Dr. Georg Herzwurm, Prof. Dr. Werner Mellis, Sixten Schockert
Lehrstuhl für Wirtschaftsinformatik, Systementwicklung der Universität zu Köln
Albertus Magnus Platz
50923 Köln

Tel.: + 49 (0)221 / 470-5368
Fax: + 49 (0)221 / 470-5386

E-Mail: herzwurm@informatik.uni-koeln.de
WWW: http://www.informatik.uni-koeln.de/winfo/prof.mellis/welcome

Kundenorientierte Planung von Softwareprodukten und -prozessen mit Quality Function Deployment (QFD)

Zusammenfassung

Software-Produkte entsprechen häufig nicht den Erwartungen der Kunden. Die Japaner haben v. a. in der Automobilindustrie mit Hilfe des sogenannten Simultaneous Engineering infolge der hierdurch verbesserten Kundenorientierung bedeutende Marktvorteile erlangt. Beim Simultaneous Engineering werden, ausgehend von den Kundenwünschen, über eine interdisziplinäre Gemeinschaftsarbeit im Unternehmen Produktgestaltung und Herstellungsprozeß synchron gestaltet. QFD ist das Rückgrat der simultanen Produktplanung. Der Beitrag berichtet über ein Projekt zur Planung einer Adreßdatenbank, mit dem geprüft werden sollte, inwiefern das für die Fertigungsindustrie entwickelte QFD-Verfahren auch für Software-Produkte eingesetzt werden kann.

Inhaltsverzeichnis

1. Problemfelder der kundenorientierten Softwareherstellung 2

2. Quality Function Deployment als Instrument zur kundenorientierten Softwareherstellung 2

3. Fallbeispiel zu Software Quality Function Deployment 3

4. Erfahrungen mit Software Quality Function Deployment 8

5. Weitere Forschung zu Software Quality Function Deployment 8

Literatur 8

1. Problemfelder der kundenorientierten Softwareherstellung

Trotz erheblicher Fortschritte im Requirements Engineering klafft in vielen Softwareprodukten noch eine erhebliche Lücke zwischen den Kunden- bzw. Benutzeranforderungen[1] und den realisierten Produktmerkmalen. Informationen über nicht-funktionale Anforderungen, Prioritäten von Anforderungen, Anforderungen an den Softwarentwicklungs*prozeß* etc. liegen teilweise nicht explizit vor bzw. sind den Kunden nicht bewußt oder gehen im Verlaufe des Softwareentwicklungsprojektes verloren; am fertigen Produkt läßt sich z. B. nicht mehr nachvollziehen, welches Produktmerkmal auf welche Kundenanforderung zurückzuführen ist.

2. Quality Function Deployment als Instrument zur kundenorientierten Softwareherstellung

Die Japaner haben v. a. in der Automobilindustrie mit Hilfe des sogenannten Simultaneous Engineering, bei dem ausgehend von den Kundenwünschen über eine interdisziplinäre Gemeinschaftsarbeit im Unternehmen Produktgestaltung und Herstellungsprozeß synchron gestaltet wird, bedeutende Marktvorteile erlangt. Quality Function Deployment (QFD) ist das Rückgrat der simultanen Produktplanung. Aus ausgesprochenen, vorausgesetzten oder latenten Anforderungen und Wünschen der Kunden an das neue bzw. zu verbessernde Produkt werden schrittweise die Produkteigenschaften entwickelt. In der ersten Phase „Qualitätsplanung“ werden die Kundenanforderungen in technische Merkmale des Produktes übersetzt. Aus den Qualitätsmerkmalen des Produktes werden anschließend in der Phase „Teileplanung“ in mehreren Stufen die entsprechenden Merkmale der Produktkomponenten (z. B. Module) entwickelt. Diese Merkmale bilden in der Phase „Prozeßplanung“ wichtige Kriterien für die Festlegung kritischer Produkt- und Prozeßparameter (z. B. Prüfobjekte und -zeitpunkte), die in der letzten Phase „Fertigungsplanung“ die Ausgangsbasis für die detaillierte Festlegung des Entwicklungsprozesses sind.

