

Die Analyse- und Definitionsphase beim Software Engineering:
Geschäftsprozessanalyse mittels Entity-Relationship-Modellierung für Lasten-
und Pflichtenhefte

VON
Frank Schmidt

Erstauflage

Diplomica Verlag 2014

Verlag C.H. Beck im Internet:
www.beck.de

ISBN 978 3 95485 169 0

Leseprobe

Textprobe:

Kapitel 2, Verwendete Methoden und Verfahren:

2.1, Die Phasen der Softwareentwicklung:

Die Entwicklung von Software ist ein Prozess, bei dem es darum geht, für ein existierendes Problem der realen Welt ein funktionierendes Computerprogramm zu schaffen und erfolgreich zu installieren. Das Programm muss dabei zieldefiniert sein, bestimmten Qualitätsanforderungen genügen und sich auf seine Wirtschaftlichkeit – sowohl in Bezug auf die Entwicklung als auch in Bezug auf den tagtäglichen Betrieb – überprüfen lassen. 'Die Softwarekosten dominieren inzwischen die Kosten der Hardware bei weitem.' (PAGEL / SIX 1994, 43). Diese Aussage stammt zwar aus dem Jahre 1994, trifft aber heutzutage mehr denn je zu. Eine Einteilung von IT-Projekten zur Softwarebeschaffung aus dem Jahre 1999 gibt als untere Größenordnung für ein einfaches Projekt (z.B. eine Abteilungsanwendung) ein Investitions-volumen bis zu Euro 15.000 vor. Mittlere Investitionssummen für Software bewegen sich bis zu Euro 150.000, von der oberen Größenordnung wird ab Euro 250.000 gesprochen (vgl. GRUPP 1999, 43 f.). Woher kommen diese Kosten? Und was haben diese Kosten mit den Phasen der Softwareentwicklung zu tun? Der Prozess der Softwareentwicklung ist ein hochgradig komplexes und arbeitsteiliges Gebilde, das neben einem materiell-technischen Ressourceneinsatz auch einen ganz erheblichen Aufwand an hochqualifiziertem Personal und Arbeitszeit verursacht. Um diesen Vorgang beherrschen zu können, hat es sich als effektiv erwiesen, den Gesamtprozess zu unterteilen, zu strukturieren und Zielvorgaben zu fixieren. Diese Phaseneinteilung im Prozess der Softwareerstellung ermöglicht es, Arbeitspakete, Ziele, Meilensteine und Verantwortlichkeiten zu definieren, Arbeitsgruppen zu koordinieren, Zeitrahmen und materielle sowie finanzielle Ressourcen festzulegen und zu kontrollieren. Die Phaseneinteilung hat demnach zwei Zielrichtungen: einmal die fachliche und zum anderen eine wirtschaftliche Seite. Alle im Zusammenhang mit der Erstellung von Software ablaufenden Aktivitäten können unter Beachtung von diversen Kriterien wie z.B. Zeit, Zwischenergebnis usw. in Phasen zusammengefasst werden. Das schafft eine organisatorisch-theoretische Grundlage, den Gesamtprozess transparenter zu machen und ihn fachlich und finanziell aktiv zu beherrschen.

Folgende Grobeinteilung ist allgemein anerkannt:

- (1) Analysephase (Problemanalyse und Planung),
 - (2) Definitionsphase (Anforderungsdefinition und Systemspezifikation),
 - (3) Entwurfsphase (System- und Komponentenentwurf),
 - (4) Implementierungsphase (Implementierung und Tests),
 - (5) Abnahme- und Einführungsphase (Einführung beim Anwender),
 - (6) Wartungsphase (Fehlerbehebung, Änderungen, Anpassungen).
- (vgl. STANIEROWSKI o. J., a, 17 f.).

Dieses 6-Phasen-Modell basiert weitgehend auf dem Vorgehensmodell zur Anwendungsentwicklung von RIEMANN, der Vorschlagsphase, Definitionsphase, Konzeptphase, Entwurfsphase, Realisierungsphase, Einführungs- und Implementierungsphase unterscheidet (RIEMANN 1996).

Vielfach ist auch eine Zusammenfassung der genannten Einzelphasen anzutreffen. So werden z.B. die beiden ersten Phasen unter dem englischen Begriff 'requirements engineering' oder 'requirements analysis and specification' als 'A&D-Phase' (Anforderungs- und Definitionsphase) zusammengefasst. Danach beginnt diese A&D-Phase bei der Ermittlung und Analyse der Aufgabenstellung und endet mit der Anforderungsdefinition für das System (vgl. KÜHNEL et al. 1987, 334 – 335). Wieder andere Einteilungen berücksichtigen die interessenengeprägte Sicht auf die eigene Involvierung in den Prozess des SE, so dass z.B. die Abnahme- und Einführungs- sowie die Wartungsphase nicht zur eigentlichen Softwareentwicklung gehören, sondern bereits dem weiteren Software-Lebenszyklus zuzuordnen wären. Wichtig ist, dass eine Phaseneinteilung ein theoretisches Grundgerüst sein kann, um den Arbeitsablauf der Softwareentwicklung zu fördern.

