

IVENA: Integriertes Vorgehen zur Erhebung nichtfunktionaler Anforderungen

Rolf Götz, Heiko Scharnweber, SOPHIST GmbH

Zusammenfassung

Im vorliegenden Beitrag wird ein Ansatz erläutert, wie man nichtfunktionale Anforderungen so erhebt, definiert und dokumentiert, dass bestimmte negative Konsequenzen für den Prozess der Systementwicklung, -einführung und -nutzung vermieden werden. Ausgehend von den sich stellenden Problemen im Betrachtungsbereich leiten die Autoren Forderungen an eine Vorgehensweise zur Erhebung von nichtfunktionalen Anforderungen ab und stellen eine von ihnen entworfene Lösung vor. Beispiele ergänzen die Argumentation. Die Tauglichkeit des gewählten Ansatzes wird derzeit in der Praxis überprüft. Erste Erfahrungen zeigen, dass der skizzierte Weg gangbar ist.

Abstract

The paper describes a methodology to elicit, define and document non-functional requirements. The question placed is how to avoid negative consequences for the system engineering process (including testing), the introduction of a system to an organization and the use of a system. Starting at the problems met in the domain the authors derive how a solution to the problems should look like. Supported by examples the solution designed is given. That the solution is suitable is shown by first experiences in a large system engineering project.

1 Einleitung

„The product shall be easy for 11 year-old children to use.“

Diese „Qualität“ einer nichtfunktionalen Anforderung an ein zu erstellendes System ist heute durchaus noch üblich. Wer solche Anforderungen akzeptiert, darf sich nicht wundern, welches Produkt er bekommt.

Was hat der Leser zu erwarten?

Wir SOPHISTen sind Experten für die Konzeptionierung und den Einsatz von Vorgehensmodellen in der Praxis, nicht für das Testen von Soft- und Hardwaresystemen. Deshalb geht es in diesem Bericht *nicht* darum, ein Verfahren zum Testen von Anforderungen darzustellen. Wir haben es uns vielmehr zum Ziel gemacht, die *Voraussetzungen* zu schaffen, um nichtfunktionale Anforderungen¹ überhaupt testen zu können. Dazu entwickelten wir IVENA, das Integrierte Vorgehen zur Erhebung nichtfunktionaler Anforderungen. Zwei Großprojekte, die derzeit von uns begleitet werden, sollten bis zum Herbst 1999 bestätigen oder widerlegen, dass unser Ansatz praxistauglich ist.

Zum Aufbau: In diesem Artikel werden wir in **Kapitel 2** zunächst die Situation darstellen, die uns in unserer täglichen Arbeit beim Kunden begegnet und die Vermutungen äußern, welches die Ursachen für „schlechte“ nichtfunktionale Anforderungen sind. Daraus werden dann in **Kapitel 3** die Aufgaben ableiten, die sich zur Konzeptionierung eines Vorgehens stellen, das die genannten Probleme angeht. In **Kapitel 4** folgt zunächst ein kurzer Überblick über unsere Lösung IVENA, gefolgt von einer etwas eingehenderen Betrachtung ihres Aufbaus und Vorgehens. Ein ausführliches Praxisbeispiel demonstriert in **Kapitel 5** den Einsatz von IVENA. Zum Schluß nennt **Kapitel 6** die Erfahrungen, die wie bisher mit dem Ansatz machten, **Kapitel 7** verzeichnet die wenigen Quellen.

2 Situationsdarstellung

2.1 Was passiert, wenn nichtfunktionale Anforderungen in der Anforderungserhebung vernachlässigt werden?

Funktionale Anforderungen eindeutig, vollständig und widerspruchsfrei zu dokumentieren wird in vielen Anwendungsbereichen zusehends üblicher. Werden jedoch die *nichtfunktionalen* Anforderungen nicht zusätzlich angemessen berücksichtigt, so hat das unserer Erfahrung nach eine Reihe von möglichen negativen Konsequenzen für das System:

- Das System erfüllt nicht die „erhofften“ Anforderungen des Auftraggebers oder der Anwender.
- Es gibt Schwierigkeiten bei der Systemeinführung, was dazu führt, dass das neue System ganz oder in Teilen unbrauchbar ist.

¹ Siehe [DIN94]

- Die definierten Anforderungen sind unbrauchbar als Vorgabe für den Entwurf und werden somit in der weiteren Systementwicklung nicht berücksichtigt.
- Die Anforderungen eignen sich nicht als Grundlage für das Testen des Systems und fehlen dann in entsprechender Qualität.

Daraus leitet sich zunächst ein Problemkreis ab, der die betroffenen Personen in den Mittelpunkt stellt:

- Die Anwender lehnen das neue System aufgrund mangelhafter Berücksichtigung ihrer Bedürfnisse (z.B. „leichte Bedienbarkeit“) ab. Die Folge davon ist oft eine „back to the pen“-Lösung für Teilbereiche des Systems.
- Die Systemeinführung scheitert unter Umständen schleichend aufgrund des passiven Protests der Anwender.
- Die Designer und Entwickler wissen nicht sicher, ob sie ihre Lösung richtig dimensionieren, was die nichtfunktionalen Anforderungen angeht.
- Spätestens die Testmannschaft beginnt, aus den verschiedenen „stake holdern“ des Systems Anhaltspunkte für einen praxisnahen Test herauszubekommen.

Das aber hat selbstverständlich weitreichende Folgen für die Organisationen, die ein System einführen, als Ganzes.

- Eine u.U. schleichend gescheiterte Systemeinführung oder ein „schlecht“ entwickeltes System bedeutet langfristig einen hohen finanziellen Aufwand.
- Teure Nachbesserungen (oft im IT-Infrastrukturbereich oder in aufwendigen Ausbesserungsarbeiten am System) sind nötig, um die „geforderten“ Qualitätseigenschaften doch noch nutzen zu können.