Die Festlegung aller wichtigen technischen, wirtschaftlichen und marktstrategischen Eckdaten erfolgt durch ein abteilungsübergreifendes Expertenteam aus Marketing, Produktplanung, Produktentwicklung, Forschung, Qualitätsmanagement etc. Ein Hauptproblem beim Produktplanungsprozeß stellt die effektive und effiziente Kommunikation zwischen den verschiedenen Abteilungen dar. Hierfür hat sich die QFD-Methode in der industriellen Praxis bewährt. In der QFD-Matrix werden alle Planungsschritte zusammengestellt, die somit eine Art Übersichtsplan (Bewertungsprofile mit Darstellung von Wechselbeziehungen der einzelnen Anforderungen bzw. Lösungen; Vergleiche mit der Konkurrenz bzw. mit alternativen Lösungen) des Projektes ergeben. Abb. 2-1 zeigt das „House of Quality“ als Ergebnis der Phase Qualitätsplanung. Je nach Produktart kommen noch weitere Felder hinzu.

Erfahrungen in den U.S.A. und Japan zeigen, daß sich QFD grundsätzlich auch für Dienstleistungs- und Softwareprodukte eignet. Allerdings sind die Besonderheiten des Softwareerstellungsprozesses bei der QFD-Anwendung zu beachten und führen deshalb zu einem modifizierten Vorgehen gegenüber der o. a. klassischen Vorgehensweise. Ferner sind die Unterschiede zwischen Standard- und Individualsoftwareentwicklung zu berücksichtigen.

---

1 Die nachfolgenden Aussagen gelten grundsätzlich sowohl für Standard- als auch für Individualsoftwareentwicklung. Zur Vereinfachung werden daher auch interne Benutzer/Anwender/Auftraggeber als Kunden bezeichnet. Das an einigen Stellen erwähnte Marketing ist natürlich nur bei Standardsoftware relevant.

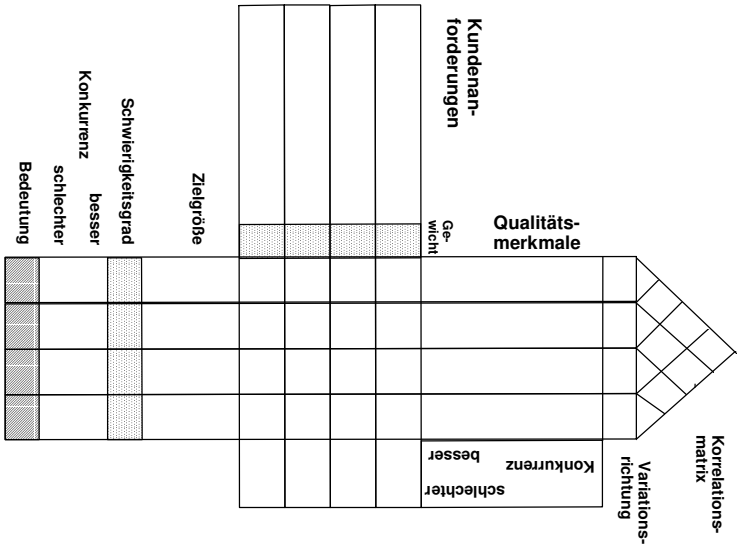

Abb. 2-1: House of Quality

3. Fallbeispiel zu Software Quality Function Deployment

Funktionsweise, Probleme und Nutzen von Software-QFD werden nachfolgend anhand eines realen Praxisbeispiels erläutert. Ausgangspunkt ist die zentrale, in Access entwickelte Adreßdatenbank des Lehrstuhls für Wirtschaftsinformatik, auf die ca. 20 Anwender von Ihrem lokalen PC aus zugreifen. Die Adreßdatenbank dient außer zur schnellen Auskunft über Telefon- und Faxadressen auch zum Versenden von Briefen oder Massenschreiben wie z. B. Seminarankündigungen oder Fragebogenuntersuchungen.[2]

