

Spotlight

Wie spezifiziere ich den Leistungsumfang meines Projekts?



Eine themenspezifische Zusammenstellung von Fachartikeln
aus dem Projekt Magazin

www.projektmagazin.de

Mehlbeerenstr. 4, 82024 Taufkirchen

Tel: +49 89 2420798-0

Fax: +49 89 2420798-8

Wie spezifiziere ich den Leistungsumfang meines Projekts?

Am Anfang eines jeden Projekts muss für Auftraggeber und Auftragnehmer vollständig klar sein, welche Leistungen zu erbringen sind. In diesem Spotlight erfahren Sie, wie Sie Anforderungen identifizieren und erfassen, Leistungsmerkmale priorisieren und Ergebnisse definieren. Verschiedene Herangehensweisen, wie das Requirements Engineering oder das Quality Function Deployment ermöglichen eine präzise Abgrenzung des Leistungsumfangs und helfen beim klaren Formulieren der Anforderungen. Die Methoden unterstützen Sie mit Schritt-für-Schritt-Anleitungen in der Praxis.

Inhalt

Anforderungen identifizieren

1. Anforderungen an Softwareprodukte klar definieren
Requirements Engineering für Projektleiter Teil 1 Seite 4
2. Anforderungen an Softwareprodukte klar definieren
Requirements Engineering für Projektleiter Teil 2 Seite 15
3. Anforderungen an Softwareprodukte klar definieren
Requirements Engineering für Projektleiter Teil 3 Seite 23
4. Wie Sie die natürliche Sprache bändigen
Die SOPHIST-Satzschablone: Funktionale Anforderungen präzise formulieren Seite 33
5. Agile Softwareentwicklung mit Scrum und User Stories Seite 44

Leistungsmerkmale priorisieren

6. Kundenorientierte und marktgerechte Produktentwicklung
Mit QFD zur innovativen LED-Beleuchtung Teil 1 Seite 53
7. Kundenorientierte und marktgerechte Produktentwicklung
Mit QFD zur innovativen LED-Beleuchtung Teil 2 Seite 64
8. Was ist am wichtigsten? Anforderungen mit "MoSCoW" priorisieren Seite 72
9. Pareto-Prinzip für IT-Projekte
Die Anforderungen auf das Machbare reduzieren Seite 76

Ergebnisse definieren

10. Die Produktbasierte Planung nach PRINCE2 Teil 1 Seite 82
11. Die Produktbasierte Planung nach PRINCE2 Teil 2 Seite 94
12. Klar vereinbarte Leistungen – zuverlässige Pläne
Produktorientierte Planung von IT-Projekten Seite 103
13. Vollständig und schnell
Standardisierte Auftragsklärung im Sondermaschinenbau Seite 119

Anhang: Arbeitshilfen

Gesprächsleitfaden (pdf).....	Artikel 3, S. 23
Beispieltabelle (xlsx).....	Artikel 3, S. 23

Fachbeitrag

Anforderungen an Softwareprodukte klar definieren

Requirements Engineering für Projektleiter

Teil 1: Leistungsumfang und Rahmenbedingungen festlegen

Viele Projektleiter sind für die Beauftragung von Software verantwortlich. Das gilt keineswegs nur für reine Softwareprojekte, sondern auch für Organisationsprojekte, die auf Prozessentwicklung oder -optimierung abzielen und dabei eben auch die Entwicklung und Einführung einer neuen Software zum Gegenstand haben. Ob am Ende tatsächlich die Software herauskommt, die benötigt wird, und zwar ohne kostspielige Umwege und Change Requests, hängt entscheidend davon ab, ob es dem Projektleiter gelingt, realistische, eindeutige und vollständige Anforderungen zu formulieren.

Unabhängig davon, ob die Software intern oder extern entwickelt wird – der Projektleiter nimmt hier die oft ungewohnte Rolle des Auftraggebers wahr. Selbst wenn er über tiefes fachspezifisches Know-how verfügt und eine genaue Vorstellung hat, wie die Software aussehen soll, fehlt oft die Erfahrung, wie die Vorgaben in die "IT-Sprache" der Entwickler zu übersetzen sind. Dann ist es sinnvoll, sich mit den grundlegenden Regeln eines professionellen Requirements Engineering auseinanderzusetzen, um realistische, eindeutige und vollständige Anforderungen zu erstellen.

Bettina Zastrow und Elisabeth Wagner stellen in diesem dreiteiligen Beitrag die wesentlichen Elemente eines professionellen Requirements Engineerings vor und beschreiben, wie auch Projektleiter ohne tief gehende IT-Kenntnisse Schritt für Schritt zu einem guten Ergebnis kommen. Ein durchgängiges Beispiel erläutert die verschiedenen Schritte und gibt einen Überblick über die strukturellen Inhalte, die im Rahmen einer Softwareentwicklung zu definieren sind.

Dieser erste Teil zeigt, wie Sie eine eindeutige und vollständige Definition des Systems erstellen, indem Sie die Merkmale des Produkts und dessen Systemumgebung festlegen. Dazu gehören z.B. der fachliche Inhalt, der Systemkontext und die Stakeholder.

Ausgehend von diesen Vorgaben können Sie die Definition, wie im zweiten und dritten Teil beschrieben, immer weiter verfeinern, indem Sie das System aus der Nutzerperspektive beschreiben und das Ergebnis in "Use Cases" (Anwendungsfälle) und schließlich in Anforderungen "übersetzen", die für SW-Entwickler gut verständlich sind.

Autor



Bettina Zastrow

Informatikerin; seit 2011
Geschäftsführerin Zastrow
information development

GmbH, Autorin von Software-
Handbüchern und Online-Hilfetexten

Kontakt: info@bettina-zastrow.de



Elisabeth Wagner

IPMA Level D, Kommuni-
kationsberaterin, Journa-
listin, seit 20 Jahren in der
Unternehmenskommunikation tätig

Kontakt:

wagner@projektmarketing-partner.de

Mehr Informationen unter:

projektmagazin.de/autoren

Ziel: Das angestrebte System definieren

Bevor das Projekt oder Teilprojekt „Softwareentwicklung“ starten kann, gilt es, das angestrebte Ergebnis grundlegend zu definieren, also eine **"Systemdefinition"** zu erstellen. Sie legt fest, was das Produkt können soll, wer damit arbeitet, wofür es eingesetzt werden soll, welche Einflussfaktoren berücksichtigt werden müssen und welche Personen an der Systemdefinition beteiligt sind. Der Begriff "System" bezeichnet hier das zu entwickelnde Gesamtsystem, das Hardware-, Software- und Infrastrukturkomponenten enthalten kann.

Die Systemdefinition umfasst folgende Dokumente:

- Produktdefinition
- Leistungsumfang
- Systemkontext, bestehend aus Einflussfaktoren und Rahmenbedingungen
- Stakeholderliste
- Begleitdokumente

Der Einfachheit halber werden die Begriffe "Software" und "System" hier synonym verwendet.

Die Produktdefinition erarbeiten

Zweck der Produktdefinition ist, Klarheit über das Ergebnis der Produktentwicklung zu erhalten. Eine Produktdefinition beantwortet die Fragen: "Wofür dient diese Software?" und "Warum soll diese Software realisiert werden?" Damit eignet sie sich sehr gut als Kurzdefinition, um andere Beteiligte über das Vorhaben zu informieren.

Die Produktdefinition ist der erste Schritt zur Gestaltung eines neuen Systems. Da das Produkt zu diesem frühen Zeitpunkt noch im Ideenstadium ist, ist es notwendig, die Idee zu konkretisieren und festzulegen, was die zu entwickelnde Software können soll und was der Auftraggeber damit erreichen möchte. Natürlich kann die Idee auch als Vorgabe von einer Fachabteilung, der Bereichsleitung oder der Geschäftsführung kommen – an der prinzipiellen Vorgehensweise ändert sich dadurch nichts. Im Folgenden wird davon ausgegangen, dass der Produktmanager als interner Auftraggeber die Software in Auftrag gibt.

An der Produktdefinition beteiligen sich meist der Produktmanager sowie der Projektleiter. Zur Unterstützung können sie eine Person mit Markt- und Branchenkenntnissen hinzuziehen, in der Regel ein Business Analyst des Unternehmens. Sind ein oder besser mehrere potenzielle Nutzer verfügbar, stellt deren Perspektive in jedem Fall eine wichtige und wertvolle Ressource dar.

Leitfragen zur Produktdefinition

Um die Produktdefinition zu erarbeiten, eignen sich folgende Leitfragen:

- Was ist das zu lösende Problem?
- Wie sieht die Lösung aus?

- Was ist das Ziel des Systems?
- Was ist der Zweck des Systems?
- Wie gestaltet sich das Geschäftsmodell?
- Was geschieht bei Nichtrealisierung?

Mit diesen Fragen lässt sich die Grundidee für die Software erweitern bzw. präzisieren. Erste Antworten auf diese Fragen können z. B. im Rahmen eines Brainstormings gesammelt werden, an dem der Produktmanager und der Projektleiter teilnehmen. Weitere Teilnehmer könnten ein Business Analyst, ein potenzieller Nutzer und ein Entwickler sein. Wichtig ist, beim Brainstorming eine kreative Atmosphäre zu schaffen und alle Ideen zuzulassen.

Der nächste Schritt ist eine Konkretisierung der Vorschläge, dann folgt die Bewertung und Beseitigung der Unklarheiten und Schwachstellen und als Ergebnis mündet dies in einer Zieldefinition. Es ist unerlässlich, die erarbeiteten Inhalte schriftlich zu dokumentieren.

! Typische Stolpersteine bei der Produktdefinition sind, dass das Projekt zum **Selbstzweck** wird oder dass bei der **Risikoanalyse** wichtige Aspekte nicht berücksichtigt werden.

Oft wird in dieser Phase das zu lösende Problem der anvisierten Nutzergruppe aus den Augen verloren. Aus lauter Begeisterung wird das Projekt zum Selbstzweck und die fertige Lösung geht völlig an der ursprünglichen Fragestellung vorbei. Unter Umständen wird dann ein System beauftragt, mit dem das ursprüngliche Ziel nur mit Mühe oder gar nicht erreicht wird, so dass der Produkterfolg ausbleibt. Ein weiterer wichtiger Punkt ist die Risikoanalyse: Entsteht ein Schaden, wenn das System nicht realisiert wird, oder bleibt nur eine positive Wirkung aus? Mit dieser Überlegung kann die Dringlichkeit einer Softwareentwicklung bestimmt werden – als Konsequenz ist es möglicherweise sinnvoll, die geplante Investition an dieser Stelle zu stoppen. Wird das Projekt weiterverfolgt, hat dies ggf. eine Verschwendung von finanziellen und personellen Ressourcen zur Folge.

Als Ergebnis entsteht eine Produktdefinition, die in der Art eines Management Summary die Hauptpunkte auf ca. einer Drittel oder Viertel A4-Seite zusammenfasst.

Die fertige Produktdefinition, die das Vorhaben grob beschreibt, ist gut geeignet, um das Projekt den Stakeholdern vorzustellen und grundlegende Diskussionen auf Businesssebene zu ermöglichen – etwa darüber, ob dieses Vorhaben grundsätzlich Aussicht auf Erfolg hat und ob ein Projektantrag gestellt werden soll.

Beispielprojekt "Jingleshop"

Das nachfolgende Beispiel soll die einzelnen Schritte im Requirements Engineering verdeutlichen. Ziel im Beispielprojekt ist es, ein konkretes Produkt zu realisieren und zwar einen Onlineshop für Musikclips mit dem Namen "Jingleshop". Um die Produktdefinition zu erstellen, erarbeiten in diesem Fall Projektleiter und Produktmanager in einem ersten Brainstorming zu den oben genannten Fragen (Tabelle 1, links) zunächst Antwort-Entwürfe (Tabelle 1, Mitte). In einem weiteren Durchgang präzisieren sie diese und formulieren sie genauer aus (Tabelle 1, rechts). Das Ergebnis entspricht inhaltlich der Produktdefinition.

Frage	Antwort - Entwurf	Antwort - Präzisierung
Was ist das zu lösende Problem?	Unternehmen benötigen Stimmungsmusik, die nicht ohne weiteres erhältlich ist	Professionelle Nutzer von Musikdateien haben Schwierigkeiten, passende Jingles für Telefonwarteschleifen, Fahrstuhlbeschallung usw. zu beschaffen. Rechtliche Situation ist unklar (Lizenzgebühren, Verwertungsgesellschaften).
Wie sieht die Lösung aus?	Ein Onlineshop für Musikdateien	Eine Plattform, auf der Komponisten Audioclips einstellen können. Über eine Marktplatz-Funktion können die Rechte daran direkt von Kunden erworben werden.
Was ist das Ziel des Systems?	Angebot und Nachfrage zusammenbringen	Erleichterung der Anbahnung eines Vertragsverhältnisses zwischen Komponisten und Musiknutzern
Was ist der Zweck des Systems?	Kunden können Rechte an Musik erwerben	Komponisten laden ihre Jingles als Datei hoch und kategorisieren sie. Nutzer durchsuchen die Plattform nach Musik zu einem bestimmten Zweck. Mit einem Bestellvorgang werden die Rechte an der Musikdatei erworben.
Wie gestaltet sich das Geschäftsmodell?	Verkauf von Musikdateien	Der Vertrag kommt direkt zwischen dem Käufer und dem Verkäufer zustande. Für die Vermittlung erhebt COMPANY eine Provision.
Was geschieht bei Nichtrealisierung?	Geschäft geht verloren	Chance zur Erweiterung und Optimierung wird nicht genutzt, Innovationsfähigkeit und Konkurrenzfähigkeit wird nicht unter Beweis gestellt, entgangener Gewinn.

Tabelle 1: Die anhand der Eingangsfragen erarbeitete Produktdefinition im Beispiel "Jingleshop".

Für den konkreten Businessplan und die Planung der Softwareentwicklung ist dieser erste Überblick noch längst nicht ausreichend, als Vorlage für die Anwendungsentwicklung ebenso wenig. Deshalb folgt als nächster Schritt eine möglichst umfassende Beschreibung des Leistungsumfangs.

Den Leistungsumfang festlegen

Der Leistungsumfang dient dazu, die Ideen aus der Produktdefinition zu konkretisieren und zu verfeinern. Nach der Festlegung des Leistungsumfangs kann das Vorhaben den Stakeholdern oder der Geschäftsführung vorgestellt oder zum Beispiel ein Marktvergleich durchgeführt werden.

Im Leistungsumfang werden die Zielgruppe, die Funktionalität und der Einsatzbereich des Systems beschrieben. Der Leistungsumfang umfasst nicht nur die einzelnen Funktionen, sondern auch die Abläufe. Hilfreich ist hier auch die Abgrenzung zu anderen Systemen, die sich bereits auf dem Markt befinden, z.B. wenn ein Geschäftszweig für neue Kundengruppen erschlossen werden soll.

Um den Leistungsumfang zu erarbeiten, werden am besten die vorher genannten Rollen eingebunden: Projektleiter und Produktmanager, außerdem ggf. Markt-/Branchenspezialist, potenzieller Nutzer und SW-Entwickler. Die Rollen können auch mehrfach vertreten sein. Es empfiehlt sich ein persönliches Treffen aller Teilnehmer, wobei die Teilnehmerzahl zwischen zwei und fünf liegen sollte.

Leitfragen zum Leistungsumfang

Folgende Leitfragen eignen sich, um den Leistungsumfang zu erarbeiten:

- Wer sind die Nutzer des Systems?
- Was ist der fachliche Einsatzbereich?
- Was ist das geografische Einsatzgebiet?
- Was ist der funktionale Umfang?
- Wie lautet das Grobkonzept?
- Mit welchen Problemen ist zu rechnen? Welche Lessons Learned wurden bislang ermittelt?
- Welche ähnlichen Produkte gibt es, wie soll sich das zu erstellende System davon unterscheiden?

Wie bereits bei der Produktdefinition werden die ersten Antworten zunächst notiert und dann in einer weiteren Bearbeitungsschleife präzisiert. Stichworte und Ideen werden zunächst gesammelt, dann diskutiert, ausgearbeitet und schriftlich festgehalten.

Ein aufmerksamer Blick auf ähnliche Systeme lässt Inspirationen für eigene Funktionalitäten entstehen. In unserem Beispiel „Jingleshop“ vielleicht ein Online-Shop für CDs.

Als Ergebnis entsteht ein Leistungsumfang, der auf zwei bis vier A4-Seiten die Funktionalität und die Prozesse unter Berücksichtigung der Nutzer und ähnlicher Systeme beschreibt.

! Typische Stolpersteine in dieser Phase sind, dass die **IT-Kenntnisse** der Nutzer unter- bzw. überschätzt werden oder dass das **geografische Einsatzgebiet** unberücksichtigt bleibt.

Im ersten Fall ist es wichtig, die Zielgruppe möglichst genau zu definieren, denn je mehr Führung der Benutzer benötigt und je mehr Bedienfehler abgefangen werden müssen, desto höher ist der Programmieraufwand. Im zweiten Fall kann die Folge sein, dass rechtliche Vorgaben für den Bestellprozess, Währungen, Sprachen und Mehrwertsteuersätze anderer Länder nicht von vornherein in der Systemarchitektur angelegt werden und eine spätere Internationalisierung der Software extrem aufwendig und damit kostenintensiv ist.

Mit der Diskussion über den Leistungsumfang gelingt es, das Sachgebiet einzukreisen und zu durchdringen. So kann man sich die zukünftigen Benutzer der Software und den geplanten Mehrwert konkreter vorstellen.

Beispiel Jingleshop: Den Leistungsumfangs festlegen

Die folgende Tabelle zeigt, wie Ideen und Entwürfe zunächst gesammelt und dann zu konkreten Funktionalitäten ausformuliert werden können:

Frage	Antwort - Entwurf	Antwort - Präzisierung
Wer sind die Nutzer des Systems?	Jinglekäufer	Endkunden, die Jingles bei sich einsetzen
Was ist der fachliche Einsatzbereich?	Marketing, Unternehmen, Events	Lobbymusik, Warteschleifen in Telefonanlagen, Begleitmusik für Website-Intros und Imagevideoclips, akustische Eventbegleitung
Was ist das geografische Einsatzgebiet?	Deutschland	Deutschland, später Europa, später Welt
Was ist der funktionale Umfang?	Upload, Bestellung, Bezahlung, Download	Musikupload der Komponisten, Kategorisierung und Verschlagwortung, Suche und Produktanzeige, Detailbeschreibung, Merkzettel, Hörprobe, Bestellvorgang, Bezahlvorgang, Bestellhistorie, Download, Rückgabe
Wie lautet das Grobkonzept?	Musiker bietet Audioclips an, Kunde kauft Audioclips	Grundlage ist ein Medienverwaltungssystem. Das zu erstellende System umfasst die Produktpräsentation und ermöglicht den Kunden den Erwerb von Nutzungsrechten an der Musik. Es gibt unterschiedliche Nutzungsrechte für verschiedene Verwendungszwecke. Der Verkauf erfolgt auf verschiedenen Plattformen (Amazon, Facebook usw.) und auf der eigenen Website. Zusammen mit jedem Audioclip sind Metadaten gespeichert (technische Merkmale, Interpret, Stimmung, Instrumente, Genre, Länge etc.). Der Kunde kann die Clips suchen, anhören und dann erwerben. Mit dem Bestellprozess ist ein Lizenzvertrag verbunden. Eine Rückgabe ist möglich. Die Komponisten können selbst Musikclips hochladen. Diese werden im Media Asset Management System gespeichert, geprüft, verschlagwortet und aktiviert.
Mit welchen Problemen ist zu rechnen? Welche Lessons Learned wurden bislang ermittelt?	Kunde ist mit Lizenzverträgen, Urheberrechten oder Verwertungsgesellschaften nicht vertraut	Kunde soll entsprechend informiert und geführt werden, was die Themen Lizenzverträge und Urheberrechte angeht. Es werden nur GEMA-freie Produkte aufgenommen. (GEMA steht für die Gesellschaft für musikalische Aufführungs- und mechanische Vervielfältigungsrechte.)
Welche ähnlichen Produkte gibt es, wie soll sich das zu erstellende System davon unterscheiden?	Werbeagenturen Projektbezogene Musik Auftragskompositionen CD-Händler Musikverlage	Bisher gibt es keinen allgemeinen, einfach zu bedienenden Markt für professionell genutzte Jingles. Eine individuelle Beauftragung ist möglich, aber kostenintensiv. Die Nutzung von gekauften CDs ist rechtlich bedenklich und Eigenkompositionen wirken oft unprofessionell.

Tabelle 2: Fragen zum Leistungsumfang.

Im nächsten Schritt wird ermittelt, wie das zu erstellende System in den fachlichen, rechtlichen und technischen Gesamtkontext einzubetten ist.

Den Systemkontext bestimmen

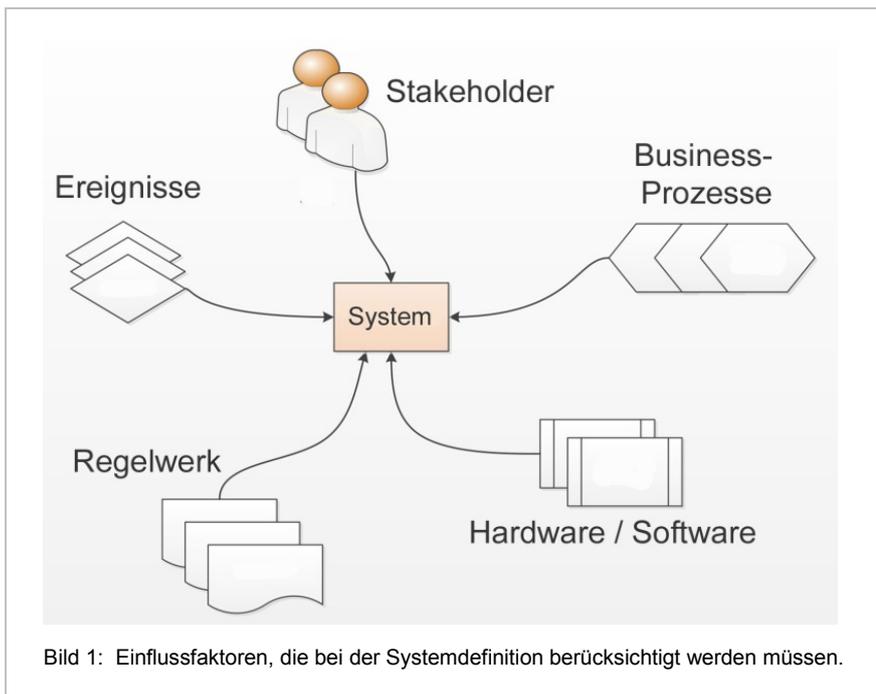
Der Systemkontext zeigt auf, in welchem rechtlichen, fachlichen und technischen Umfeld das System eingebunden werden soll. Dies sind zum einen **Einflussfaktoren**, die erkannt, niedergeschrieben und berücksichtigt werden müssen, zum anderen **Rahmenbedingungen**, die Projektleiter oder Produktmanager festlegen können. Die angenommene Anzahl an Nutzern kann z.B. ausschlaggebend für die Wahl der Hardware und des Betriebssystems sein.

Der Systemkontext benennt, was genau zum System gehört und was nicht. Er beantwortet die Fragen, welche Einflussfaktoren – wie bestehende Geschäftsprozesse, Stakeholder, rechtliche Rahmenbedingungen oder Ereignisse –

existieren und beachtet werden müssen und wie das neue System in die bestehende Systemlandschaft einzugliedern ist, d.h. welche Schnittstellen zu berücksichtigen sind. Dabei wird der angestrebte Zielzustand betrachtet.

Einflussfaktoren erfassen

Der ersten Teilschritt "Einflussfaktoren erfassen" macht eine umfangreiche Recherche notwendig. Insgesamt wirken fünf verschiedene Typen von Einflussfaktoren auf das System (Bild 1).



Die Einflussfaktoren im Einzelnen sind:

- Stakeholder
- Interne Geschäftsprozesse
- Eingetretene oder zu erwartende Ereignisse, wie z. B. Gesetzesänderungen, Produktinnovationen
- Systemlandschaft, bestehend aus Hardware, Software und Infrastruktur
- Interne und externe Regelwerke

Den ersten Teilschritt "Einflussfaktoren erfassen", kann der Projektleiter an einen Beauftragten delegieren, der vor Ort ist und im Unternehmen fachlich berät, recherchiert oder unterstützt. Dieser benötigt Kenntnisse des Marktes, der Branche und der internen Geschäftsprozesse. Für die Recherche ist außerdem eine Liste von Ansprechpartnern in den verschiedenen Abteilungen sowie externer Quellen hilfreich: So liegen der IT-Abteilung Informationen zu Hardware, Software und Infrastruktur vor, die Regelwerke sind in der Rechtsabteilung bekannt, Auskunft über Ereignisse können der Business Analyst oder die Fachpresse geben, die Stakeholder werden in der Regel vom Produkt-

manager definiert und die benötigten Informationen über Geschäftsprozesse kann der Business Process Designer des Unternehmens bzw. die betroffene Abteilung (im Beispiel: Vertriebsabteilung) beisteuern.

Zunächst bestimmt der Produktmanager die Ansprechpartner, dann werden die Informationen vom Projektleiter recherchiert und zusammengestellt. Zum Abschluss werden die Informationen vom Produktmanager und dem Projektleiter gemeinsam sortiert, ergänzt und bewertet.

! Ein typischer Stolperstein ist es, offenzulassen, was ausdrücklich **nicht zum System gehört**. Im Beispielprojekt wäre das z.B. der Mediaplayer, um die Musikdateien im Browser abzuspielen. Wird dies niedergeschrieben, können Unklarheiten, was Teil des Entwicklungsprojekts ist und was nicht, bereits zu Beginn unterbunden werden.

Als Ergebnis des Teilschritts "Ermitteln der Einflussfaktoren" entsteht eine Liste, Tabelle oder Grafik der Einflussfaktoren im Umfang von ca. einer bis zwei A4-Seiten. Diese beinhalten auch eine Tabelle der Nutzerrollen.

Rahmenbedingungen festlegen

Während der erste Schritt "Einflussfaktoren erfassen" Marktkenntnis erfordert, ist für den zweiten Schritt vor allem Sachkenntnis notwendig. Das Festlegen der Rahmenbedingungen ist Aufgabe des Produktmanagers und des Projektleiters. Zu den Rahmenbedingungen gehören z.B. das Mengengerüst und die Zielvorgaben, damit diese anschließend bei der Konzeptionierung der Systemarchitektur berücksichtigt werden können. Fehlen Informationen zu genauen Rahmenbedingungen, können Annahmen getroffen und Schätzungen gemacht werden.

! Ein typischer Stolperstein ist es, den **anvisierten Nutzertypen** und dessen realistisches Umfeld unberücksichtigt zu lassen. Handelt es sich um einen künstlerisch-kreativen Nutzer mit hoher Social-Media-Aktivität, der von unterwegs mit einem mobilen Gerät arbeitet? Oder kann davon ausgegangen werden, dass der Nutzer an einem Schreibtisch sitzt und ihm ein leistungsfähiger Computer und ein hochauflösender Bildschirm zur Verfügung stehen?

Als Ergebnis des Teilschritts "Festlegen von Rahmenbedingungen und Annahmen" entsteht ein Text oder eine Tabelle der Rahmenbedingungen und Annahmen im Umfang von ca. einer bis drei A4-Seiten.

Mit der Bestimmung des Systemkontextes ist der Rahmen gesteckt, in dem das Projekt durchgeführt werden soll. Es sind die Aspekte benannt, die bei der Softwareentwicklung zu berücksichtigen sind und fachliche, technische und organisatorische Rahmenbedingungen festgelegt. Damit erhöht sich gleichzeitig auch die Aufmerksamkeit für Änderungen im Systemumfeld, was sich im Verlauf des Softwareprojekts als nutzbringend erweisen kann. Dies können sein: Presseankündigungen von Mitbewerbern und zu Messen und Kongressen, Veröffentlichungen von Wirtschaftsverbänden und Neuerungen technischer Art sowie geplante Gesetzesänderungen, im konkreten Beispiel in Bezug auf das Urheber- oder Datenschutzrecht.

Beispiel Jingleshop: Den Systemkontext bestimmen

Frage	Antwort - Entwurf	Antwort - Präzisierung
Welche Ereignisse sind zu berücksichtigen?	Markteinführung Konkurrenzprodukt Änderungen im Urheberrecht	Markteinführung PHÖNIS der Firma Hendricks 10.10.2015 ACTA-Vereinbarung 04.04.2012 (ACTA steht für Anti-Counterfeiting Trade Agreement, dt. Anti-Produktpiraterie-Handelsabkommen)
Welche Stakeholder haben Einfluss auf das System?	Kunde, Komponist, Buchhaltung, Administrator	Kunde (Endkunde, Agentur), Komponist bzw. Musiker, Buchhaltung, Report-Ersteller, fachlicher Administrator (Produktpflege)
In welche Business-Prozesse ist das System einzugliedern?	Vertrieb	Direktvertrieb von Musikdateien
Welche Hardware und Software ist im Einsatz, zu welcher Software muss eine Schnittstelle bereitgestellt werden?	Server und Clients	Server und Datenbanken: Linux Server, Windows Server, Microsoft SQL-Datenbank Clients: PC, Tablet, Smartphone Betriebssysteme: Windows 8.x, iOS, Android, Blackberry. Browser: Internet Explorer, Firefox, Safari, Chrome Unternehmenssoftware: CRM-System ETUSK (CRM steht für Customer Relationship Management System), Buchhaltungssystem MIRAMAN, Produktinformationssystem CIDF, Marketingsystem DART, Medien Asset Managementsystem ADOT, Redaktionssystem ONEWO Drittanbieter: Zahlungsdiensteanbieter, Geldinstitute
Welche Regelwerke sind zu beachten?	Gesetze, Normen, Standards, Firmenrichtlinien	Bundesdatenschutzgesetz, GmbH-Gesetz (GmbHG), Telekommunikationsgesetz (TKG), Telemediengesetz (TMG), Gesetz gegen den unlauteren Wettbewerb (UWG), Grundsätze ordnungsgemäßer Buchführung (GoB), Grundsätze ordnungsmäßiger DV-gestützter Buchführungssysteme (GoBs), Grundsätze zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen (GDPdU) DIN EN ISO 9000 Kundenansprache (Du, Sie)

Tabelle 3: Fragen zum Systemkontext, Teilschritt 1, Einflussfaktoren.

Frage	Antwort - Entwurf	Antwort - Präzisierung
Welche Rahmenbedingungen gelten?	Budget Termine, Milestones Voraussetzungen	1,8 Mio EUR 6 Monate Freigabe durch Geschäftsführung
Welche Annahmen werden gemacht?	Mengengerüst Risiken Abhängigkeiten	Mengengerüst: <ul style="list-style-type: none"> 5.000 Jingles bei Markteinführung 800 Zeichen pro Produkttext 200 Komponisten 30.000 Nutzer Risiken: <ul style="list-style-type: none"> Änderungen im Urheberrecht Konkurrenzplattformen Abhängigkeiten: <ul style="list-style-type: none"> Keine

Tabelle 4: Fragen zum Systemkontext, Teilschritt 2, Rahmenbedingungen.

Da bei einem gelungenen Softwareprodukt das Hauptaugenmerk auf den Bedürfnissen und Wünschen der Nutzer liegt, ist der nächste Schritt, eine vollständige und aussagekräftige Liste der Stakeholder zu erstellen.

Eine Stakeholderliste erstellen

Ein Stakeholder ist eine Person oder ein Mitglied einer Interessengruppe, die Einfluss auf das zu erstellende System hat. In erster Linie sind dies die potenziellen Hauptnutzer, die künftig das System verwenden sollen – in unserem Beispiel die Kunden in den Rollen Komponisten/Musiker und Jingle-Käufer. Weitere Stakeholder können z.B. sein: der Budgetverantwortliche, das kaufmännische Personal für Buchhaltung und Auswertungen, sowie diejenigen, die für die Produktpflege verantwortlich sind.

Eine frühe Einbindung der künftigen Nutzer ermöglicht es, die Handhabung und die Abläufe von vornherein gemäß deren Bedürfnissen zu gestalten. Damit kann der Entwicklungsaufwand erheblich reduziert werden.

Es empfiehlt sich, die Nutzerrollen aus der Definition des Systemkontextes als Basis für die Benennung von Stakeholdern zu verwenden. Der Produktmanager legt für jede Rolle konkrete Personen als Key User fest, die bei der Ermittlung von Anforderungen befragt werden sollen, ggf. in Absprache mit deren Vorgesetzten.

! Ein typischer Stolperstein ist es, dass es versäumt wird, die **Stakeholder** über ihre Mitarbeit im Projekt **ausreichend zu informieren**. Es muss sichergestellt sein, dass die benannten Personen auch tatsächlich Zeit für eine Befragung aufwenden können. Handelt es sich um Mitarbeiter anderer Firmen oder Privatpersonen, ist deren Zustimmung einzuholen.

Das Ergebnis dieses Schritts ist eine Liste mit Ansprechpartnern, die Rolle, Name, Kontaktdaten und Firma/Abteilung (sofern zutreffend) enthält. Je nachdem, wie viele Rollen beteiligt sind, kann die Liste bis zu zwei DIN A4-Seiten umfassen. Sinnvollerweise ist auch eine Angabe zur Erreichbarkeit (z. B. Abwesenheiten wegen Urlaub) aufzunehmen.

