

Wissensarten und Techniken im Anforderungsmanagement

Matthias Goeken

Institut für Wirtschaftsinformatik
Philipps-Universität Marburg
35032 Marburg
goeken@wiwi.uni-marburg.de

Abstract: Das Anforderungsmanagement ist eine wissensintensive Aktivität bei der Entwicklung von Anwendungssystemen. In diesem Beitrag werden Wissensarten und typische Probleme im Rahmen des Anforderungsmanagements diskutiert. Techniken, die die Anforderungserhebung unterstützen, werden den Wissensarten und Problemen zugeordnet, sodass sich ein Kontingenzmodell zur Auswahl von Techniken im Anforderungsmanagement ergibt.

1 Einleitung

Anwendungssysteme so zu entwickeln, dass sie den Erwartungen der zukünftigen Benutzer entsprechen, stellt nach wie vor eine der wichtigsten Herausforderungen im Entwicklungsprozess dar. Typischerweise wird der Gesamtprozess der Systementwicklung in die Phasen Vorphase, Analyse (Anforderungsmanagement), Entwurf, Realisierung und Einführung zerlegt [SH05]. Insbesondere die frühen Phasen sind durch einen intensiven Austausch von Informationen und Wissen zwischen den beteiligten und von dem zu erstellenden System betroffenen Stakeholdern gekennzeichnet.

Dieser Aspekt wird jedoch von vielen Entwicklungsmethoden vernachlässigt, speziell von solchen, die im Wesentlichen eine Modellierungssprache (Notation) bereitstellen. Solche Methoden betrachten allzu häufig nicht die Fragen, wie das Wissen erlangt werden kann, das nötig ist, um ein Anwendungssystem so zu entwickeln, dass es die Erwartungen und Anforderungen der Benutzer und Anwender erfüllt. Es ist demnach keineswegs gewährleistet, dass das Wissen über die fachlichen Anforderungen im ausreichenden Maße Eingang in die im Entwicklungsprozess erstellten Modelle findet.

Die Angemessenheit und Akzeptanz eines Systems hängt jedoch im hohen Maße davon ab, dass die fachlichen Anforderungen tatsächlich umgesetzt werden, d. h. „If you don't understand the user's requirements, it doesn't matter how you code it.“ [Yo92]

Im Folgenden wird – nach Darstellung einiger typischer Probleme des Anforderungsmanagements – der Frage nachgegangen, welche Wissensarten für die Systementwicklung relevant sind. Darauf aufbauend werden Techniken, die die Anforderungserhebung unterstützen sollen, sowohl den Wissensarten als auch den typischerweise auftretenden Problemen zugeordnet, sodass sich ein Kontingenzmodell ergibt, das in Abhängigkeit einer konkreten Situation eine begründete Technikauswahl unterstützen kann.

2 Problembereiche im Anforderungsmanagement

Ziel des Anforderungsmanagements ist es, die Fähigkeiten und Bedingungen zu spezifizieren, die ein Benutzer benötigt und die ein System erfüllen bzw. besitzen muss, um ein fachliches Problem zu lösen [IE90]. Üblicherweise wird das Anforderungsmanagement in Aktivitäten zerlegt, wobei sich häufig eine Unterscheidung von Anforderungserhebung (Requirements Elicitation¹), Analyse, Spezifikation und Verifikation/Validierung findet [HD03]. Requirements Elicitation beschreibt die Aktivität des Anforderungsmanagements, die für das Auffinden, das Verstehen und das Sammeln von Informationen über Bedarfe und Anforderungen verantwortlich ist.