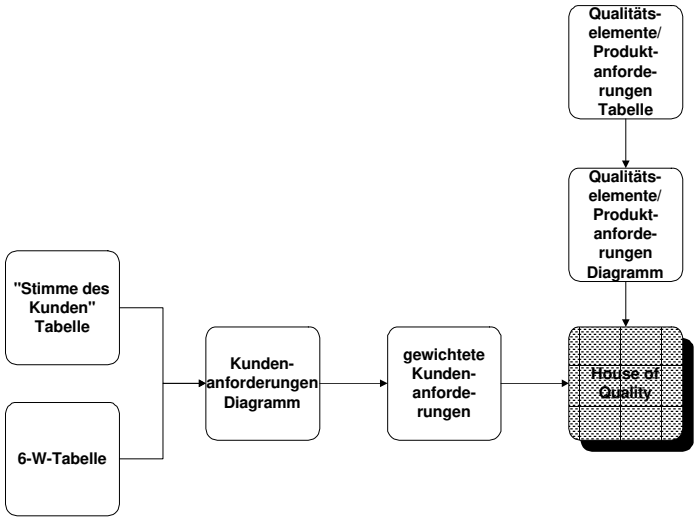

Abb. 3-1: Vorgehensweise Software-QFD bis zum House of Quality

---

2 Bei der Betrachtung des Beispiels ist zu beachten, daß jeweils nur Ausschnitte aus der QFD-Matrix betrachtet werden. Tatsächlich war die Anzahl der Kunden- bzw. Produktanforderungen etwa um den Faktor zehn höher.

Ermittlung von Kundenbedürfnissen und Kundenzufriedenheit

Ausgangspunkt für die erste QFD-Sitzung waren die Ergebnisse einer Fragebogenaktion zur Ermittlung der Kundenbedürfnisse und der Kundenzufriedenheit. Dabei stellte sich heraus, daß die Adreßdatenbank bei den Kriterien gut beurteilt wurde, die von den Kunden als eher unwichtig eingestuft wurden. Dagegen führten die als besonders wichtig eingeschätzten Kriterien wie Performance zu einer sehr schlechten Bewertung und gaben letztlich auch den Ausschlag für die Entscheidung, ein neues Relaese zu entwickeln. Außerdem kristallisierten sich durch die Anwendung statistischer Methoden (v. a. Korrelationskoeffizientenanalyse) verschiedene Kundentypen heraus: Bei den „Phlegmatiker" war eine relativ gleichmäßige Verteilung von Wichtigkeit und Zufriedenheit zu den verschiedenen Kriterien (z. B. Zuverlässigkeit/Fehlverhalten) zu beobachten. Eine nähere Betrachtung führte zu dem Ergebnis, daß es sich hierbei eher um die Gelegenheitsbenutzer handelte. Die „Hektiker" waren durch eine deutliche Priorisierung des Merkmals Antwortzeitverhalten charakterisiert, während die „Bequemen" eher Wert auf Zuverlässigkeit/Fehlverhalten und Benutzerfreundlichkeit legten.

Da die Adreßdatenbank letztlich für alle Kundentypen geeignet sein muß (bei einem Standardsoftwareprodukt würde man sich wahrscheinlich eher auf einen bestimmten Kundentyp spezialisieren), wurden für die QFD-Sitzungen jeweils zwei Repräsentanten aus den Gruppen ausgewählt. Der Lehrstuhlinhaber, der QM-Beauftragte (der auch die Kundenzufriedenheitsanalyse durchgeführt hatte) sowie der Entwickler der Adreßdatenbank waren ebenfalls Teilnehmer dieser Sitzungen. Der Kundenzufriedenheitsanalyse kam im Rahmen dieses Projektes also zum einen eine Auswahlfunktion für repräsentative Kunden und zum anderen (aus Forschungssicht) eine Kontrollfunktion bezüglich der Qualität der moderierten QFD-Sitzung zu.

Alle QFD-Sitzungen wurden nach der Metaplantechnik als moderierte Diskussionen durchgeführt. Per Kartenabfrage wurde zunächst die „Stimme des Kunden" ermittelt. Ausgangspunkt waren hierbei die „Geschäftsprozesse" der Kunden, d. h. es standen die Aufgaben, die der Benutzer erledigen will und die das Programm erfüllen sollte, im Mittelpunkt: „Brief schreiben" oder „Adreßliste erstellen" waren solche typischen Aufgaben. Eine wichtige Regel bei der Diskussion lautete, daß der Entwickler zwar mit den Kunden sprechen konnte, aber keine (wertenden) Kommentare bezüglich der Kundenanforderungen tätigen durfte. Die Kunden waren dazu angehalten, offen ihre Meinung über das Produkt zu äußern. Die Karten wurden mit Hilfe von Affinitäts- bzw. Baumdiagrammen strukturiert. Die „Stimme des Kunden" wurde dabei zunächst unverändert übernommen. Die Transformation dieser meist ungenauen Kundenstimme in detailliertere Kundenanforderungen geschah mit Hilfe der sogenannten 6W-Methode, bei dem der Moderator sich die Stimme des Kunden mit Hilfe von Leitfragen konkretisieren läßt (siehe Abb. 3-2)