Begleitdokumente

Glossar

Da die beteiligten Personen aus völlig unterschiedlichen Bereichen, unter Umständen aus verschiedenen Branchen kommen und ihr Hintergrund heterogen ist, ist es sehr empfehlenswert, ein Glossar anzulegen. Hier werden Begriffe in wenigen kurzen Sätzen erläutert, so dass alle Beteiligten ein gemeinsames Verständnis für das Projekt und die genutzten Begrifflichkeiten entwickeln können. Dies trägt dazu bei, dass Missverständnisse und Fehlinterpretationen gar nicht erst entstehen. Unter Umständen kann dies den Aufwand bei der Entwicklung und der Implementierung begrenzen, da die Zeit für immer wiederkehrende Begriffserläuterungen eingespart wird.

Rollendefinitionen

Um Klarheit zu schaffen, welcher Nutzer welche Aufgaben durchführt, bietet sich an, eine Liste der Definitionen für die verschiedenen Nutzerrollen anzulegen. Insbesondere gilt dies für Wortkreationen wie "Einsteller", womit der Urheber des Jingles gemeint ist. Auch können darin die Beziehungen untereinander verdeutlicht werden, z. B. können mehrere Rollen Teil einer übergeordneten Gruppe sein: Die Gruppe Fachadministratoren kann den kaufmännischen Administrator und den Newsletterversender beinhalten. Mit einer eindeutigen Rollendefinition kann Mehrarbeit infolge von Missverständnissen abgewendet werden.

Eine Rollendefinition hat den weiteren Vorteil, dass die beauftragten Entwickler sich ein Bild machen können, welche Rechte einem Benutzer bzw. einer Benutzergruppe mit dieser Rolle zugewiesen werden sollen. Sie dient also als Basis für ein Rechtekonzept. So kann z.B. die Rolle des Administrators in einen "Newsletterversender", einen "Reportdesigner", einen "Buchhaltungsverantwortlichen" und weitere aufgesplittet werden, um die Voraussetzung für eine spätere Skalierung zu schaffen.

Namenskonventionen und Dateiablage

Hilfreich ist ebenfalls eine Verständigung über Namenskonventionen und Standorte für Dateien und Verzeichnisse sowie eine Einigung über Vorgehensweisen für die Dokumentenbearbeitung. Im Laufe des Projekts werden mehr und mehr Dokumente verändert und ergänzt. Diese Konventionen dienen dazu, stets den Überblick zu bewahren. Sinnvoll ist die gemeinsame Festlegung einer Verzeichnisstruktur, die externe Dokumente (beispielsweise Flyer von Mitbewerbern), Ressourcen (z. B. Bildmaterial, Vorlagen), Dokumentation und Kontaktdaten enthält. Wichtig ist eine Einigung darüber, wer die Hoheit über die jeweiligen Dokumente hat, wie Dateien benannt und versioniert werden sollen (siehe auch: "**Benennen Sie Projektdateien sinnvoll!**", Projekt Magazin 15/2012) und vor allem, wie der Lebenszyklus eines Dokuments vom Entwurf bis zur Freigabe abgebildet werden soll.

Ausblick

Nachdem die zu entwickelnde Software inklusive der Rahmenbedingungen vollständig beschrieben ist, wird diese Beschreibung in zwei weiteren Schritten in die Anforderungssprache der Softwareentwicklung "übersetzt".

Der erste Schritt ist das Modellieren der Use Cases. Ein Use Case beschreibt, was ein Akteur – also ein Benutzer oder ein anderes System (Hardware, Software) – vom System erwartet. Das System wird dabei als "Black Box" betrachtet und es werden die Handlungen beschrieben, die ein Akteur mithilfe des Systems ausführt. Als Black Box wird ein System bezeichnet, dessen Funktionsweise unbekannt ist und lediglich dessen Verhalten, zur Beobachtung herangezogen wird, in diesem Fall Eingabe – Ausgabe. Wie Sie vorgehen, um die erforderlichen Use Cases zu modellieren, lesen Sie im nächsten Teil dieser Artikelserie.

Der dritte und abschließende Teil beschreibt schließlich, wie Sie aus den Use Cases Anforderungen ableiten. Anforderungen geben Auskunft darüber, ob es sich um eine Nutzeraktion, eine autonome Systemaktivität oder ein Schnittstelle handelt. Anforderungen enthalten nachprüfbar Bedingungen, Eigenschaften und Prozesse. Sie sind so verfasst, dass sie von einem SW-Entwickler klar und eindeutig verstanden und umgesetzt werden können.

Fachbeitrag

Anforderungen an Softwareprodukte klar definieren

Requirements Engineering für Projektleiter

Teil 2: Mit Use Cases den Funktionsumfang beschreiben

Im ersten Teil haben Sie erfahren, wie Sie eine eindeutige und vollständige Definition einer Software erstellen. Sie legen damit fest, was das Produkt können soll und beschreiben, was der Auftraggeber damit erreichen möchte, welche Einflussfaktoren und Rahmenbedingungen gelten und wer die Stakeholder sind. Dieser zweite Teil zeigt, wie es Ihnen auch als Nicht-ITler gelingt, sogenannte Use Cases bzw. Anwendungsfälle zu erstellen, mit denen Sie den Funktionsumfang eines Systems aus Anwenderperspektive konkreter erfassen.

Beim Erstellen der Use Cases stehen vor allem die zukünftigen Nutzer im Mittelpunkt: Was erwarten diese vom System, welche Aufgaben möchten sie erledigen und wie gehen sie dabei vor? Für jede Nutzerrolle, die im Rahmen der Produktdefinition bestimmt wurde, muss der Projektleiter die zugehörigen Aktivitäten ermitteln – in der Regel gehören dazu die Rollen "Kunde", "Einsteller" (im Beispielprojekt „Jingleshop“ aus Teil 1 sind das die Komponisten) und "Nutzer der kaufmännischen und organisatorischen Verwaltungsfunktionen" (z.B. Buchhaltung, Produktpflege). Es kann hilfreich sein, sich den gesamten Ablauf ohne das System zu vergegenwärtigen: Wie würde z.B. ein Verkauf in einem Ladengeschäft aussehen, welche Personen wären im Gesamtablauf beteiligt?

Use Cases und User Stories

Use Cases sind nicht mit den User Stories im agilen Entwicklungsprozess zu verwechseln, haben aber Berührungspunkte. Denn aus den Use Cases können meist mehrere User Stories (Anforderungen) abgeleitet werden, sofern Auftraggeber und Projekt sich darauf geeinigt haben, auf Basis einer solchen Spezifikation zu arbeiten. Weitere Erläuterungen dazu enthalten die Artikel "[Agile Softwareentwicklung mit Scrum und User Stories](#)" (Projekt Magazin 2/2010) sowie "[So vermeiden Sie Stolpersteine bei User Stories](#)" (Projekt Magazin 17/2012).

Eine Liste mit Nutzeraktivitäten erstellen

Es ist sinnvoll, zunächst eine formlose Liste von Aktivitäten für jede Nutzerrolle zu erstellen. Handelt es sich bei den Nutzern um "Normalverbraucher", ist es von Vorteil, deren Blickwinkel einzunehmen und sich das fertige System vorzustellen. Welche Aufgaben möchte dieser mit Hilfe des Systems erledigen? Sind spezielle Branchenkenntnisse gefragt, ist der Business Analyst des Unternehmens ein passender Ansprechpartner.

Autor



Bettina Zastrow

Informatikerin; seit 2011
Geschäftsführerin Zastrow
information development

GmbH, Autorin von Software-
Handbüchern und Online-Hilfetexten

Kontakt: info@bettina-zastrow.de

Mehr Informationen unter:

projektmagazin.de/autoren



Elisabeth Wagner

IPMA Level D, Kommuni-
kationsberaterin, Journa-
listin, seit 20 Jahren in der
Unternehmenskommunikation tätig

Kontakt:

wagner@projektmarketing-partner.de

Mehr Informationen unter:

projektmagazin.de/autoren

Use Cases werden im Format <Rolle> <Aktivität> <Objekt> beschrieben. Beispiele: "Kunde registriert sich am System", "Kunde sucht Produkt", "Kunde kauft Produkt".

Mit einer systematischen Herangehensweise kann der Projektleiter schnell feststellen, ob die Liste vollständig ist. Dazu gehört erstens die Überprüfung auf komplementäre Aktivitäten: Gibt es z.B. "Nutzer meldet sich am System an", gehört auch "Nutzer meldet sich vom System ab" dazu. Eine zweite Vollständigkeitsprüfung sollte die Vorgänge Erstellen, Lesen, Ändern und Löschen umfassen, die im Englischen mit CRUD bezeichnet werden (Create, Read, Update, Delete). Diese lassen sich unter anderem auf Produkte (im Beispiel: Jingles) anwenden. Drittens gibt es Prozessfunktionen, die sich aus der Natur des Verkaufs ergeben: "Kunde kauft Produkt", "Kunde bezahlt Produkt", "Kunde gibt Produkt zurück" usw.

Es entsteht zunächst ein Entwurf, in dem für jede Nutzerrolle eine Sammlung von Aktivitäten beschrieben ist. Ob dies in Textform oder mit Hilfe einer Mindmap geschieht, bleibt dem Projektleiter überlassen. Der Aufwand bewegt sich im Bereich von mehreren Stunden, bei sehr komplexen Systemen ggf. von mehreren Tagen.

Beispiel Jingleshop: Use-Case-Liste

Nutzerrolle	Use Case
Kunde	... registriert sich am System
Kunde	... hebt die Registrierung auf
Kunde	... sucht Produkt
Kunde	... zeigt Produkt an

Tabelle 1: Auszug aus einer Use-Case-Liste im Beispiel "Jingleshop".

! Oft wird versäumt, die Nutzerperspektive einzunehmen. Dies führt zu Use Cases, bei denen die Information fehlt, woraus die Aktion besteht und wer sie ausführt – wie z.B. bei der Formulierung "Merkzettel anlegen". Dieser kann präziser mit Hilfe von zwei Use Cases ausgedrückt werden: "Kunde fügt Artikel dem Merkzettel hinzu" und "Kunde löscht Artikel vom Merkzettel".

In jeden Fall ist es hilfreich, sich ähnliche Systeme genauer anzusehen, im Beispiel des Jingleshops z.B. einen Onlineshop für CDs. Im Zweifelsfall hilft ein potenzieller Nutzer weiter: Im Unternehmen der designierte Key User, ein Mitglied des Testteams, der Business Analyst oder ein Kundenbetreuer, je nachdem, für welchen Nutzerkreis das System entwickelt wird. Handelt es sich um das Nachfolgeprodukt eines bestehenden Systems, ist es sehr aufschlussreich, einen Nutzer einen oder zwei Tage bei seiner Arbeit mit dem abzulösenden System zu beobachten. Ein idealer Key User wäre im Beispiel des Jingleshops ein Mitarbeiter, der Erfahrung in der Beschaffung von Aufzugsmusik oder Warteschleifenjingles hat oder sich zumindest mit Onlineshops gut auskennt.

Schnittstellen zu angrenzenden Systemen berücksichtigen

Nicht nur die Nutzer, sondern auch die Schnittstellen zu angrenzenden Systemen sind Bestandteil der Use-Case-Modellierung. Mit diesen Use Cases ergänzt der Projektleiter die Liste entsprechend. Beispiel: "Abrechnungssystem übernimmt die Bestellung für die Rechnungserstellung".

Use Cases grafisch darstellen

Die vollständige Liste mit den Use Cases kann im nächsten Schritt in eine grafische Darstellung überführt werden. Das System wird dabei in Beziehung zu seiner Umgebung gesetzt und die Abhängigkeiten der Use Cases zueinander werden besser erkennbar.

Um ein Diagramm zu erzeugen, das zeigt, wer was mit dem System tun möchte, hat sich die UML-Notation (Unified Modelling Language) bewährt. Diese stellt das System als umgebenden Kasten, die Akteure als Strichmännchen bzw. Rechtecke und die Use Cases als Ellipsen dar (siehe Bild 2: Use-Case-Diagramm für das Beispiel Jingleshop). Damit die Übersichtlichkeit nicht verloren geht, werden Reihenfolge, Bedingungen und konkrete Ausprägungen (z.B. "Was passiert im Nichtfall?") zunächst ausgeklammert. Auf dem Markt existieren zahlreiche UML-Tools, die sich für das Erstellen eines solchen Use-Case-Diagramms eignen.

Wird nicht mit einem UML-Tool gearbeitet, lässt sich das Diagramm auch mit einem anderen System, z.B. Microsoft Visio oder PowerPoint darstellen.

Fehlt die entsprechende Kompetenz im Projektteam, kann ein Facharchitekt des Unternehmens bei der Modellierung der Use Cases hinzugezogen bzw. externe Beratung in Anspruch genommen werden. Sind die Use Cases einfach und überschaubar, kann auf die grafische Darstellung verzichtet werden.

Beispiel Jingleshop: Use-Case-Diagramm

Der Auszug aus einem Use-Case-Diagramm in Bild 1 zeigt exemplarisch einige Aktivitäten der Nutzerrollen "Kunde", "Komponist" sowie der externen Systeme "CRM" (Customer Relationship Management) bzw. "Buchhaltung". Zur besseren Übersicht sind die Aktivitäten nicht als vollständige Sätze ("Kunde sucht Produkt"), sondern in der Infinitiv-Form formuliert ("Produkt suchen"). Der zugehörige Akteur lässt sich aus dem Diagramm leicht ablesen. Zusammenhängende Tätigkeiten werden entsprechend verknüpft dargestellt, z.B. ist im Diagramm der Use Case "Produkt bezahlen" im Use Case "Produkt kaufen" eingeschlossen.

Im Beispiel des Jingleshops würde das fertige Use-Case-Diagramm ca. 70 bis 100 Use Cases enthalten. Bei diesem Umfang bietet sich eine Auftrennung in mehrere einzelne Diagramme an, die jeweils einen Themenbereich abdecken.

Während der Modellierung der Use Cases kristallisieren sich die betroffenen Akteure deutlicher heraus. Es ist an dieser Stelle sinnvoll, Rollendefinitionen für die namentlich bekannten Nutzer zu erstellen und diese z.B. im Glossar abzulegen. Beispiele für Rollendefinitionen sind "Ein 'Kunde' ist ein Nutzer, der einen Bestellvorgang abgeschlossen hat" oder "Ein 'Gast' ist ein Nutzer, der eine Bestellung ohne Registrierung vornimmt".

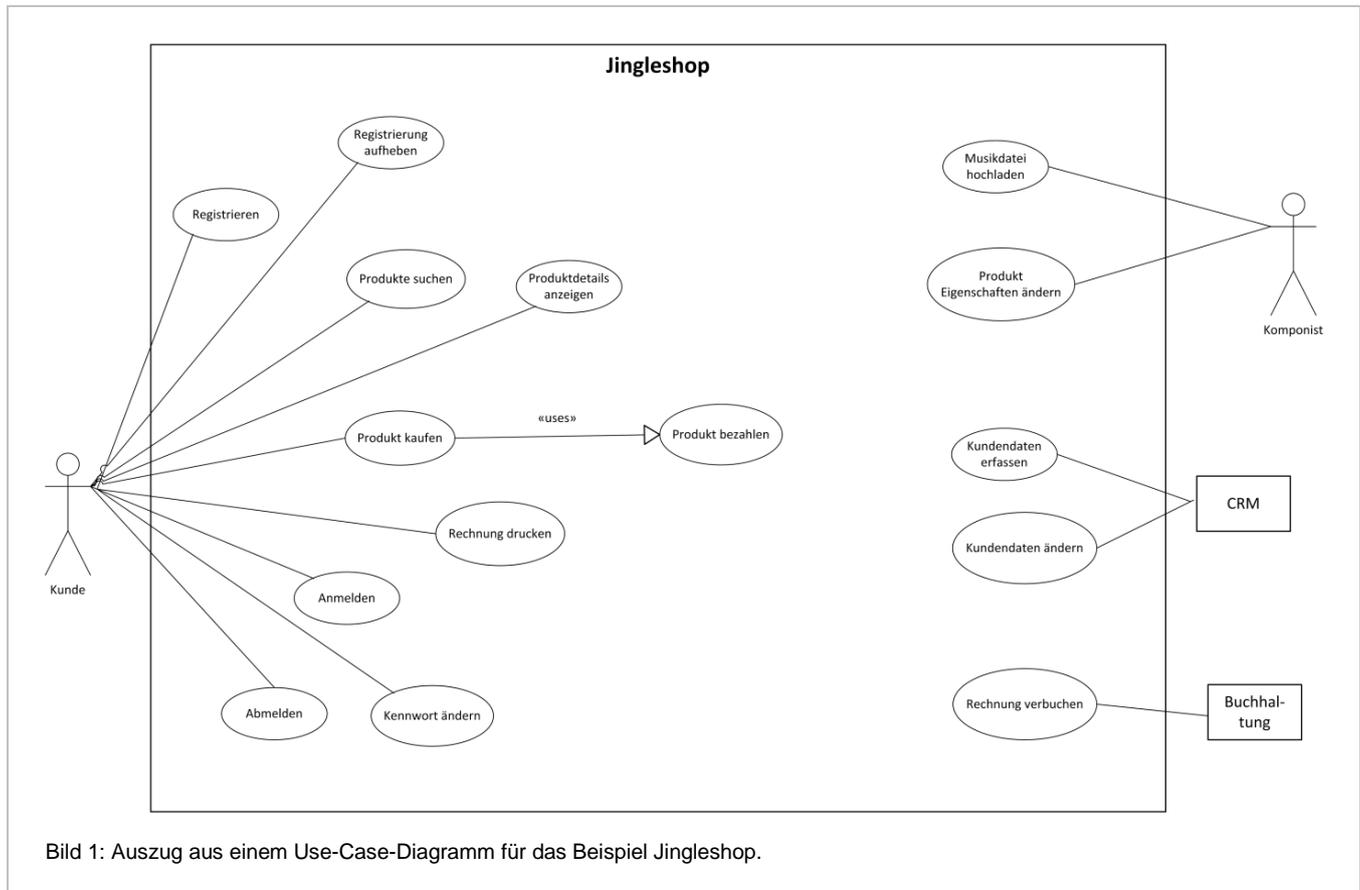


Bild 1: Auszug aus einem Use-Case-Diagramm für das Beispiel Jingleshop.

Mit Toolunterstützung gelingt das Erstellen des Diagramms in der Regel in wenigen Stunden, ohne Toolunterstützung sollte man mit zwei oder mehr Tagen rechnen. Wie hoch der Aufwand tatsächlich ist, hängt jedoch stark von der vorhandenen analytischen und schöpferischen Kompetenz und der Komplexität des Systems ab.

Als Ergebnis dieses Schritts entsteht ein Use-Case-Diagramm, das die Information enthält, welcher Nutzer und welches angrenzende System welche Aktivität ausführt. Darüber hinaus enthält es Prozesse bzw. Aktivitäten, die miteinander in Beziehung stehen.

Gruppierung der Use Cases in Features

Als nächstes bietet es sich an, die Use Cases zu Leistungsmerkmalen (Features) zusammenzufassen, also nach Merkmalen, die als in sich geschlossene Einheit betrachtet werden können. Beispiele sind die Suchfunktion oder der Registrierungsvorgang. Bild 2 verdeutlicht den Zusammenhang zwischen Features, Use Cases und Anforderungen.

Zweck von Features

Die Gruppierung von Use Cases in Features hat mehrere Vorteile:

- Sie ermöglicht, die Liste der Use Cases zu vervollständigen, da durch die themenzentrierte Sicht des Featurekatalogs evtl. noch weitere Use Cases offensichtlich werden.

- Sie hilft den Entwicklern, eine grobe Aufwandsschätzung vorzunehmen.
- Sie erleichtert die Umsetzung, weil die Entwicklungsarbeit in größere Funktionsblöcke unterteilt wird.

Dem entstandenen Featurekatalog lassen sich nicht nur die Use Cases eindeutig zuordnen, sondern auch die Verantwortlichkeiten innerhalb des Entwicklerteams. Das vereinfacht zum einen die standortübergreifende Zusammenarbeit, zum anderen ermöglicht es dem Projektleiter einen Überblick über den Fortschritt der Implementierung.

Features erarbeiten

Um einen Featurekatalog zu erstellen, fasst der Projektleiter thematisch gleichartige Use Cases zusammen und versieht diese mit einem geeigneten Oberbegriff. Als Grundlage kann die Liste der Use Cases aus der Entwurfsphase am Kapitelanfang dienen.

Beim Formulieren von Use Cases kommt es oft vor, dass weitere Features zu Tage treten, die bislang noch nicht explizit thematisiert wurden. Im Beispiel "Jingleshop" könnte es z.B. sein, dass beim Durchleuchten der Suche die Erkenntnis gewonnen wird, dass die Produkte mit Metadaten wie Kategorien und Keywords versehen werden müssen. So entsteht ein neues Feature: die Kategorisierung nach Genres.

! Ein typischer Stolperstein ist eine zu starke oder zu schwache Granularität. Ein guter Richtwert sind fünf bis fünfundzwanzig Use Cases pro Feature.

Als Ergebnis dieses Schritts entsteht eine Aufstellung aller Features mit einer Zuordnung zu einem oder mehreren Use Cases.

Beispiel Jingleshop: Features

Die aus den Use Cases hervorgehenden Features im Beispiel Jingleshop (Tabelle 2) sind charakteristisch für Shoplösungen.

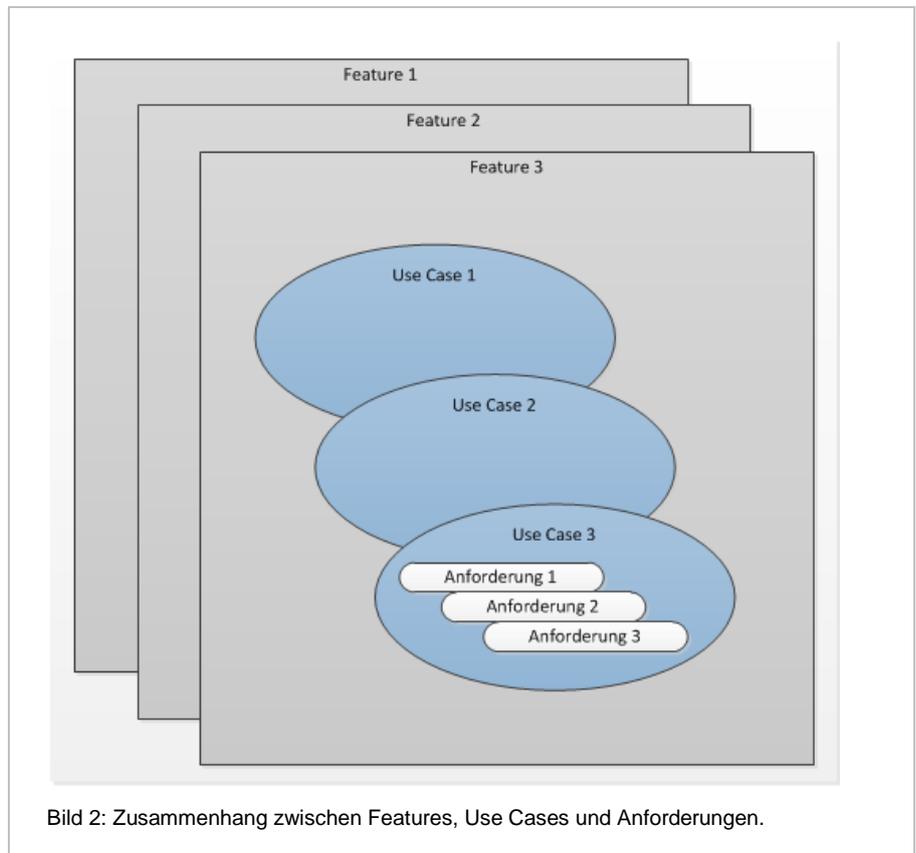


Bild 2: Zusammenhang zwischen Features, Use Cases und Anforderungen.

Feature		
Registrierungsvorgang	Merkfunktion	Upload/Download (Kunde, Komponist)
Anmelden/Abmelden/Kennwort vergessen	Bestellvorgang	Cross-Selling
Suche/Filterfunktion	Bezahlvorgang	Nutzerprofil
Produktliste/Detailanzeige	Mailversand	Auswertungen
Kategorien	PDF-Druck der Rechnung	
Hörprobe	Export/Import (Backend)	

Tabelle 2: Featureliste im Beispiel Jingleshop.

Feature	Use Case
Registrierungsvorgang	Nutzer registriert sich am System
Registrierungsvorgang	Nutzer hebt die Registrierung auf
Cross-Selling	Nutzer zeigt Cross-Selling-Produkte an
Cross-Selling	Nutzer ruft Cross-Selling-Produkte auf
Cross-Selling	Nutzer pflegt Cross-Selling-Produkte

Tabelle 3: Features mit zugehörigen Use Cases im Beispiel Jingleshop (Auszug).

Use Cases formell dokumentieren

Die formelle Dokumentation ist der letzte Schritt bei der Erarbeitung der Use Cases, nachdem diese zunächst in einer formlosen Liste gesammelt und anschließend in einem Diagramm visualisiert und untereinander in Beziehung gesetzt wurden. In diesem letzten Schritt erfolgt die feine Ausarbeitung und Dokumentation, wobei die Use Cases um Metainformationen angereichert werden.

Für die Dokumentation von Use Cases gibt es diverse Methoden und Formulare. Seit einigen Jahren sind Softwareanwendungen auf dem Markt, in denen Use Cases, Anforderungen und Test Cases modelliert werden können, die sich dann zusammen mit den einleitenden und beschreibenden Texten direkt als Fachkonzept exportieren lassen. Dadurch entfällt mehrfacher Pflegeaufwand, die entstandenen Dokumente lassen sich dennoch ausdrucken und damit rechtssicher freigeben und unterschreiben. Steht eine solche Anwendungen nicht zur Verfügung, können Microsoft Word oder Excel verwendet werden.

Je nach Komplexität beträgt der Aufwand zur formellen Dokumentation eines Use Cases ca. 15 bis 60 Minuten.

Element	Beschreibung
Nr.	Eindeutige Kennung für die Identifizierung des Use Cases
Name	Kurzname des Use Cases aus dem Use Case-Diagramm für Referenzzwecke
Kurzbeschreibung und Zielsetzung	Beschreibung der Aktivität in maximal zwei Sätzen, aus der hervorgeht, was der Akteur mit dessen Ausführung erreichen möchte
Priorität	Rangfolge für die Umsetzung
Akteure	Beteiligte Akteure (Nutzer und/oder Systeme)
Vorbedingungen	Bedingung, die erfüllt sein muss, damit der Use Case zur Ausführung kommen kann.
Ablauf <ul style="list-style-type: none"> • Start • Normaler Ablauf • Varianten • Ausnahmen • EPK (Ereignisgesteuerte Prozesskette) 	Detaillierter Ablauf innerhalb des Use Cases: Wann beginnt er, wie ist der übliche Ablauf, gibt es Varianten, gibt es Ausnahmen, und welche ereignisgesteuerte Prozesskette wird ggf. in Gang gesetzt?
Anforderungen	Kennung der Anforderungen, die mit diesem Use Case in Zusammenhang stehen. Diese Zeile bleibt zunächst frei und wird später ausgefüllt.
Umsetzungshinweise	Vorgaben oder Wünsche zur Umsetzung, z. B. als Menüpunkt oder Button
Offene Punkte	Sachverhalte, die noch zu klären sind

Tabelle 4: Typische Bestandteile einer Use-Case-Definition.

Mit diesem Katalog von Merkmalen ist der Projektleiter gut gerüstet, um vollständige Angaben zu einem Use Case zu erarbeiten. Die Priorität legt der Produktmanager fest. Die Vorbedingungen ergeben sich aus dem fachlichen oder technischen Kontext. Konkretes Beispiel aus dem Jingleshop: Ist der Use Case "Nutzer meldet sich vom System ab", lautet die Vorbedingung "Nutzer ist am System angemeldet".

! Ein typischer Stolperstein ist, die Umsetzungshinweise mit dem Use Case zu verwechseln. Insbesondere wenn das neue System an ein bestehendes angelehnt werden soll, haben Benutzer bestimmte Vorstellungen, wie bestimmte Dinge realisiert werden sollen.

Ein Negativbeispiel ist der Use Case "Ähnliche Produkte in der rechten Seitenleiste anzeigen". Tatsächlich handelt es sich hierbei um einen Umsetzungshinweis. Ein besserer Vorschlag für die Formulierung des Use Cases wäre: "Kunde ruft Cross-Selling-Produkte auf". Im ersteren Fall fällt es dem Entwickler bei der Realisierung schwer, den Sinn und Zweck der Funktion nachzuvollziehen, und es entsteht möglicherweise nur eine rein textuelle Auflistung ähnlicher Produkte – ohne hinterlegte Hyperlinks. Für die Information über die Platzierung in der rechten Seitenleiste ist die Zeile "Umsetzungshinweise" vorgesehen.

! Es hat sich als praktisch erwiesen, die Kennung für den Use Case mit einem Präfix zu versehen, z. B. "UC-". So können Use Cases später schneller aufgefunden, sortiert und gruppiert werden.

Für jeden identifizierten Use Case werden wie oben beschrieben Details wie Akteure, Priorität sowie der normale Ablauf, dessen Varianten und Ausnahmen spezifiziert.

Ergebnis ist eine Tabelle mit Merkmalen für jeden Use Case. Je nach Anzahl der Use Cases kann die entstehende Unterlage fünf oder zweihundert A4-Seiten stark sein.

Beispiel Jingleshop: Use-Case-Tabelle

Merkmal	Inhalt
Nr.	UC-8
Name	Anmelden
Kurzbeschreibung und Zielsetzung	Der Kunde meldet sich am System an, um erweiterte Funktionen bearbeiten zu können (z. B. eigene Adressangaben pflegen, Zahlungsarten pflegen, Kennwort ändern)
Priorität	1 - kritisch
Akteure	Kunde
Vorbedingungen	Der Kunde muss als Benutzer im System bekannt sein
Ablauf <ul style="list-style-type: none">StartNormaler AblaufVariantenAusnahmenEPK	Start: Der Use Case beginnt, wenn der Kunde Benutzername und Kennwort in die erforderlichen Felder eingetragen und bestätigt hat. Normaler Ablauf: <ul style="list-style-type: none">Der Kunde wird im System angemeldetDer Benutzername wird an geeigneter Stelle angezeigtDie Rolle des Benutzers wechselt zu "angemeldeter Benutzer" Fehlerfälle: <ul style="list-style-type: none">Benutzer ist im System nicht bekannt: Anzeige einer Fehlermeldung und Ermöglichung einer erneuten EingabeKennwort ist falsch: Anzeige einer Fehlermeldung und Ermöglichung einer erneuten EingabeKennwort wird 3x falsch eingegeben: Sperren des Nutzers und Umleiten auf eine Hilfeseite
Anforderungen	21, 22, 23, 24, 25
Umsetzungshinweise	Das Kennwort muss nach einer kurzen Verzögerung verdeckt angezeigt werden (Sternchen oder Punkte)
Offene Punkte	Definition der zulässigen Benutzernamen: E-Mail-Adresse oder selbst gewählter Name

Tabelle 5: Beschreibung des Use Case „Anmeldung“ im Beispiel "Jingleshop".

Ausblick

Als nächstes müssen die Use Cases mit einem weiteren Detaillierungsgrad versehen werden: Klare, für den Entwickler verständliche Anforderungen. Der dritte Teil des Artikels „Requirements Engineering für Projektleiter“ beschreibt, wie Sie dabei Schritt für Schritt vorgehen.

Fachbeitrag

Anforderungen an Softwareprodukte klar definieren

Requirements Engineering für Projektleiter

Teil 3: Anforderungen für die Software-Entwicklung ableiten

Die Merkmale der neuen Software sind festgelegt, der Funktionsumfang ist in Form von Use Cases modelliert. Als nächstes geht es darum, die Use Cases weiter zu detaillieren und daraus klare, für die Entwickler verständliche Anforderungen abzuleiten. Wie Sie dazu vorgehen, erklären Bettina Zastrow und Elisabeth Wagner in diesem dritten und abschließenden Teil der Artikelfolge.

Die Formulierung realistischer, eindeutiger und vollständiger Anforderungen hilft dem Projektleiter, der eine Software in Auftrag gibt, tatsächlich das System zu bekommen, das er will. Der erste Teil dieser Serie zeigt, wie die Software und die Rahmenbedingungen aus der Perspektive der Nutzer definiert werden. Der zweite Teil beschreibt die Formulierung der Use Cases: Was will welcher Akteur mit dem System tun? In diesem dritten und letzten Teil der Artikelfolge geht es darum, das Ziel der Softwareentwicklung aus Sicht der Entwickler zu beschreiben.

Sind die Use Cases fertig modelliert und zu Features zusammengefasst, lassen sich daraus die Anforderungen ableiten. Jedem Use Case werden dabei üblicherweise eine oder mehrere Anforderungen zugeordnet.

Mit Anforderungen sind hier standardisierte Sätze gemeint, die Ansprüche und Wünsche der Benutzer an die Funktionalitäten einer Software beschreiben. Auch Schnittstellen zu angrenzenden Systemen und eigenständige Systemprozesse sind zu berücksichtigen. Darüber hinaus können Vorgaben von außen existieren, z.B. aus firmeninternen Regelwerken, die ebenfalls als Anforderungen formuliert werden.

Definition: funktionale und nichtfunktionale Anforderungen

Ein Use Case beschreibt Funktionen, die ein Akteur mithilfe des Systems ausführen möchte. Daraus abgeleitete Anforderungen werden als "*funktionale* Anforderungen" bezeichnet. Sie beschreiben jeweils einen der folgenden drei Fälle:

- eine Interaktion zwischen dem Benutzer und dem System,
- eine Interaktion zwischen dem System und einem anderen System oder
- eine selbstständige Systemaktion aufgrund eines zeitlichen oder ereignisgesteuerten Auslösers.