2.2 Warum werden nichtfunktionale Anforderungen „schlecht“ erhoben?

Um zu lernen, was es bei der Definition nichtfunktionaler Anforderungen besser zu machen gilt, kann man zunächst untersuchen, welche Missstände im Allgemeinen dabei auftreten. Mindestens fünf Aspekte spielen dabei eine Rolle: schlechte Quantifizierbarkeit, fehlende Bewertungsmaßstäbe, unzureichende Strukturierung der Qualitätsmerkmale, die unterschiedlichen Sichtweisen der an der Anforderungserhebung Beteiligten und die mangelnde Integration von neuen Vorgehensweisen in bestehende und in existierende Begriffssysteme.

2.2.1 Quantifizierbarkeit nichtfunktionaler Anforderungen

Anforderungen können erst getestet werden, wenn sie quantifiziert, also messbar gemacht sind. Die Quantifizierung (vermeintlich schwammiger, subjektiver) nicht-funktionaler Anforderungen ist dabei für die IT-Branche überdurchschnittlich schwierig, da sie dabei mit dem „soft factor“ *Mensch* umgehen muss.

2.2.2 Bewertungsmaßstäbe für nichtfunktionale Anforderungen

Entscheidend für die Qualität der definierten nichtfunktionalen Anforderungen ist das Festlegen eines Bewertungsmaßstabs. Ohne einen Richtwert und eine

Werteskala mit objektiven Maßeinheiten lassen sich Qualitätsforderungen nicht objektiv quantifizieren und damit später nicht sinnvoll testen.

2.2.3 Einteilung der Qualitätsmerkmale

Die Erhebung nichtfunktionaler Anforderungen *verlangt* eine eindeutige und widerspruchsfreie Ordnung der zu betreffenden Qualitätsaspekte. Gleichzeitig *ermöglicht* der Komplex nichtfunktionale Anforderungen dies besonders leicht, im Gegensatz zu den funktionalen Anforderungen. Oft wird aus den gegebenen Qualitätsmerkmalen keine geeignete, d.h. organisations- oder projektspezifische Auswahl getroffen.

2.2.4 Unterschiedliche Qualitätssichten

Eine wichtige Ursache für eine unzureichende Anforderungsdefinition ist die Trennung zwischen denen, die die Anforderungen an ein zu entwickelndes System definieren und denen, die später damit arbeiten. So hat der Systemverwalter andere Ansprüche (z.B. Stabilität) an die Qualität als der Anwender (etwa Bedienbarkeit). Diese Trennung wird gerne übersehen.

2.2.5 Integration in bestehende Vorgehensweisen und Begriffssysteme

Neue Vorgehensweisen betten sich häufig nicht in bestehende Vorgehensweisen ein, die die Definition der funktionalen Anforderungen oder von Anforderungen allgemein regelt. Existierende Begriffssysteme von Organisationen oder Projekten werden gerne umgewälzt.

3 Zu lösende Aufgaben

Im Rahmen eines internen Entwicklungsprojektes der SOPHISTen sollte aus den in 2.1 genannten Gründen eine umsetzungsfähige Vorgehensweise geschaffen werden, nichtfunktionale Anforderungen zu erheben. Gleichzeitig mussten die unter 2.2 aufgeführten Problemkreise berücksichtigt werden. Folgende Aufgaben stellten sich:

- Nichtfunktionale Anforderungen müssen quantifizierbar werden (2.2.1).
- Aufklärungsarbeit ist zu leisten (2.2.1)
 - Die verantwortlichen Entscheidungsträger müssen um die Wichtigkeit der nicht funktionalen Anforderungen in der Anforderungsdefinition wissen. Dadurch kann die Gefahr des Scheiterns einer System-einführung aufgrund mangelnder Umsetzung der nichtfunktionalen Anforderungen minimiert werden.
 - Den Anwendern muss gezeigt werden, wie sie ihre von subjektivem Empfinden geprägten Anforderungen in eine Form bringen können, die sich als Vorgabe für die Entwicklung des Systems eignet.
- Objektive Bewertungsmaßstäbe für Qualitätsmerkmale müssen geschaffen werden (2.2.2).
- Überschneidungsfreie Qualitätsmerkmale müssen gefunden werden (2.2.3).
- Es muss möglich sein, bestimmte Qualitätsmerkmale außer Acht zu lassen, wenn die Situation das erfordert (2.2.3).

- Eine Möglichkeit zur Einbeziehung von allen Beteiligten muss entwickelt werden (2.2.4).
- Der Ansatz ist so zu wählen, dass bestehende Vorgehensweisen nicht behindert, besser sogar unterstützt werden und sich das Ergebnis in das organisations- oder projektspezifische Bild einfügt (2.2.5).

4 Lösungsfindung: „Integriertes Vorgehen zur Erhebung nichtfunktionaler Anforderungen“

Mit IVENA ist ein Ansatz für eine umsetzungsfähige Vorgehensweise geschaffen, nichtfunktionale Anforderungen zu erheben, zu definieren und zu dokumentieren. Sie erfüllt die unter Kapitel 3 genannten Aufgaben.

IVENA unterstützt den Analytiker in seinem Vorgehen beim Erheben, Definieren und Dokumentieren von nichtfunktionalen Anforderungen. Es bietet einen verbindlichen Rahmen für die zu definierenden nichtfunktionalen Anforderungen, basierend auf der DIN 66272 „Bewerten von Softwareprodukten“ (siehe Abbildung 1). Dieser Rahmen bietet maximale Freiheit ohne Verzicht auf Eindeutigkeit, Vollständigkeit und Widerspruchsfreiheit in den Anforderungen und erscheint deshalb als geeignet. Jedoch spricht nichts dagegen, Vorgaben aus anderen Normen oder auch freie Regelungen an dieser Stelle vorzusehen.

