

Die Zusammenführung von Bedürfnissen und Lösungen

Sixten Schockert

Universität Stuttgart, Betriebswirtschaftliches Institut, Lehrstuhl für ABWL und Wirtschaftsinformatik II
schockert@wius.bwi.uni-stuttgart.de

Beitrags-Kategorie, Zielgruppe des Beitrags

Dieser Beitrag fokussiert auf die Zusammenführung von Bedürfnissen der Stakeholder mit ihren möglichen Lösungen. Diese dient der begründeten Ableitung von Produktanforderungen als genaue und abgestimmte Entwicklungsvorgaben und stellt damit eine elementare Aufgabe im Requirements Engineering (RE) dar. Es werden methodische Ideen zur Bewältigung dieser Aufgabe vorgestellt. Der Beitrag spricht sowohl Praktiker als auch Wissenschaftler im Umfeld des RE an, insbesondere solche, die an methodischen Verbesserungen bzw. Alternativen interessiert sind.

Motivation der Fragestellung und Kontext

Lauenroth hinterfragt in einem kürzlich erschienenen Beitrag die alten RE-Weisheiten der zwingenden Trennung von Problemen und Lösungen sowie der möglichst lösungsneutralen Anforderungen. Er bringt seine Gedanken in folgender Aussage auf den Punkt:

„Das Requirements Engineering als gestaltende Disziplin ist dafür verantwortlich, dass mit allen relevanten Stakeholdern die Problemstellungen, die eine Software lösen soll, definiert und abgestimmt werden, Lösungen für die identifizierten Problemstellungen definiert und abgestimmt werden (sowie) Anforderungen an die Lösungen definiert und abgestimmt werden.“ [1]

Er konstatiert damit sogar mehr als die Trennung von Problemen und Lösungen, er fordert eine Dreiteilung von Problemen, Lösungen und Anforderungen. Nötig für die Vorgabe von möglichst konkreten Produktanforderungen für die Entwicklung sind das Verständnis von Problemen und möglichen Lösungen zu den Problemen. Oder anders gesagt: Produktanforderungen können nur dann zuverlässig formuliert werden, wenn zuvor Probleme bzw. allgemein die Bedürfnisse der relevanten Stakeholder von allen Beteiligten verstanden und die möglichst besten Lösungen für diese gefunden wurden. Erst dann lassen sich Produktanforderungen im Sinne ihrer Definition [2] von Bedingungen oder Fähigkeiten zur Lösung von Problemen bzw. zur Erreichung von Zielen (eben zur Bedürfnisbefriedigung) festlegen.

Dies verlangt allerdings nicht nur die Trennung sondern vor allem die Zusammenführung von Bedürfnissen mit entsprechenden Lösungen. Sie liefert die notwendige Grundlage bzw. Begründung für die zu definierenden Anforderungen. Entsprechend sollte diese wichtige Aufgabe im RE systematisch und nachvollziehbar für alle Beteiligten erfolgen.

Behandelte Fragestellung

Wie kann nun diese begründete Zusammenführung von Bedürfnissen und Lösungen erfolgen? Welche Möglichkeiten bietet das RE für diese Aufgabe?

Lösung und Ergebnisse

Da es sich um eine elementare Aufgabe des RE handelt, sollte man annehmen, dass diese Aufgabe in Theorie und Praxis ausgiebig behandelt wird. Dem ist allerdings nicht so, sie wird eher implizit verfolgt.

So ist bereits die Einordnung dieser Aufgabe in die Haupttätigkeiten des RE d. h. Anforderungen ermitteln, (natürlichsprachig und modellbasiert) dokumentieren, prüfen und abstimmen sowie verwalten (hier gemäß [3]) nicht eindeutig. Prinzipiell kann sie jeder dieser Tätigkeiten zugeordnet werden. Da es um Bedürfnisse der Stakeholder geht, liegt es nahe sie als Teil der Anforderungsermittlung zu begreifen. Es gibt eine Fülle von Ermittlungstechniken (z. B. [4]), doch gibt es welche, die explizit Bedürfnisse der Stakeholder aufnehmen und von möglichen Lösungen trennen sowie dann in einem nächsten Schritt begründet (wieder) zusammenzuführen, um auf diese Weise zu Produktanforderungen zu gelangen?

