

- **Seiten 32–33** Wie abstrakt dürfen Anforderungen sein?
Seiten 34–35 eHealth Schweiz – ein Markt mit grossen Chancen
Seiten 36–37 SaaS: Lizenzmodelle im Wandel



BILD: FOTOLIA

Wie abstrakt dürfen Anforderungen sein?

Formale Spezifikationen sind häufig nicht nachvollziehbar und verfehlen ihren Zweck. Der Mensch orientiert sich an Beispielen – ein hilfreiches Konzept zur Erarbeitung aussagekräftiger Spezifikationen.

→ VON MICHAEL RICHTER

Spezifikationen beschreiben die Anforderungen und Rahmenbedingungen für die Entwicklung einer neuen Software oder eines neuen Produkts. Sie bilden die Basis für die Implementierung einer neuen Lösung. Zudem sind sie ein wichtiges Kontrollinstrument: Anhand der Spezifikationen können Beteiligte und Auftraggeber überprüfen, ob alle benötigten Funktionen und Anforderungen enthalten und korrekt beschrieben sind.

Michael Richter ist Berater für Usability und Requirements Engineering bei Zühlke und Lehrbeauftragter an der Hochschule für Technik Rapperswil.

Oft sind die Beschreibungen und Modelle aber dermassen abstrakt, dass ihre Bedeutung für die Praxis kaum mehr vorstellbar ist. Dies birgt die Gefahr, dass falsche oder fehlerhafte Anforderungen erst spät im Entwicklungsprozess aufgedeckt werden; Mängel werden erst erkannt, wenn die Anwendung implementiert und damit sichtbar ist. Nachträgliche teure Änderungen sind die Folge. Zudem besteht ein grosses Risiko, dass sich die entwickelte Lösung schliesslich nicht in die Geschäftsprozesse integriert und die Benutzer ihre Aufgaben nicht wie geplant oder gar nicht erfüllen können.

Es ist und bleibt Wunschdenken, dass eine detaillierte Auflistung von Anforderungen ausreicht, um eine Software-Anwendung vollstän-

dig zu definieren. Die Verantwortlichen für die Entwicklung müssen in jedem Fall Entscheidungen treffen. Im besseren Fall stützen sie sich dabei auf ihr gutes Verständnis der Benutzerbedürfnisse, im schlechteren Fall basieren die Entscheidungen auf falschen Annahmen. Eine gute Spezifikation sollte deshalb nicht nur formale Anforderungen enthalten, sondern auch ein Verständnis für die Anwendung im Einsatz vermitteln.

Selbstverständlich ist es nicht möglich, alle erdenklichen Fälle der täglichen Arbeit auszuformulieren, welche durch das neue System unterstützt werden sollen. Es ist auch nicht sinnvoll, technische Details der vorgesehenen Lösung vorwegzunehmen. Generalisierungen und Abstraktionen in einer Spezifikation sind deshalb durchaus notwendig.

SPEZIFIKATION AM BEISPIEL

Abstrakt und dennoch konkret und verständlich – ein Widerspruch? Nicht unbedingt. Dazu eine Betrachtung des menschlichen Denkens: Konzepte, Abläufe, Funktionen – diese Konstrukte kann unser Gehirn nur mithilfe von Beispielen verarbeiten. Wir gewinnen ein Verständnis für einen Sachverhalt anhand von Einzelfällen. Erst damit wird in einem zweiten Schritt die Abstraktion möglich. Genauso geht ein Analyst vor, der eine Spezifikation für ein neues System schreibt: Er untersucht verschiedene einzelne Situationen und generalisiert die erhaltenen Informationen zu einem abstrakten

MEHR ZUM THEMA

Zühlke → www.zuehlke.com/requirements-engineering
 Usability Engineering kompakt → www.benutzbar.ch
 Michael Richter → www.michaelrichter.ch

KNOW-HOW FÜR CIOs

Projektmanagement bei der Software-Entwicklung
 → www.cio-channel.ch, Suche: 1894618

Modell. Hat der Leser der Spezifikation allerdings nur dieses abstrakte Modell zur Verfügung, kann er die konkreten Einzelfälle nicht herleiten. Denn diese Informationen sind in der Abstraktion nicht mehr enthalten. Umgekehrt gelingt es dem Menschen jedoch gut, aus Beispielen Regeln und Zusammenhänge abzuleiten. Diese Fähigkeit kann bei der Erarbeitung von Spezifikationen genutzt werden. Beispiele sind hervorragende Instrumente, um Zusammenhänge oder Abläufe aufzuzeigen. In den meisten Fällen genügt es, wenige stellvertretende Beispiele – etwa den typischen Arbeitsablauf eines typischen Benutzers – möglichst realitätsnah wiederzugeben.

Wie aber können Beispiele zu diesem Zweck effizient erarbeitet und eingesetzt werden? Was zeichnet ein gutes Beispiel aus? Aus dem Usability Engineering sind einige praktische Methoden bekannt, die ein beispielorientiertes Vorgehen in der Software- und Produktentwicklung unterstützen und Antworten auf diese Fragen liefern können.

PERSONAS: ZUKÜNFTIGE BENUTZER

Personas sind beispielhafte, prototypische Beschreibungen von zukünftigen Anwendern eines Systems. Sie werden aufgrund von Informationen über die vorgesehenen Benutzergruppen erarbeitet. Die Beschreibungen widerspiegeln die Eigenschaften der Benutzer, die für die neue Software oder das neue Produkt relevant sind, und machen Benutzerbedürfnisse deutlich.