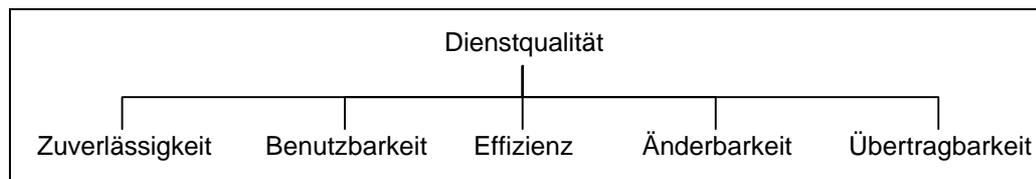


Abbildung 1: Merkmale der Dienstqualität nach DIN 66272

Die folgenden beiden Abschnitte 4.1 und 4.2 geben einen kurzen Überblick über den Aufbau von und das Vorgehen mit IVENA. Abschnitt 4.3 widmet sich einer eingehenden Betrachtung.

4.1 Der Aufbau von IVENA

IVENA besteht aus drei Teilbereichen (vgl. Abbildung 2):

Leitfaden zur Vorgehensweise im Interview/bei der Erhebung. Der Leitfaden unterstützt den Analytiker dabei, strukturiert und damit übersichtlich vorzugehen, und nichts außer Acht zu lassen.

Dieser Leitfaden besteht und dürfte unserer Erfahrung nach nur noch geringfügig durch Praxiserfahrungen modifiziert werden, da er sich stark an der praktikablen Logik anlehnt, die der DIN 66272 innewohnt.

Aktuelle Referenz-Beispiel-Datenbank (RBD). Die Referenz-Beispiel-Datenbank dient als Nachschlagewerk bei der Erhebung der nichtfunktionalen Anforderungen. Anhand der RBD sollen bereits organisations- oder projektspezifisch erarbeitete Lösungen zur Quantifizierung eines Merkmals aus der bestehenden Datenbasis übernommen, oder Ideen daraus abgeleitet werden können. Eine ständige Erweiterung der RBD stellt einen stetig zunehmenden Gebrauchswert sicher.

Für die der Norm² entnommenen Qualitätsmerkmale und ihre Submerkmale existieren bereits Lösungen, die ihre Quantifizierung ermöglichen.

Interviewergebnisse/erhobene Informationen. Die zum Beispiel in Interviews erhobenen Daten können direkt in das dafür vorgesehene Medium (z.B. eine Anforderungsdatenbank) geschrieben werden.

Im vorliegenden Fall existieren weitgehend vollständige Musteranforderungen für jedes Qualitätsmerkmal, die nur um konkrete Zahlenwerte zu ergänzen sind.

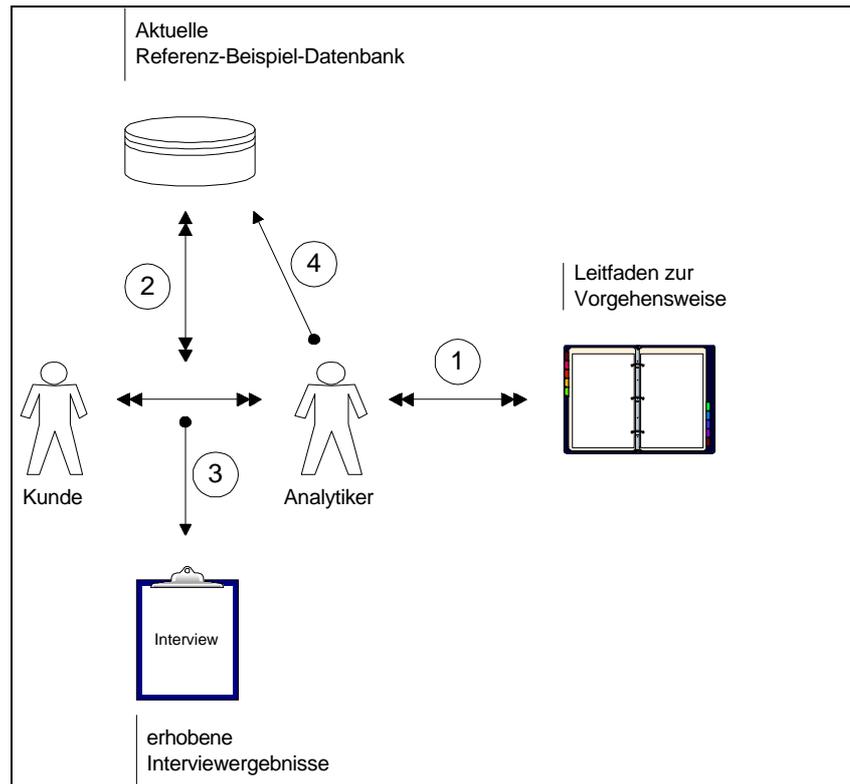


Abbildung 2: Komponenten und Vorgehen IVENA

4.2 Das Vorgehen mit IVENA

Folgende Schritte bieten sich beim Vorgehen mit IVENA an (vgl. auch Abbildung 2):

Schritt 1. Der Analytiker erhält mit dem Leitfaden konkrete Anhaltspunkte für sein Interview mit den Vertretern des Auftraggebers oder das Vorgehen, das er statt dessen für die Anforderungserhebung wählt. Anhand der Aufteilung der Qualitätsmerkmale, zum Beispiel nach DIN 66272, und deren Submerkmalen wird ihm eine Anleitung für jeden einzelnen Schritt angeboten, von den grundlegenden Begriffsfestsetzungen, über die Aufstellung von Bewertungsmaßstäben und Gewichtungsfaktoren bis hin zur fertigen Anforderung (siehe 4.3).