Die Anforderungs- und die Definitionsphase sollen kurz näher erläutert werden, weil sie die vorliegende Arbeit besonders betreffen. Es wird anerkannt, dass ein Scheitern eines Softwareprojektes zum größten Teil bereits in der A&D-Phase verursacht wird. So sollen ca. 60% des Gesamtrisikos bei der Softwarebeschaffung auf unzureichende oder fehlerhafte Anforderungsdefinitionen entfallen (vg. GRUPP 1999, 28). In einer europaweiten Marktuntersuchung durch das IT-Beratungsunternehmen Diebold wurde festgestellt, dass in der Praxis etwa ein Viertel aller IT-Projekte scheitern; bei jährlichen Gesamtausgaben in Europa von 140,5 Milliarden Dollar sind das immerhin ca. 35 Mrd. Dollar Verlust! Als wesentliche Ursache dafür wird ein Basieren des Projekts auf falschen Geschäftsmodellen identifiziert (vgl. o. V., a, 2002). Die Universität Dublin kommt in einer anderen Studie zu dem Ergebnis, dass sogar jedes zweite IT-Projekt misslingt oder zum Abbruch führt. Von den verbleibenden erfolgreichen Projekten werden 40% nur unter beträchtlichen Folgeinvestitionen zum Erfolg geführt (vgl. o. V., b, 2002). Welche Zahlen auch immer richtig sind, eine Grundaussage ist erkennbar: IT-Projekte sind mit einem enormen Risiko verbunden! Die finanzielle Belastung des Unternehmens durch ein misslungenes Projekt kann dabei gewaltige Ausmaße annehmen. So musste z. B. ein gemeinsames Projekt der Hotelketten Hilton und Marriott mit der Mietwagenfirma Budget im Jahre 1994 abgebrochen werden, übrig blieben Stranded Costs von 125 Millionen Dollar ! Die britische Child Support Agency verlor durch den Abbruch eines Projektes 1997 allein 600 Millionen Pfund (vgl. o. V., b, 2002). Leider sind keine genauen Angaben verfügbar, inwieweit diese gescheiterten IT-Projekte softwarebedingt waren, auch ist nicht belegt, welche Bedeutung letztendlich eine mangelhafte A&D-Phase für die genannten Fehlschläge hatte. Wenn jedoch die Aussagen zu den 'falschen Geschäftsmodellen' usw. auch nur ansatzweise zutreffen, dann muss man die besondere Bedeutung der A&D-Phase anerkennen!

Während in der Entwurfs-, Implementierungs-, Einführungs- und Wartungsphase ein Qualifikationsschwerpunkt des verantwortlichen Personals auf informatikspezifischem Gebiet liegt, so sind insbesondere in der A&D-Phase Fähigkeiten gefragt, die es ermöglichen, die im späteren

Programm abzubildende Realität zu durchschauen und modellhaft zu beschreiben. Zwei Fragen sind dabei signifikant:

- a) Wie sieht die Ausgangssituation aus, was läuft derzeit wie, wann, wo und womit ab?
- b) Was ist das Wunschziel, was will der Anwender zukünftig eigentlich mit der IT-Lösung machen, was soll sie ihm bringen?

Hier geht es um ein 'Analysieren des Problems und das Definieren (bzw. Spezifizieren) des Produkts.' (PAGEL / SIX 1994, 77). Auf der Basis dieser Informationen kann dann die eigentliche Umsetzung des Problems erfolgen. Zusammenfassend kann man festhalten:

- 1) Die A&D-Phase beschäftigt sich mit den realen Prozessen, die beim Anwender stattfinden.
- 2) Die A&D-Phase schafft die Grundlage für die weitere Tätigkeit des IT-Spezialisten.
- 3) Die A&D-Phase wird vorrangig nicht vom IT-Spezialisten, sondern vom Analytiker und Anwender geprägt.
- 4) Damit der IT-Spezialisten zielführend arbeiten kann, muss die A&D-Phase qualitativ ausreichende Ergebnisse bereitstellen.
- 5) Sind die Ergebnisse der A&D-Phase unvollständig oder anderweitig mangelhaft, so sind Spätfolgen in Form von unzureichenden Lösungen zu erwarten.
- 6) Nur so exakt, wie das Produkt und seine zu erfüllenden Aufgaben definiert werden, kann auch die Umsetzung erfolgen.

Beendet ist die A&D-Phase, 'wenn die Anforderungen an das Produkt in der Produktdefinition (synonym: Anforderungsdefinition, Aufgabendefinition, Pflichtenheft, Requirementkatalog) dokumentiert und verabschiedet sind. Adressaten der Produktdefinition sind neben den Entscheidungs-trägern bei Auftraggeber und Auftragnehmer auch die Entwurfsspezialisten, die auf Basis der Produktdefinition das Innenverhalten und die Architektur der Software festlegen.' (PAGEL / SIX 1994, 85).