Autor



Bettina Zastrow

Informatikerin; seit 2011
Geschäftsführerin Zastrow
information development

GmbH, Autorin von Software-
Handbüchern und Online-Hilfetexten

Kontakt: info@bettina-zastrow.de



Elisabeth Wagner

IPMA Level D, Kommuni-
kationsberaterin, Journa-
listin, seit 20 Jahren in der
Unternehmenskommunikation tätig

Kontakt:

wagner@projektmarketing-partner.de

Mehr Informationen unter:

› projektmagazin.de/autoren

Darüber hinaus ergeben sich weitere Anforderungen, z.B. rechtlicher, kultureller, sicherheitstechnischer Art, usw., die für die Anwendungsentwicklung bekannt sein müssen. Diese Anforderungen außerhalb der Use Cases werden als "*nichtfunktional*" bezeichnet (siehe Abschnitt "Nichtfunktionale Anforderungen berücksichtigen").

Funktionale Anforderungen ermitteln

Wünsche und Vorgaben der Stakeholder lassen sich am besten in Einzelinterviews ermitteln. Als Diskussionsgrundlage kann eine Liste der Use Cases dienen. Projektleiter und Stakeholder nehmen dabei die gleiche Perspektive auf das System ein und erarbeiten gemeinsam die Anforderungen aus der Nutzerperspektive, z.B. indem sie die Workflows zu den jeweiligen Use Cases durchgehen: Was sind z.B. für den Use Case "Nutzer registriert sich am System" die Voraussetzungen, was passiert im Hintergrund und was ist das Ergebnis?

Die Schwierigkeit ist hier häufig, exotische Anforderungen einzudämmen und die maßgeblichen Aspekte im Auge zu behalten, wie z.B. bei obigem Beispiel "Systemregistrierung" die Regeln für die Kennwortsicherheit.

Das Arbeitsergebnis ist eine Liste von Anforderungen aus dieser Nutzerrolle. Soweit möglich, sind die Anforderungen direkt im Format der SOPHIST[®] Satzschablone zu erfassen (siehe Abschnitt "Qualitätskriterien"). Andernfalls reichen auch komplette Sätze aus, welche die Rolle, die Aktion und ggf. die Bedingung für die Ausführung beschreiben. Das Augenmerk ist dabei auf die vollständige Erfassung aller Anforderungen zu richten, die sich aus dem jeweiligen Use Case ergeben. Geschulte SW-Entwickler können hier gut unterstützen. Prüfung, Konsolidierung und Ergänzungen erfolgen in einem späteren Schritt.

! Um die Nachvollziehbarkeit zu gewährleisten, ist es ratsam, jede Anforderung bei der Erfassung mit einer Quellenangabe zu versehen. Dies ist auch später nützlich bei der Vergabe von Prioritäten.

Ein typischer Stolperstein ist die Schwierigkeit, Anforderungen am Rande des Spektrums zu erkennen und einzubeziehen. Am unteren Ende sind dies die banal erscheinenden Anforderungen, wie z.B. die Druckfunktion oder die Onlinehilfe, die so selbstverständlich sind, dass vergessen wird, sie explizit zu erwähnen. Am oberen Ende steht die Bedienbarkeit, auch als "Usability" bezeichnet. Zur Usability gehört es, die Benutzeroberfläche nach den Gesetzen der visuellen Wahrnehmung aufzubauen und effiziente Funktionen zu gestalten, die mit wenigen Schritten ausführbar sind.

Als Königsdisziplin der Benutzerfreundlichkeit gilt der "Joy of Use". Dieser Fachbegriff bedeutet, dass die Bedienung nicht nur angenehm sein, sondern Freude bereiten soll. Soll die Software diese Bedingung erfüllen, ist frühestmöglich der Rat von speziell geschulten Usability-Experten einzuholen. Diese kommen aus unterschiedlichen Fachrichtungen und sind entweder im Bereich der Psychologie oder im Umfeld von Web- und Grafikdesignern zu finden.

Wenn mit allen Stakeholdern ein Gespräch geführt wurde, entsteht als **Ergebnis** dieses Schritts eine **Liste von Anforderungen**, die den einzelnen **Use Cases zugeordnet** sind.

Der nächste Schritt umfasst eine Präzisierung der vorliegenden Anforderungen. Sie werden mit Eigenschaften wie Vorbedingungen und einer Priorität versehen. Die Erfassung kann in einer tabellarischen Form erfolgen.

Merkmale funktionaler Anforderungen

Tabelle 1 zeigt die typischen Bestandteile einer Anforderungsdefinition. Die in der linken Spalte aufgeführten Merkmale – von der Zuweisung einer eindeutigen Kennung für jede Anforderung bis hin zu dem Datum der Aufnahme einer Anforderung – sind ein häufig eingesetzter Standard im professionellen Requirements Engineering.

Element	Beschreibung
Nr.	Eindeutige Kennung für die Identifizierung der Anforderung
Name	Kurzname der Anforderung
Beschreibung und Zielsetzung	Langtext der Anforderung
Priorität	Rangfolge für die Umsetzung
Vorbedingungen	Bedingung, unter der die beschriebene Aktion ausgeführt werden soll
Fallbeispiel	Beispiel zur Veranschaulichung, wie diese Anforderung in den Gesamtkontext einzubetten ist
Offene Punkte	Sachverhalte, die noch zu klären sind
Quelle	Namenskürzel des Anforderers
Hinzugefügt am	Datum der Erstellung

Tabelle 1: Elemente einer funktionalen Anforderung.

Qualitätskriterien

Um bei Anforderungen eine hohe Qualität zu erreichen, kann auf den internationalen Standard ISO/IEC/IEEE 29148:2011 zurückgegriffen werden.

Dieser definiert für Anforderungen folgende Qualitätskriterien:

- **eindeutig:** frei von unklaren Formulierungen und Fehlern
- **vollständig:** enthält alle relevanten Sachverhalte
- **bewertet:** nach Priorität und Verbindlichkeit geordnet
- **konsistent:** in sich widerspruchsfrei sowie ohne Konflikte zu anderen Anforderungen
- **prüfbar:** enthält konkrete, messbare Daten und Eigenschaften, testbar
- **verfolgbar:** nachvollziehbar, wer die Anforderung eingebracht hat

Die ersten drei Kriterien lassen sich durch Verwendung der SOPHIST® Satzschablone erfüllen (siehe "**Die SOPHIST-Satzschablone: Funktionale Anforderungen präzise formulieren**", Projekt Magazin, Ausgabe 15/2014). Diese stellt eine Methodik zur Verfügung, um Anforderungen in natürlicher Sprache zu formulieren. Ein Satz wird strukturiert in sechs Schritten aufgebaut. Dabei werden das System, die rechtliche Verbindlichkeit, das Prozesswort, die Prozessart, das Objekt und die Bedingungen berücksichtigt.

Die Anforderung "Es soll Cross-Selling-Optionen geben" erfüllt z.B. die oben genannten Qualitätskriterien nur unzureichend (siehe Tabelle 2). Eine präzisere Formulierung, in der alle enthaltenen Fehler korrigiert sind, zeigt Tabelle 3.

Kriterium	Bewertung der Anforderung "Es soll Cross-Selling-Optionen geben"
bewertet	Es liegt keine Angabe zur Verbindlichkeit und zur Priorisierung der Anforderung vor ("soll")
eindeutig	Der Begriff "Cross-Selling" ist nicht näher definiert
konsistent	Ja
prüfbar	Es fehlt die Angabe, wem wo und wann Cross-Selling-Optionen angeboten werden sollen
verfolgbar	Es liegt keine Angabe zur Urheberschaft und zum Erstellungsdatum der Anforderung vor
vollständig	Es fehlt die Angabe, welche Produkte als Cross-Selling-Produkte klassifiziert werden sollen

Tabelle 2: Bewertung einer Anforderung nach der Norm ISO/IEC/IEEE 29148:2011.

Merkmal	Inhalt
Nr.	Req-11
Name	Cross-Selling-Option: "Andere Kunden haben auch angesehen"
Beschreibung und Zielsetzung	Der Kunde soll die Möglichkeit haben, ausgehend von einem Produkt weitere Produkte nach dem Kriterium "Andere Kunden haben auch angesehen" aufzurufen.
Priorität	4 - niedrig
Vorbedingungen	Benutzer befindet sich auf der Detailansicht eines Produktes
Fallbeispiel	Keine
Offene Punkte	Keine
Quelle	SMI
Hinzugefügt am	19.05.2015

Tabelle 3: Korrekte Anforderungsbeschreibung über Standardmerkmale.

Anforderungen, die mehrere Aspekte beinhalten, sollten in mehrere einzelne Anforderungen aufgeteilt werden. Das vereinfacht die Aufgaben der Entwickler wie auch der Tester. Erkennbar sind solche Anforderungen an dem Wort "und", z.B.: "Der Kunde muss bei der Registrierung AGB *und* Datenschutzerklärung bestätigen." Hier ist es sinnvoll, zwei separate Anforderungen zu formulieren. Dadurch können verschiedene Teammitglieder mit der Umsetzung betraut werden und diese können die Aufgaben einzeln abarbeiten und als erledigt markieren. Dies erleichtert die Kontrolle des Projektfortschritts.

Beispiel Jingleshop: Funktionale Anforderungen

Zur Verdeutlichung wird am Beispiel "Jingleshop" gezeigt, wie der Use Case "Anmelden" in einzelne technische Anforderungen aufgeschlüsselt wird, bis Missverständnisse nahezu ausgeschlossen sind. Für jeden Teilaspekt des Use Case wird dabei eine eigene Anforderung formuliert. Folgende fünf Anforderungen wurden ermittelt:

- Anmeldung mit Benutzername und Kennwort
- Eigenschaften Feld Benutzername: Länge
- Eigenschaften Feld Kennwort: Länge
- Eigenschaften Feld Kennwort: Darstellung
- Eigenschaften Feld Kennwort: Zwischenablage

Tabellen 4 bis 8 zeigen die vollständigen Anforderungsbeschreibungen.

Merkmal	Inhalt
Nr.	Req-21
Name	Anmeldung mit Benutzername und Kennwort
Beschreibung und Zielsetzung	Der Kunde muss jederzeit in der Lage sein, sich mit Benutzername und Kennwort am System anzumelden.
Priorität	1 - kritisch
Vorbedingungen	Anmeldung ist noch nicht erfolgt. Benutzer ist registriert
Fallbeispiel	Der Kunde befindet sich in einem Bestellprozess und möchte sich anmelden.
Offene Punkte	Keine
Quelle	SMI
Hinzugefügt am	19.05.2015

Tabelle 4: Beispiel für funktionale Anforderung "Anmeldung".

Merkmal	Inhalt
Nr.	Req-22
Name	Eigenschaften Feld Benutzername: Länge
Beschreibung und Zielsetzung	Das System muss fähig sein, einen Benutzernamen mit der Länge von 130 Zeichen zu verarbeiten
Priorität	3 - niedrig
Vorbedingungen	Keine
Fallbeispiel	Keine
Offene Punkte	Keine
Quelle	SMI
Hinzugefügt am	19.05.2015

Tabelle 5: Beispiel für funktionale Anforderung "Länge des Felds Benutzername".

Merkmal	Inhalt
Nr.	Req-23
Name	Eigenschaften Feld Kennwort: Länge
Beschreibung und Zielsetzung	Das System muss fähig sein, ein Kennwort mit der Länge von 20 Zeichen zu verarbeiten
Priorität	3 - niedrig
Vorbedingungen	Keine
Fallbeispiel	Keine
Offene Punkte	Keine
Quelle	SMI
Hinzugefügt am	19.05.2015

Tabelle 6: Beispiel für funktionale Anforderung "Länge des Feldes Kennwort".

Merkmal	Inhalt
Nr.	Req-24
Name	Eigenschaften Feld Kennwort: Darstellung
Beschreibung und Zielsetzung	Das System muss fähig sein, ein Kennwort nach einer Verzögerung von 1 Sekunde verdeckt anzuzeigen
Priorität	2 - hoch
Vorbedingungen	Keine
Fallbeispiel	Keine
Offene Punkte	Keine
Quelle	SMI
Hinzugefügt am	19.05.2015

Tabelle 7: Beispiel für funktionale Anforderung "Darstellung des Feldes Kennwort".

Merkmal	Inhalt
Nr.	Req-25
Name	Eigenschaften Feld Kennwort: Zwischenablage
Beschreibung und Zielsetzung	Das System muss verhindern, dass der Benutzer den Inhalt des Feldes "Kennwort" in die Zwischenablage kopiert
Priorität	2 - hoch
Vorbedingungen	Es wurde eine Eingabe im Feld Kennwort getätigt. Benutzer kopiert den Inhalt in die Zwischenablage per Maus oder Tastatur
Fallbeispiel	Keine
Offene Punkte	Keine
Quelle	FRT
Hinzugefügt am	21.06.2015

Tabelle 8: Beispiel für funktionale Anforderung "Kopierschutz für das Feld Kennwort".

Als **Ergebnis** entsteht in diesem Schritt eine Zusammenstellung **aller funktionalen Anforderungen**. Für eine leichtere Handhabung können die Zeilen und Spalten der Einzeltabellen getauscht und in einer Gesamttabelle zusammengeführt werden. Der Umfang des Projektdokuments hängt von der Größe des Projekts und der Anzahl der Anforderungen ab. Zugleich können die Zusammenhänge zwischen den Use Cases und den Anforderungen hergestellt werden. Dies geschieht entweder mit der Platzierung der Anforderungen bei den zugehörigen Use Cases oder durch Querverweise.

! Es hat sich als praktisch erwiesen, die einmal vergebene Nummerierung für Anforderungen beizubehalten, d.h. einmal vergebene Nummern nicht mehr zu ändern. Das erleichtert das Erkennen von neuen Anforderungen. Weiterhin sind anhand dieser Nummern eindeutige Querbezüge zwischen den Anforderungen untereinander sowie zwischen den Anforderungen und Use Cases möglich. Da der Text einer Anforderung im Lauf des Projekts ggf. verändert wird, bleibt der Bezug über die Nummerierung, die eine eindeutige Kennung darstellt, erhalten. Im Verlauf des Projekts neu hinzugekommene Anforderungen können fortlaufend weiternummeriert werden.

Nichtfunktionale Anforderungen berücksichtigen

In den nichtfunktionalen Anforderungen wird vorgegeben, welche Eigenschaften die Funktionen der Software haben sollen, um die gegebenen Randbedingungen zu erfüllen. Hierbei handelt es sich um Eigenschaften wie Benutzbarkeit, Systemstabilität, Leistung, Betreibbarkeit usw. Betrachtet wird das Umfeld der Software, wie die Daten, die Benutzer, die Systemumgebung sowie begleitende Prozesse wie Rollout, Dokumentation und Schulung.

Die nichtfunktionalen Anforderungen lassen sich aus dem Systemkontext und aus der Norm für Softwarequalität nach ISO/IEC 2500 ableiten. Eine mögliche Gliederung ist wie folgt:

- **Systemtechnische Anforderungen:** Anforderungen aus bestehender oder genutzter Hardware, Software und Infrastruktur (z. B. Netzwerk)
- Anforderungen der **Erfüllung bestimmter Vorschriften:** Anforderungen aus externen und internen Regelwerken
- **Softwarequalitäts-Anforderungen:** Anforderungen, die die Performanz, Zuverlässigkeit, Usability, Stabilität und Angemessenheit betreffen
- **Datenanforderungen:** Anforderungen technischer Natur zur Datenspeicherung und -verarbeitung
- **Archivierungsanforderungen:** Anforderungen, die aus Vorgaben zur Archivierung hervorgehen
- **Sicherheitsanforderungen:** Anforderungen aus Datenschutz und Datensicherheit, sowie Arbeitssicherheit
- **Betriebsanforderungen:** Anforderungen für die Betriebsführbarkeit, beispielsweise Vorgaben zur Datensicherung und -wiederherstellung
- **Lizenzierungsanforderungen:** Anforderungen, die darauf abzielen, das Softwareprodukt lizenzierbar zu machen
- **Portierungs- und Migrationsanforderungen:** Anforderungen zur Plattformunabhängigkeit
- **Rollout- und Schulungsanforderungen:** Anforderungen, die die Einführung der Software in den Regelbetrieb ermöglichen oder erleichtern

- **Dokumentationsanforderungen:** Anforderungen an Entwicklungs-, Benutzer-, Administrations- und Betriebsdokumentation
- **Sonstige Anforderungen:** branchen- oder unternehmensspezifische Anforderungen, die nicht von den oben genannten Themengebieten abgedeckt wurden

Diese Themengebiete sind für alle Softwareprojekte relevant und erfordern tiefer gehende Kenntnisse der IT, der Branche und der anwendbaren Regelwerke. In größeren Unternehmen kann in der Regel auf existierende Vorgaben zurückgegriffen werden. Falls nicht, können Sie die nichtfunktionalen Anforderungen auf gleiche Weise wie die funktionalen Anforderungen mittels eines Interviews erheben. Mit Vorlagen für die Anforderungsdefinition, sog. "Requirements Specification Templates", die auf dem Markt erhältlich sind, ist der Projektleiter gut gerüstet, um eine Vollständigkeit der nichtfunktionalen Anforderungen herzustellen.

In der Word-Datei, die Sie zusammen mit dem Artikel herunterladen können, finden Sie einen Gesprächsleitfaden sowie eine Liste der Ansprechpartner (Rollen) für die einzelnen Interviews sowie Empfehlungen, zu welchen Themen diese jeweils Auskunft geben können. Die ebenfalls beigefügte Excel-Datei dient zur Dokumentation der erhobenen Anforderungen und enthält zu jeder der oben genannten Themenkategorien ein exemplarisches Beispiel. Die Beispiele können Sie genau so oder mit geringfügigen Änderungen für jedes beliebige Softwareprojekt übernehmen. Überdies stellen die Beispiele eine Schablone dar, um mit den genannten Ansprechpartnern die relevanten nichtfunktionalen Anforderungen für jeden der genannten Bereiche zu erarbeiten und zu formulieren. Die Systematik für die Dokumentation gleicht jener der funktionalen Anforderungen.

Typische Stolpersteine bei diesem Schritt sind, Anforderungen festzulegen, die die Zukunft betreffen, z.B. "Unterstützung aller Betriebssysteme ab Windows 10.x aufwärts". Hier hilft eine einfache Frage weiter: "Ist die Anforderung testbar?" Die Antwort in diesem Fall würde "nein" lauten, da zum Zeitpunkt der Publikation dieses Artikels noch kein Nachfolger von Windows 10 verfügbar ist.

Als **Ergebnis** entsteht in diesem Schritt eine Zusammenstellung **aller nichtfunktionalen Anforderungen**. Diese können in das bestehende Projektdokument unter einer separaten Überschrift nach den funktionalen Anforderungen eingefügt werden.

Das dokumentierte und genehmigte **Ergebnis aller Schritte** dieser Artikelserie **entspricht dem Fachkonzept**. Um formale und organisatorische Kriterien ergänzt, entsteht daraus ein Lastenheft.

Wurde im Unternehmen bereits ein Softwareprojekt durchgeführt, ist es von großem Nutzen, die nichtfunktionalen Anforderungen zentral abzulegen und für jedes weitere Softwareprojekt wiederzuverwenden. Ist das noch nicht geschehen, kann der betroffene Projektleiter das beim Leiter der IT-Abteilung anregen (siehe "**Nichtfunktionale Anforderungen strukturiert erfassen und wieder verwenden**", Projekt Magazin, Ausgabe 8/2007)

Es zahlt sich aus, den nichtfunktionalen Anforderungen einen ebenso hohen Stellenwert einzuräumen wie den funktionalen Anforderungen. Erfüllt das fertig entwickelte Produkt z.B. die rechtlichen Vorgaben nicht, kann das Produkt im schlimmsten Fall nicht auf den Markt gebracht werden. Bleiben bestimmte Vorgaben unberücksichtigt, z.B. zur Architektur, zu möglichen Clients, zu Antwortzeiten und zur Bedienbarkeit, kann dies zur Folge haben,

dass das Produkt nur unter hohem Aufwand eingeführt, betrieben, lokalisiert, portiert, migriert oder weiterentwickelt werden kann, oder das Produkt nicht bedienerfreundlich ist und der finanzielle Erfolg ausbleibt.

Fazit

Dieser dreiteilige Beitrag zeigt, wie auch Projektleiter, die keine Ausbildung oder Erfahrung im professionellen Requirements Engineering haben, mithilfe eines systematischen Prozesses zu eindeutigen und vollständigen Anforderungen für eine neue Software gelangen. Der Prozess folgt zwei grundsätzlichen Vorgehensweisen: Zuerst verläuft die Denkweise vom Allgemeinen zum Konkreten. Damit wird gewährleistet, dass das große Bild stimmt, bevor man – sorgfältig und umfassend – die Details beschreibt. Danach verläuft der Prozess von der Nutzerorientierung zur IT-Orientierung, von der Anwendersprache zur IT-Notation. So gelingt die Kommunikation zwischen Technikern und Nichttechnikern, obwohl sie grundsätzlich unterschiedliche Sprachen sprechen.

Ausblick

Professionelles Requirements Engineering hat neben der Formulierung von Anforderungen viele weitere Facetten. Zum Beispiel gibt es professionelle Methoden, um Anforderungen auf Konsistenz zu prüfen und Widersprüche aufzulösen oder nach Wichtigkeit und Zeit zu priorisieren. Darüber hinaus ist es notwendig, auf Ereignisse während der Umsetzungsphase zu reagieren, die zu einer Änderung, Ergänzung oder Umpriorisierung von Anforderungen führen. Ein weiterer Artikel zu diesen Themen ist geplant.

Methode

Wie Sie die natürliche Sprache bändigen

Die SOPHIST-Satzschablone: Funktionale Anforderungen präzise formulieren

Einen entscheidenden Grundstein für den Erfolg eines Entwicklungsprojekts stellen richtig dokumentierte Anforderungen dar: Sie sorgen dafür, dass alle Projektbeteiligten das gleiche Verständnis von den Kundenwünschen haben, sodass der Kunde am Ende genau das bekommt, für das er bezahlt hat.

In Software-Entwicklungsprojekten werden funktionale Anforderungen an zu erstellende Systeme auf unterschiedlichste Art und Weise dokumentiert: Sie können z.B. in Form von konzeptuellen Modellen wie UseCase- oder Aktivitätsdiagrammen festgehalten werden. Diese haben den Vorteil, dass Anforderungen aufgrund des hohen Formalitätsgrades der Modellierungssprache sehr kompakt und eindeutig modelliert bzw. dokumentiert werden können. Voraussetzung ist allerdings, dass die Projektbeteiligten die verwendete Modellierungssprache beherrschen (s. Pohl, Rupp 2010, S. 46).

Die in der Praxis am häufigsten verwendete Dokumentationsform aber ist die natürliche Sprache. Sie hat gegenüber den konzeptuellen Modellen zwei entscheidende Vorteile (s. Pohl, Rupp 2010, S. 45):

1. Sie wird von allen Projektbeteiligten beherrscht und muss nicht erst erlernt werden.
2. Sie ist vielseitig einsetzbar, wodurch jegliche Anforderung ausgedrückt werden kann.

Stolpersteine beim Dokumentieren von Anforderungen

Allerdings hat die natürliche Sprache als Dokumentationsform auch einen großen Nachteil, der nicht selten zum Scheitern eines Entwicklungsprojekts führt: Die natürliche Sprache überlässt den Projektbeteiligten einen großen Spielraum für Interpretationen.

Dieser Interpretationsspielraum birgt die große Gefahr, dass dokumentierte Anforderungen vom Projektteam missverstanden und folglich falsch umgesetzt werden. Die Konsequenz ist, dass nicht geplante Systemanpassungen notwendig werden und das Projekt folglich nicht innerhalb des geplanten Zeit- und Budgetrahmens umgesetzt werden kann.

Die natürliche Sprache "bändigen"

Genau diese Situation habe ich als IT-Berater leider des Öfteren erlebt: Die Anforderungen werden natürlichsprachig in ausschweifender Prosa formuliert, ohne dass dabei auf Eindeutigkeit, Vollständigkeit und Korrektheit geachtet wird. Dies führt zu Unklarheiten, sodass das Entwicklungsteam in den meisten Fällen die Anforderungen

Autor



Matthias Kulke

Dipl.-Wirtsch.-Inf. (DH),
Senior Consultant bei
CGI, zertifiz. Professional

Scrum Product Owner und Professional Requirements Engineer

Kontakt: matthias.kulke@cgi.com

Mehr Informationen unter:
projektmagazin.de/autoren

anders umsetzt, als der Kunde dies will, woraufhin dieser unzufrieden reagiert. In der Folge werden zahlreiche Diskussionen darüber geführt, wie die Anforderungen zu verstehen sind. Am Ende dieser aufreibenden Diskussionen sind in der Regel nicht nur Time, Scope und Budget des Projekts in Gefahr, sondern es ist zudem die Stimmung aller Beteiligten gedrückt und die Standpunkte sind verhärtet.

Letztendlich sind in vielen Fällen umfangreiche und aufwändige Systemanpassungen notwendig, die Projektverzögerungen und Mehrkosten verursachen. Um dem entgegenzuwirken, muss die natürliche Sprache als Dokumentationsform für Anforderungen "gebändigt" werden. Das heißt, es muss dafür gesorgt werden, dass die dokumentierten Anforderungen genau das aussagen, was der Kunde sich wünscht.

Anforderungen pragmatisch erstellen

Falls Sie ähnliche Probleme als Requirements Engineer, Entwickler oder Projektleiter bereits erlebt haben oder sich derzeit in einer solchen Situation befinden, kann Ihnen die SOPHIST-Satzschablone weiterhelfen, die in diesem Artikel vorgestellt wird. Dabei handelt es sich um eine leicht anwendbare und effektive Methode, mit der Sie in der Lage sind, die natürliche Sprache als Dokumentationsform richtig einzusetzen. Die Schablone ermöglicht es Ihnen, strukturierte Anforderungssätze in fünf Schritten eindeutig, vollständig und verständlich zu erstellen.

Bei dem Unternehmen CGI, einem kanadischen Anbieter für IT- und Geschäftsprozess-Dienstleistungen, für den ich tätig bin, wird diese Methode bereits seit Jahren erfolgreich zur Anforderungsdokumentation eingesetzt. Die Methode hat sich dort besonders in Entwicklungsprojekten bewährt, die auf einem schwergewichtigen Vorgehensmodell wie z.B. dem Wasserfallmodell basieren, und gilt seitdem als Standard für die Dokumentation von funktionalen Anforderungen.

Gründe für den Einsatz der SOPHIST-Satzschablone waren vor allem die einfache Handhabung, der effektive und effiziente Einsatz sowie die Möglichkeit, eine pragmatische und einheitliche Projektsprache zur Dokumentation von Anforderungen zu schaffen.

Ein Beispiel aus der Praxis

Ein IT-Dienstleistungsunternehmen verfügt über ein webbasiertes Reisekostensystem, über das die Reisekosten der Mitarbeiter erfasst, freigegeben und abgerechnet werden. Beim Erstellen eines Reisekostenberichts muss der Mitarbeiter die Belege über die entstandenen Reisekosten im Vorfeld einscannen und dem Reisekostenbericht als Anhang hinzufügen. Da die Mitarbeiter während einer Firmenreise nur sehr selten oder gar nicht die Möglichkeit haben, die Belege einzuscannen, werden die Reisekosten meistens mit einem Zeitverzug von mehreren Wochen erfasst. Dies führt dazu, dass die monatlichen Geschäftszahlen des Unternehmens verfälscht werden und nachträglich korrigiert bzw. angepasst werden müssen.

Um dem entgegenzuwirken, möchte das Unternehmen für das bestehende Reisekostensystem eine Smartphone-Applikation entwickeln: Mit dieser App können die Mitarbeiter ihre Reisekosten noch während der Firmenreise direkt erfassen, ohne dabei die Belege über die entstandenen Reisekosten einscannen zu müssen. Eine der zentralen und wichtigsten Funktionen der App ist, dass die Mitarbeiter ihre Belege mit dem Smartphone abfotografieren und dem Reisekostenbericht während der Erstellung als Anhang hinzufügen können.

Der erste Blick trügt

Im Zuge der Anforderungsanalyse wurde die Funktionalität genauer untersucht. Es wurde dabei bewusst festgelegt, dass an einen Reisekostenbericht maximal 20 abfotografierte Belege angehängt werden können. Anschließend wurde hierzu folgende Anforderung formuliert:

"Es ist erforderlich, dass 20 Dateianhänge hochgeladen werden können."

Die formulierte Anforderung mag auf den ersten Blick korrekt erscheinen, ist bei genauerem Hinsehen jedoch missverständlich formuliert. So ist in keiner Weise ersichtlich, dass

- der Mitarbeiter die Belege an den Reisekostenbericht anhängen können muss, während er diesen erstellt,
- es sich um Belege handelt, die über das Smartphone abfotografiert wurden und
- dass maximal 20 Belege an einen Reisekostenbericht angehängt werden können.

Durch die unklare Anforderungsformulierung ist es sehr wahrscheinlich, dass der Entwickler die Anforderung ganz anders interpretiert und umsetzt. Z.B. könnte der Entwickler die Anforderung so umsetzen, dass zwar Dateianhänge über die App hochgeladen, diese aber nicht mit dem Reisekostenbericht verknüpft werden können. Ebenso kann es passieren, dass der Entwickler die notwendige Verknüpfung mit dem Fotoalbum des Smartphones nicht berücksichtigt, sodass die abfotografierten Belege gar nicht als Dateianhang ausgewählt werden können.

Darüber hinaus wird bei der aktuellen Formulierung nicht ersichtlich, dass es sich um eine zentrale Anforderung der App handelt. Dies birgt die Gefahr, dass die Anforderung von der Projektleitung als weniger wichtig eingestuft wird und bei einem Budget-Engpass hinten herunterfällt. Auch auf die Qualitätssicherung wirkt sich unklares Formulieren negativ aus: Ähnlich wie bei den Entwicklern besteht auch bei den Testern die Gefahr, dass sie die Anforderung anders interpretieren oder sogar die falsche Umsetzung als korrekt ansehen.

Das Risiko ist also groß, dass die App am Ende nicht das kann, was sie können sollte und das Projekt somit zum Scheitern verurteilt ist. Im Folgenden erfahren Sie, wie Sie die skizzierte Anforderung richtig dokumentieren, um die geschilderten Stolperfallen zu vermeiden.

Der Baukasten der SOPHIST-Satzschablone

Bei dem hier beschriebenen Vorgehen handelt es sich um eine von der SOPHIST GmbH entwickelte Schablone bzw. Methode zur syntaktischen Dokumentation von Anforderungen (s. Rupp 2009, S.159 ff.). Diese wurde mit dem Ziel entwickelt, funktionale Anforderungen eindeutig, vollständig, verständlich und korrekt zu dokumentieren. Hierzu stellt die Methode einen Baukasten sowie einen Bauplan bereit, mit denen sich strukturierte Anforderungssätze erstellen lassen.

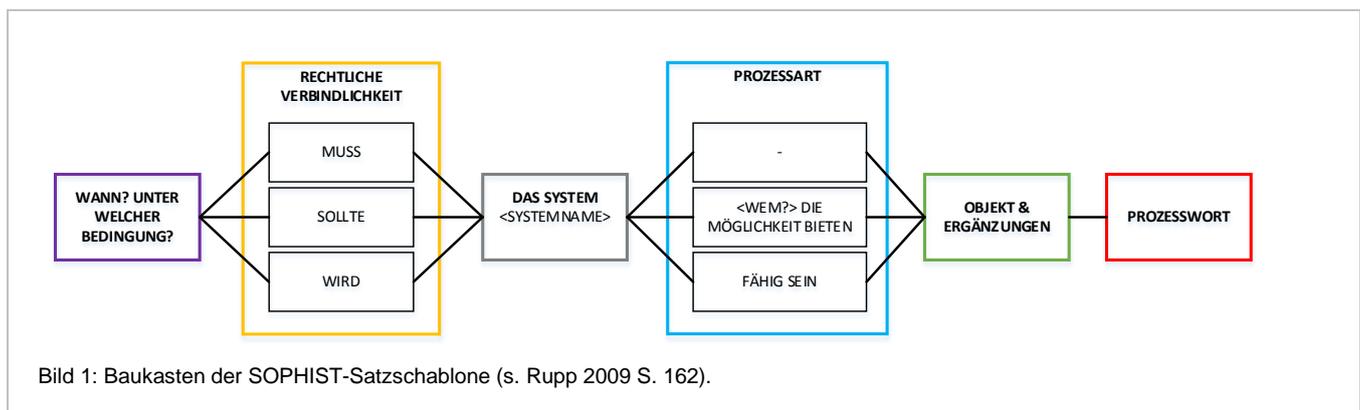
Die Methode basiert auf den Qualitätskriterien für gut dokumentierte Anforderungen, welche vom Institute of Electrical and Electronics Engineers (IEEE) erhoben und im Standard IEEE Std 830-1998 festgehalten wurden (vgl. IEEE Std 830-1998).

Sechs Bausteine zur präzisen Formulierung

Der Baukasten der Schablone besteht aus sechs Bausteinen, die jeweils eine bestimmte semantische Bedeutung innerhalb des Anforderungssatzes haben. Diese sind der Systemname, die rechtliche Verbindlichkeit, das Prozesswort, die Prozessart, das Objekt und dessen Ergänzungen sowie die Bedingung.

- Der Systemname identifiziert das System, für das die Anforderung erfasst wird.
- Die rechtliche Verbindlichkeit legt fest, wie wichtig die Anforderung ist.
- Das Prozesswort identifiziert und beschreibt die geforderte Systemfunktionalität.
- Die Prozessart bestimmt die Art der geforderten Systemfunktionalität.
- Das Objekt und dessen Ergänzungen beschreiben die Daten, die durch die geforderte Systemfunktionalität verarbeitet werden.
- Die Bedingung legt den Rahmen fest, in dem die geforderte Systemfunktionalität ausgeführt wird.