² [DIN94]

Schritt 2. Er unterfüttert die Vorstellungen des Befragten mit konkreten Beispielen für eine quantifizierbare Umsetzung dieser Vorstellungen, die er der RDB entnimmt und gegebenenfalls anpasst. Wenn sich im Gespräch ganz neue Ansätze ergeben, so werden diese verwendet und in Schritt 4 zur RDB hinzugefügt.

Schritt 3. Der Analytiker dokumentiert die zusammen mit dem Befragten definierten Begriffe, Bewertungsmaßstäbe und Gewichtungsfaktoren und bildet daraus unter Umständen sogar weitgehend mechanisch eine Anforderung, die er zum Beispiel direkt in einer Anforderungsdatenbank ablegt.

Schritt 4. Nach getaner Arbeit mit den Befragten wird die RDB um neue Ansätze erweitert. Jede Möglichkeit, eine Anforderung zu quantifizieren und alle Varianten davon werden strukturiert aufbereitet und stehen damit für jede weitere Anwendung von IVENA zusätzlich zur Verfügung.

4.3 Nähere Betrachtung des Aufbaus

4.3.1 Leitfaden zur Vorgehensweise *Teilbereich I* und *Teilbereich II*

Der Leitfaden zur Vorgehensweise unterstützt den Analytiker dabei, strukturiert und übersichtlich bei der Erhebung der nichtfunktionalen Anforderungen vorzugehen, und dabei nichts zu vergessen.

Der Leitfaden zur Vorgehensweise besteht aus *zwei* Teilbereichen, dem *Definitionsteil* und dem *Fragenteil* (vgl. Abbildung 3).

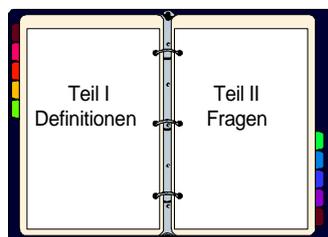


Abbildung 3: Aufbau des Leitfadens zur Vorgehensweise

4.3.1.1 Teil I: Definitionen

Der Teilbereich I des Leitfadens zur Vorgehensweise in der Anforderungserhebung besteht aus einer *Sammlung der Definitionen*³ aller Merkmale und Submerkmale⁴, die Eigenschaften von zu erstellenden Systemen definiert.

4.3.1.2 Teil II: Fragen

Der *Teilbereich II* des Leitfadens zur Vorgehensweise in der Erhebung, Definition und Dokumentation von nichtfunktionalen Anforderungen umfaßt *Fragen*, die dem Analytiker dazu dienen sollen, das zu erarbeitende Themengebiet von

³ angelehnt an [DIN94]

⁴ angelehnt an [DIN94]

Merkmale und Submerkmalen einzugrenzen und soll sogleich als „Roter Faden“ zur Vorgehensweise im Interview dienen.

Die *zwei Typen* von Fragen bauen aufeinander auf. Es sind die *erschließenden Fragen* und die *Detailfragen*.

4.3.1.2.1 Erschließende Fragen:

Anhand der *erschließenden Fragen* wird mit dem Befragten zunächst grundsätzlich geklärt, ob ein Bedarf oder die Notwendigkeit besteht, die betreffende Eigenschaft für das zu erstellende System zu fordern. Ein gezieltes Fragen von Seiten des Analytikers fordert den Kunden zunächst heraus, sich detailliertere Gedanken zur Definition der nichtfunktionalen Anforderungen zu machen, seine Bedürfnisse zu ergründen und explizit im Rahmen der Anforderungserhebung zu äußern (siehe hierzu auch das Beispiel unter 5.1.1).

4.3.1.2.2 Detailfragen

Anhand der *Detailfragen* wird mit dem Befragten gemeinsam erarbeitet, wie weit und wie umfangreich er die entsprechenden Merkmale beschreiben möchte. So kann zum Beispiel der Befragte die Notwendigkeit der Definition der nichtfunktionalen Eigenschaft „Zuverlässigkeit“ erkannt haben, mit seiner detaillierten Umsetzung aber nur für das Submerkmal „Systemreife“ einverstanden sein (siehe hierzu auch das Beispiel unter 5.1.2).

4.3.2 Aktuelle Referenz-Beispiel-Datenbank (RBD)

Die Referenz-Beispiel-Datenbank ist das innovative Kernstück von IVENA. Es soll dem Analytiker als *Ideengeber und Nachschlagewerk* bei der Umsetzung der Quantifizierung von nicht funktionalen Anforderungen dienen.

Die Funktion des Ideengebers und Nachschlagewerkes der RBD besteht darin, daß jeder Analytiker seine Erfahrungen und organisations- oder projekt-spezifische Lösungen zur Quantifizierung von Merkmalen in den jeweils genau spezifizierten Teil der generisch aufgebauten RBD einträgt.

Die eingetragenen Lösungen und Ideen stehen somit ihm selbst und anderen Analytikern zu Verfügung. Die Analytiker können im Idealfall bereits erarbeitete Lösungen komplett übernehmen oder Ideen für eine eigene Lösungsfindung aus der RBD für ihre konkrete Projektsituation ableiten.

Durch eine stetige und geordnete Erweiterung der Datenbasis der RBD, soll ein maximaler Synergieeffekt aus den Projekten der unterschiedlichsten Bereiche erzielt werden. Ziel ist dabei, den Gebrauchswert der RBD ständig zu steigern.