Kundenstimme:	Wer:	Wie(viel):	Wo:	Wann:	Warum:	Was:	Kundenanforderung:
schnelles Antwortzeitverhalten	ständiger Benutzer	<2sec	Bildschirm	Abfrage von Adressen	schnelle Antwort	Ausgabe	schnelle Auskunft über Person
einfache Bedienung	gelegentlicher Benutzer	selbsterklärend	Bildschirm	Eingabe von Adressen	zügiges Arbeiten, Fehlervermeidung	Eingabe	leichte Eingabe von Adressen

Abb. 3-2: Beispiele zur 6W-Methode

Zur Gewichtung der Kundenanforderungen wurde der paarweise Vergleich eingesetzt, bei dem der Kunde (für alle Kombinationen) immer eine Kundenanforderung mit der anderen bezüglich ihrer Wichtigkeit vergleicht. Auf der Basis dieses Ergebnisses kann man dann die Kundenanforderungen auf einer Skala von eins bis fünf gewichten.

Bei der Ermittlung der Kundenzufriedenheit wurde mit Hilfe einer Punktbewertung zunächst die aktuelle Adreßdatenbank bezüglich aller Kundenanforderungen auf einer Skala von eins bis fünf bewertet. Außerdem wurde die Adreßdatenbank bezüglich der einzelnen Kundenanforderungen mit konkurrierenden Lösungen (das waren in diesem Fall einfache Excel-Tabellen, wie sie beispielsweise von der Sekretärin immer noch benutzt wurden) bezüglich der Zufriedenheit verglichen.

Die Kunden werden i. d. R. bei solchen Sitzungen (oder auch bei den klassischen Interviews im Rahmen des Requirements Engineering) sogenannte Qualitäts- und Leistungsanforderungen stellen (z. B. „diverse Suchmöglichkeiten“, „schnelles Antwortzeitverhalten“); sind diese Anforderungen erfüllt, wird auch die Kundenzufriedenheit zunehmen. Darüber hinaus existieren aber auch sogenannte Grund- bzw. Basisanforderungen, bei deren Nichterfüllung die Kundenzufriedenheit sehr rasch absinkt. Auf der anderen Seite bedeutet die Erfüllung dieser Basisanforderungen nicht zwangsläufig Kundenzufriedenheit, da die geforderten Merkmale als selbstverständlich angesehen werden. Sind sogenannte Begeisterungsanforderungen des Kunden (sprachgesteuertes Telefonieren, Einscannen von Visitenkarten etc.) nicht erfüllt, so führt dies keineswegs zur Unzufriedenheit, da diese Merkmale gar nicht erwartet werden. Werden die Begeisterungsanforderungen dagegen erfüllt, nimmt die Kundenzufriedenheit stark zu, gewisse andere Mängel (hinsichtlich der Qualitäts- und Leistungsanforderungen) werden dafür in Kauf genommen. Hierbei spielt also der Innovationsgrad eine wesentliche Rolle. Basis- und v. a. Begeisterungsanforderungen werden aber oftmals vom Kunden nicht geäußert, da dem Kunden möglicherweise nicht alle Anforderungen bewußt sind oder er auch oftmals keine Kenntnis über die technischen Möglichkeiten hat.