Die folgende Abbildung veranschaulicht den Aufbau eines Anforderungssatzes mit den sechs Bausteinen der SOPHIST-Satzschablone.



Die Anforderung aus dem Praxisbeispiel sieht gemäß der SOPHIST-Satzschablone demnach wie folgt aus:

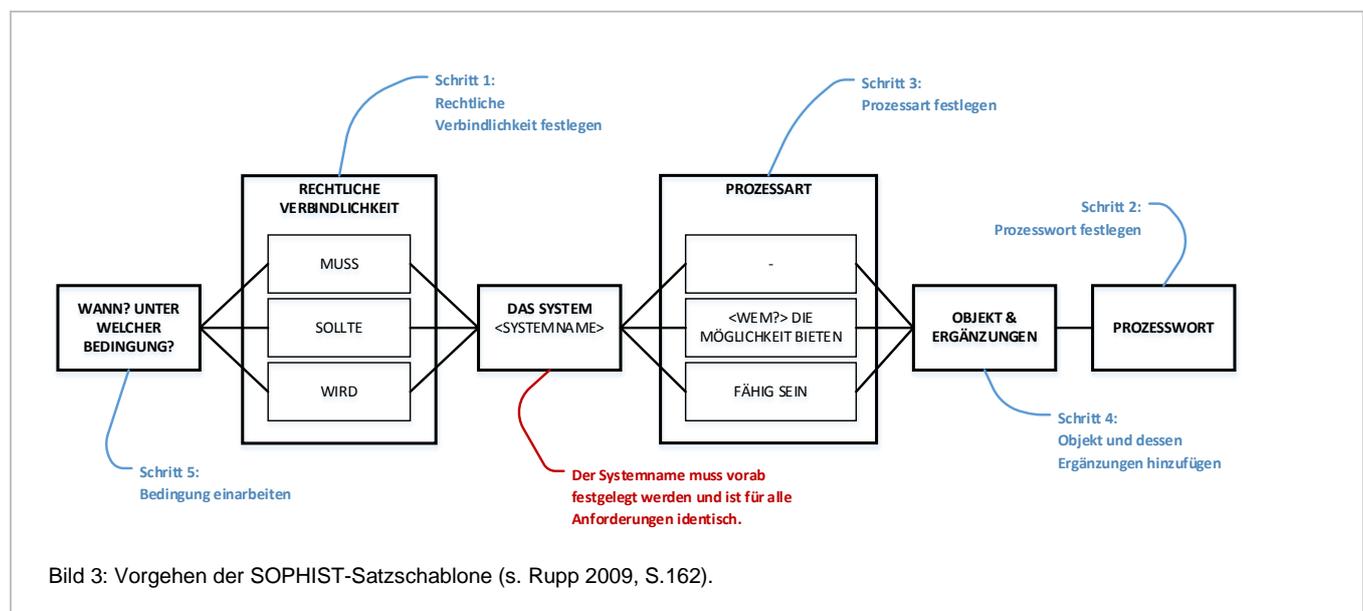


Wie Sie zu dieser syntaktischen Dokumentation der Anforderung kommen, erfahren Sie im nächsten Abschnitt.

Der Bauplan der SOPHIST-Satzschablone

Der Bauplan der Schablone erklärt, wie ein Anforderungssatz mit den sechs Bausteinen erstellt wird. Die Anleitung besteht aus fünf aufeinanderfolgenden Schritten, in denen die einzelnen Bausteine der Satzschablone bestimmt und zusammengesetzt werden. Diese fünf Schritte sind:

1. die rechtliche Verbindlichkeit festlegen
2. das Prozesswort festlegen
3. die Prozessart festlegen
4. das Objekt und dessen Ergänzungen hinzufügen
5. die Bedingung einarbeiten



Für den Systemnamen wird vorab ein geeignetes Subjekt festgelegt; er ist für alle Anforderungen des Systems identisch. Wenn kein sprechender Systemname gefunden wird, können Sie "das System" oder ähnliche Umschreibungen verwenden.

Für die Anforderung aus dem Praxisbeispiel wähle ich als Systemnamen "die App".

Schritt 1 – die rechtliche Verbindlichkeit festlegen

Als erstes wird die rechtliche Verbindlichkeit, also die vertragliche Verpflichtung der Anforderung festgelegt. Üblicherweise unterscheidet man dabei zwischen "Pflicht", "Wunsch" und "Absicht".

- **Pflicht:** Es handelt sich um eine zwingend zu erfüllende Anforderung. Eine solche Anforderung ist juristisch verbindlich (vertraglich einklagbar).
- **Wunsch:** Es handelt sich um eine Anforderung, von deren Umsetzung unter bestimmten Umständen abgesehen werden kann, z.B. bei hoher Komplexität und hohem Aufwand während der Implementierung. Eine solche Anforderung ist juristisch nicht verbindlich.
- **Absicht:** Es handelt sich um eine Anforderung, die die Berücksichtigung einer künftigen Entwicklung fordert. Künftige Entwicklungen sind z.B. kommende Standards, Richtlinien oder Erweiterungen an das System. So kann z.B. gefordert werden, dass die App zur Erfassung der Reisekosten künftig in der Lage sein muss, den erstellten Reisekostenbericht via Email zu versenden. Diese Funktionalität wird im aktuellen Release nicht umgesetzt, muss aber in der Umsetzung vorbereitend mit berücksichtigt werden, sodass sie in einem der nächsten Releases realisiert werden kann. Eine solche Anforderung ist juristisch verbindlich.

Neben Klarheit über die vertraglichen Verpflichtung schafft das Festlegen der rechtlichen Verbindlichkeit einen entscheidenden Vorteil: Bei Ressourcen-, Budget- und terminlichen Engpässen hilft es bei der Entscheidung darüber welche Anforderungen unverzichtbar sind und welche erst in einem späteren Release umgesetzt werden können.

Um die rechtliche Verbindlichkeit auszudrücken, können die Modalverben "muss", "sollte" und "wird" verwendet werden. Dabei steht das Modalverb

- "muss" für eine Pflicht-Anforderung,
- "sollte" für einen Wunsch und
- "wird" für eine Absicht.

Die rechtliche Verbindlichkeit muss in jedem Fall mit dem Kunden abgestimmt und dokumentiert werden, sodass alle Beteiligten das gleiche Verständnis von der Bedeutung der Verbindlichkeit bekommen.

Da es sich bei der Anforderung aus unserem Beispiel um eine der zentralen und wichtigsten Anforderungen handelt, wird "muss" als rechtliche Verbindlichkeit verwendet. Das erste Fragment der Anforderung ist somit fertig:

"Die App muss ..."

Schritt 2 – das Prozesswort festlegen

Im zweiten und zugleich wichtigsten Schritt bestimmen Sie das Prozesswort. Das Prozesswort drückt die geforderte Systemfunktionalität aus und bildet damit den Kern der Anforderung. Um die Systemfunktionalität (einen bestimmten Prozess oder Vorgang) auszudrücken, werden ausschließlich Vollverben als Prozesswörter verwendet. Vollverben sind Verben, die alleine das Prädikat eines Satzes bilden, z.B. "speichern".



Bild 4: Varianten für die rechtliche Verbindlichkeit.

Das Verwenden von Vollverben hilft, schwammige Formulierungen zu Funktionalitäten zu vermeiden, welche die Anforderung mehrdeutig, unvollständig sowie inkonsistent machen würden und dadurch dem Leser der Anforderung Spielraum für Interpretationen lässt.

Schwammige oder mehrdeutige Formulierungen erhalten Sie meist dann, wenn Sie Substantivierungen oder Funktionsverbgefüge verwenden. Substantivierungen bedeutet in diesem Zusammenhang meist, dass Verben, **die Prozesse beschreiben**, in Substantive umgewandelt werden, z.B. "Bestellung" oder "Archivierung". Bei Funktionsverbgefügen handelt es sich dagegen um Kombinationen aus sogenannten **inhaltsarmen Verben** (machen, können, haben, sein, stellen ...) und sinngebenden Substantiven, z.B. "etwas zur Verfügung stellen".

Sowohl bei Substantivierungen als auch bei Funktionsverbgefügen entsteht der Effekt, dass Anforderungen oder gar ganze Prozesse getilgt – d.h. verschleiert – werden. So verbergen sich z.B. hinter dem Begriff "Bestellung" Teilprozesse wie z.B. das Zusammenstellen des Warenkorbs und die Berechnung des Gesamtpreises mit wiederum eigenen Anforderungen. Die Formulierung "etwas zur Verfügung stellen" kann die Prozesse "Anzeigen", "Drucken" oder "Versenden" beinhalten, an die ebenfalls eigene Anforderungen gestellt werden. Diese Formulierungen machen also nicht eindeutig klar, was die Kernfunktionalität der Anforderung ist.

Daher sollten Sie bei der Bestimmung des Prozessworts unbedingt darauf zu achten, aussagekräftige Vollverben zu verwenden. Dies sind solche Vollverben, die einfache, elementare Prozesse darstellen, z.B. "drucken", "anzeigen", "speichern" oder in unserem Fall "anhängen". Oftmals lässt es sich nicht vermeiden, Vollverben zu verwenden, die komplexere Prozesse widerspiegeln wie z.B. "archivieren". In diesen Fällen ist es ratsam, das Vollverb in einem Glossar eindeutig zu beschreiben und damit abzugrenzen.

Nachdem das Prozesswort festgelegt ist, kann die Anforderung um den Kern erweitert werden: die Systemfunktionalität. Unser Praxisbeispiel sieht nun folgendermaßen aus:

"Die App muss etwas anhängen."

Schritt 3 – die Prozessart festlegen

Der dritte Schritt beim Erstellen einer gut dokumentierten Anforderung ist das Festlegen der Prozessart. Hierbei wird die Art der geforderten Systemfunktionalität bestimmt, die in drei Arten klassifiziert werden kann: "selbständige Systemaktivität", "Benutzerinteraktion" und "Schnittstellenanforderung".

- **Selbständige Systemaktivität** besagt, dass das System die Funktionalität selbständig, also ohne Benutzeraktion und unabhängig von Nachbarsystemen durchführt.
- Bei einer **Benutzerinteraktion** stellt das System dem Benutzer eine Funktionalität zur Verfügung. Bei dieser Funktionalität findet eine Interaktion zwischen dem System und dem Benutzer des Systems statt.
- Bei einer **Schnittstellenanforderung** führt das System eine Funktionalität nicht selbständig, sondern in Abhängigkeit zu einem Nachbarsystem aus. Im Falle der App könnte eine solche Funktionalität das Anzeigen einer Meldung über die Genehmigung der Reisekosten sein. Nachdem die Reisekosten vom verantwortlichen Manager genehmigt wurden, schickt das Reisekostensystem eine Nachricht an das Smartphone des

Mitarbeiters. Die App empfängt diese Nachricht, verarbeitet sie und blendet anschließend eine entsprechende Meldung auf dem Display des Smartphones ein.

Die Art der Systemfunktionalität definieren Sie, indem Sie einen bestimmten Ausdruck vor das Prozesswort stellen:

- Bei einer **Benutzerinteraktion** geben Sie an dieser Stelle eine Antwort auf die Frage "Wem" bietet das System die Möglichkeit, etwas zu tun?". Sie setzen den Ausdruck "<wem?> die Möglichkeit bieten" vor das Prozesswort. Für <wem?> setzen Sie eine bestimmte Benutzerrolle oder aber eine allgemeine Benutzerbezeichnung, wie z.B. "Benutzer" ein.
- Für eine **Schnittstellenanforderung** setzen Sie den Ausdruck "fähig sein" vor das Prozesswort.
- Wenn keiner der beiden Ausdrücke vor dem Prozesswort zu finden ist, handelt es sich um eine **selbständige Systemaktivität**.



Demnach bieten sich zum Charakterisieren der Funktionalität "anhängen" folgende Möglichkeiten:

- **Selbstständige Systemaktivität** – Die App muss etwas anhängen.
- **Benutzerinteraktion** – Die App muss dem Benutzer die Möglichkeit bieten, etwas anzuhängen.
- **Schnittstellenanforderung** – Die App muss fähig sein, etwas anzuhängen.

Indem Sie die Prozessart festlegen, geben Sie den Projektmitarbeitern einen Hinweis über die Komplexität und damit über den Umsetzungsaufwand der geforderten Funktionalität. So verbirgt sich hinter **Schnittstellenanforderung** meist eine hohe Komplexität, da Daten und Nachrichten zwischen zwei Systemen ausgetauscht werden und unter Umständen Prozesse des Nachbarsystems angepasst werden müssen. **Selbstständige Systemaktivitäten** sind in der Regel weit weniger komplex, da die Entwickler für diese keine externen Faktoren berücksichtigen müssen. **Benutzerinteraktionen** können einen ähnlich hohen Umsetzungsaufwand mit sich bringen wie Schnittstellenanforderung. Das liegt daran, dass bei Benutzerinteraktionen immer auch nicht-funktionale Aspekte wie z.B. Benutzbarkeit, Performance, Fehlertoleranz oder Barrierefreiheit berücksichtigt werden müssen.

Nachdem die Prozessart ermittelt wurde, in unserem Fall eine Benutzerinteraktion, ist das Grundgerüst der Anforderung fertig:

"Die App muss dem Benutzer die Möglichkeit bieten, etwas anzuhängen."

Schritt 4 – das Objekt & dessen Ergänzung hinzufügen

Als viertes fügen Sie der Anforderung das Objekt – das "etwas" – sowie dessen Ergänzungen hinzu, für das die Systemfunktionalität gefordert wird.

Bei dem Objekt handelt es sich um die Daten, die durch die Systemfunktionalität verarbeitet werden. Ein Objekt kann z.B. ein Systembenutzer mit bestimmten Benutzerdaten, ein Produkt mit bestimmten Produktdaten oder ein Bericht mit bestimmten Berichtsdaten sein. Im vorliegenden Praxisbeispiel sind das Objekt die Belege, die an den Reisekostenbericht angehängt werden.

Durch die Ergänzungen wird das Objekt näher beschrieben, um ein möglichst genaues Bild vom Objekt selbst, aber auch von der geforderten Systemfunktionalität zu erhalten. In unserem Fall werden die folgenden drei Ergänzungen verwendet:

- Es handelt sich um **abfotografierte** Belege.
- Die Belege werden an einen **Reisekostenbericht** angehängt.
- An einen Reisekostenbericht können maximal **20 Belege** angehängt werden.

Mit dem Hinzufügen des Objektes und dessen Ergänzungen ist die Anforderung fast vollständig:

"Die App muss dem Benutzer die Möglichkeit bieten, maximal 20 über das Smartphone abfotografierte Belege an einen Reisekostenbericht anzuhängen."

Schritt 5 – die Bedingung einarbeiten

Mit dem letzten Schritt hin zu einer gut dokumentierten Anforderung wird die Bedingung festgelegt, unter der die geforderte Systemfunktionalität ausgeführt wird. Die Bedingung präzisiert, wie auch schon das Objekt, die geforderte Systemfunktionalität, da diese für gewöhnlich nicht fortlaufend, sondern nur unter gewissen Umständen ausgeführt wird. Wir unterscheiden dabei zwischen logischen und zeitlichen Bedingungen:

- Logische Bedingungen werden mit der Konjunktion "falls" eingeleitet.
- Für zeitliche Bedingungen wählen Sie die Konjunktion "wenn"; alternativ können Sie auch Konjunktionen wie "bevor", "nachdem", "um" oder "während" verwenden.

In unserem Fall soll der Mitarbeiter Belege an einen Reisekostenbericht anhängen können, während er diesen erstellt. Die Bedingung wird als Nebensatz an den Anfang der Anforderung gestellt, wodurch die rechtliche Verbindlichkeit in der Satzstellung vor den Systemnamen wandert. Nachdem die Bedingung festgelegt wurde, ist die Anforderung aus dem Praxisbeispiel komplett:

"Während der Benutzer einen Reisekostenbericht erstellt, muss die App dem Benutzer die Möglichkeit bieten, maximal 20 über das Smartphone abfotografierte Belege an den Reisekostenbericht anzuhängen."

Ein klarer Unterschied...

Im Gegensatz zu der Anforderung aus dem Praxisbeispiel, die ohne die SOPHIST-Satzschablone formuliert wurde ("Es ist erforderlich, dass 20 Dateianhänge hochgeladen werden können."), geht unserer Anforderung deutlich hervor, dass

- der Mitarbeiter über das Smartphone abfotografierte Belege an einen Reisekostenbericht anhängen können muss,
- er dies tun können muss, während er den Reisekostenbericht erstellt,
- an einen Reisekostenbericht maximal 20 Belege angehängt werden können und
- dass es sich um eine der zentralen und wichtigsten Anforderungen der App handelt.

Grundregeln für die Anwendung

Bei der Anwendung der Methode gilt es gewisse Grundregeln zu beachten, um nicht aus einer gut dokumentierten Anforderung eine schwer lesbare Anforderung zu machen (s. Rupp 2009, S. 123 sowie Pohl, Rupp 2010 S. 56):

Regel 1: Anforderungen in vollständigen Sätzen beschreiben – "Stichpunkte sind tabu"

Jede Anforderung muss vollständig ausformuliert werden. Dies trägt dazu bei, dass jede Anforderung vollständig ist und keine Informationen verloren gehen oder hineininterpretiert werden können.

Regel 2: Anforderungen in kurzen Sätzen beschreiben – "Schachtelsatz ade"

Jede Anforderung muss kurz und prägnant beschrieben werden. Lange und komplizierte Schachtelsätze sollten Sie vermeiden. Wenn eine Anforderung sehr komplex ist, muss diese in kurze und prägnante Unteranforderungen unterteilt werden. Dies erhöht die Lesbarkeit und die Verständlichkeit der Anforderung.

Regel 3: Nur eine Anforderung pro Satz formulieren – "Weniger ist mehr"

Jede Anforderung darf nur aus einem Prozesswort – das heißt aus einer geforderten Systemfunktionalität – bestehen. Jede geforderte Systemfunktionalität muss einzeln beschrieben werden. So ist gewährleistet, dass jede Anforderung vollständig, eindeutig und korrekt dokumentiert ist.

Regel 4: Begriffe konsistent verwenden – "Gleiches schreiben, gleiches meinen"

Die Begrifflichkeiten in den Anforderungen (bspw. die Prozesswörter) müssen für alle Anforderungen einheitlich verwendet werden. Verwenden Sie wenn möglich keine Synonyme und Homonyme (Wörter, die für verschiedene Begriffe stehen) wie z.B. Kohle. Beides fördert die Eindeutigkeit der Anforderungen.

Zusammenfassung

Die Vielseitigkeit der natürlichen Sprache bringt die Gefahr mit sich, dass so dokumentierte Anforderungen unterschiedlich interpretiert und folglich missverstanden werden können. Das hier vorgestellte Praxisbeispiel hat aufgezeigt, welche Missverständnisse entstehen können, wenn Anforderungen mehrdeutig und unvollständig formuliert werden. Die Konsequenz sind Systeme, die nicht den Kundenerwartungen entsprechen.

Mit der SOPHIST-Satzschablone sorgen Sie dafür, dass in ihrem Projekt die funktionalen Anforderungen eindeutig, vollständig, verständlich sowie korrekt gemäß dem Standard IEEE Std 830-1998 dokumentiert werden und legen so einen Grundstein für den Erfolg Ihres Projekts.

Alternativ können funktionale Anforderungen auch mit Hilfe der aus der agilen Softwareentwicklung bekannten User Stories oder Snow Cards aus dem Volere-Template natürlich-sprachig dokumentiert werden. Im Vergleich zur SOPHIST-Satzschablone liegt der Fokus bei diesen beiden Methoden allerdings weniger auf der Bildung von vollständigen und aussagekräftigen Anforderungssätzen, sondern vielmehr auf der vollständigen und strukturierten Erfassung bestimmter Informationen rund um die Anforderung, wie z.B. Priorität, Abnahmekriterien oder Verweise auf Zusatzinformationen.

Literatur

- Institute of Electrical and Electronics Engineers: IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998). IEEE Computer Society, New York 1998.
- Pohl, Klaus; Rupp, Chris: Basiswissen Requirements Engineering – Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level, dpunkt.verlag, Heidelberg 2010.
- Rupp, Chris: **Requirements-Engineering und -Management – Professionelle, iterative Anforderungsanalyse für die Praxis**, Hanser Verlag, München 2009.

Agile Softwareentwicklung mit Scrum und User Stories

Das Vorgehensmodell Scrum erfreut sich durch wenige Regeln und klare Verantwortlichkeiten in der Softwareentwicklung zunehmender Beliebtheit. So erlaubt Scrum ein einfaches und flexibles Vorgehen – wie die Beteiligten den Entwicklungsprozess gestalten und organisieren, bleibt ihnen selbst überlassen. (Siehe auch "Agiles Projektmanagement. Scrum – eine Einführung", Projekt Magazin, Ausgabe 21/2009.)

Allerdings enthält Scrum keine konkreten Vorgaben, wie die Anforderungen an die zu entwickelnde Software erfasst und spezifiziert werden sollen. Die agile Softwareentwicklung verfolgt jedoch den Ansatz, die Anforderungen auf eine Art und Weise zu beschreiben, die es dem Entwicklungsteam ermöglicht, nach jedem Sprint nutzbare Software für den Kunden zu liefern.

Ein etablierter Weg, dies zu tun, ist die Verwendung von User Stories. Dabei handelt es sich um einfache Formulierungen der Anforderungen aus Sicht des späteren Anwenders. Dieser Artikel beschreibt, wie man die Anforderungen eines Scrum-Projekts mit Hilfe von User Stories beschreiben und verwalten kann.

Praxisbeispiel

Zum besseren Verständnis soll das Prinzip der User Stories innerhalb eines Scrum-Projekts am Beispiel eines webbasierten Job-Portals erläutert werden: Eine Gruppe von Investoren möchte ein Portal für die Stellenvermittlung im High Professional-Umfeld entwickeln, das Firmen und Recruitern Zugriff auf nachgewiesenen hoch qualifiziertes Fachpersonal ermöglichen soll. Im Gegenzug bietet das Portal Job-Suchenden die Möglichkeit, Kontakte zu renommierten und interessanten Arbeitgebern aufzubauen.

Ein Auszug aus der zentralen Anforderungsliste des Job-Portals, dem sog. Product Backlog, könnte wie folgt aussehen:

- Datenbankmodell entwerfen
- OR-Mapping-Framework (Mapping zwischen den Objekten des Systems und der unterliegenden relationalen Datenbank) entwickeln
- User Interface für die Job-Suche entwerfen
- Security Framework (Authentifizierung und Autorisierung von Nutzern) implementieren
- Suchmaschine entwickeln und integrieren

Autor



Ralf Wirdemann

Zertifizierter Scrum Master
und zertifizierter Scrum
Practitioner, Coach für
agiles Projektmanagement

Kontakt:

business@ralfwirdemann.de

Mehr Informationen unter:

projektmagazin.de/autoren

Das Problem: Scrum enthält keine Spezifikation der Anforderungen

Das Product Backlog in Scrum ist eine priorisierte Liste mit sämtlichen Anforderungen des Projekts. Die Einträge im Backlog werden als Backlog Items bezeichnet. Für jeden Eintrag liegt eine Schätzung des Aufwands vor; die Priorität des Eintrags bestimmt seine Position in der Liste. Während Scrum zwar mit dem Product Backlog sehr genau vorgibt, wie und womit Anforderungen verwaltet werden, trifft es keinerlei Aussagen darüber, wie die einzelnen Einträge des Backlogs formell genau auszusehen haben. Von "Die Software muss in Java programmiert werden" bis hin zu nicht-funktionalen Anforderungen, wie "Die Applikation muss skalierbar sein" ist alles erlaubt, was irgendwie getan werden muss.

Beide Beispiele sind notwendige Anforderungen, liefern aber keinen offensichtlichen Mehrwert für die Nutzer der Anwendung. Und darum geht es schließlich bei Scrum – der Entwicklung und regelmäßigen Auslieferung von nutzbarer Software für den Kunden. Entsprechend sollte das Product Backlog nur Einträge enthalten, die einen klar ersichtlichen Wert für die späteren Nutzer der Anwendung liefern. Hier hat der Product Owner die Aufgabe, ein Backlog zu erstellen, das diesen Kriterien genügt.

Ein Beispiel für Anforderungen ohne praktische Funktion für den Nutzer ist das oben genannte Backlog Item "OR-Mapping-Framework entwickeln". Die Entwicklung eines solchen OR-Mappers ist sinnvoll und notwendig, nur bietet der OR-Mapper für sich allein genommen keinerlei Mehrwert für die späteren Anwender des Portals. Entsprechend schwierig ist es für den Product Owner, der ja die Sicht des Kunden repräsentiert, dieses Backlog Item im Verhältnis zu den anderen Anforderungen im Sinne des Geschäftswerts zu priorisieren. Der Begriff Geschäftswert beschreibt in diesem Zusammenhang den Wert, den ein neues Feature für das Geschäftsfeld liefert, für das die Software entwickelt werden soll. Dieses kundenorientierte Anforderungsmanagement zielt folglich auf die regelmäßige Produktion und Lieferung von Features mit dem größtmöglichen Geschäftswert ab.

Die Lösung: User Stories im Kontext von Scrum

User Stories sind ein Werkzeug der agilen Softwareentwicklung, mit dem sich Software-Anforderungen kurz und prägnant aus Sicht des Nutzers beschreiben lassen. Viele Teams in der Softwareentwicklung setzen aktuell auf Scrum, so dass sich ein genauerer Blick auf die Verwendung von User Stories im Kontext von Scrum lohnt.

Ein wichtiges Prinzip von Scrum ist, dass die Auftraggeber regelmäßig in kurzen Abständen die aktuelle Version der Software zu sehen bekommen, um möglichst früh Kurskorrekturen vornehmen zu können. Deshalb muss der Product Owner die wichtigsten Anforderungen so früh wie möglich vom Team umsetzen lassen, um dem Kunden funktionsfähige Software präsentieren zu können. Damit diese relevanten Anforderungen auch hoch priorisiert werden, eignen sich User Stories, da sie für den Product Owner sehr viel greifbarer sind und sich gegeneinander abwägen lassen.

Im Kontext des Job-Portals könnte eine User Story z.B. das Einstellen oder die Suche nach Stellenangeboten sein. An diesem Beispiel wird außerdem offensichtlich, dass das Einstellen von Angeboten vor einer möglichen Suche stattfinden muss und folglich höher priorisiert werden sollte.

Im Gegensatz zu den vorher exemplarisch genannten Anforderungen zielt eine User Story immer auf einen konkreten Mehrwert für den Kunden ab: Statt der Entwicklung eines OR-Mappers beschreibt eine User Story immer

eine konkrete und für den Nutzer sinnvolle Funktion. Technische Details für die Programmierer werden bei dieser Sichtweise nicht ignoriert, sondern zu einer notwendigen Voraussetzung für die Entwicklung einer User Story. Der OR-Mapper liefert für sich genommen keinen Mehrwert, ist aber erforderlich für die Entwicklung eines sinnvollen Features für den Nutzer. Während im Backlog nur die User Story auftaucht, wird das Team den OR-Mapper im Rahmen der Entwicklung aufgreifen und als technisches Detail der Story implementieren.

Was ist eine User Story?

Eine User Story beschreibt eine Anforderung an das zu entwickelnde Softwaresystem. Jede User Story besteht aus drei Teilen: Einer Story Card, ihren Akzeptanzkriterien und die für ihre konkrete Umsetzung erforderliche Konversation zwischen Product Owner und Entwicklungsteam.

Story Card

Der schriftliche Teil einer User Story, die sogenannte Story Card, besteht häufig nur aus einem einzigen Satz, der den Kern der Anforderung auf den Punkt bringt. Beispiele für User Stories sind

- "Als Nutzer kann ich nach Stellenangeboten suchen" oder
- "Als Arbeitgeber kann ich Stellenanzeigen aufgeben".

Beide Beispiele bringen die jeweilige Anforderung auf den Punkt und machen in einem Satz klar, um wen es geht und was derjenige mit dem System machen kann. Der Product Owner kann dadurch den Geschäftswert der jeweiligen Anforderung bestimmen und sie entsprechend priorisieren.

Story Cards lassen sich am einfachsten mit Hilfe von Karteikarten oder großen Post-Its verwalten. Diese lassen sich schnell schreiben und für alle sichtbar an die Wand hängen. Sie sind weder zu groß, noch zu klein und bieten ausreichend Platz für den Titel der Story, vermeiden dabei aber das Schreiben von zu langen Texten. Physische Story Cards funktionieren allerdings nur für zentral arbeitende Teams; verteilt sich das Team über mehrere Orte, bietet sich der Einsatz elektronischer Tools, wie Scrumy, Mingle oder ScrumNinja (siehe Referenzen) an.

Akzeptanztests

Die Story Card ist nur einer von drei Teilen einer User Story. Über die Card hinaus besteht eine User Story aus ihren Akzeptanztests. Dabei handelt es sich um beispielhafte Beschreibungen der Funktionen einer User Story. Mit den Akzeptanztests legt der Product Owner fest, welche konkreten Funktionen die Story liefern muss. Einige Akzeptanztest-Beispiele für die Suche nach Stellenangeboten im Kontext des Jobportals:

- Eine Stichwortsuche nach "Java" liefert alle Angebote, die den Begriff "Java" entweder im Titel oder in der Beschreibung des Angebots enthalten.
- Eine Suche nach "Ort" liefert alle Angebote, die sich entweder direkt auf den gesuchten Ort oder auf einen Umkreis von 50 Kilometer außerhalb dieses Orts beziehen.

Akzeptanztests werden vom Product Owner formuliert und sind gleichermaßen Test und exemplarische Spezifikation. Für den Product Owner sind sie Abnahmekriterien, die er nach der Fertigstellung der Story überprüfen muss.

Für das Team sind Akzeptanztests während des Sprints eine Richtschnur, was genau zu entwickeln ist und wann die Story aus Geschäftssicht fertig ist.

Ein geeigneter Platz für das Festhalten der Akzeptanztests ist die Rückseite der jeweiligen Story Card. Elektronische Tools bieten i.d.R. gesonderte Eingabebereiche, um die Tests zu notieren.

Konversation zwischen Product Owner und Entwicklungsteam

Der dritte und wichtigste Teil einer User Story ist die Konversation zwischen dem Product Owner und dem Team. Wenn der schriftliche Teil einer User Story nur aus einem einzigen Satz und den zugehörigen Akzeptanztests besteht, wo bleiben dann die Details, d.h. die konkreten Ausprägungen der jeweiligen Story? Ganz einfach: Sie werden vom Product Owner dem Team erzählt und von ihm während des Sprints maßgeblich mitbestimmt. Anstatt die Details der Anforderungen im Vorfeld der Entwicklung sehr genau aufzuschreiben, verlagern User Stories die Anforderungsbeschreibung von der schriftlichen auf die verbale Kommunikation. Dadurch rückt der Product Owner während des Sprints nah an das Team heran, sieht die entwickelte Software sehr früh und erhält so viel bessere Steuerungsmöglichkeiten, die Software im Sinne des Kunden zu entwickeln.

Der typische Ablauf für die Planung und Umsetzung einer Story im Rahmen eines Sprints gestaltet sich wie folgt: Der Product Owner erklärt dem Team im Sprint Planning seine Vorstellungen von der Story "Als Job-Suchender will ich nach Stellenangeboten suchen" und das Team entscheidet die Story in den anstehenden Sprint zu nehmen. Anschließend startet das Team in den Sprint und beginnt mit der konkreten Umsetzung der Story; es arbeitet Aufgabe für Aufgabe (sog. Tasks) ab und entwickelt möglichst schnell eine erste vorzeigbare Version der Story.

Zusammen mit dem Product Owner geht das Team diese erste Version direkt vor dem Rechner durch und diskutiert die konkreten Details ihrer Weiterentwicklung. Die Story wird Schritt für Schritt und unter ständiger und sehr enger Einbeziehung des Product Owners weiter entwickelt. Durch diese Art der engen Zusammenarbeit wird von vornherein sichergestellt, dass das Team etwas entwickelt, was der Product Owner auch haben will.

Der Anforderungsworkshop

Um User Stories zu sammeln, sollte zu Beginn des Projekts ein Anforderungsworkshop durchgeführt werden. Der Product Owner initiiert den Workshop und lädt neben dem Entwicklungsteam alle Stakeholder, wie z.B. Marketing, Vertrieb, Investoren oder auch die Geschäftsführung ein. Jeder, der in irgendeiner Form etwas zu den Anforderungen der anstehenden Produktentwicklung sagen kann, sollte an diesem Workshop teilnehmen.

Der Workshop beginnt damit, dass der Product Owner den Teilnehmern seine zuvor in Zusammenarbeit mit dem Kunden ausgearbeitete Produktvision präsentiert. Die Produktvision ist ein kurzes und knappes Statement, das beschreibt, was entwickelt werden soll. Die Vision bringt den Kern des Produkts auf den Punkt und soll möglichst alle Stakeholder überzeugen und ins Boot holen. Die Produktvision des Job-Portals könnte z.B. folgender Satz sein: "Wir wollen das einzige Job-Portal für die Stellenvermittlung im High Professional-Umfeld in Europa werden."

Wurde die Vision vermittelt und von den Teilnehmern verstanden, ist der nächste Schritt im Workshop die Modellierung der Nutzerrollen. Eine Nutzerrolle ist eine Gruppe von Nutzern mit identischen Zielen in Bezug auf die

Nutzung des Systems. Die zentrale Frage bei der Modellierung der Nutzerrolle ist also, welche Bedürfnisse der Anwender der Software hat. Denken Sie beispielsweise an eine der diversen Job-Seiten im Internet, dann drängen sich einige offensichtliche Nutzerrollen unmittelbar auf: Job-Suchende, als Gruppe von Nutzern die über die Plattform neue Stellen finden wollen, oder Job-Anbieter, als Gruppe von Nutzern, die Stellenangebote ausschreiben wollen, um neue Mitarbeiter zu finden.