4.3.2.1 Übersicht über den Aufbau der RBD

Um den stetig wachsenden und wechselnden Anforderungen aus der Theorie und Praxis gerecht zu werden, ist die RBD möglichst generisch gehalten. Es soll dadurch neuen Impulsen möglichst viel Raum gegeben werden.

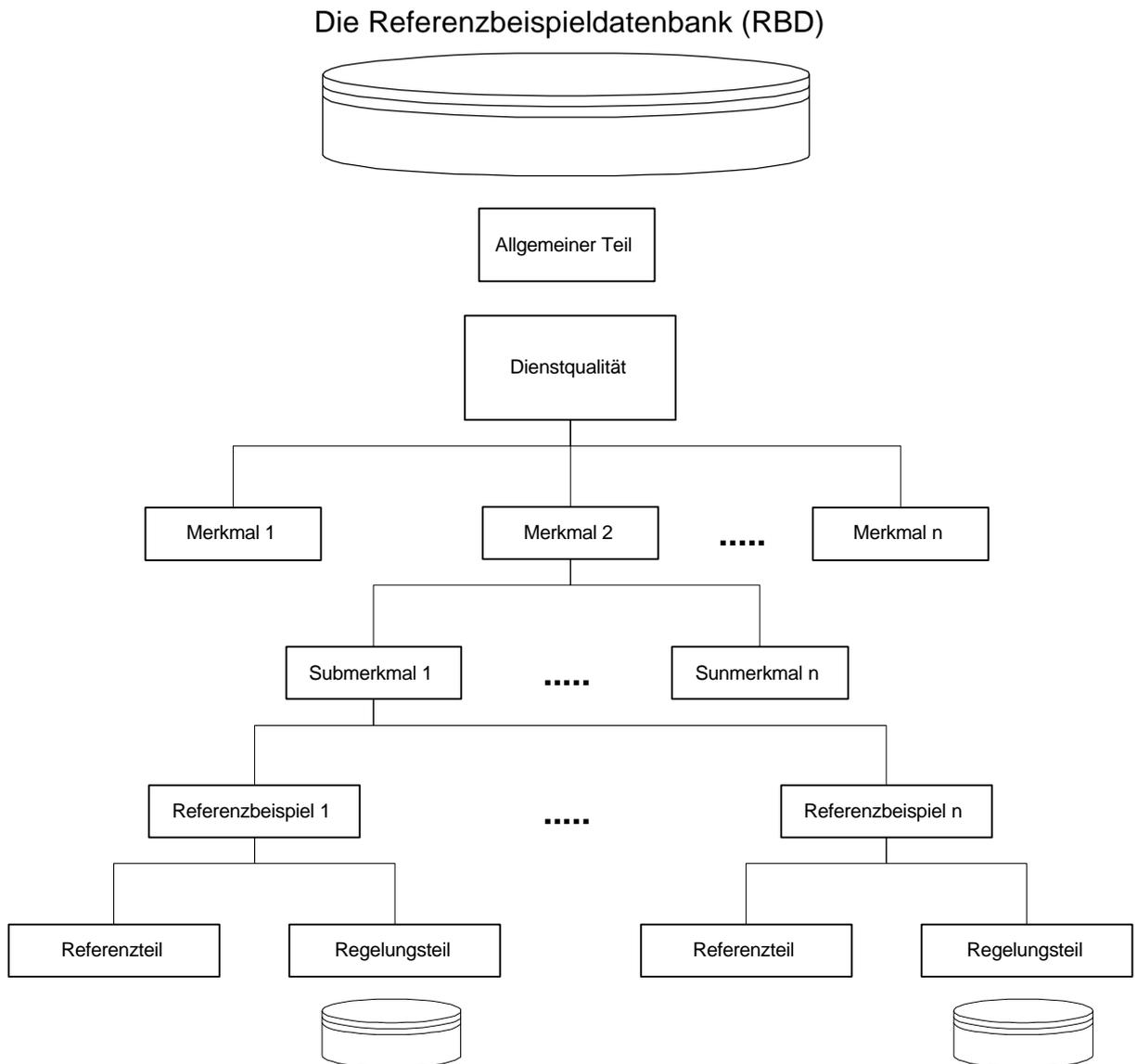


Abbildung 4: Die Referenz-Beispiel-Datenbank

4.3.2.2 Allgemeiner Teil der RBD

Im *Allgemeinen Teil* der RBD sind allgemeine und merkmalsübergreifende Informationen und Anmerkungen hinterlegt, die es bei der Erhebung, Definition und Dokumentation der nichtfunktionalen Anforderung im Rahmen von IVENA zu berücksichtigen gilt.

4.3.2.3 Das Referenzbeispiel

Für jede in die RBD einzustellende Lösung wird ein in sich geschlossenes *Referenzbeispiel* erstellt und dem entsprechenden Submerkmal zugeführt. Ein Referenzbeispiel besteht aus einem *Referenz-* und einem zugehörigen *Regelungsteil*.

4.3.2.3.1 Referenzteil

Im *Referenzteil* sind die Kenngrößen zur Berechnung der jeweiligen Submerkmale dargestellt und näher erläutert. Siehe dazu auch das Beispiel in Abschnitt 5.2.1.

4.3.2.3.2 Regelungsteil

Im *Regelungsteil* sind anhand von Beispielen der Lösungsweg und die konkreten Lösungen beschrieben. Am Ende des Regelungsteils kann die konkrete Anforderung an die gewünschte Eigenschaft stehen. In Abschnitt 5.2.2 findet sich ein vollständig ausgearbeitetes Beispiel dazu.