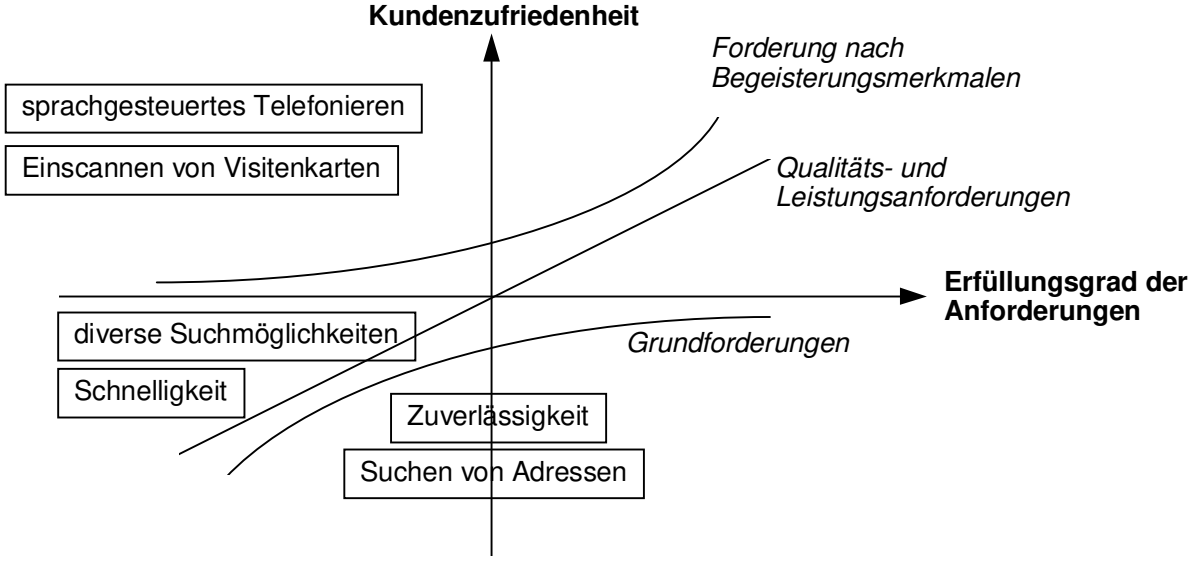

Abb. 3-3: Klassifikation von Kundenanforderungen[3]

3 In Anlehnung an Kano et. al.: Attractive quality and must-be quality. In: Quality. Nr. 2, 1984, S. 39-44

Um dies im Rahmen von QFD zu berücksichtigen, wurde in einer separaten Sitzung aus Entwickler- und das Qualitätsmanagementsicht über mögliche Produktmerkmale ein Brainstorming durchgeführt („Stimme des Entwicklers“) und diese dann zu den geäußerten Kundenanforderungen in Beziehung gesetzt.[4]

Ermittlung von Qualitätselementen bzw. Produktanforderungen

Während die Kundenanalyse eher die Frage, *warum* etwas zu entwickeln ist, beantwortet, mußte im nächsten Schritt der Frage nachgegangen werden, *was* zu entwickeln ist. Bei dieser Fragestellung nach den Produktanforderungen treten die Entwickler in den Vordergrund. Zunächst wurde wieder per Kartenabfrage „Wie erfüllen wir die Kundenanforderungen?“ nach implementierungsunabhängigen Aussagen über die Fähigkeiten des Produktes gesucht. Diese Produktanforderungen sollten noch keine detaillierten Lösungen enthalten, aber möglichst überprüfbar, d. h. quantifizierbar, sein (z. B. Antwortzeitverhalten; das Vorhandensein bzw. die Qualität dieser Produktanforderung kann später in Sekunden gemessen werden). Auch hier ist eine Hierarchisierung z. B. mit Hilfe von Baumdiagrammen sinnvoll.

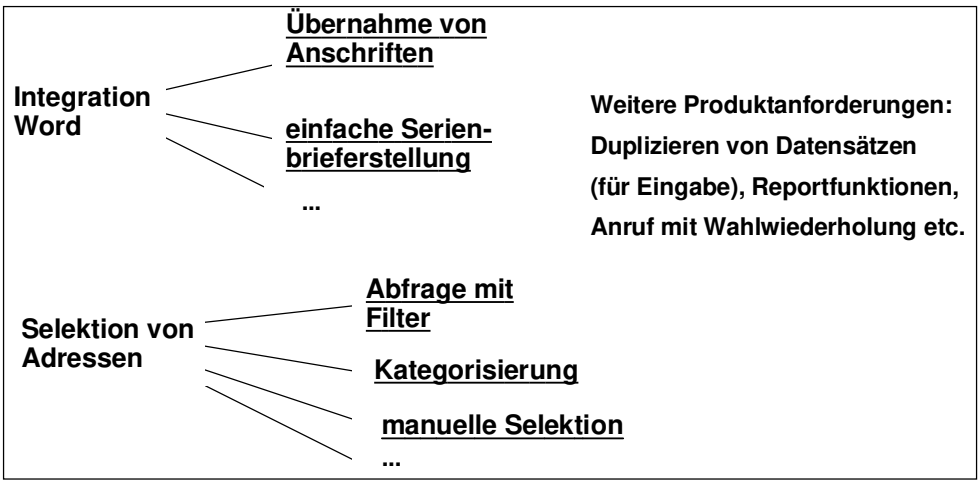

Abb. 3-4: Beispiele für hierarchisierte Produktanforderungen an die Adreßdatenbank

