

Wertanalyseverfahren für Kundenanforderungen

David Kuhlen
NORDAKADEMIE
david.kuhlen@datenlotsen.de

Abstract: In dieser Arbeit wurde ein Verfahren zur Anforderungsbewertung entwickelt, das für die Probleme heterogener Bewertungsmethoden ein kombinierten Lösungsansatz darstellt. Dabei entstand ein Vorgehensmodell, das auf drei Säulen ruht: Einem Phasenmodell, das Akteuren je Dimension Methoden zuordnet; einem Prozessmodell, das den Prozess mit dem Phasenmodell verbindet und Szenarien, die einem Hersteller die Anwendung der Methoden ermöglichen.

1 Einleitung

Softwaresysteme werden profitabel, wenn ihre Qualität maximiert, ihre Kosten minimiert und ihre Entwicklungszeit gekürzt werden. Die Praxis zeigt jedoch, dass diese Ansprüche sich oft widersprechen und zwischen konkurrierenden Zielen abzuwägen ist (Vgl. [Ho08]). In vielen Fällen reicht die Entwicklungszeit nicht aus, um alle sinnvollen Features umzusetzen (Vgl. [Bo84]). Ohne eine Wertanalyse besteht die Gefahr, dass gewinnbringende Anforderungen nicht als solche erkannt und nicht umgesetzt werden (Vgl. [SR13]). Dazu ist die Forschungsfrage zu beantworten: „Wie lässt sich der Wert einer Kundenanforderung durch einen Softwarehersteller ermitteln?“. Es ist ein Verfahren zu entwickeln, das den Wert einer Kundenanforderung aus der Sicht eines Softwareherstellers zeigt. Dieses Verfahren soll die Umsetzung von Softwareanforderung mit einer betriebswirtschaftlichen Planung absichern. Es wurde für den Einsatz in Scrum-Projekten optimiert, weil bei Scrum die Untersuchung des Wertes aufgrund knapper Spezifikationen vor besonderen Herausforderungen steht.

2 Investitionen in Software und die Bewertung von Anforderungen

Investition ist definiert, als Bindung von Geldmitteln in Gütern des Anlagevermögens, wobei es sich bei Investitionen um Zahlungsreihen handelt, die mit einer Auszahlung beginnen (Vgl. [DG07]). Die Umsetzung einer Anforderung bindet Arbeitsstunden, zur Entwicklung von Funktionalität. Die entstehende Software kann, als immaterielles Vermögen, in der Bilanz aktiviert werden. Die Entwicklung erfordert zunächst Auszahlungen (Lohn der Entwickler) die später durch Einzahlungen beglichen wird. Eine Methode zur Investitionsrechnung muss das Verhältnis von Aus- und Einzahlungen bewerten. Bei dynamischen Methoden zur Investitionsrechnung wird die Zeitpräferenz, der Zahlungen berücksichtigt. Weitere Methoden erlauben es zusätzlich, Unsicherheiten bei der Investitionsrechnung zu berücksichtigen. Für die Bewertung von IT-Investitionen ist die Berücksichtigung qualitativer Effekte von besonderer Bedeutung. Insgesamt

müssen zur Bewertung von Anforderungen darum die Dimensionen Einzahlungen, Auszahlungen, Unsicherheiten, qualitative Effekte und die Zeitstruktur bewertet werden.

Eine Anforderung beschreibt ein Merkmal, das die Lösung erfüllen muss. Sie bilden die Brücke zwischen der Ist- und der Soll-Situation (Vgl. [Br05]).

Indem eine Anforderung ein Lösungsmerkmal beschreibt, hat sie zwei Charakteristika. In erster Linie hilft sie bei der technischen Orientierung. So legt die Anforderungsdefinition fest, was das Softwaresystem leisten soll und bündelt gleichzeitig ein Zeit- und Ressourcenbudget im Projektplan (Vgl. [GM05]). Um die Soll-Situation zu definieren müssen Anforderungen ausreichend genau spezifiziert werden. So werden in einer Anforderung die späteren Schritte des Projektvorgehens detailliert spezifiziert (Vgl. [FLT92]). Die Spezifikation ist der Ausgangspunkt für die Bewertung.