Die Hindernisse, die bei dieser Aktivität auftreten, beschreiben Valusek/Fryback als „obstacles which can be categorized as WITHIN an individual user, AMONG users, and BETWEEN the individual user and those responsible for system development“ [VF85]. „*Within-Obstacles*“ beschreiben das Problem, dass der Benutzer sich über seine Anforderungen selbst nicht im Klaren ist oder dass er sein Wissen nicht angemessen artikulieren kann, bspw. weil es sich um implizites Wissen handelt. „*Among-Obstacles*“ entstehen, weil verschiedene Benutzer und Anwender abweichende und möglicherweise gegensätzliche Vorstellungen und somit Anforderungen haben. Die „*Between-Obstacles*“ ergeben sich dadurch, dass Entwickler und Benutzer unterschiedliche Sprachen sprechen. Während der Anwender in der gewachsenen, natürlichen Sprache kommuniziert, werden von Entwicklern künstliche, konstruierte Sprachen (z. B. Diagrammsprache) wegen ihrer formalen Eindeutigkeit bevorzugt. Hieraus resultiert eine „Sprachlücke“, die zu Kommunikationsproblemen und damit zu Problemen bei der Übertragung entwicklungsrelevanten Wissens führen kann [Or95].

Um zu einer Spezifikation zu gelangen, ist insbesondere die Überwindung der genannten „Obstacles“ von entscheidender Bedeutung. Im übernächsten Abschnitt werden Techniken diskutiert, die ihre Überwindung ermöglichen sollen.

3 Wissensarten im Anforderungsmanagement

Während das entwicklungsrelevante fachliche Wissen i. d. R. bei den zukünftigen Benutzern liegt, bringen die Entwickler Wissen über die technologischen Möglichkeiten in den Entwicklungsprozess ein.

Bei Kensing/Munk-Madsen findet sich eine erweiterte Systematisierung verschiedener *Wissensarten* im Entwicklungsprozess [KM95], die sich aufgrund von zwei Dimensionen ergeben (Abbildung 1). Zum einen unterscheiden sie zwischen konkretem und abstraktem Wissen, wobei ersteres den konkreten Erfahrungen der Benutzer und Analysten entspricht und letzteres die Beschreibung und Darstellung mittels Modellen und Spezifikationen beinhaltet. Zum anderen legen sie drei entwicklungsrelevante Wis-

¹ Leider findet sich keine befriedigende Übersetzung für „Requirements Elicitation“. „Elicit“ wird mit „entlocken“ und „herauskitzeln“ übersetzt.

sensgebiete zugrunde. Hierbei wird das Wissen über die aktuellen Tätigkeiten des Benutzers, über die technologischen Möglichkeiten sowie über das zu erstellende System unterschieden. Das Wissen über das zu erstellende System ergibt sich im Entwicklungsprozess aus den beiden anderen Wissensgebieten.

		Wissensgebiete		
		Aktuelle Tätigkeiten und Aufgaben des Benutzers	Neues, zu erstellendes Anwendungssystem	Technologische Möglichkeiten
Wissensebenen	Abstraktes Wissen	II Relevante Strukturen der aktuellen Arbeitssituation des Benutzers - Modell des Istzustands -	V Visionen und Modelle - Modell des Sollzustands -	IV Überblick über technologische Möglichkeiten
	Konkrete Erfahrung	I Konkrete Erfahrungen mit der aktuellen Tätigkeit des Benutzers - Domänenwissen -	VI Konkrete Erfahrungen mit den zu erstellenden System	III Konkrete Erfahrungen mit den technologischen Möglichkeiten

Abbildung 1: Wissensarten im Entwicklungsprozess (Nach [KM95])

Auf Seiten der Benutzer ist typischerweise Domänenwissen (I) vorhanden, während Analysten über Wissen in III und IV verfügen. Infolgedessen muss Wissen zwischen den Beteiligten übertragen werden (bspw. bzgl. I vom Benutzer zum Analysten und bzgl. III vom Analysten zum Benutzer). Darüber hinaus muss Wissen der Arten II (abstraktes Domänenwissen), V, und VI gemeinsam entwickelt werden. Abstraktes Wissen über die aktuelle Arbeitssituation liegt häufig nicht vor, ebenfalls fehlen auf beiden Seiten konkrete Erfahrungen und abstraktes Wissen über das zu erstellende System. Der Analyst muss daher geeignete Techniken heranziehen, die die Artikulation, die Übertragung und die Entwicklung von Wissen in den genannten Bereichen unterstützen.