4.3.3 Interviewergebnisse / erhobene Informationen

Die in Interviews erhobenen Informationen (die tatsächlichen Regelungen des Regelunsteiles) können direkt in das dafür vorgesehene Medium (zum Beispiel eine Anforderungsdatenbank) geschrieben werden. Nach Abschluß des Projekts sollen diese als Grundlage für die Referenz-Beispiel-Datenbank dienen.

Die SOPHISTen realisieren IVENA selbst und das Management der erhobenen Informationen erfolgreich mit Hilfe einer C.A.R.E.-Datenbank⁵.

⁵ C.A.R.E (Computer-Aided Requirements Engineering) ist eine Anforderungsdatenbank, die ein koordiniertes, methodisch fundiertes Vorgehen beim Requirements Engineering mit räumlich verteilten Projektgruppen wirksam unterstützt.

5 Ein Beispiel aus der Praxis

Es folgt ein Beispiel zum Thema Quantifizierung von nichtfunktionalen Anforderungen. Es handelt sich um einen Auszug aus einem von den SOPHISTen begleiteten Projekt.

Da die Demonstration des gesamten Sachverhalts den Rahmen dieses Artikels sprengen würde, wird an dieser Stelle nur auf den *Leitfaden zur Vorgehensweise II (Fragen)* eingegangen. Des Weiteren wird ein vollständig ausgearbeitetes *Referenzbeispiel* für das Qualitätsmerkmal „Änderbarkeit“⁶ präsentiert. Die angeführten Querverweise ermöglichen es, den Bezug zur theoretischen Betrachtung von IVENA in Abschnitt 4.3 herzustellen.

5.1 Beispiel zum Leitfaden Teil II: Fragen

Dieser Abschnitt demonstriert die erschließenden und die Detailfragen zur Änderbarkeit des verwendeten Referenzbeispiels. (vgl. Abschnitt 4.3.1.2). Die Fragen sind bewußt sehr abstrakt gehalten, um auf die verschiedensten Situationen anwendbar zu sein. Es obliegt dem Analytiker, die Fragen projekt- und personenspezifisch anzupassen.

5.1.1 Erschließende Frage zur Änderbarkeit (vgl. 4.3.1.2.1)

< Wollen Sie Kenngrößen zu den Merkmalen definieren, die sich auf den Aufwand beziehen, der zur Durchführung vorgegebener Änderungen des zu entwickelnden Systems notwendig ist? >

5.1.2 Detailfragen zur Änderbarkeit (vgl. 4.3.1.2.2)

5.1.2.1 Analysierbarkeit

< Wollen Sie Kenngrößen für die Merkmale definieren, die sich auf den Aufwand beziehen, um Mängeln oder Ursachen von Versagen des zu erstellenden Systems zu diagnostizieren oder um änderungsbedürftige Teile zu bestimmen? >

5.1.2.2 Modifizierbarkeit

< Wollen Sie Kenngrößen für die Merkmale definieren, die sich auf den Aufwand beziehen, der zur Ausführung von Verbesserungen, zur Fehlerbeseitigung oder zu Anpassung an Umgebungsveränderungen notwendig ist? >

5.1.2.3 Stabilität

< Wollen Sie Kenngrößen für die Merkmale definieren, die sich auf das Risiko unerwarteter Wirkungen von Änderungen beziehen? >

⁶ Zum Verständnis muß man wissen, dass laut DIN 66272 das Merkmal Änderbarkeit in die Submerkmale Analysierbarkeit, Modifizierbarkeit und Stabilität und Testbarkeit untergliedert ist.

5.1.2.4 Prüfbarkeit

< Wollen Sie Kenngrößen für die Merkmale definieren, die sich auf den Aufwand beziehen, der zur Prüfung der geänderten Software des zu erstellenden Systems notwendig ist? >

5.2 Auszug aus der Referenz-Beispiel-Datenbank (vgl. 4.3.2.3)

Dieser Abschnitt enthält einen *Referenzteil* aus der RBD zum Thema Änderbarkeit. Anschließend werden im *Regelungsteil* die projektspezifischen Festlegungen demonstriert.

5.2.1 Der Referenzteil (vgl. Abschnitt 4.3.2.3.1)

5.2.1.1 Definition von Metriken und deren Grenzwerten

Es sind bestimmte Grenzwerte für Metriken festzulegen.

Die Metriken und die dazugehörigen Grenzwerte, die zur Bestimmung der Analysierbarkeit, Modifizierbarkeit, Stabilität und Testbarkeit herangezogen werden können lauten⁷:

Bez.	Metrik	Wertebereich	
		Min	Max
NSTAT	Anzahl der ausführbaren Anweisungen	1	50
DCAL	Anzahl der direkt aufgerufenen Komponenten	0	7
VG	Komplexität einer Funktion nach McCabe (Cyclomatic Number)	1	15
UJMP	Anzahl der direkten Sprünge (z. B. goto-Anweisungen)	0	0
NOUT	Anzahl der Ausgänge (z. B. return-Anweisungen)	1	1
LEVL	Anzahl der Ebenen (Schachtelungstiefe + 1)	0	5
COMF	Anzahl der Kommentare dividiert durch die Anzahl der Anweisungen	0,2	1
VOCF	Vokabel-Frequenz nach Halstead	1	5
AVGS	Durchschnittliche Länge einer Anweisung nach Halstead	1	10

Hinweise:

Befindet sich der jeweilige Wert der Metriken innerhalb des spezifizierten Bereichs, wird in der Formel zur Berechnung der Änderbarkeit der Wert "1" angenommen, sonst „0“.

⁷ Grundlage dieses Beispiels ist [DFS97].

5.2.1.2 Zuordnung der definierten Werte zu den zu berechnenden Submerkmalen

Metriken (siehe 5.2.1.1) sind Submerkmalen (Analysierbarkeit, ...) zuzuordnen.