3 Probleme mit der Wertanalyse von Anforderungen

Softwarehersteller entscheiden sich für die Umsetzung von Anforderungen wegen ihrer Nutzeneffekte, die sich in direkten und indirekten Einzahlungen auswirken. Gegenüber direkten Einzahlungen (etwa aus dem Verkauf einer Software), ist bei indirekten Einzahlungen teilweise unklar, wo und in welcher Höhe Erträge entstehen (Vgl. [DG07]). Erträge aus einer IT-Investition entstehen verteilt über die Organisation (Vgl. [II01]). Unternehmen wissen, dass es durch IT-Investitionen zu unerwartbaren Erträgen kommen kann (Vgl. [BS98]). Verglichen mit dem Problem der Identifikation von Ertragschancen gilt, die Quantifikation des Ertrags, als das größere Problem (Vgl. [BS98]). Der Ertrag einer IT-Investition kann schwer gemessen- und in finanziellen Größen quantifiziert werden (Vgl. [Bo84], Vgl. [FLT92], Vgl. [BH96], Vgl. [Ok06], Vgl. [SR13]). Nach der Umsetzung einer Anforderung können Erträge ausbleiben oder anders als geplant anfallen (Vgl. [Bo84], Vgl. [Br93], Vgl. [Br05]). So sind Nutzeneffekte unsicher weil verschiedener Faktoren die Anforderung beeinflussen können (Vgl. [SR13]). Die Kosten einer Software richtig einzuschätzen ist ein bedeutendes Problem der Wirtschaftsinformatik und somit bei Investitionsentscheidungen relevant (Vgl. [Bo84]). In den meisten Fällen unterschätzen Unternehmen die Kosten eines IT-Projektes (Vgl. [Ho90]). Für Softwarehersteller hängen die Kosten einer Anforderung maßgeblich vom Zeitaufwand ab, den die Entwickler zur Umsetzung benötigen. IT-Investitionen haben keine deterministischen Wirkungen und bergen Risiken, die den Wert beeinflussen (Vgl. [Hi05]). Typischerweise besteht während Entwicklungsprojekten das Risiko, dass Kunden ihre Anforderungen verändern. Fehler bei der Anforderungsdefinition wirken auf die Projektplanung. Wenn die Verteilung von Ressourcen zur Anforderungsumsetzung falsch ist, wird die langfristige Flexibilität des Entwicklungsprojektes gefährdet (Vgl. [II01]). Weil die Fehlerfreiheit einer Software kaum wirtschaftlich nachweisbar ist, bleibt weiter das Risiko von Softwarefehlern bestehen (Vgl. [KP96]). Die Erfüllung eines übergeordneten Kundenwunsches, dem mehrere Anforderungen zugeordnet sind, kann zu Abhängigkeiten zwischen den Anforderungen führen. Die Rendite einer Anforderung kann von den Risiken einer anderen Anforderung abhängen (Vgl. [WHS05]). Investitionen in Software haben den Ruf, besonders riskant zu sein,

weswegen höhere Renditen erwartet werden (Vgl. [BH96]). Aus diesem Grund verteuern sich Anforderungen künstlich und die Wirtschaftlichkeit der Anforderung leidet. IT-Investitionen haben vielfach qualitative Wirkungen, die per Definition schwer in monetäre Größen überführt werden können. Der Nutzen von IT-Investitionen kann z. B. in den Größen Effizienz, Qualität oder Effektivität ausgedrückt werden (Vgl. [Po93]). Klassische Methoden zur Investitionsrechnung lassen sich nur für quantifizierbare Nutzenwirkungen (Zahlungen) einsetzen (Vgl. [BS98]). Weil qualitative Effekte schwer zu berücksichtigen sind, müssen ihre Zahlungswirkungen geschätzt werden. Investitionsentscheidungen basieren in diesen Fällen auf Annahmen, wobei subjektive Einflüsse oder gezielte Manipulationen das Ergebnis verändern können (Vgl. [II01]). Ferner fehlt bei der Investitionsrechnung vielfach der Überblick über alle Effekte um diese in der Entscheidung zu berücksichtigen (Vgl. [SR13]). Dieses führt dazu, dass gewisse Wirkungen nicht in die Wirtschaftlichkeitsanalyse einbezogen werden. Kostenabschätzungen fallen im Zeitablauf des Projektes, von der Planung bis zur Umsetzung, immer leichter (Vgl. [Bo84]). Oft ist die Veränderung der Kundenprozesse abzuwarten, bevor der Nutzen aus einer umgesetzten Anforderung vollständig sichtbar wird (Vgl. [BH96]). Oft fehlen Lebenszyklusmodelle und wichtige Informationen, um den Zahlungsverlauf bei der Wertanalyse zu berücksichtigen (Vgl. [ZSB04]).