Bestimmung der Korrelationen von Kunden- und Produktanforderungen

Der nächste Schritt versuchte den Zusammenhang zwischen den vornehmlich durch die Kunden formulierten Kundenanforderungen und den überwiegend durch die Entwickler beschriebenen Produktanforderungen zu analysieren. Der Moderator stellte die Frage „Wie stark unterstützt die Erreichung der Produktanforderung x die Erfüllung der Kundenanforderung y?“ Die Unterstützungsstärke konnte in mehreren Bewertungstufen (z. B. 0, 1, 3, 9) angegeben

---

4 Auf eine detaillierte Schilderung dieser Sitzung und die Zusammenführung mit der Kundensitzung muß an dieser Stelle aus Platzgründen verzichtet werden.

werden. Bei der Diskussion bzw. Argumentation der Teilnehmer sollten Zahlen, Daten und Fakten vor Vermutungen rangieren. Nachdem die Matrix mit Zahlen gefüllt war, konnte eine Konsistenzanalyse erfolgen: leere Zeilen (bzw. Kundenanforderung ohne Korrelation) deuten darauf hin, daß Produktanforderungen übersehen wurden, leere Spalten (bzw. Produktanforderung ohne Korrelation) sind ein Indiz, daß möglicherweise ein überflüssiges Produktmerkmal definiert wurde.

Technische Beurteilung der Produktanforderungen

In diesem Schritt wurde v. a. aus der Sicht des Entwicklers zum einen der technische Schwierigkeitsgrad der Erfüllung der Produktanforderungen sowie der technische Vergleich mit der Konkurrenz bewertet (die Bewertungsstufen erfolgten analog zu der Bewertung der Kundenanforderungen).

Auswertung und Interpretation des House of Quality

Für jedes Produktmerkmal wurde die Bedeutung als ∑ Gewichte * Korrelationsstärke (absolut und relativ) ermittelt. Hieraus ergab sich dann eine Rangfolge bezüglich der wichtigsten nachfolgend zu realisierenden Produktmerkmale. Die Auswahl der wichtigsten Produktmerkmale erfolgte jedoch nicht alleine auf der Basis dieser einfachen Berechnung. Vielmehr mußten auch die Daten über den Konkurrenzvergleich bzw. den Schwierigkeitsgrad berücksichtigt werden. So landete z. B. das Produktmerkmal Datensätze duplizieren rein rechnerisch nur auf Platz 5, da aber dieses Merkmal sehr leicht zu realisieren war und die „Konkurrenz“ hier gleichzeitig besser abschnitt, wurde dieses Produktmerkmal dennoch mit in die nächste QFD-Stufe übernommen.

