität und nichtfunktionale Anforderungen, insbesondere Leistungsparameter erfasst
- Ausführbarer Architektur-Prototyp
- Alle wesentlichen Anwendungsfälle sind mit ihren Akteuren erfasst
- Geschäftsfälle und Risikoanalyse sind überarbeitet
- Grob-granularer Projektplan mit Iterationen und Erfolgskriterien für die einzelnen Iterationen
- Vorläufiges Nutzerhandbuch

Konstruktionsphase

Ziel: Alle (weiteren) Komponenten und Anwendungsmerkmale werden entwickelt, integriert und umfassend getestet. Wird als *Herstellungsprozess* ausgerollt, in dem Ressourcenverwaltung, Kostenoptimierung, Zeit- und Qualitätspläne im Vordergrund stehen.

Übergang vom Konzept-Fokus auf den Produkt-Fokus

Ergebnisse:

- Produkt, das den Endnutzern übergeben werden kann, mindestens
 - Das in einer adäquaten Plattform integrierte Softwareprodukt
 - Das Nutzerhandbuch
 - Eine Beschreibung der aktuellen Version der Software
- Im Meilenstein ist zu entscheiden, ob
 - Das Produkt in die Nutzergemeinde ausgerollt werden kann
 - Die beteiligten Partner (stakeholders) für die Übergabe bereit sind
 - Die Kostenrahmen eingehalten wurden und werden

Übergabephase

Ziel: Ausrollen der aktuellen Iterationsstufe in der Nutzergemeinde, Ausgangspunkt eines weiteren Entwicklungszyklus. Oft überlappen sich die Konzeptionsphase des nächsten Zyklus mit der Übergabephase des aktuellen. Diese Phase besteht typischerweise selbst aus mehreren Iterationen. Rücklaufinformationen werden primär für das Tuning der aktuellen Version verwendet.

Voraussetzungen:

- Betatest hat gezeigt, dass das Produkt die Nutzererwartungen erfüllt
- Konzept zu parallelem Einsatz mit Vorgängerversion
- Konversionskonzept für die operationalen Datenbestände
- Training der Nutzer und Systemverwalter
- Roll-out des Produkts in den Bereichen Marketing, Vertrieb und Service

Ergebnisse:

- Nutzer haben das Produkt akzeptiert und können sich weitgehend selbst helfen
- Die beteiligten Partner sind davon überzeugt, dass die mit der Iteration angestrebte Basislinie vollständig und konsistent mit den Bewertungskriterien der Vision umgesetzt wurde
- Die finale Basislinie der weiteren Iterationen ist in den geplanten Zeit- und Kostenhorizonten weiterhin erreichbar.

Vergleich zum Wasserfall-Ansatz

Im Vergleich zum Wasserfallansatz hat ein solches iteratives Vorgehen grundsätzliche Vorteile:

- Risiken werden deutlich eher aufgedeckt und abgefangen (Prinzip der frühzeitigen Fehlerentdeckung)
- Änderungen sind deutlich besser zu verwalten. Damit wird Flexibilität und Anpassungsfähigkeit deutlich gestärkt.
- Entwicklungsbausteine können auf hohem Abstraktionsniveau wiederverwendet werden.
- Das Projektteam lernt deutlich intensiver
- Es wird eine deutlich bessere Gesamtqualität erreicht.

Abstimmung von Geschäfts- und IT-Strategie

RUP ist ein erster Ansatz, um Geschäfts- und IT-Strategien miteinander abzustimmen. Eine solche Abstimmung ist ein generelles Problem

- Horizonte beider Strategien sind verschieden
- Quelle der Herausforderungen primär extern / primär intern
- Problem der Überdimensionierung von IT-Systemen
- Problem der geringen Flexibilität von IT-Systemen

Diese fehlende Abstimmung wurde in den letzten Jahren als das „Problem Nummer 1 des Managementbereichs“ identifiziert. Nimmt zu mit der wachsenden Kritikalität einer qualitativ hochwertigen und verfügbaren IT für den geschäftlichen Erfolg.

Ziel einer solchen Abstimmung ist die Anwendung der IT in angemessener und zeitlich konsistenter Weise, so dass die geschäftlichen Bedürfnisse, Ziele und Strategien harmonisch unterstützt werden.

Abstimmung von Geschäfts- und IT-Strategie

Bedeutung der Abstimmung:

- Maximiert den Return von IT-Investitionen
- Erlaubt es, Wettbewerbsvorteile durch adäquate IT-Unterstützung zu erlangen
- Größere Flexibilität bei der Reaktion auf sich ändernde externe Anforderungen

Wichtige Erfolgsfaktoren:

- Klare Organisationsziele
- Verständnis für die Relation zwischen Prioritäten der Geschäftsziele und der IT-Ziele
- Schließen der Lücke zwischen Geschäfts- und IT-Strategie
- Kommunikation zwischen IT-Mitarbeitern und dem mittleren Management
- Kontrolle, Vertrauen, Verständnis gegenüber der IT-Abteilung

Lösungsansatz: Information Technology Service Management

Ansatz: Umformung der Geschäftsstrategie in eine Serie von aufeinander bezogener Dienstleistungen (Service-Orientierung)

- Dies entspricht der Umsetzung von CMM Stufe 3 auf der Ebene der Geschäftsprozesse.
- Führt zu einer weiteren *Integration des Qualitätsaspekts* in die unmittelbaren Geschäftsprozesse (Service Level Agreements)
- Fokuswechsel auf hochqualitative Produkte und Dienste für Kunden zu geringen Kosten (ähnlich TQM)

Brücke zwischen GP und IT wird in der Führung der IT-Abteilung als spezieller Geschäftsfunktion gesehen, die sich als Support-Komponente in die Ausführung entsprechender Services einordnet.

- “ITSM befasst sich mit der Erbringung und Unterstützung von IT-Diensten, die angemessen sind für die geschäftlichen Ziele der Organisation.“

Verdichtung der Erfahrungen zu einer **Information Technology Infrastructure Library – ITIL**

- Methodologie zur Verwaltung und Erbringung von IT-Diensten in einem Geschäftsumfeld
- Überdeckt alle Aspekte von Entwicklung über das Ausrollen, Verfügbarkeit, Verwaltung bis hin zu Systemen der Hilfestellung
- Menge von Best Practises für IT Infrastruktur-Management und IT-Dienste-Erbringung
- Serie von Büchern, in denen diese Erfahrungen verdichtet sind

Entwicklung eines **Strategic Alignment Model – SAM**

- Komplexerer Zugang der Abstimmung zwischen
 - Geschäftsstrategie
 - IT-Strategie
 - Entwicklung der Organisations-Infrastruktur und -Prozesse
 - Entwicklung der IT-Infrastruktur und -Prozesse

Die IT rückt dabei aus dem Rang einer Supportfunktion in den Rang einer kritischen Funktion