Mit Nutzerrollen die Anforderungen festlegen

Nutzerrollen, deren Beschreibung und Ziele sind ein wichtiger Ausgangspunkt für das Schreiben der ersten User Stories. Das funktioniert gut, indem sich das Anforderungsteam Rolle für Rolle vornimmt und anhand der jeweiligen Ziele die User Stories überlegt, die für die Zielerreichung notwendig sind. Das Anforderungsteam besteht idealerweise aus Vertretern sämtlicher am Projekt beteiligten Stakeholder, wie dem eigentlichen Scrum-Team, der Marketingabteilung, dem Vertrieb oder auch der Geschäftsführung. Auf diese Art wird sichergestellt, dass alle relevanten Parteien von Anfang an beteiligt sind und den Product Owner beim Schreiben der User Stories unterstützen.

Nehmen wir als Beispiel die Nutzerrolle "Job-Suchender". Das primäre Ziel dieser Rolle ist es, einen neuen Job zu finden. Dazu muss der Job-Suchende den Stellenmarkt nach bestimmten Kriterien durchsuchen können. Enthält die Ergebnisliste interessante Angebote, dann will er die Angebote im Detail betrachten. Interessiert ihn ein Angebot besonders, dann will er sich schließlich darauf bewerben. Ein einfacher Blick auf das Ziel "Job finden" führt zu drei offensichtlichen User Stories:

1. Als Arbeitssuchender will ich die Angebote nach bestimmten Kriterien durchsuchen.
2. Als Arbeitssuchender will ich die Details der gefundenen Angebote sehen.
3. Als Arbeitssuchender will ich mich auf ein Angebot bewerben.

Beim Betrachten dieser User Stories fällt auf, dass die Stories einem Muster folgen: Als <Nutzerrolle> will ich <das Ziel> [, so dass <Grund für das Ziel>].

Die Verwendung dieses Musters hat sich in der Praxis bewährt und ermöglicht es dem Product Owner mit sehr wenigen Worten sehr viel auszudrücken: Für wen ist die Anforderungen gedacht, welches Ziel verfolgt derjenige mit der Anforderung und was ist der Grund für dieses Ziel. Außerdem zwingt das Muster den Product Owner, den Kern der Anforderung und damit deren Wert für den Kunden auf den Punkt zu bringen.

Der letzte Teil des Musters, der "Grund für das Ziel", ist optional und sollte immer dann verwendet werden, wenn das Ziel alleine nicht ausreicht, um den Geschäftswert der Anwendung zu formulieren. Ein Beispiel ist die Story: "Als Job-Anbieter will ich mich einloggen". Die Funktion "Einloggen" stellt keinen offensichtlichen Geschäftswert dar. Erweitert man hingegen die Story um den Grund "so dass kein anderer Anbieter unter meinem Namen Job-Angebote erstellen kann", werden der Wert und damit der Grund fürs Einloggen deutlich.

Ergebnis des Workshops

Das Ergebnis des ersten Anforderungsworkshops ist ein initiales Product Backlog, gefüllt mit den wesentlichen User Stories des Projekts. Die initialen User Stories sind häufig noch zu groß, als dass sie sich für die direkte

Umsetzung in einem der ersten Sprints des Projekts eignen würden. Deshalb müssen sie auf die richtige Größe geschnitten werden. Hierbei kommt es darauf an, dass die resultierenden Teil-Stories weiterhin ihren Geschäftswert behalten. Ein Beispiel für eine zu große Story ist die Suche nach Stellenangeboten. Diese kann die Stichwortsuche, die Suche nach Ort, Bezahlung oder Position einbeziehen. Jede dieser Teilfunktionen sind wieder eigenständige User Stories die weiteren Wert für den Kunden schaffen und sich eigenständig umsetzen lassen.

Mit zunehmender Projekterfahrung wird relativ schnell offensichtlich, wann eine Story zu groß ist und wie sie sich am sinnvollsten zerschneiden lässt. Grundsätzlich gilt es, Stories so klein wie möglich zu schneiden, sofern sie weiterhin eine nützliche Funktion für den Anwender liefern. Die natürliche Obergrenze für die Größe einer Story ist dabei immer ein Sprint; schätzt das Team die Entwicklung einer Story auf länger als einen Sprint, ist klar, dass die Story zerschnitten werden muss.

Weitere Anhaltspunkte für zu große User Stories liefern die regelmäßig stattfindenden Estimation-Meetings. In diesen Meetings schätzt das gesamte Team die bis dahin geschriebenen User Stories.

Kommt es dabei zu sehr großen Schätzungen, oder kann der Product Owner die Story dem Team nicht ausreichend weit erklären, so dass das Team nicht in der Lage ist, die Story zu schätzen, ist es offensichtlich, die Story zu verkleinern.

Der erste Schritt vor dem Schätzen und dem daraus ggf. folgendem weiteren Zerschneiden der Stories ist ihre Priorisierung.

User Stories priorisieren und schätzen

Wie eingangs erwähnt, ist es ein wesentliches Scrum-Prinzip, die wichtigsten Dinge zuerst zu erledigen, um dem Kunden früh funktionsfähige Software zu präsentieren. Folglich muss sich der Product Owner für sein Backlog überlegen, welches die wichtigsten Stories sind und sie entsprechend priorisieren. Die wichtigsten Stories wandern im Backlog nach oben und werden damit zu Kandidaten für den nächsten Sprint.

Eine einfache und schnelle Variante der Priorisierung ist die sogenannte "MuSCoW-Priorisierung", die Stories hinsichtlich der folgenden vier Kriterien bewertet:

Must Have: Diese User Stories sind zwingend erforderlich. Ohne sie würde das System nicht funktionieren.

Should Have: Diese User Stories sind sehr wichtig, das System funktioniert aber auch ohne sie, weil z.B. eine provisorische Lösung (Workaround) existiert.

Could Have: User Stories dieser Kategorie haben eine geringe Bedeutung und werden nur umgesetzt, wenn neben Must Have- und Should Have-Stories noch Kapazitäten zur Verfügung stehen.

Won't have this time: Diese User Stories sind zurzeit nur vorgemerkt, werden aber aktuell nicht umgesetzt.

Nach dem ersten Anforderungsworkshop enthält das Backlog meistens noch viele zu große Stories. Mit dem Prinzip MuSCoW kann der Product Owner eine schnelle und einfache Priorisierung des initialen Product Backlog

durchführen und die Stories hinsichtlich ihrer Notwendigkeit für ein Funktionieren des Systems bewerten. Alle Must Have- und Should Have-Stories werden weiterverfolgt und auf konkretere Stories "heruntergebrochen". Die resultierenden Teil-Stories werden ihrerseits nach MuSCoW priorisiert. Auf diese Weise ist es möglich, relativ schnell zu einem sinnvoll priorisierten Product Backlog zu kommen, das vom Team geschätzt und als Grundlage für die Planung des ersten Sprints genutzt werden kann.

Stories schätzen

Nach der Priorisierung steht das Schätzen der Stories auf dem Programm. Zusammen mit dem Team geht der Product Owner das Backlog von oben nach unten durch und lässt das Team gemeinsam den Aufwand der jeweiligen User Story schätzen. Das relative Schätzen findet im Rahmen agiler Vorgehensweisen und damit auch im Rahmen von Scrum zunehmende Verbreitung. Bei dieser Art des Schätzens werden Anforderungen nicht in Personentagen, sondern ausschließlich in ihrer Größe geschätzt.

Der entscheidende Vorteil des relativen Schätzens ist die Trennung von Dauer und Größe. Während man beim traditionellen Schätzen versucht, in Personentagen vorherzusagen, welches Feature wie viel Entwicklungszeit benötigen wird, trifft man beim größenbasierten Schätzen ausschließlich Aussagen über die relative Größe von Features zueinander. So lässt sich sehr viel einfacher bestimmen, dass Feature A kleiner als Feature B ist, als vorherzusagen, dass Feature A genau einen Personentag beansprucht, während Feature B zwei Personentage dauern wird. Das Schätzen in relativen Größen erhöht die Wahrscheinlichkeit, dass die Schätzungen zutreffen, erheblich.

Der offensichtliche Nachteil relativer Schätzungen ist, dass Aussagen über die Entwicklungsdauern der jeweiligen Features fehlen. Schließlich wollen Projektleiter und Geschäftsführung nicht wissen, was wie groß im Verhältnis zueinander ist, sondern wann welches Feature geliefert wird. Ein Punkt, auf den ich später im Artikel zurückkommen werde.

Für die Schätzungen verwendet man "Story Points", die die relativen Größenverhältnisse der Stories zueinander ausdrücken. Eine Story von zwei Punkten ist doppelt so groß, wie eine 1-Punkt-Story. Eine 8-Punkte-Story ist viermal so groß, wie eine 2-Punkte-Story, und so weiter. Größe ist dabei zunächst ein abstrakter Begriff, der nichts darüber aussagt, wie lange die Entwicklung der jeweiligen Story dauern wird. Im Abschnitt "Sprints messen und planen" (s. unten) erkläre ich, wie Story Points für das Messen der Entwicklungsgeschwindigkeit des Teams und so für die Release-Planung genutzt werden können.

User Stories im Sprint

Hat der Product Owner die Stories priorisiert und liegen ausreichend geschätzte Stories für mindestens einen Sprint vor, dann kann das Team in seinen ersten Sprint starten. Am Anfang jedes Sprints steht das Sprint Planning Meeting, das in zwei Teile gegliedert ist: das "Sprint Planning 1" und das "Sprint Planning 2". Das Sprint Planning 1 ist ein Analyse-Meeting, in dem der Product Owner dem Team die User Stories der Reihe nach vorstellt. Das Team stellt Fragen, diskutiert die Story und versucht die Anforderungen ausreichend weit zu verstehen, um entscheiden zu können, ob es die Story im anstehenden Sprint vollständig umsetzen kann. Sieht das Team anschließend Raum für eine weitere Story, setzt der Product Owner das Vorstellen fort.

Einzig und allein das Team bestimmt, wie viele Stories es in den Sprint aufnimmt. Der Product Owner gibt ausschließlich die Reihenfolge der Stories, jedoch nicht deren Anzahl vor. Diese Regel ist wichtig im Hinblick auf die Teilung der Verantwortlichkeiten in Scrum: Das Team ist für die Software verantwortlich. Wenn das Team bestimmen darf, wie viel Arbeit es in den anstehenden Sprint nimmt, dann übernimmt das Team damit auch die Verantwortung dafür, die Software am Ende des Sprints zu liefern. Die Übernahme von Verantwortung basiert also auf Freiwilligkeit, was nicht gegeben wäre, wenn der Product Owner bestimmen würde, wie viele Stories das Team im anstehenden Sprint zu liefern hat.

Das im Anschluss an das Sprint Planning 1 folgende Sprint Planning 2 ist ein Design-Meeting. In diesem Meeting entwirft das Team das Software-Design der Stories und zerbricht jede Story in die konkret auszuführenden Arbeitsaufgaben, die sogenannten Tasks. Die Task-Liste der Angebotssuche könnte z.B. wie folgt aussehen:

- Domain-Modell entwerfen und implementieren
- Controller-Logik programmieren
- Layout entwerfen und Basis-CSS erstellen
- Suchtechnologie evaluieren
- ...

Stories und Tasks werden auf Karteikarten notiert und ans Taskboard gehängt. Das Taskboard ist die To-Do-Liste des Teams, das sämtliche noch ausstehende Arbeit des Sprints visualisiert.

Während des Sprints arbeitet das Team Story für Story ab. Die Stories werden im Sprint in der Reihenfolge ihrer Prioritäten nacheinander umgesetzt. Alle am Ende des Sprints nicht fertigen User Stories nimmt der Product Owner zurück ins Product Backlog. Ist die Story bei der nächsten Priorisierung noch genauso wichtig wie im gerade abgeschlossenen Sprint, schafft sie es unmittelbar zurück in den nächsten Sprint. Hier hat Scrum einen großen Vorteil: Die noch nicht umgesetzten Anforderungen werden ständig darauf überprüft, ob die ursprüngliche Fassung noch gültig ist.

Sprints messen und planen

Auch Scrum-Teams reihen nicht Sprint um Sprint aneinander und gucken mal, wann sie fertig sind. Sowohl der Kunde als auch der Product Owner haben ein berechtigtes Interesse an Planung und einer darauf basierenden Kosten- und Fertigstellungsvorhersage. Planung in Scrum basiert auf der sogenannten "Velocity". Diese bezeichnet die Entwicklungsgeschwindigkeit des Teams gemessen an der tatsächlich entwickelten und am Ende des Sprints gelieferten Funktionalität.

Dazu ein Beispiel: Hat das Team im Sprint Planning Meeting Stories mit einer Summe von 20 Story Points angenommen, schafft während des Sprints aber nur drei statt vier Stories, dann beträgt die tatsächliche Velocity dieses Sprints 20 minus der Story Points der vierten Story. Angenommen die vierte Story wurde vom Team auf fünf Punkte geschätzt, dann liefert das Team 15 Story Points ab, was die tatsächliche Velocity dieses Sprints ist. Basierend auf diesem Wert kann der Product Owner eine erste Releaseplanung vornehmen, indem er das Product

Backlog in Häppchen zu je 15 Punkten aufteilt. Die sich daraus ergebende Anzahl an Sprints multipliziert mit der ursprünglich geplanten Sprint-Dauer (z.B. vier Wochen) liefert das geschätzte Fertigstellungsdatum. Voraussetzung dafür ist, dass sämtliche der für das anstehende Release geplanten Stories vom Team geschätzt wurden.

Die Velocity wird von Sprint zu Sprint genauer. Während die tatsächliche Velocity nach dem ersten Sprint noch ein sehr grober und ungenauer Wert ist, pendelt sie sich nach einigen Sprints auf einen mittleren und damit für die Planung geeigneten Wert ein. Für eine sichere Planung empfiehlt es sich, mit einer niedrigeren als der mittleren Velocity zu planen, da auch eine mittlere Velocity durchaus in dem einen oder anderen Sprint unterschritten werden kann. Außerdem ist zu beachten, dass ein Product Backlog für Änderungen offen ist und Änderungen am Backlog folglich zu Änderungen am Release-Plan führen.

Fazit

Dieser Beitrag zeigt die Vorteile der Verwendung von User Stories im Rahmen von Scrum. User Stories fokussieren Team und Product Owner auf den Geschäftswert: Jeder Sprint liefert sichtbaren Mehrwert im Sinne des Kunden. Das Product Backlog enthält mit den User Stories ausschließlich werthaltige Items und kann vom Product Owner entsprechend priorisiert werden. User Stories verlagern den Fokus von der schriftlichen auf die verbale Kommunikation und erzwingen die Vollzeit-Verfügbarkeit des Product Owner. Statt genau aufzuschreiben, was das Team zu liefern hat, bleibt er eng an der Entwicklung dran und kann die Entwicklung des Teams sehr zeitnah und in kurzen Feedback-Schleifen steuern. Auf diese Art wird sichergestellt, dass am Ende jedes Sprints etwas heraus kommt, was der Product Owner wirklich haben will, und nicht etwas, was er vor zwei Monaten aufgeschrieben hat.

Literatur

- <http://scrumy.com>
- <http://www.thoughtworks-studios.com/mingle-agile-project-management>
- http://scrumninja.com/scrum_software
- Wirdemann, Ralf: **Scrum mit User Stories**, Hanser 2009

Methode

Kundenorientierte und marktgerechte Produktentwicklung

Mit QFD zur innovativen LED-Beleuchtung

Teil 1: Die Stimme des Kunden verstehen

Das Erkennen des Kundennutzens und seine optimale Umsetzung in neue Produkte sind für Unternehmen mehr denn je nicht nur Erfolgs- sondern sogar Überlebensfaktoren. Aber wie können Produktmanager den Kundennutzen erkennen und wie können Entwickler ihn optimal umsetzen? Noch dazu bei einer innovativen Technologie, die unzählige neue Möglichkeiten der Produktgestaltung bietet?

Insbesondere neue Technologien verleiten dazu, viele Funktionen in ein Produkt zu integrieren, die der Kunde zum Teil gar nicht benötigt und deshalb auch kein Geld dafür ausgeben möchte. Die Produktkosten steigen unnötig, der Gewinn schrumpft. Oder das Produkt trifft die Nutzenerwartung des Kunden generell nicht und findet nicht den erhofften Absatz. Die Entwicklungen von Digitalkameras oder Smartphones in den letzten Jahren sind für diese Herausforderungen bei der Produktentwicklung typische Beispiele: Eine neue Technologie ersetzt in kürzester Zeit eine bestehende und die Hersteller konkurrieren mit extrem kurzen Produktzyklen um die Gunst der Kunden.

Autor



Dr. Christine Knorr
Dipl.-Physik, seit 2011
selbstständige
Unternehmensberaterin
für technisches Marketing und PM

Kontakt: info@knorr-techmarket.de

Mehr Informationen unter:
projektmagazin.de/autoren

LEDs als Beleuchtung – jede Menge Fragen für die Entwickler

Genau in dieser Situation befinden sich die Hersteller von Leuchtmitteln seit der Erfindung einer lichtstarken, blauen LED 1993 durch den Japaner Shuji Nakamura, die auch den Weg zur weißen LED als Lichtquelle der Zukunft bereitete. Bereits 1997 kam die erste weiße LED auf den Markt. Aus LEDs ein marktfähiges Leuchtmittel zu konstruieren stellt die Entwickler aber vor eine Vielzahl von Fragen. Wenn Sie einen Blick auf die derzeit angebotenen LED-Leuchtmittel werfen, sehen Sie, dass sie sich in Anzahl und Anordnung der LEDs erheblich unterscheiden. Und dies sind nur zwei der von außen sichtbaren Parameter, die es zu optimieren gilt!

Ich begleitete einige Entwicklungen neuer LED-Produkte und möchte Ihnen hier ein Fallbeispiel mit realem Hintergrund vorstellen – Details (v.a. technische), die für das Verständnis nicht nötig sind, wurden vereinfacht. In dem Fallbeispiel sollte eine linienförmige Lichtquelle konstruiert werden, die Leuchtenhersteller für dekorative Wandbeleuchtungen benötigen. Schnell wurde uns klar, dass wir eine Vielzahl technischer Möglichkeiten und z.T. widersprüchlicher Kundenanforderungen miteinander in Einklang bringen mussten – und dies möglichst schnell! Wir entschieden uns deshalb, mit Quality Function Deployment (QFD) zu arbeiten, einer Methode der Produktplanung, die den Weg von den Kundenanforderungen bis zum Funktionsumfang des Produkts systematisch und zielführend abbildet.

Was ist QFD?

Quality Function Deployment (QFD) ist eine Methode, um die Funktionsmerkmale eines Produkts optimal auf die Kundenanforderungen abzustimmen. Der Japaner Yoji Akao entwickelte QFD in den sechziger Jahren, wo sie unter anderem bei Mitsubishi und Toyota einen wesentlichen Beitrag für deren Markterfolge leistete. In den achtziger Jahren fand sie in den USA Verbreitung, zuerst in der Automobilbranche dann in vielen anderen Branchen. Seit Ende der 80er Jahre wird QFD in Deutschland eingesetzt, seit einigen Jahren vermehrt auch in der Software- und Dienstleistungsbranche.

Die drei wichtigsten Merkmale der Methode sind:

- **Trennung von Anforderungen (das "Was") und Lösungen (das "Wie"):** Anforderungen sind stabiler als Lösungen. Die getrennte Analyse erhöht die Flexibilität sowohl beim Produktmanagement, das die Marktperspektive vertritt, als auch bei der Entwicklung, die für die technische Umsetzung verantwortlich ist.
- **Systematisches Vorgehen:** Entscheidungen über Produkteigenschaften erfolgen methodisch und werden dokumentiert. Der Prozess ist deshalb für alle Beteiligten stets nachvollziehbar.
- **Teamarbeit:** Alle Personengruppen, die Knowhow über Kundenanforderungen besitzen, arbeiten zusammen (Kunde, Vertrieb, Marketing, Produktmanagement, Entwicklung, Qualität, Produktion, Controlling, Kundenservice, ...).

Kern des Verfahrens ist das sogenannte "House of Quality" (HoQ), eine Matrix, mit deren Hilfe das Entwicklungsteam die Wechselbeziehungen zwischen Anforderungen und Umsetzung analysiert (Bild 1).

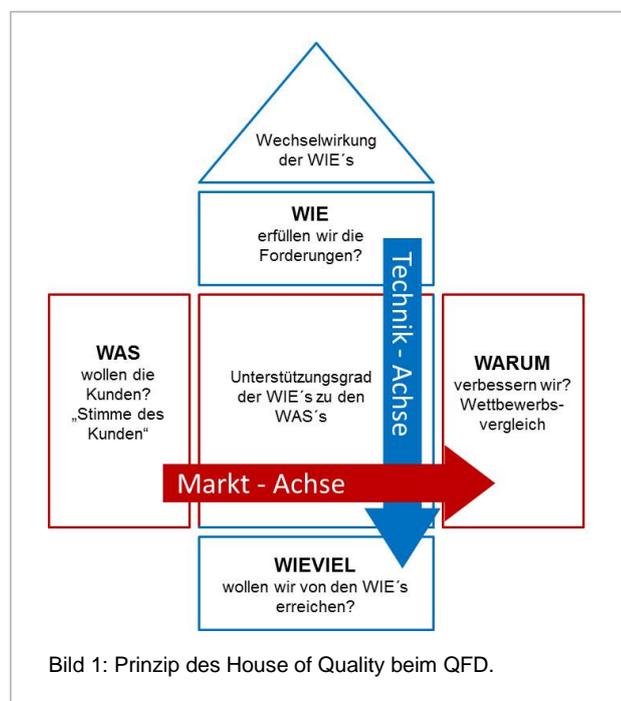


Bild 1: Prinzip des House of Quality beim QFD.

Grundlegendes Prinzip des House of Quality (HoQ) ist die konsequente Trennung der beiden Perspektiven "Markt" und "Technik" auf das Produkt (Bild 1). Die Marktachse "Was mache ich?" beinhaltet die Kundenanforderungen ("WAS"), den Wettbewerb und die eigene strategische Ausrichtung ("WARUM"). Erst danach werden mit der Frage "Wie mache ich es?" die dazugehörigen technischen Produktmerkmale ("WIE") gesucht. In der dann zu füllenden Matrix werden die Wechselwirkungen deutlich. Im Dach zeigen sich Widersprüche und Verstärkungen. Am Ende steht unter der Matrix eine Bewertung der Produktmerkmale ("WIEVIEL"), die eine für alle transparente und nachvollziehbare Entscheidungsgrundlage für die weitere Entwicklung des Produkts liefert.

Auch wenn die Darstellung auf den ersten Blick komplex aussieht: QFD ist keine schwierige Methode. Sie bringt lediglich Denkvorgänge und Prozesse, die im Produktentwicklungsteam sowieso stattfinden, in eine systematische und gut dokumentierte Form und stellt sicher, dass alle relevanten Personen zum jeweiligen Thema beitragen. Dadurch verhin-

dert sie auch "Holzwege", d.h. nicht zielführende Lösungsansätze, da sie sehr schnell unüberwindbare Hindernisse aufzeigt. QFD vereinfacht und beschleunigt somit den Produktentwicklungsprozess.

Warum QFD für die Entwicklung eines "einfachen" LED-Lichts?

Im Beispiel war aus einer Spezialanwendung eine qualitative hochwertige, sehr eng fokussierende Linsenoptik vorhanden, die eine Bündelung des LED-Lichts auf einen sehr kleinen Öffnungswinkel ermöglicht. So entstand die Idee eine lineare, d.h. sehr schmale aber beliebig lange Lichtquelle zu konstruieren, was mit herkömmlichen "Punkt-Lichtquellen", die zudem eine hohe Abwärme haben, nur mit hohem Aufwand realisierbar ist. Anwendung finden solche Lichtquellen bei der dekorativen Effektbeleuchtung durch Streiflicht: Stark fokussiertes Licht streift über eine Wandfläche und wird von der Wand in den Raum diffus reflektiert. Der Betrachter nimmt die eigentliche Lichtquelle, die z.B. in der Fußbodenleiste integriert ist, gar nicht wahr, sondern sieht nur eine "leuchtende" Wand.

Der Markt wartet auf das Produkt

Wir analysierten den Markt: Einige zig Leuchtenhersteller kamen als Abnehmer in Betracht. Erste grobe Abschätzungen zu Absatzmenge, Preis und Kosten lieferten attraktive Ergebnisse. Die neue Linsenoptik versprach ein gutes Alleinstellungsmerkmal gegenüber den Produkten der Wettbewerber.

Die Unternehmensleitung genehmigte daraufhin das Produktentwicklungsprojekt, ernannte einen Projektleiter und genehmigte die benötigten Ressourcen für das Entwicklungsteam. Der nächste Projektschritt stand an: die Produkthanforderungen mussten detailliert festgelegt werden, um anschließend Produktionskosten, Absatzmenge, und Verkaufspreis genauer bestimmen zu können.

Viele Möglichkeiten, unbekannte Anforderungen

Das Team aus Entwicklung, Produktmanagement, Qualitätsabteilung, Applikation und Vertrieb diskutierte intensiv: Wie viele LEDs sollen auf die Lichtquelle? In welchem Abstand? Sollten die Kunden, d.h. die Leuchtenhersteller, löten, um das LED-Modul elektrisch anzuschließen? Ist die Spezialoptik einsetzbar? Und viele, viele andere Fragen waren zu beantworten. Ein Vertriebsmitarbeiter favorisierte sehr konkrete Wünsche seines Hauptkunden. Das Team zweifelte jedoch an deren Akzeptanz bei anderen Kunden. Nach zwei langen und ergebnislosen Besprechungen war das Resümee: Wir müssen die Kunden fragen, deren Anforderungen verstehen und diese umsetzen.

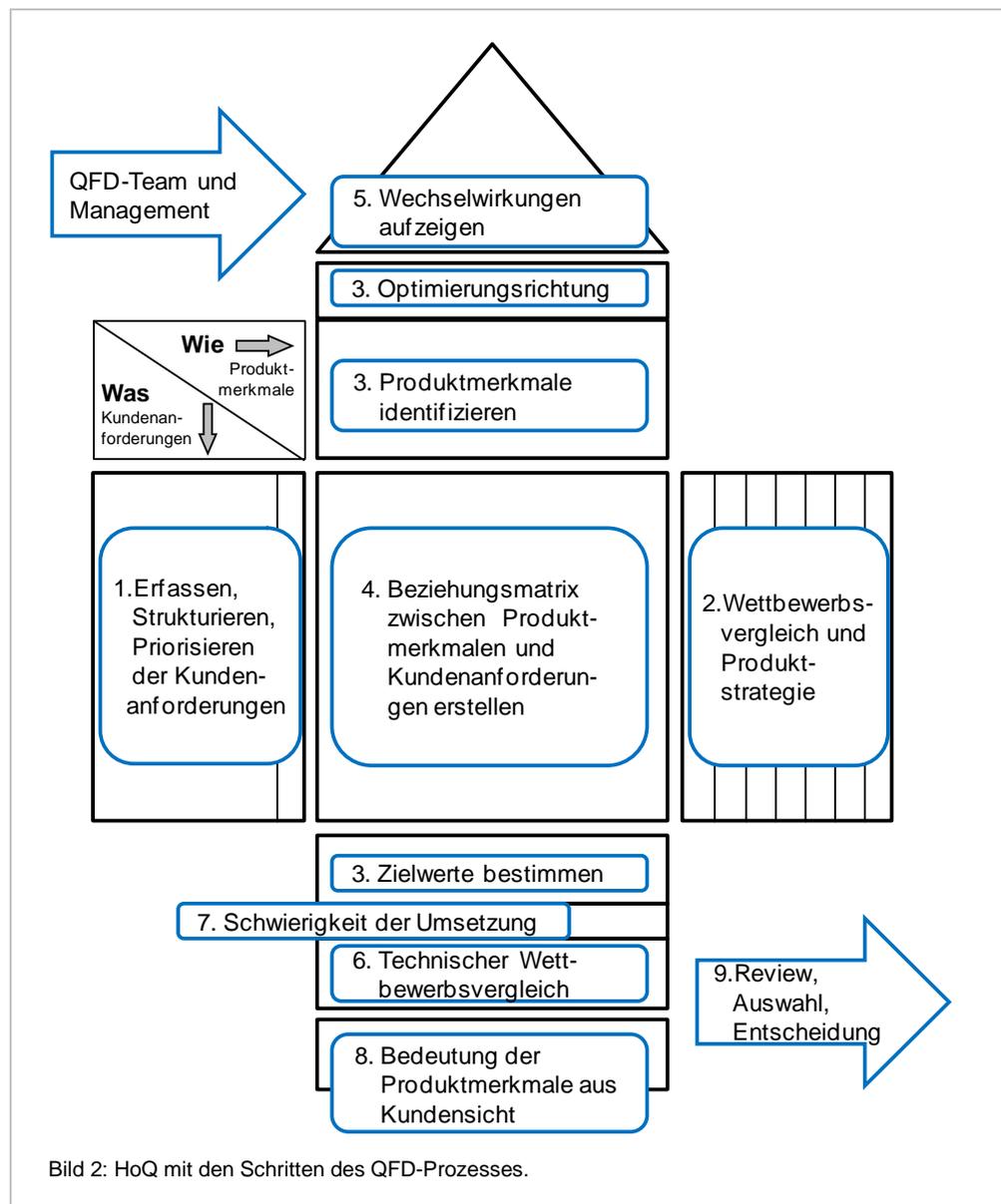
In dieser Situation schlug ich zusammen mit einem Kollegen, der ebenfalls entsprechende Methodenkenntnisse hatte, den Einsatz von Quality Function Deployment (QFD) vor. Der Standard-Entwicklungsprozess sah QFD zwar nicht vor, aber das Team erkannte, dass QFD für diese Situation ein optimales Werkzeug darstellte. Das Projektteam beschloss daher, für die LED-Lichtquelle das erste "House of Quality" zu erarbeiten und mit dem Ergebnis im regulären Entwicklungsprozess weiterzuarbeiten.

Der QFD-Prozess

QFD vollständig eingesetzt beschreibt in einer Folge von Houses of Quality den kompletten Entwicklungsprozess von Produktdefinition über Entwicklung und Produktion bis hin zum After Sales Service (Saatweber, 2011). Wir haben uns auf

das erste "Haus" beschränkt, das die Kundenanforderungen in die Sprache der Technik übersetzt. Meiner Meinung nach ist eine gute Produktdefinition der entscheidende Erfolgsfaktor bei der Produktentwicklung. Alle Optimierungsmaßnahmen in der Entwicklungs- und Produktionsphase nützen wenig, wenn das falsch definierte Produkt optimiert wird.

Bild 2 zeigt den typischen Aufbau eines HoQ, in das ich die einzelnen Prozessschritte eingetragen habe. Alle relevanten Informationen sind am Ende des Prozesses im Haus eingetragen und gut dokumentiert.



Voraussetzungen dafür, dass QFD "funktioniert" sind die Einrichtung eines QFD-Teams und die Unterstützung der übergeordneten Managementebene, denn nur so können die oben genannten Eigenschaften von QFD realisiert werden. Sobald diese Voraussetzungen geschaffen sind, kann der eigentliche QFD-Prozess durchgeführt werden, den ich anhand des Beispiels in den folgenden neun Schritten beschreiben will:

1. Erfassen, Strukturieren und Priorisieren der Kundenanforderungen
2. Wettbewerbsvergleich und Produktstrategie
3. Produktmerkmale identifizieren, Zielwerte und Optimierungsrichtung bestimmen
4. Beziehungsmatrix zwischen Kundenanforderungen und Produktmerkmalen erstellen
5. Wechselwirkungen zwischen Produktmerkmalen aufzeigen
6. Technischer Wettbewerbsvergleich
7. Schwierigkeit der Umsetzung
8. Bedeutung der Produktmerkmale aus Kundensicht
9. Review, Auswahl, Entscheidung

Installation des QFD-Teams und Managementunterstützung

Projektleitung, Produktmanagement und Qualitätsmanagement erarbeiteten eine Entscheidungsvorlage für den Einsatz von QFD und unterbreiteten diese der Unternehmensleitung, in der sie die Gründe dafür ausführlich darstellten. Neben den situationsbezogenen Gründen sprechen allgemein folgende Argumente für den Einsatz von QFD (Saatweber, 2011):

- QFD trägt dazu bei, die Kundenerwartungen an das Produkt sicher zu erfüllen. Aus verbalen Kundenwünschen wird mit Hilfe des QFD eine klare technische Spezifikation erstellt.
- Mit QFD wird in allen Bereichen des Unternehmens ein detailliertes Produktwissen aufgebaut. Die Kommunikation zwischen den Abteilungen wird gefördert und ein gemeinsames Verständnis für das Unternehmensziel geschaffen.
- QFD trägt dazu bei, die Entwicklungszeiten zu verkürzen, da der spätere Änderungsaufwand reduziert wird. Fehlentwicklungen, die am Markt vorbei gehen, werden vermieden.
- Der höhere Zeit- und Ressourcenaufwand einer Produktdefinition mit QFD ist absolut zu rechtfertigen, da in Summe die Entwicklungskosten am Ende niedriger sind.