ANALYSIERBARKEIT = $xx + xx + xx + \dots$

MODIFIZIERBARKEIT = $xx + xx + xx + \dots$

STABILITÄT = $xx + xx + xx + \dots$

TESTBARKEIT = $xx + xx + xx + \dots$

5.2.1.3 Gewichtung der Werte

Anschließend ist eine Gewichtung der zu messenden Werte vorzunehmen.

ANALYSIERBARKEIT = $\%*xx + \%*xx + \%*xx + \dots$

MODIFIZIERBARKEIT = $\%*xx + \%*xx + \%*xx + \dots$

STABILITÄT = $\%*xx + \%*xx + \%*xx + \dots$

TESTBARKEIT = $\%*xx + \%*xx + \%*xx + \dots$

5.2.2 Der Regelungsteil (vgl. 4.3.2.3.2)

Hier nun ein Beispiel für die Reihe von konkreten Festlegungen zum Referenzteil, die in einem Interview vereinbart wurden. Zugegebenermaßen ergab sich ein recht komplexes Vorgehen:

Zuerst wurden die verwendeten Metriken und deren Grenzwerte festgelegt (5.2.2.1). Dabei orientierte man sich stark an den Vorschlägen des Referenzteiles. Anschließend ordnete man den Submerkmalen der Änderbarkeit jeweils verschiedene Metriken zu (5.2.2.2), die außerdem gewichtet wurden (5.2.2.3). Abschließend errechnete man eine Mindestpunktzahl (5.2.2.4, 5.2.2.5), die mit Hilfe eines Notenschemas (5.2.2.6) in eine natürlichsprachliche Anforderung umgesetzt wurde (5.2.2.7).

5.2.2.1 Definition von Metriken und deren Grenzwerten

Folgende Zuordnungen haben sich nach Abschluß des Interviews ergeben:

Bez.	Metrik	Wertebereich	
		Min	Max
NSTAT	Anzahl der ausführbaren Anweisungen	1	40
DCAL	Anzahl der direkt aufgerufenen Komponenten	0	8
VG	Komplexität einer Funktion nach McCabe (Cyclomatic Number)	1	20
UJMP	Anzahl der direkten Sprünge (z. B. goto-Anweisungen)	0	0
NOUT	Anzahl der Ausgänge (z. B. return-Anweisungen)	1	1
LEVL	Anzahl der Ebenen (Schachtelungstiefe + 1)	0	2
COMF	Anzahl der Kommentare dividiert durch die Anzahl der Anweisungen	0,2	1
VOCF	Vokabel-Frequenz nach Halstead	1	5
AVGS	Durchschnittliche Länge einer Anweisung nach Halstead	1	15

Eine Metrik wird beim Errechnen der Änderbarkeit des fertigen Systems mit „1“ bewertet, wenn der Messwert innerhalb des Wertebereichs liegt und mit „0“, wenn nicht.

5.2.2.2 Zuordnung der definierten Metriken und deren Grenzwerte zu den Submerkmalen

Folgende Zuordnung der verfügbaren Metriken zu Submerkmalen hat sich nach Abschluß des Interviews ergeben:

$$\text{ANALYSIERBARKEIT} = \text{VG} + \text{NSTAT} + \text{COMF} + \text{AVGS}$$

$$\text{MODIFIZIERBARKEIT} = \text{AVGS} + \text{LEVL} + \text{UJMP} + \text{VOCF}$$

$$\text{STABILITÄT} = \text{NOUT} + \text{DCAL} + \text{UJMP} + \text{VOCF}$$

$$\text{TESTBARKEIT} = \text{LEVL} + \text{UJMP} + \text{VG}$$

5.2.2.3 Gewichtung der gemessenen Werte für jedes Submerkmal

Der Beitrag der einzelnen Metriken zur Bestimmung der Submerkmalswerte wurde folgendermaßen festgelegt:

$$\text{ANALYSIERBARKEIT} = 25 \cdot \text{VG} + 25 \cdot \text{NSTAT} + 25 \cdot \text{COMF} + 25 \cdot \text{AVGS}$$

$$\text{MODIFIZIERBARKEIT} = 25 \cdot \text{AVGS} + 25 \cdot \text{LEVL} + 25 \cdot \text{UJMP} + 25 \cdot \text{VOCF}$$

$$\text{STABILITÄT} = 25 \cdot \text{NOUT} + 25 \cdot \text{DCAL} + 25 \cdot \text{UJMP} + 25 \cdot \text{VOCF}$$

$$\text{TESTBARKEIT} = 33 \cdot \text{LEVL} + 33 \cdot \text{UJMP} + 34 \cdot \text{VG}$$

Damit können für jedes Submerkmal Soll-Summen errechnet werden.

5.2.2.4 Bewertung der Submerkmale

Aus den Soll-Summen leiten sich bestimmte Punktezahlen ab. Dadurch erreicht man im vorliegenden Fall, dass ein schlechtes Ergebnis einer Metrik durch gute Ergebnisse anderer Metriken kompensiert werden kann.

Folgendes Bewertungsschema wurde im Interview festgelegt:

Summe pro Submerkmal	Punktzahl für die Berechnung der Änderbarkeit
100	20
67 – 99	15
34 – 66	10
0 – 33	0

5.2.2.5 Bestimmung der geforderten Punktzahl

Nun können einzelne Anforderungen für jedes Submerkmal geschrieben werden. Alternativ lassen sich mit dem o.a. Punktebewertungsschema die Anforderungen an die Werte der Submerkmale auch zu einer einzigen Anforderungen an das Merkmal selbst zusammenfassen, was im vorliegenden Fall auch gemacht wurde:

$$\begin{aligned} \text{ÄNDERBARKEIT} = & \text{ANALYSIERBARKEIT} + \\ & \text{MODIFIZIERBARKEIT} + \\ & \text{STABILITÄT} + \\ & \text{TESTBARKEIT} \end{aligned}$$

Auch hier ist denkbar, dass einzelne Submerkmale stärker gewichtet werden als andere.