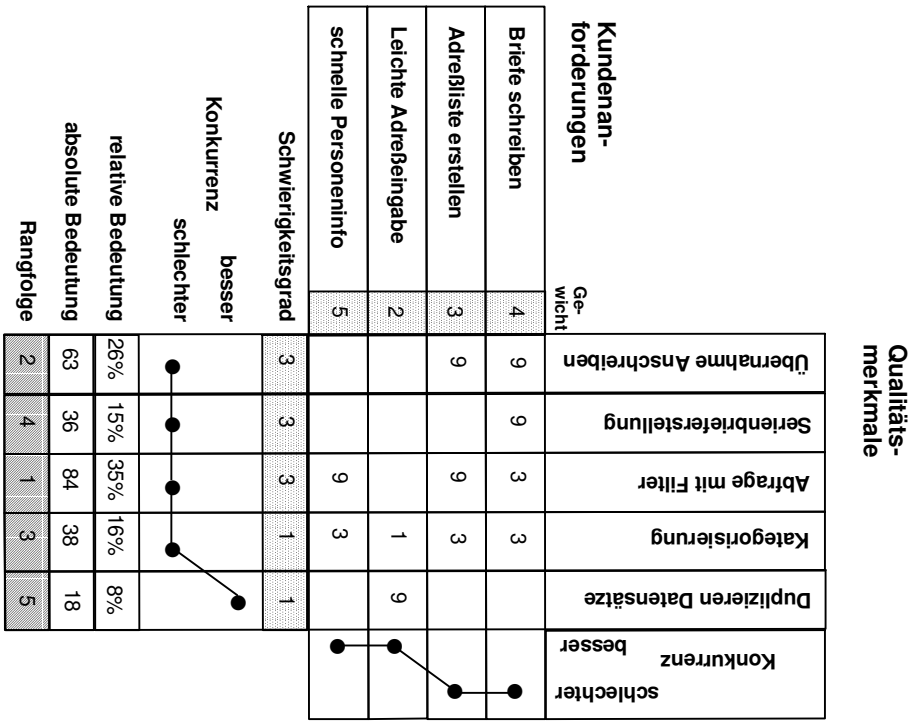

Abb. 3-5: Vereinfachtes House of Quality für die Adreßdatenbank

Weitere Vorgehensweise

Die nächsten Schritte bei der Anwendung der QFD-Methode sind das Herunterbrechen der Produktanforderungen auf Subsysteme und Module. Diese können dann ähnlich wie beim QFD der industriellen Fertigung weiter auf Prozeßmerkmale, d. h. bestimmte „Arbeitsanweisungen" in Form von Programmiervorgaben etc. abgebildet werden. Aber selbst wenn man sich auf das erste House of Quality beschränkt, bietet QFD eine hervorragende Möglichkeit der kundenorientierten Fokussierung der weiteren Softwareentwicklung auf die wichtigsten Aspekte.

4. Erfahrungen mit Software Quality Function Deployment

Erste Erfahrungen in den U.S.A. und Japan mit der Anwendung von QFD für die Softwareentwicklung zeigen, daß der Nutzen von QFD v. a. in der bereichsübergreifenden Kommunikation, dem klaren Verständnis von Kunden-/Benutzeranforderungen, der Übereinstimmung über die gefundenen Lösungswege, der geringeren Anzahl von Änderungen nach der Auslieferung, der kompletten Dokumentation der einzelnen Schritte, einem profitablen Produkt und zufriedenen, oft begeisterten Kunden liegt. Als Nachteile von QFD sind die erhebliche Komplexität verbunden mit einem erheblichen Zeitaufwand für Vorbereitung, Durchführung und Nachbereitung der Sitzungen zu nennen.

5. Weitere Forschung zu Software Quality Function Deployment

Um die Kundenstimme wirklich durch den gesamten Entwicklungsprozeß zu tragen, (Deployment) ist eine komplexe Matrixkette erforderlich, die sich für Software erheblich von der traditionellen Fertigungs-QFD Kette unterscheidet. Ausgehend von Geschäftsprozeßanforderungen, die nicht immer alle durch Software abgedeckt werden können oder sollten, muß der Softwareentwicklungs*prozeß* insgesamt „requirements-driven" gestaltet werden.

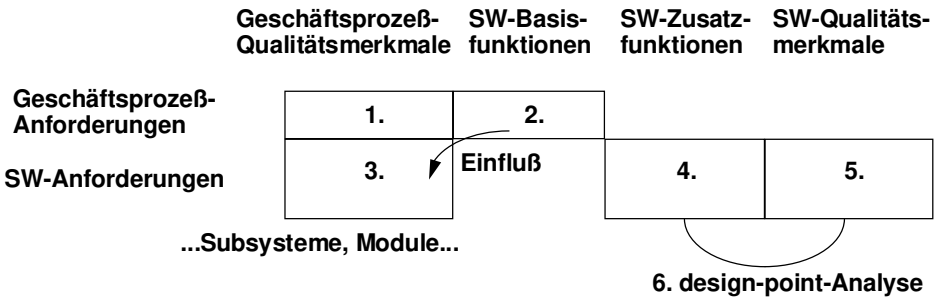

Abb. 5-1: Prozeßanalyse beim Software-QFD

Literatur

Yoji Akao: QFD - Quality Function Deployment - Wie die Japaner Kundenwünsche in Qualität umsetzen. Landsberg/Lech 1992

Akira Ohmori: Software quality deployment approach: framework design, methodology and example. In: Software Quality Journal. Nr. 3, 1993, S. 209-240