Die Unternehmensleitung genehmigte das Vorgehen und benannte aus den Bereichen Produktmanagement, Entwicklung, Controlling und Anwendungstechnik konkrete Mitarbeiter, die als Mitglieder des QFD-Teams am Projekt mitwirken sollten. Der Qualitätsmitarbeiter, der über ausreichende QFD-Kenntnisse verfügte, wurde als neutraler Moderator des QFD-Projekts bestimmt, so dass kein externer Spezialist erforderlich war. Je nach Bedarf wurden Vertrieb, Logistik, Kundenservice, Produktion und ggfs. weitere Bereiche mit eingebunden.

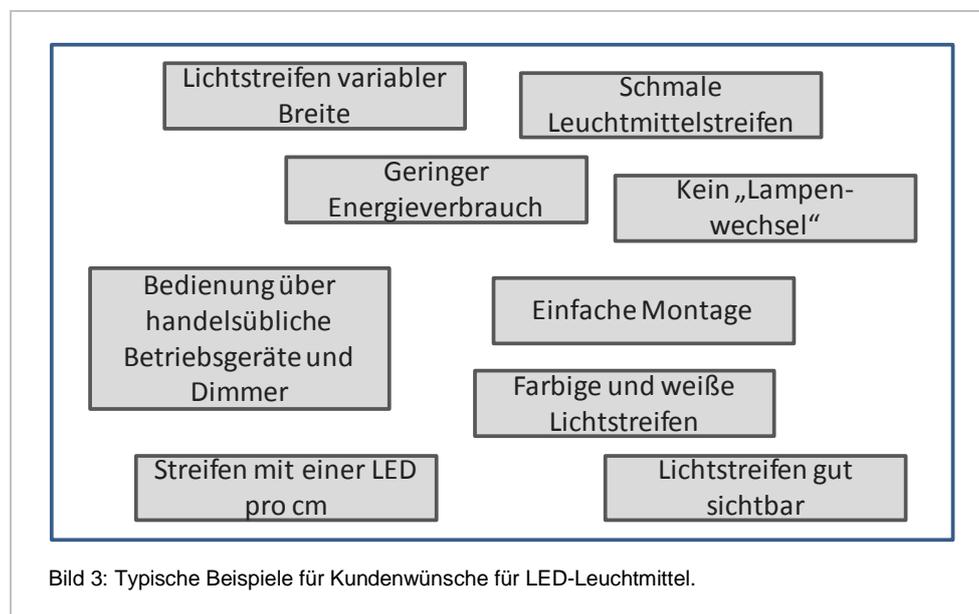
! Wichtig ist, dass alle Personen involviert werden, die zum Produkt beitragen können, und so das im Unternehmen vorhandene Knowhow vollständig genutzt wird. Der Moderator benötigt entsprechendes Geschick, um für Besprechungen genau die Personen einzuladen, die zum Thema beitragen können. Nehmen Personen an einem Workshop teil, an dem sie kaum mitwirken können, sinkt deren Bereitschaft rapide, künftig am QFD-Projekt mitzuarbeiten.

Als Arbeitsmittel legten wir Microsoft Excel fest. Das HoQ lässt sich ohne weiteres mit einer Tabellenkalkulations-Software abbilden. Es gibt auch Spezialsoftware dafür, dies ist aber kein Muss.

Erfassen, Strukturieren und Priorisieren der Kundenanforderungen

Um die Kundenanforderungen – im QFD-Jargon "Stimme des Kunden", "Voice of the Customer" oder kurz "VoC" genannt – zu erfassen, gibt es eine Vielzahl von Methoden. Aufgrund der innovativen Technologie entschieden wir uns, zunächst strukturierte Interviews mit rund zehn wichtigen und innovativen Kunden durchzuführen. Mitarbeiter aus den Bereichen Vertrieb, Produktmanagement, Anwendungstechnik und Entwicklung besuchten über einen Zeitraum von zwei Monaten diese Schlüsselkunden und präsentierten dabei Prototypen als Demonstrationsmuster. Die Kunden erhielten auf diese Weise eine anschauliche Vorstellung davon, was mit der neuen Technologie möglich ist, so dass ihre Antworten konkreter und fundierter wurden. Als Zielanwendung kristallisierte sich bei diesen Einzelinterviews die Effektbeleuchtung in repräsentativen Gebäuden (Hotel, Wellness-Bereich, Tagungshäuser usw.) heraus in Form der Erzeugung von Lichtstreifen an Wänden und Säulen.

Auf Basis der Ergebnisse dieser Einzelinterviews entwarfen wir einen zweiseitigen Fragebogen, den der Vertrieb an einen größeren Kundenkreis von ca. 50 Leuchtenherstellern verschickte. Wie zu erwarten erhielten wir eine Vielzahl von Kundenwünschen. Bild 3 zeigt eine Auswahl typischer Kundenaussagen, die sich gut als Beispiele zur Demonstration des QFD-Prozesses eignen.



Damit der QFD-Prozess optimal funktioniert, sollen Kundenanforderungen

- für alle verständlich und nachvollziehbar,
- eindeutig,
- möglichst quantifiziert und damit messbar,
- nicht aus Lösungen abgeleitet,
- möglichst vollständig,
- und möglichst unabhängig voneinander sein.

Die Aufgabe besteht also darin, aus den Kundenwünschen Kundenanforderungen zu destillieren, die diese Eigenschaften erfüllen. Ich möchte exemplarisch einige der Wünsche aus Bild 3 herausgreifen und anhand dieser Kriterien für den QFD-Prozess taugliche Anforderungen daraus ableiten.

"Lichtstreifen variabler Breite" – wie breit genau?

Mit dieser Anforderung ist gemeint, dass die Länge der LED-Einheiten variierbar sein muss, um unterschiedlich breite Lichtstreifen an der Wand erzeugen zu können. Diese Anforderung ist verständlich und nachvollziehbar, allerdings ist sie nicht quantifiziert, da keine Maßzahlen für die geforderten Breiten angegeben sind. Der Produktmanager fragte deshalb bei ausgewählten Schlüsselkunden nach konkreten Maßen und konnte die Anforderung dadurch präzisieren: Für den schmalsten Lichtstreifen reichen 10-15 cm aus, auch für die Stückelung ist dieses Maß eine sinnvolle Schrittgröße.

"Schmale Leuchtmittelstreifen" – wofür braucht man das?

Für diese Anforderung wurden bereits bei der Umfrage konkrete Werte zwischen 0,5 cm und 3 cm genannt. Dies sind zwar konkrete Zahlen, allerdings ist nicht nachvollziehbar, was der Grund für diese Anforderung ist und woher diese Werte kommen. Ein Hinterfragen ergab: Die Kunden möchten die LED-Module in U-Profile verschiedener Breite montieren. Im weiteren Gespräch erfuhr der Produktmanager zusätzlich, dass für die Anwendung hauptsächlich Profile der Breite 3 cm geplant sind.

"Geringer Energieverbrauch" – eh klar!

Bei der Diskussion um konkrete Zahlenwerte stellte sich heraus, dass ein genauer Zielwert nicht interessant ist. LED-Lichtquellen benötigen sowieso bedeutend weniger Energie als Glühlampen. Die Lichtquelle sollte allerdings mindestens in die Energieeffizienzklasse A+ fallen.

"Farbige und weiße Lichtstreifen" – mehrere Produkte?

Hier wünscht der Kunde verschiedene Farbvarianten. Das ist eine klare und eindeutige Anforderung, stellt aber das QFD-Team vor eine wichtige Entscheidung: Sollen rote, grüne, blaue und weiße LED-Module als jeweils eigene Produkte betrachtet werden oder können sie gemeinsam analysiert werden? Wir beschlossen, diese Varianten in einem einzigen HoQ zu betrachten. Dies vereinfacht einerseits das Vorgehen, birgt aber andererseits die Gefahr, wichtige funktionelle Unterschiede zu übersehen.

"LED Modul mit einer LED pro cm" – nicht lösungsneutral

Dieser Kundenwunsch stellt bereits eine mögliche Lösung dar! Ein direktes Umsetzen einer vom Kunden vorgeschlagenen Lösung widerspricht dem Gedanken von QFD, da die Suche nach technischen Lösungsmöglichkeiten nicht unnötig eingeschränkt werden soll. Deshalb müssen solche Kundenwünsche sorgfältig hinterfragt werden: Warum schlägt der Kunde diese Lösung vor? Die Idee des Kunden war, dass das LED Modul einen homogenen Lichtstreifen an der Wand produzieren soll, in dem das Licht einer einzelnen LED nicht mehr auflösbar ist. Wir formulierten deshalb als Anforderung: "homogener Lichtstreifen, mit dem Auge sind keine Intensitäts- und Farbschwankungen wahrnehmbar".

Den Kunden wirklich verstehen – nicht nur interpretieren!

! Dieses unermüdliche Hinterfragen und Verstehen der Kundenwünsche ist Basis für eine gute und richtige Formulierung der Kundenanforderungen. Wichtig ist, dass Sie die Wünsche nicht nach eigenen Vorstellungen interpretieren, sondern wirklich verstehen, was der Kunde damit meint. Durch diesen Prozess erhält man einen tiefen Einblick in die Denkweise der Kunden und versteht die Anwendung immer besser. Dies sind wertvolle Informationen für die aktuelle und die zukünftige Produktdefinition.

Es gibt darüber hinaus Wünsche, die der Kunde als selbstverständlich erwartet und daher nicht explizit äußert. In der Terminologie des sog. **Kano-Modells** sind dies "Basisanforderungen". Ein Autokäufer geht z.B. davon aus, dass ein Airbag vorhanden ist, ohne dies extra zu benennen. Auch Normen, die einzuhalten sind, wird der Kunde vermutlich nicht ansprechen. Natürlich müssen diese jedoch bekannt sein und eingehalten werden.

Je nach Produkt ergeben sich sehr viele Kundenanforderungen. Bei über 20 Anforderungen wird die Tabelle unübersichtlich und die weitere Bearbeitung mühsam. Abhilfe schafft eine Aufteilung in Untersysteme: Soll z.B. eine neue Kamera definiert werden, so kann für das optische System ein eigenes QFD durchgeführt werden.

Wir strukturierten die erhaltenen Kundenanforderungen nach Oberbegriffen. Das Ergebnis zeigt Tabelle 1.

Es ist nicht möglich, alle Anforderungen gleichzeitig vollständig zu erfüllen. Um die richtigen Schwerpunkte zu setzen, müssen sie priorisiert werden. Die QFD-Bewertungsskala umfasst 10 Stufen im Bereich von 1 = "unwichtig" bis 10 = "sehr wichtig".

Die Priorisierung erfolgte in einer Gruppendiskussion im QFD-Team mit einstimmiger Vergabe der Bewertungen durch die Mitglieder, dabei gehen die aus den vorangegangenen Gesprächen bekannten Prioritäten einzelner Kunden über ihre Kontaktpersonen ein. Wenn in der Gruppendiskussion keine Einstimmigkeit erzielt wird, empfiehlt sich die Methode des paarweisen Vergleichs. Ist die Priorisierung dann immer noch nicht möglich, müssen die Anforderungen nochmals genauer analysiert werden. Die Spalte "Prio" in Tabelle 1 zeigt das Ergebnis unserer Bewertung im Beispiel.

	Kundenanforderung	Messgrößen	Zielwert	Prio
Performance	Lichtstreifen variabler Breite	Breite des Lichtstreifens	ca. 10-15 cm "Auflösung"	10
	homogener Lichtstreifen	Intensitäts-/Farbverteilung im Lichtstreifen	Schwankungen nicht mit Auge wahrnehmbar	8
	farbiges Licht und weiß	Farbtemperatur (weiß) und Position in Farbdreieck	warmweiß, rot, blau, grün	10
	Farben kombinierbar, Streifenmuster	beliebige Farbkombinationen	ca. 10-15 cm "Auflösung"	2
	Streifen gut sichtbar über Raumhöhe	Lichtintensität über Streifenlänge	mit Auge gut sichtbar bis Höhe 3 m	9
Bedienung/ Wartung	mit gängigen Betriebsgeräten und Dimmern betreibbar	marktübliche Betriebsgeräte und Dimmer	Abdeckung der 3 Hauptanbieter	3
	keine Lichtabnahme mit dem Betrieb	Lichtintensität	mit Auge keine Abnahme wahrnehmbar im Zeitraum 5 Jahre	6
Montage	in Standardprofile montierbar	Breite der häufigsten Standardprofile	2 - 3 cm	7
	einfache Montage	Standardwerkzeug	Klebebänder und Akkuschrauber	6
	einfache elektrische Verbindung	Standardwerkzeug	Stecken oder Klemmen	6
Umwelt	geringer Energieverbrauch	Energy label	Kategorie A+	3

Tabelle 1: Strukturierte und priorisierte Anforderungsliste.

Wettbewerbsvergleich und Produktstrategie

Für den Unternehmer ist neben der reinen Kundenaussage wichtig, bis zu welchem Grad Wettbewerbsprodukte die Kundenanforderungen erfüllen. Dies ist ein Maß dafür, wie gut das eigene Produkt die Anforderung erfüllen muss. Oft reicht es aus, etwas besser als der Wettbewerb zu sein. Wer nach einem zu hohen Maß an Perfektion strebt, läuft Gefahr, entweder ein zu teures Produkt herzustellen oder zu viel Zeit und damit Marktanteile zu verlieren.

Dazu wählt man die wichtigsten Wettbewerbsprodukte aus und beurteilt den Erfüllungsgrad der Kundenanforderungen anhand eines Stärken-Schwächen-Profiles. Die Skala spannt sich von 1= "schlechter" bis 5 = "besser". Das Ergebnis wird in die Spalten rechts neben der zentralen Matrix des HoQ eingefügt (vgl. Bild 1 und Bild 2).

Wir bewerteten zwei Wettbewerbsprodukte (Spalten "W1" und "W2" in Tabelle 2). Die Bewertung basierte auf Kundenaussagen in vorangegangenen Gesprächen, ergänzt durch die eigene Einschätzung mit der "Brille" des Kunden. Um das Stärken-Schwächen-Profil zu veranschaulichen, kann man zusätzlich zu den numerischen Werten eine Liniengrafik erstellen.

Ebenso wird das Entwicklungsziel des eigenen Produkts (Spalte "SOLL") basierend auf dem Wettbewerbsvergleich eingetragen. Ist ein früheres Vergleichsprodukt ("IST") vorhanden, so kann auch dieses Produkt eingetragen werden. Der Quotient aus Entwicklungsziel "SOLL" zum "IST" kann dann als Korrekturfaktor die Priorisierung der Anforderungen aus reiner Kundensicht neu bewerten: $Prio(neu) = Prio * SOLL/IST$. Da wir ein neues Produkt entwickelten, entfiel bei uns dieser Schritt.

Im Praxisbeispiel waren die Wettbewerbsprodukte für einen sehr breiten Anwendungsbereich vorgesehen, wodurch sich ein direkter Vergleich mit dem geplanten Produkt schwierig gestaltete. Die Daten der Wettbewerber verbleiben in der Tabelle, da sie eine wesentliche Information für das HoQ darstellen. Sie gehen jedoch nicht in eine Modifikation der Gewichtung ein.

	Kundenanforderung	Prio	MS	Gesamt	W1	W2	SOLL
Performance	Lichtstreifen variabler Breite	10	1	10	4	3	5
	homogener Lichtstreifen	8	1,2	10	3	2	4
	farbiges Licht und weiß	10	1,2	12	4	3	5
	Farben kombinierbar, Streifenmuster	2	1	2	2	2	3
	Streifen gut sichtbar über Raumhöhe	9	1,2	11	5	2	4
Bedienung/ Wartung	mit gängigen Betriebsgeräten und Dimmern betreibbar	3	1	3	4	4	4
	keine Lichtabnahme mit dem Betrieb	6	1,2	7	3	3	5
Montage	in Standardprofile montierbar	7	1	7	4	3	4
	einfache Montage	6	1	6	2	3	4
	einfache elektrische Verbindung	6	1	6	2	2	4
Umwelt	geringer Energieverbrauch	3	1	3	3	3	3

Tabelle 2: Vergleich mit Wettbewerbsprodukten. Der Faktor "MS" beschreibt die Bedeutung für die Marketingstrategie: "W1" und "W2" sind Produkte des Wettbewerbs; "Soll" beschreibt die Qualität des neuen Produkts im Vergleich zu den Wettbewerbsprodukten.

Neben Kundensicht, Wettbewerbssicht und Entwicklungspotential des eigenen Produkts kann die Marketingstrategie Einfluss auf die Bewertung nehmen. Soll das Produkt als lichtstärkstes Produkt vermarktet werden? Liegt der Verkaufsschwerpunkt auf Langlebigkeit und Zuverlässigkeit? In der Spalte "MS" = "Marketingstrategie" kann die Bedeutung der einzelnen Anforderung mit Faktoren von 1 (normaler Verkaufsschwerpunkt) bis 1,5 (hohe Bedeutung) modifiziert werden, wodurch Werte größer als 10 entstehen können (s. Tabelle 2, "Streifen gut sichtbar über Raumhöhe"). Wir rundeten die erhaltenen Werte, denn Kommazahlen suggerieren eine Genauigkeit, die in der Realität nicht existiert.

Der Auftraggeber – in unserem Fall die Unternehmensleitung – muss regelmäßig über den Stand des QFD-Projekts informiert werden. Die Faktoren der Marketingstrategie sollten zusammen mit bzw. vom Auftraggeber festgelegt werden, da hier grundsätzliche, strategische Überlegungen mit eingehen. Im Praxisfall lag der

Schwerpunkt auf der Qualität des Lichtstreifens: gut sichtbare und homogene Streifen in passender Farbe mit nicht wahrnehmbarer Alterung (Lichtabnahme mit der Zeit) sollten das Produkt besonders auszeichnen (s. Spalte "MS" in Tabelle 2). Die Priorität der Kundenanforderung wird modifiziert, indem die reine Kundenbewertung mit dem Faktor "Marketingstrategie" multipliziert wird. Das Ergebnis sehen Sie in der Spalte "Gesamt" der Tabelle 2. Damit ist die Markt-Achse des HoQ komplett.

Wie geht es weiter?

Mit dem Sammeln, Präzisieren und Priorisieren der Kundenanforderungen ist schon ein erster, sehr wichtiger Schritt hin zum Produkt getan: Wir wissen nun das "Was" – aber noch nicht das "Wie". Der übliche Weg der Produktentwicklung besteht nun darin, dass Experten ein Fachkonzept erstellen, in dem sie den Lösungsweg beschreiben und die technische Spezifikation erstellen.

QFD systematisiert und dokumentiert den Prozess der Spezifikationserstellung vollständig und bewirkt allein dadurch bereits einen großen Mehrwert. Der Nutzen von QFD geht jedoch weit darüber hinaus: Es entsteht ein ganzheitliches und zugleich analytisches Bild des Produkts: In ihm sind alle technischen Funktionen mit ihren Wechselwirkungen, ihrer Bedeutung für die Qualität des Produkts bis hin zu ihren Auswirkungen auf die Produktionskosten enthalten.

Im zweiten Teil erfahren Sie, wie Sie mit Hilfe von QFD ein optimal auf die Bedürfnisse des Markts ausgerichtetes Produkt spezifizieren.

Literatur

- Saatweber, Jutta: Kundenorientierung durch Quality Function Deployment, Symposion Publishing GmbH, Düsseldorf, 2011

Methode

Kundenorientierte und marktgerechte Produktentwicklung

Mit QFD zur innovativen LED-Beleuchtung

Teil 2: Effizient das richtige Produkt spezifizieren

Im ersten Teil dieses Beitrags: "Die Stimme des Kunden verstehen" lernten Sie das Prinzip des Quality Function Deployments (QFD) kennen und erfuhren anhand eines Fallbeispiels, wie Sie aus Kundenwünschen konkrete Anforderungen ableiten und diese zum Zweck der Produktentwicklung priorisieren. Dies waren die ersten beiden der vorgestellten neun Schritte des QFD-Prozesse (siehe Teil 1).

Dieser Teil beschreibt die nächsten sieben Schritte zur Spezifikation des Produkts. Mit ihnen stellen Sie die möglichen technischen Funktionen so zusammen, dass das neue Produkt sowohl die Kunden überzeugt als auch mit der eigenen Produktstrategie in Einklang steht.

Autor



Dr. Christine Knorr

Dipl.-Physik, seit 2011
selbstständige

Unternehmensberaterin

für technisches Marketing und PM

Kontakt: info@knorr-techmarket.de

Mehr Informationen unter:

projektmagazin.de/autoren

Produktmerkmale identifizieren, Zielwerte und Optimierungsrichtung bestimmen

Jetzt verschiebt sich der Schwerpunkt der Aktivitäten auf den Aspekt "Technik". Neben der Entwicklung sind dabei auch Anwendungstechnik, Produktion, Produktmanagement und – je nach Unternehmen und Organisationsform – weitere Bereiche involviert. Das Team sucht zuerst nach Produktmerkmalen, die Lösungsansätze für die Kundenanforderungen sind. Die so erhaltenen charakteristischen Merkmale werden in die Tabelle eingetragen und nach Oberbegriffen gruppiert (vgl. Tabelle 1). Optimierungsrichtung und Ziel-, bzw. Grenzwerte quantifizieren die Merkmale.

! Team und Moderator müssen darauf achten, dass hier keine denkbaren Lösungen eingesetzt werden, sondern Leistungsmerkmale. Ich möchte dies anhand eines Beispiels verdeutlichen:

Die Kundenanforderung "Lichtstreifen gut sichtbar über Raumhöhe" kann z.B. durch die Spezifikation "10 helle LED des Typs X vom Hersteller Y" gut erfüllt werden. Dies stellt allerdings bereits eine mögliche Lösung dar und schränkt die Anzahl der Lösungsmöglichkeiten an dieser Stelle unnötig ein. Vielleicht findet sich ja ein anderer Hersteller, bei dem bereits 8 LEDs genügen. Ein besseres Qualitätsmerkmal ist "Lichtstärke in Achsrichtung", das im weiteren Schritt durch die Optimierungsrichtung ("höhere Lichtstärke") und einen messbaren Ziel- oder Grenzwert (z.B. ">1000 cd") beschrieben wird.

Die Suche nach den Produktmerkmalen kann länger dauern. Das Festlegen von Ziel- bzw. Grenzwerten erfordert eine genaue Betrachtungsweise, die das Hinzuziehen von Spezialisten aus Entwicklung oder Produktion erforderlich machen kann. Bei der Suche nach einem Zielwert für die Lichtstärke stellten wir z.B. fest, dass Informationen vom Kunden bzw. der Anwendung fehlten: Die nötige Lichtstärke, um den Lichtstreifen über eine Raumhöhe von 3 m sichtbar zu machen, hängt wesentlich von der Rauheit der Wand ab und von der Umgebungshelligkeit. Hier hilft nur eines: Gehe zurück zu Schritt 1 und beschreibe die Kundenanforderung genauer!

Um das Vorgehen in diesem Schritt zu verdeutlichen, möchte ich einige Produktmerkmale aus dem Beispiel herausgreifen.

"Farbschwankungen über die Streifenbreite"

Dieses Merkmal korreliert direkt mit der Kundenanforderung "homogener Lichtstreifen". Farb- und Intensitätsschwankungen sollten mit dem Auge nicht wahrnehmbar sein. Die Optimierungsrichtung war daher "niedriger". Der technische Zielwert war der mit dem Auge wahrnehmbare Farbunterschied, messbar definiert als zulässige Schwankung der Spektralzusammensetzung und der Intensität.

"Eingangsspannung"

Die Hauptanbieter von Betriebsgeräten liefern Netzgeräte mit 24 V Gleichspannung. Dies war damit auch der Zielwert für die Eingangsspannung des Lichtmoduls. Eine Optimierungsrichtung gab es hier nicht.

Zu den Produktmerkmalen gehört auch die Erfüllung relevanter Normen (Produktsicherheit, Umweltrichtlinien, etc.), die hier nicht genannt sind. Die Erfüllung dieser Normen ist Voraussetzung dafür, dass das Produkt in Umlauf gebracht werden kann.

Produktmerkmale	Optimierungsrichtung	technische Ziel- u. Grenzwerte
optische Eigenschaften		
Farbtemperatur warmweiß		3200 K
Wellenlänge des Lichtes (rot/blau/grün)		620/520/467 nm
Lichtstärke in Achsrichtung	▲	> 1000 cd
Abstrahlwinkel Längsachse	▼	< 6°
Farbschwankungen über Streifenbreite	▼	< 200 K (weiß), < 8 nm (rot), < 5 nm (blau, grün)
Intensitätsschwankungen über Streifenbreite	▼	< 50% des max. Werts
elektrische Eigenschaften/Anschluss		
Eingangsspannung		24 V +/- 1V (DC)
Module elektrisch durchverdrahtbar	▲	bis max. 50 W Gesamtleistung
Module verschiedener Farben elektrisch kompatibel		(alle Farben)
Modulunterseite elektrisch isolierend		el. isolierend, Normwert
Stromaufnahme	▼	< 200 mA
el. Anschluss über Steckerverbinder		Spezialstecker
geometrische Maße		
Modullänge	▲	< 60 cm
Teilbarkeit	▼	> 10 cm, < 15 cm
Modulbreite		< 3 cm
Schraublöcher	▼	2
Modulunterseite flach		glatte Fläche, keine sichtbaren Kanten
Betrieb		
Lichtabnahme im Betrieb durch Alterung (Lebensdauer)	▼	50% der LED-Module müssen nach 50.000 Std. Betriebsdauer noch 70% des ursprünglichen Lichtstroms liefern.
zusätzliche Kühlmaßnahmen		keine
Energieaufnahme gemäß Effizienzklasse		gemäß A+

Tabelle 1: Produktmerkmale, gruppiert nach technischen Gesichtspunkten.

Beziehungsmatrix zwischen Kundenanforderungen und Produktmerkmalen erstellen

Jetzt wird die zentrale Matrix gefüllt, die zwischen den Anforderungen und den Produktmerkmalen vermittelt. Das Team fragt für jedes Produktmerkmal, wie stark es die einzelnen Forderungen des Kunden unterstützt und vergibt hierfür einen Zahlenwert. In der Literatur finden sich hierfür verschiedene Bewertungsskalen. Aus Erfahrung empfehle ich die am meisten eingesetzte, nichtlineare Skala (Tabelle 2):

Stärke der Beziehung	Zahlenwert für Matrix
Keine Beziehung	0 bzw. kein Eintrag
Geringe Beziehung	1
Mittlere Beziehung	3
Starke Beziehung	9

Tabelle 2: Empfohlene Skala für die Bewertung der Beziehung zwischen Anforderungen und Funktionen.

Diese nichtlineare Skala erzeugt eine starke Spreizung und damit eine bessere Differenzierung bei der Auswertung. Es geht bei QFD nicht um exakte Zahlenwerte, sondern um ein Herausarbeiten der kritischen und wichtigen Merkmale. Diese Skala betont Merkmale überproportional, die Kundenanforderungen stark unterstützen, was letztlich das Ziel von QFD ist.

Für diese Aufgabe ist das komplette QFD-Team gefragt. Wir gehen spaltenweise vor:

"Farbtemperatur weiß mit 3200 K" unterstützt stark die Forderung nach warmweißem Farbton ("9"), es trägt mittel zu möglichen Streifenmustern bei ("3") und durch den Farbton leicht zur Sichtbarkeit der Streifen ("1") an der Wand. Für die anderen Farben gilt dies analog.

Die hohe Lichtstärke in Achsrichtung trägt stark dazu bei, die Lichtstreifen gut über die ganze Raumhöhe sichtbar zu machen ("9") und unterstützt die Möglichkeit der variablen Breiten ("3"). Aber inwieweit trägt dieses Merkmal dazu bei, den Streifen homogen erscheinen zu lassen? Diese Frage beschäftigte das Team lange: Irgendwie schon, denn mehrere Licht-Verteilungen, die überlappen und jeweils hohe Lichtstärken in Achsrichtung haben, können den Streifen mit dem Auge homogen erscheinen lassen. Es wurde eine "1" vergeben und die Begründung als Kommentar hinterlegt. Ähnlich verlief die Diskussion bei der Beziehung zum Streifenmuster.

! Der Moderator muss bei endlosen Diskussionen unbedingt eingreifen. Im Zweifelsfall muss er die Diskussion beenden und den Stand der Diskussion als Kommentar vermerken. So kann, falls nötig, später der Faden wieder aufgenommen werden.

Wichtig sind vor allem die starken Beziehungen ("9"), da diese die endgültige Bewertung des Produktmerkmals deutlich verändern. So geht man Spalte für Spalte vor, bis die Matrix gefüllt ist. Bild 1 zeigt die vollständig ausgefüllte Matrix.

Check: Was haben wir vergessen, was ist überflüssig?

Eine Kurzcheck wird folgendermaßen durchgeführt: Gibt es eine Kundenanforderung (Zeile), die durch kein Merkmal stark oder zumindest mittel unterstützt wird? Dann ist die Liste der Produktmerkmale nicht komplett. Gibt

Viele Produktmerkmale haben nichts miteinander zu tun. Es geht also nicht darum, jedem Feld mit Gewalt eine positive oder negative Korrelation zuzuordnen. Der wirklich wichtige Aspekt im Dach ist das Aufzeigen stark positiver und stark negativer Wechselwirkung. Starke negative Korrelationen können auf technische Grenzbereiche hinweisen. Eine starke positive Wechselwirkung zeigt Chancen für die Produktentwicklung auf. Eine leicht negative Wechselwirkung heißt nicht, dass die verglichenen Merkmale nicht umgesetzt werden können. Es bedeutet lediglich, dass bei der Umsetzung besonders darauf geachtet werden muss, dass die Werte in Summe kundengerecht spezifiziert werden.

! Als Moderator müssen Sie bei diesem Schritt ganz besonders darauf achten, dass sich das Team nicht in technische Detaildiskussionen verliert.

Das typische QFD-Satteldach, wie es die Bilder in Teil 1 diese Beitrags zeigen, ist ohne Programmieraufwand in Excel nicht darstellbar. Man kann den Vergleich aber genauso gut in einem Pultdach durchführen. Bild 1 zeigt das Haus mit Pultdach. Um die Wechselbeziehungen leicht identifizieren zu können, ist es bei dieser Darstellung empfehlenswert, die Produktmerkmale entlang der "Dachschräge" erneut aufzutragen.

Bei unserem Beispiel korrelierte das Merkmal "Stromaufnahme" mit Optimierungsrichtung "niedrig" stark positiv mit langer "Lebensdauer" und "keine zusätzliche Kühlmaßnahme". Solche Synergien liefern gute Verkaufsargumente für ein Produkt.

Leider gab es zwei stark negative Korrelationen "Abstrahlwinkel < 6°" und "keine sichtbaren Farbschwankungen im Lichtstreifen", die bei "Farbe weiß" auftreten. Eine technische Erklärung zum Verständnis: Eine weiß leuchtende LED besteht aus einem blauen LED-Chip, der von einem Konverter bedeckt ist. Dieser wandelt einen Teil des blauen Lichts zu einem Großteil in gelbes Licht um. Die Mischung ergibt weißes Licht. Leider fallen die räumlichen Ursprünge des konvertierten Lichts und des blauen Lichts nicht zusammen. Optische Systeme bilden beide Anteile deshalb leicht verschieden ab. Vor allem eng fokussierende Optiken spalten das LED-Licht in einen eher kaltweißen bis bläulichen Strahl und einen warmweißen bis gelblichen Strahl auf, der im Vergleich zum ersten auch noch "schmutzig" aussieht. Ein Streiflicht ist ein dekoratives Lichtelement und darin schmutzig aussehende Streifen zu finden, ist sehr unschön.

Gibt es stark negative Korrelationen, dann ist das gesamte QFD-Team gefordert: Kann durch technische Maßnahmen die negative Korrelation beseitigt werden? Kann das Produkt auch auf dem Markt platziert werden, wenn es weniger Anforderungen erfüllt? Oft ist hier eine Eskalation an den Auftraggeber erforderlich.

Technischer Wettbewerbsvergleich

Diesen Schritt beschränkten wir auf die Untersuchung der optischen Eigenschaften, da die Wettbewerbsprodukte nur zum Teil für diese Anwendung ausgelegt waren. Die Skala für die Beurteilung der Wettbewerbsprodukte spannt sich wieder von 1="schlechter" bis 5="besser" als die definierten Zielwerte der Produktmerkmale. Bei diesem Schritt wurde uns klar, dass ein enger Abstrahlwinkel (< 6°) von keinem der Wettbewerbsprodukte realisiert wird, obwohl dieses Merkmal hoch priorisierte Kundenanforderungen stark unterstützt. Dies war also für das neue Produkt die wichtigste Chance im Wettbewerb – andere Eigenschaften dürften deshalb getrost in den Hintergrund treten.

Schwierigkeit der Umsetzung

Diese Spalte gibt eine Einschätzung, wie schwierig die Umsetzung des jeweiligen Produktmerkmals ist. Die Schwierigkeit wird abgeschätzt anhand von Parametern wie benötigte Entwicklungszeit und entstehende Kosten z.B. durch Formen- oder Werkzeugbau. Es wird üblicherweise eine Skala von 1="leicht umzusetzen" bis 10="schwer umzusetzen" vergeben. Wir verwendeten hier lediglich die 1="leicht umzusetzen", 5="mittelschwer umzusetzen" und 10="schwierig umzusetzen", um unnötige Detaildiskussionen zu vermeiden.

In unserem Beispiel gab es zwei kritische Merkmale. Zum einen schätzten wir den Aufwand für das optische Design sehr hoch ein, um den engen Abstrahlwinkel für alle Farben inkl. weiß zu erreichen. Zusätzlich musste dafür ein Werkzeug gebaut werden.

Zum anderen erhielt das Merkmal "Farbschwankungen über die Streifenbreite" zwei Bewertungen. Dies war dem Vorgehen geschuldet, mehrere Varianten in einem HoQ zu betrachten (rot, blau, grün und weiß). Farbschwankungen bei einem blau, rot oder grün leuchtenden Modul zu vermeiden ist einfacher (Schwierigkeitsgrad 5) als bei einem "weißen" Modul (Schwierigkeitsgrad 10).