4 Kontingenzmodell zur Auswahl von Techniken

Mittels so genannter Kontingenzmodelle lassen sich in Abhängigkeit von der Projektsituation geeignete Techniken bestimmen, die spezifische Aufgaben im Entwicklungsprozess und speziell im Anforderungsmanagement unterstützen können (zu älteren Kontingenzmodellen vgl. bspw. [Da82, ST88]). Die Projektsituation soll hier durch das benötigte Wissen auf der einen Seite und durch die identifizierten „Obstacles“ auf der anderen Seite gekennzeichnet werden. Die Zuordnung von Techniken zu der so charakterisierten Projektsituation erfolgt aufgrund ihrer spezifischen Eigenschaften und orientiert sich an der Diskussion in der Literatur [bspw. By92; DS97; St96; SS97].

Bei der Übertragung von Wissen über die Tätigkeiten der Benutzer (I) treten gegebenenfalls die oben diskutierten „Within-Obstacles“ (Artikulationsprobleme) auf. Die nach Kensing/Munk-Madsen für diese Aufgabe geeigneten Techniken ergeben sich aus Abbildung 2. Ebenfalls finden sich dort für die übrigen Wissensarten Techniken, die die skizzierten Aufgaben bei der Entwicklung von Wissen unterstützen.

Von besonderer Bedeutung ist im Entwicklungsprozess das Sollkonzept (Gebiet V), welches aufgrund der identifizierten Schwächen in II und der technologischen Möglichkeiten (III & IV) erstellt wird. Hierbei können sich die ebenfalls oben dargestellten

Kommunikationsprobleme, die aus der Sprachlücke resultieren (Between-Obstacles), ergeben. Zusätzlich sind in Abbildung 2 die identifizierten Hindernisse der Anforderungserhebung den vorgeschlagenen Techniken zugeordnet [By92; DS97].

	Relevante Wissensarten						Obstacles
	I Erfahrungen mit der Tätigkeit des Benutzers	II Strukturen der aktuellen Arbeitssituation	III Erfahrung mit technologischen Möglichkeiten	IV Überblick über technologische Möglichkeiten	V Visionen und Modelle	VI Erfahrungen mit zu erstellenden System	Überwindung spezifischer Hindernisse
Fragebogen	•						-
Interview	•						B
Kritische Erfolgsfaktoren	•	•				•	W, A*, B
Prototyping			•	•	•	•	W, B
Rich pictures	•	•					W, A*, B
Laddering-Fragen	•	•					W, B
Szenarien	•				•		W, B
Card games/sorting	•					•	W, A*, B
Gruppendiskussion	•				•		A
Brainstorming		•			•		W, A
Cognitive Mapping	•	•					W, B
Mediation	•				•		A
Beobachtung	•						W
Dokumentenanalyse	•	•					-
Zukunftsworkshops		•			•		W, A

•: Unterstützt; W: Within-Obstacle; A: Among-Obstacle; B: Between-Obstacle; *: wenn Anwendung in Gruppe; - : keine Unterstützung

Abbildung 2: Kontingenzmodell zur Technikwahl (zu den Techniken [SS97; By92; DS97; St96])

Mithilfe der dargestellten Zusammenhänge lässt sich in einem zweistufigen Vorgehen eine Technik vor dem Hintergrund der aktuellen Projektsituation bestimmen:

- Die Systematisierung nach Wissensarten erlaubt in einem ersten Schritt eine Eingrenzung der relevanten Techniken vor dem Hintergrund der aktuell anstehenden Aufgabe im Entwicklungsprozess.
- In einem nächsten Schritt können zu überwindende Hindernisse identifiziert werden (Within-Obstacles/Artikulationsprobleme, Among- sowie Between-Obstacles).

Auf diesem Wege können darüber hinaus Technikkombinationen begründet werden. So lässt sich bspw. ablesen, dass in Interviews der Einsatz von Prototypen als Stimulus die Anforderungserhebung für ein zu erstellendes System unterstützt (Wissensart V) und die Überwindung von Artikulationsproblemen ermöglichen kann (Within-Obstacles).