4 Entwicklung eines Vorgehensmodells zur Wertanalyse

4.1 Konzeption eines Vorgehensmodells

Es ist ein Wertanalyseverfahren zu entwickeln, um Softwareherstellern zu ermöglichen die Anforderungen auszuwählen, die den Wert des Betriebsergebnisses maximieren. Keine eigenständige Methode kann zur Analyse des Wertes von Kundenanforderungen geeignet sein. Sinnvoll ist daher der Einsatz verschiedener Bewertungsmethoden, die sich jeweils für Kalkulation einer Dimension eignen. Verschiedene Methoden können in einem Vorgehensmodell in Phasen und Prozesse eingebunden werden (Vgl. [Ha12]). Dazu beschreibt ein Vorgehensmodell in welcher Reihenfolge Aktivitäten durchgeführt werden (Vgl. [Wo13]), um die Bewertungsmethoden zu orchestrieren. Auch die Verantwortlichkeiten für den Einsatz der Bewertungsmethoden und die Ergebnisse der Bewertung werden im Vorgehensmodell koordiniert (Vgl. [Ba96]) zitiert bei (Vgl. [Fä09]). Dabei sind die anzuwendenden Bewertungsmethoden austauschbar. Die Auswahl von Bewertungsmethoden sollte vom Geschäftsmodell und der Branche des Endanwenders abhängen, weil diese Methoden Dimensionsgrößen wie z. B. die Zeitstruktur maßgeblich beeinflussen. Das Vorgehensmodell sieht vor, dass die Analyse des Wertes in jeder Phase von Scrum zyklisch wiederholt wird. Scrum legt eine Unterscheidung der Phasen „Backlog“, „Sprint Planning“, „Sprint“ und „Review“ nahe.

4.2 Wertanalyse entlang des Softwareentwicklungsprozesses

In der **Backlog Phase** wird die Wertanalyse zur Ermittlung der relativen Vorteilhaftigkeit (und später zur Priorisierung) einer Anforderung eingesetzt. Grundlage

dazu ist die Anforderungsdefinition. Die relative Vorteilhaftigkeit wird maßgeblich durch den Product Owner ermittelt. Die Nutzenanalyse beginnt mit der Anforderungsaufnahme. Danach werden die einzelnen Dimensionen analysiert. Für die Anforderungsaufnahme werden Anforderungsschablonen-, und für die Analyse der einzelnen Dimensionen standardisierte Methoden zur Investitionsbewertung eingesetzt. Wesentlich ist die Ermittlung der Einzahlungen. Wenn für die Software kein Preis feststeht, können die erwartbaren Einzahlungen durch die Wirkungskettenanalyse ermittelt werden. Unabhängig von der Art der IT-Investition sind stets wiederkehrende Nutzeffekte zu beobachten (Vgl. [SK09]). Die Wirkungskettenanalyse kann mit einer Prozessanalyse kombiniert werden. Um Einzahlungen zu ermitteln, kann eine Prozessanalyse die Marktleistung des Kunden im Ist- und im Sollzustand erfassen (Vgl. [Br05a]). Die Wertanalyse in der **Planning Phase** baut auf den vorherigen Ergebnissen auf. Eine genauere Aufwands- und Risikoschätzung wird mit den bereits gewonnenen Erkenntnissen kombiniert. Für die genaue Aufwandsschätzung ist das Entwickler Team verantwortlich. Entwickler können den Aufwand am besten einschätzen, da sie das Produkt erstellen. Das Entwickler Team kann Anforderungen zunächst mit der Function Point Method bewerten. Dazu werden die Funktionen analysiert, die für die Umsetzung der Anforderungen neu entwickelt oder angepasst werden müssen. So wird gleichsam die Entwicklung vorbereitet. In der **Sprint Phase** werden die ausgewählten Anforderungen umgesetzt. Insbesondere die kritischen Werte und die Werttreiber geben bei Entwicklung eine Orientierungshilfe. Trotz guter Planung kann es bei der Ausführung zu Abweichungen kommen, etwa durch Fehler oder unzureichende Mitarbeiterleistungen (Vgl. [WD08]). Auf Basis der geplanten Ergebnisse der vorherigen Phasen können die realisierten Lösungen in die Wertanalyse einbezogen werden. In täglichen Meetings (den sog. Daily Scrums) schätzt das Entwicklungsteam seinen Fortschritt in Richtung des Sprintziels (Vgl. [SS11]). Im Daily Scrum informiert sich das Entwicklungsteam über den Fertigstellungsgrad von Anforderungen und über Hindernisse. In diesem Rahmen kann die Wertanalyse wiederholt werden. Die Ermittlung des realisierten Wertes einer Anforderung kann in der **Review Phase** erfolgen. Dazu wird die fertiggestellte Softwarelösung einer (Expost) Wertanalyse unterzogen. Dieses hat den Vorteil, dass für bestimmte Dimensionsgrößen bereits Werte feststehen (wie etwa die tatsächliche Entwicklungszeit). Das Vorgehensmodell empfiehlt, dass der Product Owner anhand der Software die erwartbaren Einzahlungen erneut untersucht. Dazu können erneut die Wirkungskettenanalysen vorgenommen- und mit Prozessanalysen kombiniert werden. Durch einen vorher/nachher Vergleich von Prozessen werden die Kosteneinsparungen sichtbar (Vgl. [Hi05]). Schließlich führt die Optimierung der Geschäftsprozesskosten zum Ergebnisbeitrag der IT-Investition (Vgl. [BES05]).