Bedeutung der Produktmerkmale aus Kundensicht

Nachdem die Beziehungsmatrix komplett gefüllt ist, wird die Gesamtbewertung der Produktmerkmale aus Kundensicht numerisch berechnet. Dazu wird jeweils die Gesamtbedeutung (siehe Spalte "Gesamt" in Bild 1) der Kundenanforderung mit dem Korrelationswert aus der Matrix multipliziert und dem jeweiligen Produktmerkmal zugewiesen. In jeder Spalte wird dann die Summe dieser Produkte addiert. Dies ergibt die Gesamtbedeutung jedes Produktmerkmals in der Zeile "absolute Bedeutung" (Bild 1). Die Zeile "relative Bedeutung" stellt die Gesamtbedeutung in Prozent dar, berechnet aus der absoluten Bedeutung geteilt durch die Summe aller absoluten Anteile.

! Es wäre ein grober Fehler, die so erhaltenen gewichteten Produktmerkmale automatisch in dieser Priorisierung in die Produktentwicklung zu übernehmen. QFD ist kein stumpfer Algorithmus, sondern ein leistungsfähiges Werkzeug, um fundierte Entscheidungen für Produktspezifikationen zu treffen. Hierzu müssen Sie stets das gesamte Haus betrachten.

Das Merkmal mit der höchsten Bewertung "Abstrahlwinkel Längsachse < 6°" war technisch schwierig umzusetzen und hatte eine starke negative Korrelation mit dem wichtigen Merkmal "Farbtemperatur warm-weiß". Die optische Bündelung des Lichts spaltet es in die zwei Anteile auf (s.o.). Eine technische Lösung hierfür zu finden ist Aufgabe des Entwicklungsteams, das dann auch für die endgültige Entscheidung Abschätzungen über Kosten und Entwicklungsdauer geben muss.

Für die Bewertung und die letztendliche Entscheidung ist jedoch der Auftraggeber zuständig. Wir forderten hier deshalb die Beteiligung der Unternehmensführung beim Review und der Produktentscheidung ein: Dies ist der letzte Schritt im QFD-Prozess, um zur endgültigen Produktspezifikation zu kommen.

Review, Auswahl, Entscheidung

Alle nötigen Informationen und das Wissen aller am QFD-Projekt Beteiligten waren nun im HoQ vorhanden und verarbeitet. Das Haus ist dabei als Landkarte zu verstehen, aus der nun Wege für das weitere Vorgehen der Pro-

duktentwicklung abgeleitet werden. Die jetzt ausgewählten Produktmerkmale gehen in die Umsetzung ein. Dies ist somit der wichtigste Schritt in einem QFD-Prozess.

Die höchste Bedeutung hatte in unserem Beispiel das Merkmal "Abstrahlwinkel < 6°". Es unterstützte stark die hoch priorisierten Kundenanforderungen "Lichtstreifen variabler Breite" und "Streifen gut sichtbar über die Raumhöhe" und mittelstark den "homogenen Lichtstreifen". Auf zwei dieser Kundenanforderungen lag ein Marketingschwerpunkt, gleichzeitig war der Wettbewerb dort schlecht aufgestellt. Leider gab es die technische Schwierigkeit bei der Lichtfarbe weiß. Wir diskutierten die von den Entwicklern vorgeschlagenen Lösungsmöglichkeiten. Die Wahl war, entweder einen "schmutzig" aussehenden, weißen Lichtstreifen oder zeit- und kostenintensive Lösungen zu akzeptieren.

Strategische Entscheidung: Nur blaue, rot und grüne LED-Module!

Unter Berücksichtigung der Umsatz- und Gewinnplanung traf das Management auf Basis dieser Informationen eine klare Entscheidung: Es sollte kein weiß leuchtendes Modul hergestellt werden! Die zur Verwendung stehende Spezialoptik aus einer anderen Anwendung sollte für die farbigen Module eingesetzt werden. Mit dieser Optik konnte die wichtigste Anforderung "Abstrahlwinkel < 6°" sehr gut erfüllt werden und sie unterstützte die zweitwichtigste Anforderung "Lichtstärke > 1000 cd".

Dies war eine richtungsweisende Entscheidung mit klarer Begründung, die nur das Management treffen konnte. Wir hielten sie in der Dokumentation des QFD-Projekts fest, so dass auch zu einem späteren Zeitpunkt klar ist, warum so gehandelt wurde.

Bei einer anderen identifizierten Schwierigkeit war die Lösung weitaus weniger dramatisch: Es gab eine leicht negative Korrelation zwischen "Lichtstärke in Achsrichtung", "geringe Stromaufnahme", "Lebensdauer" und "keine zusätzliche Kühlmaßnahme". "Keine zusätzliche Kühlmaßnahme" hatte eine niedrige Gesamtbewertung von 2% und unterstützte keine Kundenanforderung stark. Auf diese Eigenschaft hätten wir notfalls verzichtet, um die höher bewerteten Merkmale Lichtstärke, Stromaufnahme und Lebensdauer kundengerecht erfüllen zu können.

Fazit: QFD gewährleistet Effizienz und Effektivität

Wir definierten im Konsens mit allen Bereichen zielorientiert ein Produkt, hinter dem nicht nur das Entwicklungsteam, sondern auch der Vertrieb und die Unternehmensleitung standen. Das komplette Wissen des Teams war im HoQ dokumentiert.

Ja, es war Aufwand, aber alle Themen hätten ohnehin zu irgendeinem Zeitpunkt untersucht werden müssen. Jetzt wurden sie strukturiert im QFD Prozess betrachtet. Das entstehende Produkt erfüllte die Kundenerwartungen und passte zur Strategie des Unternehmens: Die Lichtleisten zaubern noch heute bunte Streifen an unzählige triste Wände.

Vielleicht haben Sie in anderen "Qualitäts-Häusern" weitere Spalten und Zeilen gesehen. Hier möchte ich auf die Philosophie des Erfinders Akao verweisen: "Copy the spirit and not the form". Es steht jedem Team frei, weitere Informationen im Markt- oder Technikblock einzutragen, wenn diese für die Landkarte der Produktdefinition wichtig sind.

Literatur

- Saatweber, Jutta: Kundenorientierung durch Quality Function Deployment, Symposion Publishing GmbH, Düsseldorf, 2011

Tipp

Was ist am wichtigsten? Anforderungen mit "MoSCoW" priorisieren

Ist zu Projektbeginn eine genaue Aufwandsschätzung möglich, können Auftraggeber und Auftragnehmer den Leistungsumfang im Rahmen der Projektplanung festlegen. Lassen sich die Aufwände und Dauern für die Realisierung der Anforderungen hingegen nicht genau schätzen, sollte der Auftraggeber die Anforderungen sorgfältig priorisieren. Denn nur so lässt sich erkennen, welche Anforderungen unbedingt notwendig sind, um das gewünschte Projektergebnis zu erreichen, welche verhandelbar und welche lediglich Nice-to-Haves sind. Eine solche Priorisierung garantiert, dass das Projektteam selbst bei Problemen und Verzögerungen im Projektverlauf zumindest die wichtigsten Anforderungen liefert.

Autor



Marko Zotschew

Dipl.-Ing. (FH), PRINCE2
Foundation, Redakteur
beim Projekt Magazin

Kontakt: mz@projektmagazin.de

Mehr Informationen unter:

› projektmagazin.de/autoren

Doch wie kann eine gute Priorisierung in der Praxis gelingen? Ein einfaches Vorgehen hierfür ist die sog. MoSCoW-Methode.

Was bedeutet "MoSCoW"?

Die MoSCoW-Priorisierung ist eine vierstufige Prioritätenskala zur Bewertung von Anforderungen (Angermeier, 2014). Ein typischer Anwendungsbereich für diese Methode sind agile Projekte, in denen Anforderungen priorisiert werden und somit die Reihenfolge ihrer Abarbeitung feststeht. Auch PRINCE2 empfiehlt diese Methode, um offene Punkte, d.h. meistens Änderungsanträge, zu priorisieren. Darüber hinaus lässt sich das Vorgehen auch für die eigene Aufgabenplanung verwenden, um z.B. bei einer großen Fülle an Aufgaben Struktur in deren Abarbeitung zu bringen.

Der Begriff "MoSCoW" ist ein Akronym und bezeichnet eine vierstufige Priorisierungsskala (vgl. Angermeier, 2014 und Wikipedia):

- M - MUST (Die Umsetzung ist für die Abnahme zwingend erforderlich.)
- S - SHOULD (Anforderungen müssen ebenfalls umgesetzt werden, sind aber im Gegensatz zu den MUST-Anforderungen durch Change Requests oder Verhandlungen veränderbar.)
- C - COULD (Diese Anforderungen werden umgesetzt, wenn alle Must- und Should-Anforderungen erfüllt sind und noch ausreichend Ressourcen und Zeit zur Verfügung stehen.)
- W - WON'T (Diese Anforderungen werden im aktuellen Projekt noch nicht umgesetzt und stattdessen in einem Ideenpool oder der Anforderungsliste für das nächste Projekt gespeichert)

Vorteil der Methode

Das Besondere an der MoSCoW-Methode ist, dass sie auch Anforderungen dauerhaft festhält, die im aktuellen Leistungsumfang nicht umgesetzt werden können. Dies ist besonders für den Auftraggeber interessant, da dieser aus Angst, dass Features verloren gehen, möglichst alle Features hoch priorisieren möchte; dadurch fällt eine klare Priorisierung in sehr wichtige, durchschnittlich wichtige und weniger wichtige Anforderungen häufig schwer. Werden jedoch weniger wichtige Anforderungen in einem Ideenpool festgehalten und ist dies dem Auftraggeber auch bewusst, fällt die Priorisierung leichter (vgl. Herwarth von Bittenfeld, 2011). Auf diese Weise ermöglicht die MoSCoW-Methode, dass die wirklich entscheidenden Dinge in jedem Fall umgesetzt werden.

Must: Keine Kompromisse!

Die Umsetzung von Anforderungen mit der Priorität "Must" ist Mindestvoraussetzung für die Abnahme des Projektergebnisses. Die Nicht-Umsetzung führt zwangsläufig zum Scheitern des Projekts. Um festlegen zu können, ob es sich bei einem Feature um ein "Must" handelt, benötigt man Entscheidungskriterien. Diese können z.B. sein (s. Kubitz, 2012 oder Herwarth von Bittenfeld, 2011):

- **Ist das Produkt ohne dieses Feature funktionsfähig?** Beispiel: Bei einer Logistiksoftware ist die Disposition des Fuhrparks nicht möglich.
- **Ist ohne dieses Feature die Kundenzufriedenheit gewährleistet?** Beispiel: Bei der Logistiksoftware ist zwar der Fuhrpark disponierbar, aber es gibt keine automatische Vorschlagsfunktion für geeignete Fahrzeuge.
- **Erreicht das Produkt ohne dieses Feature die notwendige Qualität?** Beispiel: Die Antwortzeiten der Logistiksoftware für Einbuchungen ist zu lang, sodass ein effizientes Disponieren nicht möglich ist.

Lassen sich diese Fragen mit "Nein" beantworten, deutet dies auf eine Must-Anforderung hin.

Besonders der Punkt der Kundenzufriedenheit lässt viel Spielraum zu, da aus Sicht des Auftraggebers generell jedes Feature der Kundenzufriedenheit dient. Wird die Frage zur Kundenzufriedenheit mit "Nein" beantwortet, sollte man sich deshalb genau vor Augen halten, welcher "Schaden" dem späteren Nutzer tatsächlich entsteht, wenn er dieses Feature nicht bekommt. Sollte sich herausstellen, dass dieser Schaden zunächst vertretbar wäre, ist hier evtl. auch eine Einordnung unter "Should" sinnvoll. Die Einsortierung unter "Should" bedeutet schließlich nicht, dass das Feature nicht umgesetzt wird.

Should: Darüber können wir nochmal reden

Should-Anforderungen haben einen hohen Nutzen und der Kunde erwartet auch deren Umsetzung. Sie werden in der Projektplanung vollständig berücksichtigt und bilden mit den Must-Anforderungen den gesamten Leistungsumfang.

! Kommt es jedoch im Projekt zu massiven Problemen und ist es absehbar, dass sich der gesamte Leistungsumfang (Must- und Should-Anforderungen) nicht innerhalb des gesetzten Budget- und Zeitrahmens realisieren lässt, treten die Should-Anforderungen gegenüber den Must-Anforderungen zurück. Das bedeutet im Umkehrschluss: Das Projekt wird auch dann abgenommen, wenn nicht alle Should-Anforderungen erfüllt sind.

Während Must-Anforderungen entscheidend für die Projektabnahme sind, können Should-Anforderungen in Abstimmung zwischen Auftraggeber und Auftragnehmer evtl. zu einem späteren Zeitpunkt nachgeliefert werden, vereinfacht werden oder mit Genehmigung des Auftraggebers sogar ganz entfallen (Angermeier, 2014).

Could: Wünschenswert, aber es geht auch ohne

Could-Anforderungen lassen sich auch mit "Nice to have" umschreiben. Sie sind nicht Bestandteil des Leistungsumfangs (Angermeier, 2014). Eine Umsetzung erfolgt erst nach der Realisierung der Must- sowie Should-Anforderungen und wenn Zeit bzw. Budget dies auch zulassen.

Sollte die Umsetzung von Could-Anforderungen im Rahmen des Projekts noch möglich sein, ist es sinnvoll, das Augenmerk auf diejenigen Features zu legen, die kurzfristig umzusetzen sind und einen hohen Mehrwert bringen.

Beispiel

Zwei Wochen vor dem Endtermin stehen dem Team nach Umsetzung der Must- und Should-Anforderungen noch zehn Arbeitstage zur Verfügung. In einem kurzen Workshop betrachten Kunde und Vertreter des Entwicklerteams die einzelnen Could-Anforderungen. Dabei schätzt der Kunde (z.B. auf einer Skala von 1 bis 10) den Nutzen der Features und das Team den Aufwand in Teamtagen (Bild 1).

Feature 1 hätte demnach aus Kundensicht den höchsten Nutzen, der Arbeitsaufwand liegt aber mit acht Tagen auch entsprechend hoch; nur noch Feature 4 mit zwei Arbeitstagen ließe sich dann umsetzen. Gemeinsam kommen beide Features nach Einschätzung des Auftraggebers auf einen Nutzwert von 11. Die Features 2, 3 und 5 ließen sich ebenfalls im möglichen Zeitraum umsetzen, erreichen aber einen Wert von 18. Somit ist die Umsetzung dieser drei Features gegenüber der ersten Option vorzuziehen, da der Gesamtnutzen überwiegt.

Feature	Arbeitsaufwand in Tagen	Nutzen
Feature 1	8 T	9
Feature 2	3 T	5
Feature 3	4 T	7
Feature 4	2 T	2
Feature 5	3 T	6
Feature 6	3 T	3

Bild 1: Die Umsetzung der Could-Anforderungen nach Aufwand und Nutzen priorisieren.

Won't: Noch nicht!

Unter Won't-Anforderungen versteht man diejenigen Features, die sich im Hinblick auf Termin und Budget nicht im aktuellen Leistungsumfang unterbringen lassen. Sie können aber für Folgeaufträge in Betracht kommen und dienen somit als Ideenspeicher für weitere Projekte. Darüber hinaus dient die Klassifizierung einer Anforderung als "Won't" auch dazu, den Teammitgliedern die Abgrenzung des Leistungsumfangs deutlich zu machen.

Auf den ersten Blick könnte man auf die Klassifizierung von "Won't" verzichten; denn wenn Anforderungen nicht im Leistungsumfang des Projekts enthalten sind, werden sie ja auch nicht benötigt. Dennoch ist es sinnvoll, die Anforderungen in dieser Kategorie zu dokumentieren. Die Won't-Liste vervollständigt das Gesamtbild der initial definierten Anforderungen und verhindert, dass Features in Vergessenheit geraten, was besonders den Kunden interessiert.

Die 60%-Regel

Häufig fällt es schwer, das richtige Maß zu finden, wie viele Anforderungen nun unter Must, Should usw. einzuordnen sind. Grundsätzlich gilt, dass eine Must-Anforderung nur seine Bedeutung gewinnt, wenn es auch genügend Should- und Could-Anforderungen gibt. Als Orientierung hierfür kann die sog. 60%-Regel dienen (s. auch Fischbach, 2012), wobei diese Anhaltspunkte je nach Branche und/oder Projektart variieren können:

- Must: max. 60% des Gesamtaufwands,
- Should: ca. 20% des Gesamtaufwands,
- Could: ca. 20% des Gesamtaufwands,
- Won't: gehören nicht zum Umfang

Überschreiten die Must-Anforderungen die 60%-Marke, hat der Kunde nun zwei Möglichkeiten:

1. Es kommt zu Nachverhandlungen mit dem Auftragnehmer, um den Projektumfang auszuweiten.
2. Der Auftraggeber muss für sich abwägen, welche Anforderungen wirklich in die Kategorie "Must" gehören. Helfen können hierbei eine klare Zielformulierung zu Projektbeginn (s. hierzu auch "[Projektziele ermitteln, beschreiben und einordnen](#)", Projekt Magazin 15/2012) sowie das Wissen, dass ein Should-Feature mit hoher Wahrscheinlichkeit auch umgesetzt wird.

Dieses Vorgehen ist zugleich ein gutes Steuerungsinstrument für den Auftragnehmer, um den Projektumfang nicht ausufern zu lassen und dem Kunden zu zeigen, was mit vereinbartem Budget und Termin tatsächlich machbar ist.

Literatur und weiterführende Links

- Angermeier, Georg: [MoSCoW](#), Projekt Magazin Glossar, 2014
- Fischbach, Jan: [Wie Teams mit der MoSCoW-Priorisierung ihre Arbeit planen](#), 2012, zuletzt eingesehen am 02.04.2014
- Herwarth von Bittenfeld, Paul: [Priorisierung nach der MoSCoW-Methode](#), 2011, zuletzt eingesehen am 02.04.2014
- Kubitz, Eric: [Priorisierung für SEO-Projekte mit der MoSCoW-Methode](#), 2012, zuletzt eingesehen am 02.04.2014
- Peterjohann, Horst: [Projektziele ermitteln, beschreiben und einordnen](#), Projekt Magazin 15/2012
- Wikipedia: [MoSCoW-Priorisierung](#), zuletzt eingesehen am 02.04.2014

Fachbeitrag

Pareto-Prinzip für IT-Projekte

Die Anforderungen auf das Machbare reduzieren

Viele Projektmanager aus der IT kennen dieses Problem:

Knappe Ressourcen einerseits, zahlreiche Anforderungen andererseits. Ein Mobilfunkanbieter mit über 120 Millionen Kunden weltweit stand 2007 ebenfalls vor dem Problem, dass die zahlreichen Anforderungen, welche die Abteilungen an die Neu- und Weiterentwicklung der internen IT-Systeme stellten, mit dem vorhanden Budget nicht umgesetzt werden konnten. Unter dem Schlagwort "Rightsizing" wurde deshalb ein Verfahren entwickelt, mit dem Projektmanager und Auftraggeber den Projektinhalt und –umfang an die Ressourcensituation anpassen können. Das Verfahren basiert auf der so genannten Pareto-Regel und wird im Unternehmen als Pareto-Analyse bezeichnet. Diese Analyse wird im Folgenden vorgestellt.

Autor



Tim Krüger

Diplom-Kaufmann, Senior Consultant bei der 7P Consulting GmbH, Spezialist für IT-Projektmanagement, PMP

Kontakt:

tim.krueger@7p-group.com

Mehr Informationen unter:

projektmagazin.de/autoren

Ausgangslage und Problemstellung

Anfang 2007 nahmen die Controller der IT-Abteilung eine Auswertung vor. Dabei stellten sie fest, dass die Anzahl der neuen Anforderungen, die an die internen IT-Systeme gestellt wurden (Demands), kontinuierlich gestiegen war. Die Zahl der Demands, die im Januar 2007 bereits erfasst waren, ließ auf das gesamte Jahr fortgeschrieben eine weitere, extreme Erhöhung erwarten. Gleichzeitig hatte das Top-Management das Budget für die Neu- und Weiterentwicklung von IT-Systemen gekürzt. Es war somit abzusehen, dass die IT-Abteilung im Jahr 2007 nur einen sehr kleinen Anteil der Demands würde umsetzen können (Bild 1).

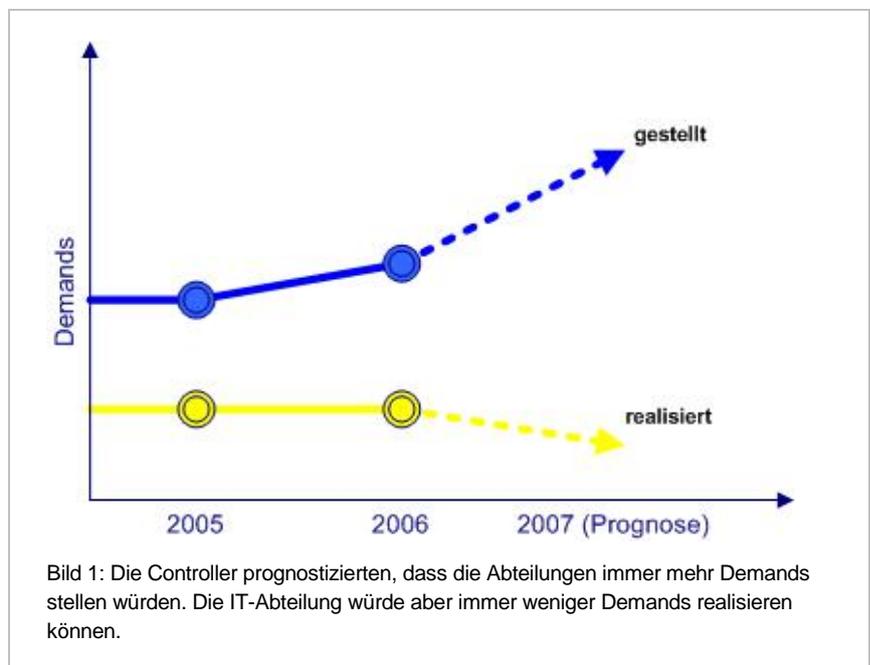


Bild 1: Die Controller prognostizierten, dass die Abteilungen immer mehr Demands stellen würden. Die IT-Abteilung würde aber immer weniger Demands realisieren können.

Die Controller der IT standen vor der Aufgabe, die reduzierten Budget- und Personalressourcen und die gestiegenen Anforderungen in Einklang zu bringen und dabei ein möglichst effizientes Ergebnis zu erzielen. Dafür mussten sie ein Verfahren finden, mit dem sich der Umfang einzelner Demands reduzieren ließ, z.B. indem ein-

zelne Funktionsanforderungen fallen gelassen wurden. Dieses Verfahren sollte leicht anzuwenden sein und die Projektteams möglichst wenig belasten.

Als Lösung wurde schließlich ein Verfahren auf Basis der so genannten Pareto-Regel entwickelt und in der deutschen Landesgesellschaft des Mobilfunkanbieters eingeführt. Mit diesem Verfahren ist es möglich, die wichtigsten Funktionsanforderungen von Demands auszuwählen und diese auch umzusetzen. Weniger nutzbringende Funktionsanforderungen werden aussortiert.

Die Pareto-Regel

Die Pareto-Regel, auch 80/20-Regel genannt, ist nach dem italienischen Ökonomen und Soziologen Vilfredo Pareto (1848-1923) benannt. Pareto fand heraus, dass 20% der italienischen Familien über 80% des italienischen Volkvermögens verfügten. Der Ingenieur Joseph M. Juran (*1904) wandte dieses Prinzip auf das Qualitätsmanagement an und prägte den Ausdruck "vital few, useful many". Gemeint sind Sachverhalte wie

- 20% der Kunden bringen einem Unternehmen 80% des Umsatzes.
- 80% der Fehler einer Software sind auf dieselben 20% der Ursachen zurückzuführen.
- In 20% seiner Arbeitszeit erreicht ein Mitarbeiter 80% seiner Ergebnisse.
- usw.

Dabei sind die Zahlen 80 und 20 keine starren Werte. Vielmehr ist gemeint, dass oft ein kleiner Anteil der Ursachen den größten Teil der Wirkung erzeugt. Bei Ressourcenknappheit ist es deshalb sinnvoll, diesen kleinen, wirkungsvollen Anteil zu identifizieren und hierauf die Energien zu konzentrieren. Auf diese Weise lässt sich die Effizienz beträchtlich steigern.

Einbettung in das Vorgehensmodell

Beim Mobilfunkanbieter werden Softwareentwicklungsprojekte nach einem intern entwickelten Vorgehensmodell durchgeführt. Ausgangspunkt ist immer eine Anforderung (Demand), die beispielsweise von einer Fachabteilung (Marketing, Kundenbetreuung usw.) gestellt wird. Ein Demand wird u.a. dann eingebracht, wenn neue Geschäftsprozesse eingeführt und durch die IT unterstützt werden sollen oder wenn es darum geht, bereits bestehende, IT-gestützte Geschäftsprozesse zu erweitern, zu verändern oder zu optimieren. Eine Anforderung kann auch technikgetrieben sein, z.B. um die IT-Architektur zu vereinheitlichen oder Geschäftsprozesse besser zu unterstützen.

Um einen Demand umzusetzen, müssen in einem bzw. in mehreren IT-Systemen Veränderungen oder Ergänzungen vorgenommen werden. Im Unternehmen sind knapp 100 IT-Systeme im Einsatz, die zudem stark miteinander vernetzt sind. Deshalb betrifft eine technische Lösung für eine Anforderung in der Regel mehrere Systeme, so dass ein hoher Koordinations- und Steuerungsaufwand für das Entwicklungsprojekt entsteht. In diesem Fall wird für das Projekt ein eigener Projektmanager abgestellt. Betrifft eine Lösung nur ein einziges System, wird sie unter der Leitung des Systemverantwortlichen erarbeitet.

Die Anforderer (z.B. Fachabteilungen) erfassen alle Demands in einer unternehmensintern entwickelten Software. Jeder Demand muss mehrere Phasen durchlaufen. Dazu gehören "Erstellung eines Fachkonzepts", "Er-

stellung der Spezifikation" und "Realisierung". Vor jeder Phase schätzt der Projektmanager den Aufwand dafür (z.B. Aufwand für die Erstellung eines Fachkonzepts). Auf Basis dieser Schätzung entscheidet das Gremium, das für die Releaseplanung zuständig ist, ob ein Demand in die nächste Phase gehen darf. Bei einem Ressourcenengpass werden einzelne Demands aussortiert und nicht weiter verfolgt. Im Gremium für die Releaseplanung sind die Bereichsleiter des IT-Controllings sowie aller Fachseiten vertreten.

Feasibility Study

Gibt es bei einem Demand mehrere, konkurrierende Ansätze für eine Lösung oder ist unklar, ob eine Anforderung überhaupt realisierbar ist, erstellt der Projektmanager zunächst eine Vorstudie (Feasibility Study). Ziel ist es hierbei, Lösungen zu identifizieren und zu bewerten und eine Entscheidungsgrundlage für das weitere Vorgehen zu liefern. Steht fest, welche Lösung umgesetzt werden soll, folgt die Planung.

Fachkonzept

Für die Planung erstellt der Projektmanager ein Fachkonzept (Detailed Solution Design). Es führt die fachlichen Anforderungen vollständig und konsistent auf und benennt alle betroffenen Systeme sowie Abhängigkeiten zu anderen Projekten.

Technische Spezifikation

Genehmigen der Auftraggeber und die Betreuer der betroffenen Systeme das Fachkonzept, fertigen die Analysten jedes beteiligten Systems eine genaue technische Spezifikation der Lösung an (System Design). Auch diese Dokumente müssen vom Auftraggeber freigegeben werden.

Entscheidung über Realisierung

Ist das Lösungsdesign abgeschlossen, können die Analysten den Aufwand für die Lösung schätzen. Auf Grundlage dieser Aufwandsschätzung entscheidet das für die Releaseplanung zuständige Gremium über die Umsetzung der Demands.

Die weiteren Phasen des Vorgehensmodells sind die Realisierung (Realisation), die System- und Verbundtests (Test & Acceptance) und schließlich die Inbetriebnahme (Deployment).

Integration der Pareto-Regel

Das Vorgehensmodell wurde um die Pareto-Regel erweitert. Wird eine Feasibility Study erstellt, sieht das Vorgehensmodell am Ende der Studie eine Kosten-Nutzen-Analyse auf Basis der Pareto-Regel vor. Diese Analyse wird im Unternehmen als Pareto-Analyse bezeichnet. Zu diesem Zeitpunkt steht bereits fest, welcher der möglichen Lösungsansätze empfohlen werden kann. Ist keine Feasibility Study notwendig, sondern wird gleich ein Fachkonzept entworfen, muss der Anforderungsumfang gleich zu Beginn dieser Phase analysiert werden. Um kleinere Projekte nicht unnötig mit Verwaltungsaufwand zu belasten, wurde beschlossen, dass die Pareto-Analyse nur bei Projekten mit einem Gesamtaufwand von über 100 Personentagen obligatorisch ist.

Ablauf der Pareto-Analyse

Die Pareto-Analyse ist als formaler Prozess definiert.

1. Schritt: Teilanforderungen definieren

Als erstes wird der Gesamtumfang des Projekts in einzelne Teilanforderungen (Sub-Demands) unterteilt. Diese Sub-Demands decken sich oft mit den Funktionsanforderungen, manchmal sind sie aber auch unterschiedlich. Wichtig ist, dass sich jede Teilanforderung einzeln bewerten, implementieren und einführen lässt – andernfalls ist es später nicht möglich, einzelne Teilanforderungen von der Umsetzung auszuschließen. Sub-Demands sind z.B. Varianten eines Geschäftsprozesses, die durch ein IT-System unterstützt werden sollen.

Die Aufteilung übernimmt der Projektmanager in Abstimmung mit dem Auftraggeber (z.B. Mitarbeiter einer Fachabteilung). Die Unterteilung ist mit einem Projektstrukturplan vergleichbar: Sie sollte an den Liefergegenständen des Projekts ausgerichtet sein und den Projektumfang zu 100% abdecken.

2. Schritt: Bewertung

Im zweiten Schritt erfolgt die eigentliche Bewertung: Der Projektmanager bewertet die Kostenanteile, der Auftraggeber die Nutzenanteile jedes einzelnen Sub-Demands im Verhältnis zum gesamten Demand. Unter "Nutzen" wird dabei ein betriebswirtschaftlicher Nutzen wie z.B. Kosteneinsparung oder Risikoverringerung verstanden, nicht etwa die persönliche Präferenz des Auftraggebers. Die Bewertungen der Projektmanager und Auftraggeber erfolgen nach keinem festen Verfahren, sondern beruhen auf Einschätzungen. Um eine belastbare Kosteneinschätzung abgeben zu können, befragen die Projektmanager oft Mitarbeiter und Experten im Unternehmen.

Die Bewertung erfolgt in einer Prozentskala, wobei die Summe aller Kosten- bzw. Nutzenbewertungen 100% ergeben muss. Pflichtanforderungen erhalten keine Nutzenbewertung (Tabelle 1). Entscheidend ist, dass Projektmanager und Auftraggeber getrennt und unabhängig voneinander ihre Bewertungen vornehmen, damit eine möglichst hohe Objektivität gewährleistet ist.

3. Schritt: Vergleich

Im dritten Schritt werden die Sub-Demands anhand des ermittelten Kosten-Nutzen-Verhältnisses verglichen. Hier kommt das Pareto-Prinzip zum Tragen: Es gilt, die Sub-Demands zu identifizieren, die nur wenig Nutzen, aber den Großteil der Kosten ausmachen. Diese Sub-Demands werden aus dem Projektumfang entfernt.

Projektmanager und Auftraggeber legen in einem Gespräch Inhalt und Umfang des Projekts fest. Sie gehen die Liste durch und klassifizieren die einzelnen Sub-Demands entweder als "in scope" oder als "out of scope". Das Ergebnis wird als formlose Liste schriftlich festgehalten und als Anhang zum Fachkonzept archiviert. In der Pareto-Analyse aus Tabelle 1 beispielsweise wurden die Sub-Demands 3B und 5B aussortiert, d.h. hier wurde eine "39/2-Regel" verwendet. Es gibt keine verbindlichen Kriterien, nach denen die Sub-Demands aussortiert werden müssen. Diese Entscheidung liegt im Ermessen des Projektmanagers und des Auftraggebers. Grundsätzlich versteht sich der Projektmanager bei der Bewertung der Sub-Demands als strategischer Berater des Auftraggebers.

#	Requirement / Component	Costs [%]	Benefit [%]
1E	E2E Monitoring: Distributed SIM cards / partner channel	4%	Mandatory
1I	E2E Monitoring: Distributed SIM cards / partner channel / module	2%	8%
1J	Replacement of SIM cards	7%	4%
2A 2B	...	25%	Mandatory
2C	...	0%	1%
3A	...	4%	15%
3B	...	35%	1%
4A	...	1%	20%
5A	...	9%	8%
5B	...	4%	1%
6A 6B	...	4%	10%
6C 6D	...	1%	12%
6E	...	2%	10%
6F 6G	...	2%	10%

Tabelle 1: Beispiel für eine Pareto-Bewertung.

Wurde die Pareto-Analyse am Ende einer Feasibility Study durchgeführt, ist der festgelegte Projektumfang für das nachfolgende Fachkonzept (Detailed Solution Design) verbindlich. Wurde das Projekt gleich mit einem Fachkonzept gestartet, wird nach der Pareto-Analyse nur noch der entsprechend reduzierte Projektumfang geplant. In beiden Fällen wird die gesamte Pareto-Analyse als separates Dokument abgelegt; im Ausgangsdokument (Feasibility Study oder Detailed Solution Design) wird ein entsprechender Verweis eingefügt.