Es muss an dieser Stelle aber erwähnt werden, dass das phasenstrukturierte Vorgehen in der Softwareentwicklung auch kritisch diskutiert wird. Als Schwachpunkte werden vor allem unscharfe Phasengrenzen, Probleme bei der Beurteilung von Phasenergebnissen, zu starke Formalisierung und Bürokratisierung des Gesamtentwicklungsprozesses und unzureichende Einflussmöglichkeiten der Anwender auf das Endprodukt und daraus resultierende hohe Wartungskosten genannt. Als Alternativen zum Phasenmodell gelten Werkzeuge wie z.B. Prototyping, Objektorientierung oder der Einsatz von CASE-Tools (vgl. SCHUMANN o. J., 21 ff.).

2.2.2, Das Lastenheft und das Pflichtenheft:

Der Schwerpunkt der vorliegenden Arbeit betrifft – unter Zugrundelegung des 6-Phasen-Modells – Ergebnisse der ersten beiden Phasen: Analyse- und Definitionsphase. Zunächst soll ein Lastenheft erstellt werden, das die anwenderseitigen Wünsche und Anforderungen an das Programm 'Holzfirma' beschreibt. Dabei wird von der Ist-Situation im Unternehmen ausgegangen und eine 'Vision' der vom späteren Programm zu erfüllenden Aufgaben entwickelt. Das Lastenheft, auch als Produktskizze bezeichnet, stellt eine systematische, strukturierte Aufzählung

der Wünsche und Anforderungen des Anwenders an das neue System dar. Ausgehend vom Bedarf des Anwenders werden innerhalb einer sogenannten Vorstudie die meist noch vagen Vorstellungen eruiert und aufgezeichnet. Ein Lastenheft ist die 'Zusammenstellung aller Anforderungen des Auftraggebers hinsichtlich Liefer- und Leistungsumfang. Im Lastenheft wird definiert, WAS und WOFÜR zu lösen ist. Es dient als Ausschreibungs-, Angebots- und/ oder Vertragsgrundlage. Hier sollten das Thema und die Ziele der Projektarbeit deutlich werden.' (THORSTENT 2002).

Darauf aufbauend wird ein Pflichtenheft erarbeitet, das die Anforderungen des Lastenheftes berücksichtigt und eine Produktdefinition liefert, die als verpflichtendes Dokument den weiteren Prozess der Softwareerstellung begründet. Dieses Pflichtenheft kann dabei – je nach Umfang des zu erstellenden Programms – große Dimensionen annehmen. 'Die Größe und Komplexität heutiger Softwareentwicklungen ziehen umfangreiche und komplizierte Produktdefinitionen nach sich, welche das menschliche Beurteilungsvermögen überfordern.' (PAGEL / SIX 1994, 47). Das Pflichtenheft als sehr komplexes Dokument dient dabei zunächst einmal als fachlicher Wegweiser bei der Realisierung des Softwareprojektes. Außerdem besitzt es auch Bedeutung als juristische Vertragsgrundlage zwischen Auftraggeber und Auftragnehmer. Sollte es nämlich im Zusammenhang mit dem Projekt zu Streitigkeiten kommen, so kann anhand des von den Vertragsparteien unterzeichneten Pflichtenheftes überprüft werden, ob eine Vertragsverletzung vorliegt oder nicht. Bei diesen Dimensionen wird deutlich, welche immense Bedeutung einem Pflichtenheft für die Klärung einer eventuellen Schuldfrage nebst Schadensersatzforderung zukommen kann.

Als das zentrale Dokument des Gesamtprojekts besitzt die Produktdefinition, also das Pflichtenheft, eine entscheidende Bedeutung (vgl. PAGEL / SIX 1994, 47 und 79 f.). Es ist eine 'Beschreibung der Realisierung aller Anforderungen des Lastenheftes. Im Pflichtenheft wird definiert, WIE und WOMIT die Anforderungen zu realisieren sind. Es dient als verbindliche Vereinbarung für die Realisierung und Abwicklung des Projektes für Auftraggeber und Auftragnehmer. Das Pflichtenheft enthält das Lastenheft. Die Ergänzung beinhaltet insbesondere einen Verlaufsplan.' (THORSTENT 2002). Inwieweit den Vorstellungen zur Einbeziehung des Lastenheftes und des Verlaufsplanes in der betrieblichen Praxis gefolgt wird, bleibt dem Anwender überlassen. GRUPP beispielsweise kombiniert Lasten- und Pflichtenheft als Ausschreibungsgrundlage zur externen Softwarebeschaffung unter dem Gesamtbegriff 'Pflichtenheft' und gibt nur einen zeitlichen Realisierungsrahmen vor (vgl. GRUPP 1999, 124 ff.). Entscheidend ist, dass das Pflichtenheft Anforderungen nach Vollständigkeit, Eindeutigkeit und Widerspruchsfreiheit genügen muss. Inwieweit formale Spezifikationen integriert werden, hängt vom jeweiligen Einzelfall ab. Alle Angaben müssen weiterhin nachvollziehbar und validierbar sein (vgl. PAGEL / SIX 1994, 85 f.). Für weitere Informationen wird auch auf die VDI / VDE-Richtlinien 2519 und 3694 verwiesen.