5 Tragfähigkeit des Verfahrens und Ausblick

Für die Evaluation der Tragfähigkeit wurde das Modell mit den Anforderungen der Scrum Akteure verglichen. Der **Product Owner erwartet**, dass mit dem Verfahren die richtigen Anforderungen gewählt werden, um den Produktwert zu maximieren. Das Vorgehensmodell ermöglicht die Anforderungsuntersuchung, indem für die Phasen der Entwicklung einzelne Methoden vorgeschlagen werden. Die Kenntnis über die

Werttreiber erleichtert es dem Product Owner die inhaltliche Ausgestaltung einer Anforderung festzulegen. Seine Kontrollmöglichkeiten werden dadurch gesteigert. Die **Erwartungen des Scrum Masters** beziehen sich auf die Unterstützung des Projektvorgehens durch die Wertanalyseverfahren. Ihm gibt das Wertanalyseverfahren die Möglichkeit auf die Erreichung der Top-Werthebel zu achten. Abweichungen werden schnell angezeigt und der Scrum Master kann Gegensteuerungsmaßnahmen einleiten. Die Erwartung des Scrum Masters, Verschwendung zu vermeiden, wird durch den modularen Aufbau des Verfahrens erfüllt. Nicht in jeder Phase müssen alle Dimensionen erneut analysiert werden. Die Prozesse der Wirtschaftlichkeitsanalyse stellen sicher, dass die entwickelte Software wirtschaftlich ist und zeigen die erreichte Wertschöpfung durch die **Entwickler**. Außerdem soll das Vorgehensmodell Entwicklern die Entscheidung über eine sinnvolle Implementierung erleichtern. Indem das Vorgehensmodell die Werttreiber einer Anforderung zeigt, gibt es den Entwicklern eine Orientierungshilfe bei ihren Entscheidungen.

Das vorliegende Verfahren hilft Softwareherstellern das Sprintziel festzulegen. In einer weiteren Forschungsarbeit ist eine genauere Untersuchung der Aufwands-Dimension angebracht. Vor der Prämisse eines steigenden Kostendrucks in der Softwarebranche müssen Softwarehersteller das Controlling ihrer Prozesskosten verbessern.

Acknowledgement

Mein besonderer Dank gilt Herrn Prof. Dr. Andreas Speck (Universität zu Kiel) und Herrn Prof. Dr. Hinrich Schröder (NORDAKADEMIE) für ihre Betreuung.

Literaturverzeichnis

[Ba96]: Balzert, H.: Lehrbuch der Software-Technik, Spektrum Akademischer Verlag Heidelberg Berlin, Oxford, 1996, zitiert bei: Fähnrich (2009).

[BES05]: Buchta, D.; Eul, M.; Schulte-Croonenberg, H.: Strategisches IT-Management: Wert steigern, Leistung steuern, Kosten senken, Gabler / GWV Fachverlag GmbH, 2. Auflage, Wiesbaden 2005.

[BH96]: Brynjolfsson, E.; Hitt, L.: Paradox Lost? Firm-level Evidence on the Returns to Information Systems Spending, in: Management Science, 42. Jg. (1996) 4, S. 541 – 558.

[Bo84]: Boehm, B.: Software Engineering Economics, in: IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-10, NO. 1 (1984), JANUARY 1984.

[Br05]: Brugger, R.: Der IT Business Case, Springer-Verlag Berlin Heidelberg, Berlin 2005.

[Br05a]: Brugger, R.: IT-Projekte strukturiert realisieren, Friedrich Vieweg & Sohn Verlag / GWF Fachverlage GmbH, 2. Auflage, Wiesbaden 2005.

[Br93]: Brynjolfsson, E.: The Productivity Paradox of Information Technology, in: Communications of the ACM, 36. Jg. (1993) 12, S. 67 – 77.

[BS98]: Ballantine, J.; Stray, S.: Financial appraisal and the IS/IT investment decision making process, in: The journal of Information Technology (1998) 13, S. 3 – 14.