Das goal-oriented Requirements Engineering [5] liefert hier einen Ansatz indem es Ziele als Absichtserklärungen, welche die Software erfüllen soll, in den Mittelpunkt der Betrachtung stellt. Damit wird zwar die Begründung von Produktanforderungen adressiert, deren Kopplung zu den Lösungen bzw. konkreten Entwicklungsvorgaben wird allerdings nicht deutlich expliziert. Ähnlich ist es beim aktuell sehr in Diskussion befindlichen Design Thinking (z. B. [6]) bei dem die Auswahl der „besten“ Ideen und in ersten Prototypen zu visualisierenden Lösungen für identifizierte Bedürfnisse ebenfalls nicht transparent erfolgt.

Eine begründete Zusammenführung oder allgemein ein „in Beziehung bringen“ von zwei Artefakten ist theoretisch nur möglich durch ein Darstellungsmittel, dass diese Relation zum Ausdruck bringt. Dies können z. B. zweidimensionale Matrixdiagramme, Beziehungsdiagramme ähnlich Ursache-Wirkungs-Diagrammen [7] oder auch entsprechende natürlichsprachige Zweck-Mittel-Formulierungen sein. Ein Blick in bewährte RE-Literatur (z. B. [3], [4], [8], [9]) zeigt aber, dass zumindest diese Diagrammtypen nicht gebräuchlich sind. Hier ist die Dominanz von modellbasierter Dokumentation mit UML und Co. eindeutig. Deren Fokus ist die präzise Beschreibung der Anforderungen, jedoch nicht deren Begründung. Dies gilt im Allg. auch für die

natürlichsprachige Dokumentation von Anforderungen mit einer Anforderungsschablone wie in [9].

Insbesondere bei agiler Entwicklung haben sich user stories als natürlichsprachige Anforderungsbeschreibungen durchgesetzt, so z. B. bei Scrum zur Formulierung der Einträge im Product Backlog. Bei konsequenter Handhabung des erweiterten Templates „als <Stakeholder/Rolle> möchte ich <Ziel/Wunsch>, um <Nutzen/Vorteil> zu erreichen“ [10] lässt sich mit ihnen auch eine Beziehung zwischen Bedürfnissen (= Nutzen/Vorteil) und Lösungen (= Ziel/Wunsch) abbilden. Solche „begründeten“ user stories berücksichtigen explizit den Grund für einen Lösungswunsch in der Beschreibung einer Anforderung. Sie repräsentieren damit einen Schritt in die Richtung der Zusammenführung von Bedürfnissen und Lösungen, auch wenn die Angabe des Grundes eher der Information und weniger der Handlungsleitung zur Definition von konkreten Produktanforderungen dient [9]. Überdies bilden user stories nur einfache 1:1-Beziehungen zwischen Bedürfnissen und Lösungen ab, ggf. vorhandene 1:n bzw. n:m-Beziehungen sind nur schwer bzw. kompliziert darstellbar. So werden sowohl Lösungen, die mehrere Bedürfnisse befriedigen als auch mögliche alternative Lösungen für dasselbe Bedürfnis nicht systematisch berücksichtigt.

Gerade die Darstellung solcher n:m-Beziehungen ist eine Stärke von Ansätzen basierend auf der Produktplanungsmethode Quality Function Deployment (QFD) [11]. QFD trennt konsequent zwischen Bedürfnissen und Lösungen, beurteilt die Lösungen im Hinblick auf ihre Nutzeffekte für die Stakeholder, ermittelt auf diese Weise die wichtigsten Lösungen, die dann weiter bewertet und zu konkreten Produktanforderungen verdichtet werden. Das alles geschieht im Wesentlichen mit dem einfachen Mittel einer Matrixdarstellung zur Gegenüberstellung von Problem- und Lösungsraum (Abb. 1).

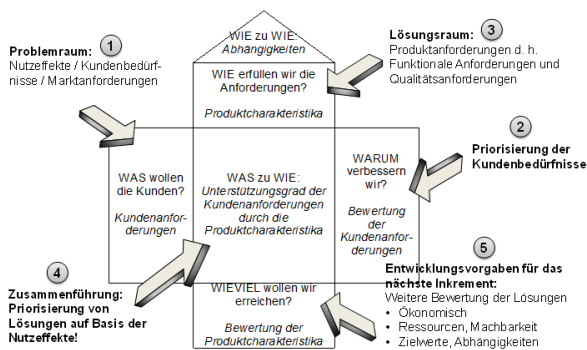


Abb. 1: Schematische Matrixdarstellung in QFD

Allerdings ist QFD im Sinne der Kundenorientierung zwar als effektiv aber vor allem auch als sehr komplex bekannt. Gerade die Zeit und der Aufwand zur Erstellung einer solchen Beziehungsmatrix können bei vielen Bedürfnissen und Lösungen unverhältnismäßig im Verhältnis zum Nutzen erscheinen. Neuere methodische Ansätze, wie die inkrementell wachsende Matrix oder die Maximum Value Table [12], reduzieren diesen

Aufwand allerdings erheblich in dem sie nicht direkt alle n:m-Beziehungen betrachten sondern zuerst auf die Wichtigsten fokussieren.