Im Rahmen eines Projekts entstehen meist mehrere Personas, welche die Anforderungen unterschiedlicher Benutzergruppen repräsentieren. Personas bilden eine hervorragende Basis für das weitere Vorgehen, indem sie die impliziten Annahmen des Projektteams über die zukünftigen Benutzer explizit machen und objektive Kenntnisse über die Benutzergruppen anschaulich darstellen.

SCENARIEN: VORGESEHENE PROZESSE

Anwendungsszenarien oder kurz Szenarien schlagen die Brücke zwischen den Anforderungen und dem Entwurf einer neuen Lösung. Ein Szenario beschreibt in Form eines realistischen Beispiels, wie ein Benutzer mit dem geplanten System interagieren wird. Mit einfachen Sätzen wird ein konkreter Ablauf aus Benutzersicht dargestellt. Dabei sollte wie auch bei Personas mehr auf inhaltlich richtige Aussagen als auf deren formale Korrektheit geachtet werden.

Ein grosser Vorteil von Szenarien ist ihre leichte Verständlichkeit. Sie können von verschiedenen Stellen wie Auftraggeber, Benutzer oder Entwickler bereits zu einem frühen Zeitpunkt überprüft, ergänzt oder korrigiert werden.

Szenario 1: Aufnehmen eines Schadensfalls

- Es ist 15:00. Bei Jakob S. klingelt das Telefon.
- Auf seinem Bildschirm erscheinen neben der Telefonnummer Name und weitere Angaben des anrufenden Kunden.
- Jakob nimmt den Anruf entgegen und begrüßt den ungeduldigen Kunden, der eine zerbrochene Fensterscheibe melden möchte.
- Da für diesen Kunden verschiedene Versicherungsverträge bestehen, wählt Jakob die entsprechende Police aus der Übersicht aus.
- Danach nimmt Jakob den Schadensfall des Kunden auf.
- ...

BILDER: ZÜHLKE

Szenarien beschreiben einen typischen Ablauf mit der vorgesehenen neuen Lösung

Die Macht des guten Beispiels

Die Spezifikation eines neuen Systems darf nur wenig Interpretationsspielraum zulassen. Um formal korrekt zu formulieren, muss zwangsläufig über verschiedene mögliche Fälle generalisiert werden. Die Gefahr dabei ist, dass die Realität auf der Strecke bleibt.

Ein Beispiel dagegen ist weder eindeutig noch vollständig. Interessanterweise ist das menschliche Gehirn allerdings hervor-

gend dafür geeignet, aus Beispielen Regeln abzuleiten. Mittels weniger, guter Beispiele kann ein Sachverhalt oft schneller, umfassender und manchmal sogar präziser dargestellt werden als mit einer formalen Spezifikation.

Buchauszug: M. Richter & M. Flückiger, «Usability Engineering kompakt – Benutzbare Software gezielt entwickeln», Spektrum Akademischer Verlag, Heidelberg, 2007

MOCK-UPS: ANWENDUNG IM EINSATZ

Um die Abläufe aus Benutzersicht darzustellen, können mit User-Interface-Prototypen erste Dialogschritte mit einfachen Mitteln wie Papierskizzen oder Drahtmodellen (Wireframes) umgesetzt und von ausgewählten Benutzern und weiteren Stakeholdern geprüft werden. Solche Attrappen der Benutzerschnittstelle werden auch Mock-ups genannt.

Ein Mock-up zeigt mithilfe der Benutzerschnittstelle, wie ein System oder Produkt verwendet wird. Dabei geht es noch nicht darum, das System zu entwerfen, sondern die Funktio-

nen mit einem einfachen Hilfsmittel erfahrbar zu machen. Mock-ups machen die Anforderungen greifbar, die in den Use Cases in Textform oder mit grafischen Modellen festgehalten sind. So lassen sich zum Beispiel Navigation und Interaktion anschaulich zeigen. Die User-Interface-Prototypen dienen zudem als einfach verständliche, gemeinsame Sprache zwischen Benutzern, Auftraggebern und Entwicklern.

Bei der Erarbeitung der Spezifikationen können Mock-ups den Funktionsumfang illustrieren, die Funktionsweise deutlich machen oder User-Interface-Elemente spezifizieren. Zudem sind Mock-ups ein Hilfsmittel, um den Entwicklungsaufwand abzuschätzen.

Work on TC Parts		OK
Part	exch.	up.
Bearings	<input type="checkbox"/>	
Rotor	<input type="checkbox"/>	
Compressor	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Turbine	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Casing	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>
Parts	Spare	Remarks

Mock-ups: Abläufe aus Benutzersicht

DAS BESTE AUS ZWEI WELTEN

Die beispielorientierten Methoden aus dem Usability Engineering und gängige Vorgehensweisen aus dem Software Engineering ergänzen sich in der Praxis ausgezeichnet:

→ Die Reflektion am konkreten Beispiel erlaubt es Auftraggebern und Benutzern, Anforderungen für wichtige Anwendungssituationen zu vergegenwärtigen, zu überprüfen und zu ergänzen.

→ Personas und Szenarien illustrieren die Anwendung im realen Kontext und dienen als Ergänzung zu den formalen Use Cases. Sie vermitteln den Entwicklern ein Verständnis der Abläufe und Zusammenhänge.

→ Mit Mock-ups kann die Interaktion des Benutzers mit der neuen Lösung modelliert und optimiert werden. Die Entwickler können dabei die technischen Anforderungen überprüfen. ←