5.2.2.6 Bewertungsskala für die Änderbarkeit

Mit Hilfe einer Zuordnung von Punkten zu Noten ähnlich dem Schulnotensystem wurde erreicht, dass in der Anforderung keine Zahlenwerte sondern nur Wörter stehen.

Punkte	Note
100	Sehr gut
70 – 99	Gut
50 – 69	Befriedigend
35 – 49	Ausreichend
0 – 34	Mangelhaft

5.2.2.7 Anforderung an die Änderbarkeit

Die Anforderung an die Änderbarkeit lautete schlussendlich:

> Das System soll über eine mindestens befriedigende Änderbarkeit verfügen. <

Was „befriedigend“ bedeutet erschließt sich beim Testen oder Abnehmen des Systems aus den Festlegungen im Regelungsteil. Dabei werden die Werte für die verschiedenen Metriken ermittelt, die verschiedenen Gewichtungen, Punkte- und Notenzuordnungen vorgenommen und der Ist-Wert mit dem Soll-Wert verglichen.

6 Erfahrungen

IVENA wurde bisher in einem sehr großen Projekt mit administrativen und technischen Inhalten angewendet, das die beiden SOPHISTen begleiteten, die für die Entwicklung des Ansatzes verantwortlich zeichnen. Die Erfahrungen lassen sich wie folgt zusammenfassen:

- Die Vorgehensweise hat sich in den Schritten 1 bis 3 als sehr praktisch erwiesen. Die zu erfüllenden Aufgaben aus Kapitel 3 wurden soweit absehbar (das Projekt läuft noch) gelöst. Insbesondere die Freiheitsgrade bezüglich der spezifischen Situation in der Organisation und im Projekt sind hervorzuheben.
- Es erscheint relativ zeitaufwendig, die Referenz-Beispiel-Datenbank im Anschluss an den Erhebungsprozess zu pflegen und zu erweitern (Schritt 4). Die Qualität der Wissensaufbereitung hängt stark von der dafür zur Verfügung gestellten Zeit ab. Eine passende Werkzeugunterstützung tut Not, um den Aufwand möglichst gering zu halten.
- Der Prozess vor Ort wird wirkungsvoll unterstützt, wenn Datenbanken sowohl für IVENA als auch für die schließlich resultierenden Anforderungen verwendet werden. Insbesondere sind dazu Werkzeuge geeignet, die die Arbeit im Team erleichtern (Workflow- und Unterschriftenkonzepte).
- Abschließend ist hervorzuheben, dass die am Erhebungsprozess beteiligten Personen noch immer den größten Einfluss auf die Qualität der Ergebnisse haben. Wie jeder methodische Ansatz auch ist IVENA nicht als Patentrezept zu verstehen, um Qualität „automatisch“ zu erzielen.

In einem weitere Projekt ähnlicher Größe steht in naher Zukunft die Erhebung der nichtfunktionalen Anforderungen an. Es bestehen Chancen, das daraus gewonnene zusätzliche Praxiswissen bis zum Herbst 1999 präsentieren zu können.

7 Quellenverzeichnis

[DIN94] Norm DIN 66272: *Bewerten von Softwareprodukten: Qualitätsmerkmale und Leitfaden zu ihrer Verwendung*. Ausg. Oktober. 1994. Deutsches Institut für Normung e.V.

[DFS97] Pieper, R.: *Definition von Software-Qualitätsmerkmalen*, internes Papier der Deutschen Flugsicherung GmbH, 1997

Kurzbiografien



Rolf Götz, geboren 1971 in Burgau bei Augsburg ist seit drei Jahren SOPHIST und dabei als Systemanalytiker und Berater für verschiedene Auftraggeber tätig. Spezialisiert auf Methoden und Vorgehensmodelle des Requirements Engineering leitete, schulte und unterstützte er mehrere Großprojekte in der Erhebung funktionaler und nichtfunktionaler Anforderungen. In einem zweiten Tätigkeitsfeld unterstützt er Organisationen bei der Konzeptionierung und Implementierung von QM-Systemen.

Seine Arbeitsmethodik umfasst u.a. natürlichsprachliche und objektorientierte Analysen sowie bestimmte Bereiche der Transaktionsanalyse, des Neuro-Lingustischen Programmierens (NLP) und der analytischen Philosophie.



Heiko Scharnweber wurde 1970 in Geesthacht bei Hamburg geboren. Er war acht Monate in Südamerika in der Organisationsberatung tätig und ist nunmehr seit einem Jahr SOPHIST. Zu seinen Aufgabenschwerpunkten gehört die Unterstützung von Kunden bei deren Systemanalysen. Spezialisiert auf Methoden und Vorgehensmodelle des Requirements Engineering begleitete er mehrere Projekte in der Erhebung und Optimierung funktionaler und nichtfunktionaler Anforderungen.

Herr Scharnweber ist zur Zeit in einem internationalen Flugsicherungsprojekt für die Spezifikation des Bereichs Flugdatenverarbeitung verantwortlich und maßgeblich an der Entwicklung und Umsetzung von IVENA beteiligt.

Kontakt

SOPHIST Gesellschaft für innovatives Software - Engineering mbH

Vordere Cramergasse 11 - 13
90478 Nürnberg

Tel.: +49 (9 11) 4 09 00 – 0

Fax: +49 (9 11) 4 09 00 – 99

mailto: sophist@sophist.de

<http://www.sophist.de>