

Schwachstellen des Requirements Engineering für Informationssysteme in der Praxis

Helmut Thoma
Ciba-Geigy AG, Basel / Schweiz

Auch wenn Methoden des Requirements Engineering bekannt sind, mehr oder weniger praktikable Hilfsmittel zur Beschreibung und Sammlung von Anforderungen verfügbar sind (z.B. Daten- und Funktionsmodelle in einer anwendungnahen Beschreibungssprache), Abbildungen aus diesen abstrakten in maschinennahe Beschreibungsebenen mehr oder weniger brauchbar sind (z.B. Generierung von relationalen Schemata aus Beschreibungen im Entity-Relationship-Modell), gibt es in der Praxis genügend ungünstige Randbedingungen, um die Konzepte des Requirement Engineering massiv zu stören. Im folgenden sollen zwei derartige Problembereiche angesprochen werden.

Wir wollen zunächst einmal das Konzept der Applikations-Architektur betrachten. Bei diesem Ansatz wird die Planung von Applikationen und von Datenbeständen für ein Unternehmen oder eine grössere organisatorische Einheit eines Unternehmens aus gesamtheitlicher Sicht durch ein top-down-orientiertes Vorgehen unterstützt. Das Verfahren ist in [Thoma 93] beschrieben und wurde in mehreren Projekten mit unterschiedlichem Erfolg eingesetzt. Die Modellierungsebene der Applikations-Architektur beinhaltet Funktions- und Dateneinheiten. Aus betrieblichen Aktivitäten und den hierbei erzeugten oder benutzten Informationen werden Applikationen und Datenbestände bestimmt. Unterschiedliche Modellierungsebenen unterschiedlicher Detaillierungsgrade, also unterschiedliche Ausschnitte der Realität, können miteinander in Beziehung gesetzt werden (Verfeinerungen in einer resp. Vergröberungen in der anderen Richtung).

Mit diesem Ansatz können sowohl lokal von einer Applikation genutzte Daten herausgefunden als auch Datenbestände modelliert werden, die durch beliebig viele Applikationen gemeinsam benutzt werden können. Konsistente und jederzeit aktuelle Datenbestände in physisch integrierten oder in verteilten Datenbanken können ebenso modelliert werden wie redundante Datenbanken. Mit den Ergebnissen einer Applikations-Architektur kann der Entwurf von Daten, die von unterschiedlichen Benutzerkreisen resp. Applikationen verwendet werden und die ähnliche Fakten der Realität in einer Datenbasis abbilden, aufeinander abgestimmt werden.

Existierende Applikationen und Datenbanken, zugekaufte Lösungen in Software-Paketen (z.B. sog. Standard-Software), zukünftige Anforderungen an Funktionen und Daten können bei diesem Ansatz berücksichtigt werden. Soll- und Ist-Architekturen können miteinander verglichen werden, interessierende Schnittstellen können bis auf einzelne elementare Datenbestandteile detailliert betrachtet werden.

In der Praxis hat sich in manchen Projekten die Ermittlung der Anforderungen als der problematischste Teil des Prozesses herausgestellt, eine Anwendungs-Architektur zu erstellen. Mit fragwürdigen Ergebnissen der Anforderungs-Erhebung werden jedoch auch die Ergebnisse der Applikations-Architektur fragwürdig.

Die Hauptursache für nicht gesicherte Anforderungen liegt in der Grösse des Realitäts-Ausschnittes, der bei solch einer Applikations-Architektur zu betrachten ist. Soll die Architektur für eine grössere organisatorische Einheit erstellt werden, sind die Anforderungen vieler Menschen mit den unterschiedlichsten Verhaltensweisen zu berücksichtigen. Da ist zum einen der mittellose Kollege, der endlich froh ist, sein Aufgaben-Spektrum, das ihn unter Umständen ohnehin überfordert, jemandem schildern zu können. Demgegenüber gibt es auch den ungedulden Kollegen, der durch eine Anforderungs-Analyse seinen Arbeitsgang unterbrochen sieht oder der nicht einsieht, dass die Erstellung einer Applikations-Architektur auch für ihn sinnvoll ist. Beide Aussagen gilt es, miteinander zu verschmelzen. Eine entsprechende Relativierung und Gewichtung von Aussagen unterschiedlichster Präzision ist häufig schwierig bis unmöglich. Eigentlich möchte man den Realitätsausschnitt des „Kollegen Ungeduld“ nicht mit berücksichtigen. Dem steht jedoch der Umfang des Modellierungs-Auftrages entgegen! Guter Rat ist schwer, eine Minderung der Qualität ist häufig die Folge.

Wenn wir die Zusammenarbeit von - in der Hauptsache bestehenden - Applikationen betrachten, haben wir es in der Praxis nicht selten mit den Folgen mangelhafter Ergebnisse von Anforderungs-Analysen zu tun. Entweder wurde eine Architektur gar nicht, zu spät oder mit fragwürdigen Ergebnissen erstellt oder aus der Architektur wurden für die System-Entwicklung keine Konsequenzen gezogen. Dann liegen unkoordiniert entwickelte Applikationen vor uns, die zumindest teilweise mit gemeinsamen Daten arbeiten müssten.

Gemeinsam aufgebaute resp. genutzte Datenbanken sind in der Regel Mangelware, selten ist ein funktionierendes Datenmanagement implementiert. Die Daten müssen irgendwie zwischen den Applikationen hin- und hergeschoben werden. Bei der Integration von Standardsoftware unterschiedlicher Provenienz oder beim Einbau von Standardsoftware in ein „Geflecht bestehender Applikationen“ bestand zudem kaum eine Chance, koordiniert Systeme aufzubauen. Auch sind nicht selten die Strukturen von Datenbeständen unbekannt, die eigentlich dieselben Daten verwalten müssten, weil ihre Applikations-Systeme mit denselben Daten arbeiten müssten.

Abhilfe kann in vielen Fällen nur ein Konzept bieten, das eigentlich für den computergestützten Austausch von Daten zwischen unterschiedlichen Geschäftspartnern entwickelt wurde. Das Konzept des „Electronic Data Interchange“ (EDI). Wenn auch nur für eine kleine Anzahl möglicher Message-Typen EDIFACT-Standards vorliegen, so können mit einigem Aufwand Konventionen im eigenen Unternehmen für den allgemeinen Austausch gewisser, fest umrissener Datenmengen geschaffen werden.

Der Vortrag wird neben der Schilderung obiger Probleme ein Konzept vorstellen, das die oben genannten Schwachstellen eines aus Applikationssicht „übergeordneten“ Requirements Engineering überwinden muss und das derzeit realisiert wird. Als Beispiel wird der Austausch von Sicherheitsdaten für Gefahrstoffe der Chemie etwas genauer beleuchtet. Es werden die Möglichkeiten eines EDI mit EDIFACT-Standard den Möglichkeiten eines EDI mit unternehmensintern entwickelten Messagetypes für den Datenaustausch zwischen unterschiedlichen Applikationen und Datenbeständen gegenüber gestellt.

Literatur:

[Thoma 93]

Thoma, H.: Integration von Applikationen und Datenbanken mit Hilfe einer Applikations-Architektur. In Müller-Ettrich, G. (Hrsg.): Fachliche Modellierung von Informationssystemen. Addison-Wesley (Deutschland), Bonn, 1993