Gleichzeitig bleibt allerdings der unschätzbare Vorteil erhalten, dass diese Ansätze die Begründung für die Anforderungen explizit und für alle Beteiligten transparent machen. Überdies dient sowohl das Füllen der einzelnen Matrixbestandteile (Abb. 1) als auch die Dokumentation selbst als wertvolles Kommunikationsmittel, nicht nur innerhalb der Ermittlung sondern auch während des Prüfens und der Abstimmung von Anforderungen. Und auch die insb. bei agiler Entwicklung notwendigen Priorisierungstätigkeiten werden inhärent unterstützt, so wie z. B. die Analyse des Product Backlogs in Scrum zur Festlegung der im nächsten Inkrement zu berücksichtigenden Anforderungen [12].

Was ist neu?

Dieser Beitrag macht eine notwendige, aber oftmals nur implizit durchgeführte Tätigkeit im RE explizit: die Zusammenführung von Bedürfnissen und Lösungen als Begründung von Produktanforderungen. Der Beitrag zeigt methodische Ansätze zur Bewältigung dieser Aufgabe auf und offenbart damit gleichzeitig, dass diese Tätigkeit oft nur implizit durchgeführt wird. Allgemeiner stellt der Beitrag die grundsätzliche Frage nach der Rechtfertigung von Produktanforderungen. Werden sie begründet und wenn ja, wie werden sie begründet? Bzw. gemäß Lauenroths Plädoyer für RE als Gestaltungsdisziplin [1] formuliert: Werden die Produktanforderungen mit Bedürfnissen und deren Lösungen begründet?

Ihre bisherigen Arbeiten/Vorträge über dieses Thema

Schockert, S.; Herzwurm, G.: Das Business setzt die Prioritäten?! In: Engstler, M; Fazal-Baqaie, M.; Hanser, E.; Mikusz, M.; Volland, A. (Hrsg.): Projektmanagement und Vorgehensmodelle 2015. Lecture Notes in Informatics (LNI) – Proceedings, Volume P-250. Bonn 2015. (erscheint noch im Oktober)

Referenzen

- [1] Lauenroth, K.: Die Legende von der lösungsneutralen Anforderung oder warum das Requirements Engineering eine Gestaltungsdisziplin ist! In: Object Spectrum – Online Themenspecial Requirements Engineering 2015.
- [2] IEEE (1990). Standard Glossary of Software Engineering Terminology. IEEE Std 610.12-1990 bzw. z. B. auch Glinz, M.: A Glossary of Requirements Engineering Terminology, Version 1.6., May 2014.
- [3] Pohl, K.; Rupp, C.: Basiswissen Requirements Engineering. Heidelberg 2015.
- [4] Pohl, K.: Requirements Engineering. Heidelberg u.a. 2010.

- [5] van Lamswerde, A.: Requirements Engineering. From system goals to UML models to software specifications. Wiley 2009.
- [6] Plattner, H., Meinel, C., Leifer, L.: Design Thinking – Understand, Improve, Apply. Springer, Heidelberg 2011.
- [7] Mizuno, S.: Management for Quality Improvement: The 7 New QC Tools. Cambridge, MA, 1988.
- [8] Ebert, C.: Systematisches Requirements Engineering. Anforderungen ermitteln, dokumentieren, analysieren und verwalten. Heidelberg 2014.
- [9] Rupp, C.: Requirements-Engineering und –Management. München 2014.
- [10] Cohn, M.: User Stories: für die agile Software-Entwicklung mit Scrum, XP u.a. mitp, Bonn 2010.
- [11] Herzwurm, G.; Schockert, S.; Mellis, W.: Joint Requirements Engineering. QFD for Rapid Customer-Focused Software and Internet Development, Vieweg – Gabler, Braunschweig – Wiesbaden 2000.
- [12] Schockert, S.; Herzwurm, G.: Das Business setzt die Prioritäten?! In: Engstler, M; Fazal-Baqaie, M.; Hanser, E.; Mikusz, M.; Volland, A. (Hrsg.): Projektmanagement und Vorgehensmodelle 2015. Lecture Notes in Informatics (LNI) – Proceedings, Volume P-250. Bonn 2015. (erscheint noch im Oktober)