- [**DG07**]: Däumler, K. -D.; Grabe, J.: Grundlagen der Investitions- und Wirtschaftlichkeitsrechnung, NWB Studium Betriebswirtschaft, Verlag Neue Wirtschaft-Briefe GmbH & Co. KG, 12. Auflage, Ettenheim 2007.
- [**Fä09**]: Fähnrich, K., P.: Vorlesung Softwaretechnik - Vorgehensmodelle, V-Modell XT -, Universität Leipzig, Institut für Informatik Betriebliche Informatik-Systeme, 2009, URL: http://bis.informatik.uni-leipzig.de/de/Lehre/0910/WS/LV/SWT/files?get=2009w_swt_v_03.pdf, [Stand: 17.01.2014].
- [**FLT92**]: Farbey, B.; Land, F.; Targett, D.: Evaluating investments in IT, in: Journal of Information Technology (1992) 7, S. 109 – 122.
- [**GM05**]: Gadatsch, A.; Mayer, E.: Masterkurs IT-Controlling, Friedr. Vieweg & Sohn Verlag / GWV Fachverlage GmbH, 2. Auflage, Wiesbaden 2005.
- [**Ha12**]: Hafenrichter, B.: Vorgehensmodelle zur Softwareentwicklung, Fachhochschule Flensburg, 2012, URL: http://fbim.fh-regensburg.de/~hab39652/SE/Kapitel_2a_Vorgehensmodelle_Kurz.pdf, [Stand: 17.01.2013].
- [**Hi05**]: Hirschmeier, M.: Wirtschaftlichkeitsanalysen für IT-Investitionen, WiKu-Verlag - Verlag für Wissenschaft und Kultur Dr. Stein & Brokamp KG, Berlin 2005.
- [**Ho08**]: Hoffmann, D., W.: Software-Qualität, Springer 2008
- [**Ho90**]: Hochstrasser, B.: Evaluating IT investment matching techniques to projects, in: Journal of Information Technology, (1990) 5, S. 215 – 221.
- [**II01**]: Irani, Z.; Love, R.: Developing a frame of reference for ex-ante IT/IS investment evaluation, in: European Journal of Information Systems, (2001) S. 1 – 9.
- [**KP96**]: Kitchenham, B.; Pfleeger, S., L.: Software Quality: The exclusive Target, in: IEEE Software 0740-7459 (1996), S. 12 – 21.
- [**Ok06**]: Okujava, S.: Wirtschaftlichkeitsanalysen für IT-Investitionen, WiKu-Verlag: Wissenschaftsverlag und Kulturedition Dr. Stein, Duisburg 2006.
- [**Po93**]: Post, B., Q.: A Business Case Framework for Group Support Technology, in: Journal of Management Information Systems, Vol. 9, No. 3, (1992), S. 7 – 26.
- [**SK09**]: Schröder, H.; Kesten, R.: Wirtschaftlichkeitsprognose und -kontrolle von IT-Investitionen: Abschlussbericht und Fallbeispiel, Arbeitspapiere der Nordakademie, Nr. 2009-03, Elmshorn 2009.
- [**SR13**]: Surka, M.; Rogers, C.: Maximizing the Value of Information Technology, in: CFO PUBLISHING, LLC, MARCH (2013), S. 9 – 20.
- [**SS11**]: Schwaber, K.; Sutherland, J.: Scrum Guide Der gültige Leitfaden für Scrum: Die Spielregeln, o. O., o. O., Oktober 2011, URL: <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum%20Guide%20-%20DE.pdf>, [Stand: 12.12.2013].
- [**WD08**]: Wöhe, G.; Döring, U.: Einführung in die Allgemeine Betriebswirtschaftslehre, Verlag Franz Vahlen, 23. Auflage, München 2008.
- [**WHS05**]: Wehrmann, A.; Heinrich, B.; Seifert, F.: Quantitatives IT-Portfoliomanagement: Risiken von IT-Investitionen wertorientiert steuern, in: Wirtschaftsinformatik, 48, 4 (2006), S. 234 – 245.
- [**Wo13**]: Wolf, M., R.: Software Engineering Phasenmodelle, Universität Wuppertal, BWL / Wirtschaftsinformatik, o. O., o. J., URL: <http://winfor.uni-wuppertal.de/fileadmin/wolff/Downloads/Grundstudium/EWI/Phasenmodelle.pdf>, [Stand: 16.01.2013].
- [**ZSB04**]: Zarnekow, R.; Scheeg, J.; Brenner, W.: Untersuchung der Lebenszykluskosten von IT-Anwendungen, in: WI – Schwerpunktaufsatz, WIRTSCHAFTSINFORMATIK 46 (2004) 3, S. 181–187.