5 Fazit und Ausblick

Das vorgestellte Kontingenzmodell erlaubt die Auswahl von Techniken zur Unterstützung des Anforderungsmanagements im Rahmen der Systementwicklung. Hierbei wird die Projektsituation, für die es eine geeignete Technik auszuwählen gilt, durch die Merk-

male Wissensart und zu überwindendes Hindernis gekennzeichnet. Die empirische Validierung der Zuordnung von Techniken zu Projektsituationen könnte den Bezugsrahmen konkretisieren. Eine umfassendere Unterstützung der Technikwahl ließe sich erzielen, wenn zusätzliche Kontingenzfaktoren als Merkmale der Projektsituation einbezogen werden. Zu diesem Zweck könnten weitere, in der Literatur diskutierte Kontingenzmodelle herangezogen werden [E198; Ha99; HD03]. Eine Verfeinerung könnte durch eine genauere Analyse der einzelnen Techniken erfolgen. In [BR01] bspw. findet sich eine detailliertere Analyse von Interviewtechniken für das Anforderungsmanagement.

Aufgrund der großen Menge verschiedenster Techniken erfolgt hier eine Konzentration auf eine reduzierte Anzahl. Der vorgestellte Bezugsrahmen scheint jedoch grundsätzlich geeignet für die Integration weiterer Techniken, die auch anderen Forschungsgebieten entliehen werden können (Marktforschung, Gesprächstherapie etc.).

Literaturverzeichnis

- [BR01] Browne, G., Rogich, M.: An Empirical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques. In: *Journal of Management Information Systems* 17 (2001) 1.
- [By92] Byrd, T. A., Cossick, K. L., Zmud, R. W.: A synthesis of research on requirements analysis and knowledge acquisition techniques, in: *MIS Quarterly* 16 (1992) 1.
- [Da82] Davis, G.: Strategies for Information Requirements Determination. In: *IBM Systems Journal* 21 (1982) 1.
- [DS97] Darke, P., Shanks, G.: User viewpoint modelling: understanding and representing user viewpoints during requirements definition. In: *Information Systems Journal* 7 (1997) 3.
- [E198] El Louadi, M., Galletta, D. F., Sampler, J. L.: An Empirical Validation of a Contingency Model for In-formation Requirements Determination. In: *DATA BASE* 29 (1998) 3.
- [Ha99] Hardgrave, B. C., Wilson, R. L., Eastman, K.: Toward a Contingency Model for Selecting an Information System Prototyping Strategy. In: *Journal of Management Information Systems* 16 (1999) 2.
- [HD03] Hickey, A. M., Davis, A. M.: Requirements Elicitation and Elicitation Technique Selection. In: *Proc. of the 36th Hawaii Int. Conf. on System Sciences, 2003 (HICSS03)*.
- [IE90] IEEE Computer Society (1990): IEEE Standard Glossary of Software Engineering Terminology. IEEE Computer Society (New York). Verfügbar unter <http://www.computer.org/certification/csdpprep/Glossary.htm>.
- [KM95] Kensing, F., Munk-Madsen, A.: PD: Structure in the Toolbox. In: *CACM* 36 (1995) 4.
- [Or95] Ortner, E.: Elemente einer methodenneutralen Konstruktionssprache für Informationssysteme. In: *Informatik – Forschung und Entwicklung* 10 (1995) 3.
- [SH05] Stahlknecht, P., Hasenkamp, U.: Einführung in die Wirtschaftsinformatik. Berlin et al. 2005.
- [SS97] Sommerville, I., Sawyer, P.: Requirements Engineering. A good practice guide. New York 1997.
- [ST88] Sethi, V., Teng, J. T. C.: Choice of an Information Requirements Analysis Method: An Integrated Approach. In: *INFOR* 26 (1988) 1.
- [St96] Struckmeier, H.: Gestaltung von Führungsinformationssystemen. Wiesbaden 1996.
- [VF85] Valusek, J. R., Fryback, D. G.: Information requirements determination obstacles within, among and between participants. In: *Proceedings of the twenty-first annual conference on computer personnel research*. Minneapolis 1985.
- [Yo92] Yourdon, E.: *The Decline and Fall of the American Programmer*. Prentice Hall, 1992.