Ergebnisse und Lessons Learned

In der täglichen Arbeit mit der Pareto-Analyse zeigten sich einige Verbesserungspotenziale. Für einige Punkte wurden bereits Lösungen gefunden, andere müssen derzeit noch für jeden Fall einzeln gelöst werden.

Alte Bekannte: Abgelehnte Sub-Demands werden erneut eingebracht

Projektmanager stellten fest, dass Sub-Demands, die bei der Analyse aussortiert worden waren, teilweise von den Auftraggebern als Anforderungen für spätere Releases erneut eingestellt wurden. Diese Praxis wurde vom Releaseplanungsgremium unterbunden. Ein abgelehnter Sub-Demand wird nur dann erneut zugelassen, wenn

vom Anforderer (z.B. der Fachabteilung) ein eigener Business Case für seine Realisierung dargelegt werden kann, der vom Gremium akzeptiert wird. Der Business Case muss belegen, dass der Sub-Demand eine eigenständige betriebswirtschaftliche Bedeutung hat, z.B. indem durch ihn Kosten gesenkt oder Kundenpotentiale erschlossen werden. Auf diese Weise wird verhindert, dass Projektmanager und Auftraggeber in jedem Projekt immer wieder über dasselbe Sub-Demand diskutieren.

Besondere Herausforderung: Unterteilung in Sub-Demands

Es zeigte sich, dass die Unterteilung in Sub-Demands eine besondere Herausforderung darstellt. Ist die Unterteilung zu grob, können viele kleine Features, die kostspielig, aber entbehrlich sind, nicht identifiziert und eliminiert werden. Ist die Unterteilung zu fein, wird es schwierig, eine Einigung zwischen Projektmanager und Auftraggeber zu erzielen. Außerdem steigt in diesem Fall der Aufwand für die Unterteilung stark an. Bisher gibt es kein Verfahren, mit dem die Unterteilung formalisiert durchgeführt werden kann. Projektmanager und Auftraggeber müssen die Sub-Demands deshalb jedes Mal im Gespräch festlegen.

Verzicht tut weh: Kooperation der Auftraggeber fördern

Obwohl die Fachabteilungen grundsätzlich die Einführung der Pareto-Analyse befürworteten, sind die Auftraggeber im Einzelfall nicht immer bereit, konkrete Anforderungen aufzugeben. Um die "Diskussionsbereitschaft" der Auftraggeber bei der Pareto-Analyse zu erhöhen, wurde eine zusätzliche Schwelle für die Umsetzung von Demands festgelegt: Das Releaseplanungsgremium lässt Projekte nicht zu, wenn diese die verfügbaren Entwicklerkapazitäten eines betroffenen Systems übermäßig belegen würden. Was konkret unter "übermäßig" zu verstehen ist, wird je nach Einzelfall entschieden. Um zu verhindern, dass der Demand vollständig abgelehnt wird, sind die Auftraggeber nun eher bereit, den Umfang mit Hilfe einer Pareto-Analyse zu reduzieren. So können sie wenigstens einen Teil der Anforderungen realisieren lassen.

Management-Attention: Präsentation vor Bereichsleitern

Um den Mitarbeitern die Aufmerksamkeit des Managements zu verdeutlichen, wurde ein regelmäßiger Präsentationstermin eingerichtet. Einmal im Quartal können Projektmanager hier ihre Projekte mitsamt der durchgeführten Pareto-Analyse auf Bereichsebene vorstellen. Die Projektmanager können darstellen, welchen Beitrag sie zur Realisierung der Demands geleistet haben und werden motiviert, die Pareto-Analyse regelmäßig und konsequent anzuwenden.

Ausblick: Ziele für die Zukunft

Um den Aufwand eines Projekts noch besser abbilden zu können, soll Schritt für Schritt auf eine Bewertung der Total Cost Of Ownership (TCO) umgestellt werden. Dies macht die Bewertung für den Projektmanager allerdings schwieriger und aufwändiger, da z.B. für die Ermittlung von Hardware- oder Betriebskosten weitere Spezialisten befragt werden müssen.

Fazit

Das Gesamturteil der Pareto-Analyse fällt positiv aus: Der Umfang von Projekten und damit ihr Aufwand konnte im Jahr 2007 im Vergleich zu 2006 um durchschnittlich 30% reduziert werden. Die Pareto-Analyse leistet einen wertvollen Beitrag zur effizienten Projektarbeit und hat sich im Arbeitsalltag bewährt.

Fachartikel

Die Produktbasierte Planung nach PRINCE2

Teil 1: Produktbeschreibung und Produktstrukturplan

Was gibt es zu tun? Diese intuitive Frage stellen sich Projektmanager und Auftraggeber zu Beginn eines Projekts. Planungsteams neigen deshalb bei der Projektplanung dazu, gleich im ersten Schritt To-do-Listen, Netzpläne oder Balkendiagramme zu erstellen. Allerdings fordern sämtliche Projektmanagement-Richtlinien, dass bei einer Projektplanung vor dem Ablaufplan zuerst ein Strukturplan erstellt wird. Dieser soll gewährleisten, dass alle Aspekte des Projekts vollständig erfasst und in die richtigen Zusammenhänge gestellt werden.

Wenn man versucht, ohne Strukturplan Projektpläne zu erstellen, entsteht im Projekt statt eines zielgerichteten Vorgehens unnötiger Mehraufwand: Arbeitsergebnisse müssen wieder verworfen werden, da verfrüht mit Arbeiten begonnen wurde, für die notwendige Voraussetzungen noch gar nicht geschaffen waren. Weiterhin laufen viele Arbeiten doppelt, da die Zusammenhänge der einzelnen Arbeitsergebnisse nicht rechtzeitig erkannt wurden und folglich Synergieeffekte nicht ausgenutzt werden konnten.

Während der PMBOK® Guide des PMI (PMI 2008) und die ICB der IPMA (IPMA, 2006) hierfür einen Projektstrukturplan (Work Breakdown Structure) fordern, geht die britische Richtlinie PRINCE2 einen etwas anderen Weg: Sie stellt den Produktstrukturplan in den Mittelpunkt des Planungsprozesses.

Auf den ersten Blick scheint der Unterschied zwischen diesen beiden Planungsansätzen nicht groß zu sein: Der Projektstrukturplan gliedert die Aufgabenstellung des Projekts in Teilaufgaben bis hinunter zu Arbeitspaketen. Der Produktstrukturplan gliedert dagegen den Leistungsumfang vom Projektendprodukt bis hinunter zu den einzelnen Teilleistungen (gleichbedeutend: Teilprodukt, Teillieferung, Liefergegenstand, Werk). Bei genauerer Betrachtung wird im Folgenden aber bei der produktbasierten Planung eine andere Philosophie deutlich, die dem ergebnisorientierten Charakter von Projekten besser entspricht als die aktivitätenorientierte Planung.

Pate der Produktbasierten Planung: Colin Bentley

Die Methode der Produktbasierten Planung geht auf Colin Bentley zurück, dem Initiator und langjährigen Lead Author des britischen Projektmanagementsystems PRINCE2 (OGC 2005; OGC, 2009). Dabei erfand er das Rad keineswegs neu – das Konzept des Produktstrukturplans gibt es schon seit jeher und findet sich in vielen Standardwerken über Projektmanagement, ausführlich dargestellt z.B. bei Burghardt (Burghardt, 2006).

Die Leistung Colin Bentleys besteht darin, mit seinem Konzept der Produktbasierten Planung (PBP) den gesamten Planungsprozess ergebnisorientiert gestaltet zu haben. Dies hilft, u.a. zwei der wichtigsten Schwierigkeiten des Projektmanagements zu überwinden:

Autor



Dr. Georg Angermeier

Selbst. Berater und Trainer
f. PM, PRINCE2
Practitioner, Autor Glossar

Projekt Magazin.

Kontakt:

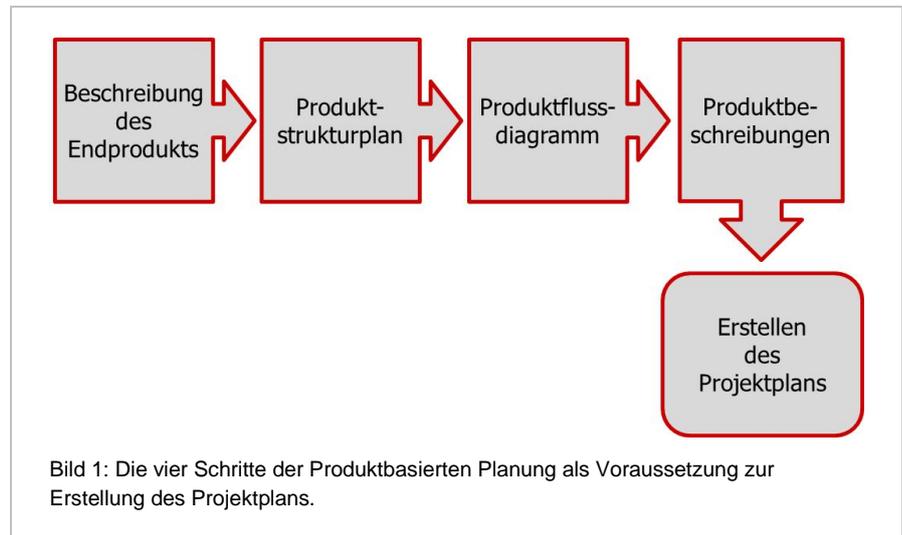
Georg.Angermeier@Ask-ASC.de

Mehr Informationen unter:

projektmagazin.de/autoren

- die unzureichende Spezifikation des Leistungsumfangs
- das sog. 90%-Phänomen, d.h. dass die Ersteller zu einem frühen Zeitpunkt im Projekt glauben, bereits "zu 90% fertig zu sein" und sie deshalb den Projektfortschritt viel zu optimistisch einschätzen.

Die Fokussierung auf die zu erstellenden Produkte bewirkt zum einen, dass durch die obligatorischen Produktbeschreibungen, den Produktstrukturplan sowie durch das Produktflussdiagramm (vgl. Bild 1) die exakte Spezifikation des Leistungsumfangs in die Projektplanung nahtlos integriert ist und sich geradezu automatisch aus ihr ergibt. Im Rahmen der PBP werden im Planungsprozess aus vagen Qualitätserwartungen der Kunden messbare Qualitätskriterien für die Abnahme.



Zum anderen bewirkt die produktorientierte Auslegung der Pläne und die daraus folgende Fortschrittsüberwachung, dass Produkte erst bei der Ermittlung des Fertigstellungsgrads berücksichtigt werden dürfen, wenn sie tatsächlich vollständig erstellt und abgenommen sind. Die Aussage "Wir sind fast fertig" zählt bei einem produktorientierten Controlling genauso viel wie "Wir haben noch gar nicht angefangen". Bei aktivitätenorientierten Plänen wird häufig der bereits angefallene Aufwand als Fortschritt gezählt wird, z.B. 70% des geschätzten Arbeitsaufwands ist erbracht.

Auch wenn die Produktorientierung ein wesentliches Merkmal von PRINCE2 ist, kann die produktbasierte Planung problemlos in andere Vorgehensweisen integriert werden (Bentley, 2012). Sie kann dann u.U. zwar nicht ihre volle Wirkung entfalten, wenn die Prozesse der Fortschrittsüberwachung und des Qualitätsmanagements nicht produktorientiert gestaltet sind. In jedem Fall ist sie für den Planungsprozess aber eine große Hilfestellung.

Im Folgenden stelle ich die produktbasierte Planung anhand eines einfachen Beispiels für den allgemeinen Einsatz vor. Ich folge im Wesentlichen der Darstellung im PRINCE2-Handbuch 2005 (OGC, 2005), da ihre Beschreibung in der aktuell gültigen Version des PRINCE2-Handbuches (OGC, 2009) leider aufgrund des allgemeinen Kürzungszwangs sehr stark reduziert wurde. Ergänzend verwende ich Material von Colin Bentley, das er mir dankenswerterweise zur Verfügung gestellt hat (Bentley, 2012).

Ich beschreibe einerseits die reine Methode, gebe andererseits aber auch aus eigener Erfahrung Anregungen für die pragmatische Interpretationen ihrer Regeln.

Warum produktorientiert planen?

Die Grundidee der produktorientierten Planung beruht darauf, dass Projekte den Charakter eines Werkvertrags haben. Es gilt, den Zustand A in einen definierten Zustand B zu überführen. Welcher Weg hierzu beschritten wird, ist von untergeordneter Bedeutung, auch wenn Rahmenbedingungen hierfür enge Grenzen setzen können. Deshalb, so die Idee von Colin Bentley, sollen bei der Projektplanung auch die zu liefernden Produkte die Struktur vorgeben und nicht die Tätigkeiten.

Wichtig ist hier die verallgemeinerte Interpretation des Begriffs "Produkt" im Sinne von PRINCE2: Produkte sind demgemäß nicht nur greifbare Gegenstände, sondern auch die Ergebnisse der im Rahmen eines Projekts erbrachten Dienstleistungen. Im Folgenden ist der Begriff "Produkt" somit gleichbedeutend zu verstehen mit "Deliverable" bzw. "Liefergegenstand", wie er im PMBOK® Guide verwendet wird, oder dem betriebswirtschaftlichen Begriff "Leistung".

Beispiel: Die an einem Messestand gesammelte Kontaktliste von Prospects (potenziellen Neukunden) ist unzweifelhaft ein Produkt, selbst wenn sie nur als elektronische Datei vorliegt. Das nach der Messe erfolgende Abtelefonieren der Kontakte ist aber kein dingliches Produkt, sondern eine Dienstleistung. Sobald diese jedoch erbracht ist, hat sich einerseits die Kontaktliste selbst verändert – zumindest um die Kennzeichnungen ihrer Bearbeiter – andererseits ist auch ein neuer Zustand eingetreten, da ein neuer Informationsstand vorliegt. Im Sinne von PRINCE2 ist also nicht "Nachtelefonieren der Messekontakte" das Produkt, sondern "abgearbeitete Kontaktliste" (inklusive Gesprächsnotizen und Folgeaufgaben im CRM-System).

Aus diesem Verständnis heraus hat die Produktorientierung bei der Planung folgende Vorteile:

- Sie gibt dem Projekt eine für alle Stakeholder intuitiv nachvollziehbare Struktur.
- Sie verhindert ins Blaue hinein gestartete Projekte. Falls eine produktorientierte Planung nicht möglich ist, weil zu große Unsicherheiten der zu liefernden Leistungen bestehen, kann es auch kein Projekt geben, da weder Planung, Steuerung noch Abnahme erfolgen kann. Stattdessen könnten sich z.B. eine Machbarkeitsstudie, ein Marktforschungsprojekt oder ein Projekt mit kleinerem Umfang als sinnvoll erweisen.
- Sie bietet der großen Gefahr der schleichenden Erweiterung des Umfangs (scope creep) Einhalt, da auch die Teilleistungen klar abgegrenzt sind.
- Sie zwingt die Planer dazu, aus den ungenau formulierten Qualitätserwartungen exakte Qualitätskriterien zu entwickeln.
- Sie gibt den Experten die Freiheit, den besten Lösungsansatz zu entwickeln, da die PBP keine Tätigkeiten und damit auch keinen Lösungsweg vorgibt.
- Sie erlaubt eine wertorientierte Planung des Leistungsumfangs und damit die Optimierung des Business Cases, da jedes definierte Teilprodukt hinsichtlich seines Beitrags zum Projektnutzen beurteilt werden kann.
- Sie ist die einzig mögliche Vorgehensweise, um eine konsistente Top-Down-Planung zu realisieren, da nur Produkte mit eindeutig vereinbarten Abnahmekriterien stabile Schnittstellen zwischen Planungsebenen liefern.

Beim Einsatz der PBP gibt es zwei typische Gefahren:

- Sie wird nicht vollständig durchgeführt. Wird z.B. das Produktflussdiagramm weggelassen, kann dies zu schwerwiegenden Fehlern führen, weil Produkte übersehen werden.
- Es wird anschließend keine Planung der Aktivitäten durchgeführt. Die PBP liefert keine Aussagen über Arbeitsaufwände, Dauern und Ressourcenabhängigkeiten, sondern nur die Vorgaben dafür. Um Aussagen über Termine und Kosten zu erhalten, müssen deshalb Arbeitspakete bzw. Vorgänge und Ressourcen im Rahmen dieser Vorgaben geplant werden.

Die Methode: In vier Schritten zur Planungsgrundlage

Die Regeln der produktbasierten Planungstechnik sind sehr einfach und intuitiv nachvollziehbar. Dennoch erweist es sich in der Praxis, dass Anfänger gerade die am einfachsten erscheinenden Regeln, wie z.B. die Beibehaltung der Produktnamen während aller Planungsschritte, bei den ersten Übungen verletzen. Vielleicht verleitet eben gerade die Einfachheit dazu, die Regeln nicht exakt einzuhalten. Zum Glück fallen diese Fehler sehr schnell auf, da dann Widersprüche bei den Plänen entstehen. Bereits nach einigen Versuchen werden Sie die Regeln so verinnerlicht haben, dass Sie über sie gar nicht mehr bewusst nachdenken werden.

Die Produktbasierte Planung erfolgt in vier Schritten (vgl. Bild 1):

1. Beschreiben Sie das Endprodukt!
2. Analysieren Sie das Endprodukt in einem Produktstrukturplan!
3. Übertragen Sie die identifizierten Produkte in ein Produktflussdiagramm!
4. Erstellen Sie mindestens für alle Einzelprodukte (d.h. nicht weiter zerlegten Produkte) eine Produktbeschreibung!

Tipp: Ich merke mir die vier Schritte der PBP ganz einfach so:

1. Was sollen wir heute essen?
2. Welche Zutaten und Küchengeräte benötige ich dafür?
3. Wie geht das Kochrezept?
4. Was brauche ich genau und wo bekomme ich alles her?

Wenn Sie während der produktbasierten Planung auf Schwierigkeiten stoßen, ist dies nicht in der Methode sondern im Projektinhalt begründet, z.B. weil Sie vorher übersehene Unklarheiten bei den Anforderungen entdecken.

! Die produktbasierte Planungstechnik ist ein Hilfsmittel, kein Selbstzweck! Es gibt niemals nur "die eine richtige Lösung" für Produktbeschreibungen, Produktstrukturplan oder Produktflussdiagramm. Je nach Rahmenbedingungen, Routine des Projektteams und vielen anderen Faktoren werden sich unterschiedliche Strukturierungsansätze als zielführend erweisen. Die Verantwortung für die fachlich richtige Bearbeitung des Projektinhalts kann nicht auf die Methode abgewälzt werden!

Zum besseren Verständnis werde ich den Ablauf der produktbasierten Planung anhand eines einfachen Beispiels erläutern.

Beispiel: Messeauftritt

Die Edeluhr GmbH ist ein mittelständisches Unternehmen, das hochwertige Armbanduhren herstellt und sowohl über den Einzelhandel als auch im Direktverkauf vertreibt. Wichtiger Vertriebsweg sind Messen rund um den Globus, bei denen die Edeluhr GmbH sowohl ihre Distributoren als auch Endkunden anspricht. Hierfür gibt es standardisierte Messestände, die qualitätsgesichert und transportbereit im Lager jederzeit auf Abruf stehen. Die Messeteams müssen sich nur noch um die Organisation des Messeauftritts und um Besonderheiten wie z.B. Ausstellungsstücke von Novitäten kümmern. Die Kontaktliste der eingeladenen Kunden und Geschäftspartner ist das wichtigste Arbeitsmittel des Messeteams. In ihr dokumentieren sie alle Messekontakte.

Die Geschäftsführung hat entschieden, das Messeteam "Charly" zu einer Messe für Luxusgüter nach Sidney zu schicken und hat den Teamleiter Hans Meier zum Projektleiter dieses Messeauftritts ernannt. Er soll den Projektplan erstellen und wählt hierfür die Technik der Produktbasierten Planung.

Schritt 1: Beschreiben Sie das Endprodukt!

Meier definiert als Endprodukt des Projekts: "Durchgeführter und nachbereiteter Messeauftritt auf der Messe Sydney 2012". Da diese präzise, aber lange Bezeichnung im alltäglichen Gebrauch sehr unhandlich wäre, wählt er als Kurzbezeichnung: "Messe Sydney 2012".

Als Ausgangspunkt der Produktbasierten Planung ist es im ersten Schritt notwendig, das Endprodukt des Projekts zu beschreiben. Eine Produktbeschreibung nach PRINCE2 ist allerdings weit mehr als eine herkömmliche Spezifikation. Sie enthält darüber hinaus die notwendigen Schnittstellen zu den Managementaufgaben, insbesondere zu den Aufgaben des Qualitätsmanagements, wie aus Tabelle 1 ersichtlich ist. Auch wenn Sie nicht nach PRINCE2 arbeiten, sind die Produktbeschreibungen ein nützliches Werkzeug für das Zusammenspiel von Plänen, Qualitätsmanagement und Konfigurationsmanagement. Insofern stellen die während der PBP entstehenden Produktbeschreibungen das Gerüst für das Projektmanagement dar.

Hans Maier erstellt zusammen mit seinem Team die Produktbeschreibung des Endprodukts, das Ergebnis sehen Sie in Tabelle 1.

Element	Inhalt	Beispiel: Messeauftritt eines Uhrenherstellers
Kennzeichnung	Jedes Produkt muss eine eindeutige Bezeichnung haben, um es identifizieren zu können. Dies ist insbesondere für das Konfigurationsmanagement unverzichtbar. Falls ein Produkt der Versionskontrolle unterliegt, sollte eine Bezeichnung deshalb auch in geeigneter Weise die Versionsnummer enthalten.	Durchgeführter und nachbereiteter Messeauftritt auf der Messe Sydney 2012
Titel	Eindeutige Kennzeichnungen sind häufig für den täglichen Einsatz ungeeignet, deshalb sollten Produkte auch einfach verwendbare Namen haben.	Messe Sydney 2012

Zweck	Damit die Ersteller eines Produkts effektiv arbeiten können und z.B. Verbesserungen vorschlagen können, ist es notwendig, dass beschrieben wird, wer das Produkt für was benötigen wird.	Bindung bestehender Kunden und Geschäftspartner in Australien sowie dem restlichen Asia-Pacific. Gewinnung von mindestens 50 qualifizierten Prospects zur anschließenden Kontaktierung durch den Vertrieb.
Zusammensetzung	Dies ist Teil der herkömmlichen Spezifikation: Was ist das Produkt, woraus setzt es sich zusammen, welche Funktionen hat es? Typischerweise wird dieser Abschnitt in der Produktbeschreibung ein Verweis auf eine ausführliche Spezifikation sein.	Team "Charly" präsentiert auf der Luxuswarenmesse Sydney 2012 die neuen Kollektionen für Damen und Herren. Das Messehandbuch in der Version vom 1.12.2011 ist anzuwenden.
Herkunft / Ableitung	Welche anderen Produkte werden benötigt, um dieses Produkt zu erstellen? Dieser Frage wird bei der PBP während der Erstellung des Produktflussdiagramms beantwortet. Umgekehrt müssen im Produktflussdiagramm alle hier aufgeführten Produkte vor dem beschriebenen Produkt stehen.	Standard-Messestand, Novitäten aus Damen- und Herrenkollektion, Kontaktliste, Dienstreise. Siehe Produktstrukturplan
Darstellung und Form	Die Eigenschaften des Produkts bzw. die Art der Bereitstellung. Diese Informationen stehen meist auch in einer herkömmlichen Spezifikation, auf die von hier verwiesen werden kann.	Im Rahmen der langfristigen Messeplanung wurde ein zentraler Standplatz direkt neben dem Event-Bereich gebucht. Für die Gestaltung des Publikumsverkehrs sind die die im Messehandbuch festgelegten Regeln für die Asia-Pacific-Zielgruppen streng zu beachten.
Notwendige Kenntnisse der Ersteller	Anforderungen an die zur Erstellung dieses Produkts benötigten Ressourcen. Dies ist die Grundlage dafür, um bei der späteren Ressourcenplanung die korrekten Zuordnungen treffen zu können.	Kenntnisse des Kundenstammes, der Zielgruppenstrategie und des lokalen Markts in Australien. Team "Charly" wurde speziell aufgrund seiner multikulturellen Zusammensetzung für diese Messe beauftragt.
Qualitätskriterien (bei der Beschreibung des Endprodukts sind dies zugleich die Abnahmekriterien des Projekts)	Die überprüfbareren Abnahmekriterien für dieses Produkt. Dies kann auch durch einen Verweis, z.B. auf die Spezifikation oder einen Abschnitt in einem Handbuch des Unternehmens geschehen.	Vgl. Messehandbuch in der Version vom 1.12.2011, insbes. Prozessbeschreibung "Messe-Auftritt"
Qualitätstoleranzen	Erlaubte Abweichungen der Ist-Werte des Produkts von den in den Qualitätskriterien beschriebenen Soll-Werten.	Der Leiter des Messestands kann je nach Kundenresonanz den Umfang des Standpersonals um bis zu zwei Personen temporär reduzieren.
Qualitätsprüfmethode	Benennung der Methoden, mit denen die Qualitätsprüfung durchgeführt wird. Dies kann auch durch Verweise auf technische Richtlinien oder Normen geschehen.	Der Leiter des Stands bzw. seine Stellvertreterin führen das Standprotokoll gemäß Messehandbuch.
Notwendige Kenntnisse der Prüfer	Benennung der Qualifikation der Prüfer. Beispielsweise kann es erforderlich sein, ein unabhängiges Prüfinstitut zu beauftragen.	Erfahrung in Vertrieb und Leitung eines Messestands, Kenntnisse der Qualitätsstandards des Unternehmens
Qualitätsverantwortlichkeiten (bei der Beschreibung des Projektendprodukts sind dies die Verantwortlichen für die Abnahme des Projekts)	Ersteller, Prüfer und Abnahmeberechtigter dieses Produkts. Diese Personen werden in der Regel erst bei späteren Planungsschritten benannt.	Leiter Vertrieb

Tabelle 1: Bestandteile der Produktbeschreibung nach PRINCE2 (OGC, 2005) für das Beispiel "Messe Sydney 2012".

Diese Form der Produktbeschreibung erscheint auf den ersten Blick sehr aufwendig, tatsächlich enthält sie aber lediglich Informationen, die im Laufe eines Projekts unverzichtbar sind. Wenn z.B. die falschen Personen mit einer Qualitätsprüfung beauftragt werden oder notwendige Tests nicht durchgeführt werden, kann es leicht geschehen, dass ein Zwischenprodukt sich im Nachhinein als ungeeignet erweist und dadurch den Projekterfolg gefährdet. Um beim Beispiel zu bleiben: Wenn ein Mitarbeiter beim Erstellen der Kontaktliste im Länderauswahlfeld "Austria" statt "Australia" anklickt und keine Kontrolle erfolgt, dann muss das Standpersonal am ersten Messetag ganz gewaltig improvisieren bis die korrekte Kontaktliste erstellt und übermittelt ist.

Wenn Sie also für Ihren spezifischen Bedarf die Produktbeschreibung aus Tabelle 1 anpassen, sollten Sie keines ihrer Elemente weglassen, sondern überlegen, welche Information für Ihre Branche oder Ihre Projekte bei den einzelnen Elementen erforderlich sind. Vielleicht müssen Sie noch weitere Aspekte ergänzen. Die Produktbeschreibungen sind die Begleitdokumente der Produkte während des Projekts und unterliegen ebenfalls dem Konfigurationsmanagement.

! Versuchen Sie als Übung, die Beschreibung für das Endprodukt eines Ihrer gerade laufenden Projekte gemäß Tabelle 1 zu erstellen!

Anmerkung: PRINCE2 (Version 2009) unterscheidet übrigens zwischen der Produktbeschreibung des Endprodukts und den Produktbeschreibungen der Teilprodukte, da die Abnahme des Endprodukts für PRINCE2 gleichbedeutend mit der Abnahme des Projekts ist. Die Beschreibung des Endprodukts enthält deshalb die Projektabnahmekriterien, die Qualitätstoleranzen für das gesamte Projekt und die Beschreibung, wie die Projektabnahme durchgeführt wird. Wenn Sie die PBP außerhalb von PRINCE2 einsetzen, sollten Sie abklären, wie die Projektabnahme in Ihrem Projektmanagementsystem geregelt ist und wie Sie die Produktbeschreibungen darin integrieren können.

Schritt 2: Analysieren Sie das Endprodukt in einem Produktstrukturplan!

Im nächsten Schritt erstellen Sie den Produktstrukturplan. Schritt für Schritt gliedern Sie dabei das Endprodukt immer weiter in Teilprodukte bis zu einem Detaillierungsgrad, der Ihnen für die Projektplanung angemessen erscheint. Die Leitfrage ist dabei durchgängig: Aus welchen Teilen besteht dieses Produkt?

Dabei müssen Sie drei einfache Regeln sorgfältig beachten:

- Der Produktplan ist streng hierarchisch, d.h. ein Teilprodukt hat immer genau ein übergeordnetes Produkt.
- Der Produktstrukturplan trifft keine Aussagen über Abfolgen, es gibt daher keine Pfeile. Die Reihenfolge, in der Teilprodukte aufgelistet werden, kann vom Planer beliebig festgelegt werden, z.B. alphabetisch.
- Eins-zu-eins-Verbindungen sind verboten, d.h. ein (Teil-)Produkt muss in mindestens zwei untergeordnete Produkte zerlegt werden.

Anmerkung zur Darstellungsform des Produktstrukturplans

In den folgenden Beispielen ist der Produktstrukturplan als Organigramm dargestellt. Weitere, gleichberechtigte Darstellungsmöglichkeiten sind die Mindmap oder die eingerückte Liste.

Hans Meier fängt ganz pragmatisch an und gliedert das Endprodukt in die vier Teilprodukte "Messestand", "Kontaktliste", "Dienstreise" und "Novitäten" (Bild 2).

Hans Meier wählte als erste Gliederungsebene diese vier Produkte, da sie seiner Erfahrung nach am besten als Grobstruktur geeignet sind. Ein

anderer hätte vielleicht eine abstraktere Gliederung in "Produkte der Vorbereitung", "Produkte der Messe", "Produkte der Nachbereitung" und "Produkte des Projektmanagements" gewählt.

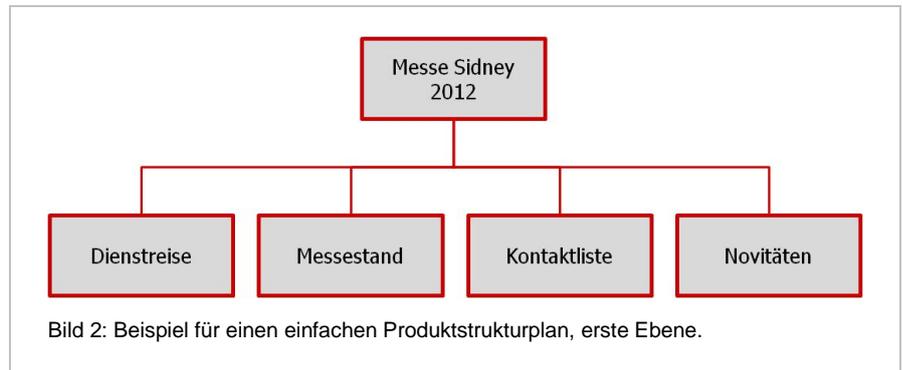


Bild 2: Beispiel für einen einfachen Produktstrukturplan, erste Ebene.

Puristen der Methode könnten hier bemängeln, dass "Dienstreise" keine saubere Bezeichnung für ein Produkt ist. Sie könnten z.B. fordern, dass hier mindestens "Produkte der Dienstreise" oder noch besser: "Vorbereitete, durchgeführte und dokumentierte Dienstreise" steht. Hier ist es für mich wichtig, die Aussage von Colin Bentley zu beherzigen: "The product breakdown structure is simply a mechanism to help us think of the products that will be needed in the plan" (Bentley 2006). Mit anderen Worten: Es geht nicht darum, einen theoretisch perfekten Produktstrukturplan zu erstellen, sondern der Produktstrukturplan soll uns optimal dabei unterstützen, alle benötigten Produkte des Projekts zu identifizieren.

Dabei unterscheidet sich der Produktstrukturplan der PBP in einem wichtigen Punkt vom traditionellen Produktstrukturplan in der deutschen Literatur: Er ist keine statische Momentaufnahme, sondern beschreibt die Produkte in ihrer Zusammensetzung während ihres Lebenswegs innerhalb des gesamten Projekts (Bentley 2012).

Beispiel

Die Kontaktliste nimmt während des Messeprojekts verschiedene Zustände an, die jeweils überprüft werden müssen. Vor Abflug muss die Kontaktliste von der Vertriebsabteilung vorbereitet werden, in ihr müssen insbesondere die zur Messe eingeladenen Kunden und Geschäftspartner stehen. Während der Messe tragen die Mitglieder des Teams "Charly" alle geknüpften Kontakte qualifiziert

in die Kontaktliste ein. Am Ende der Messe überprüfen sie die Liste auf Vollständigkeit und Hans Meier übermittelt sie wieder an den Vertrieb. Am Ende der Nachbereitung ist die Kontaktliste mit den Bearbeitungsvermerken versehen und wird komplett in das CRM-System der Edeluhr GmbH übertragen und archiviert. Dementsprechend gliedert Hans Meier das Produkt "Kontaktliste" in die drei Teilprodukte "vorbereitete Kontaktliste", "ausgefüllte Kontaktliste" und "abgearbeitete Kontaktliste" auf, wie in Bild 3 dargestellt.

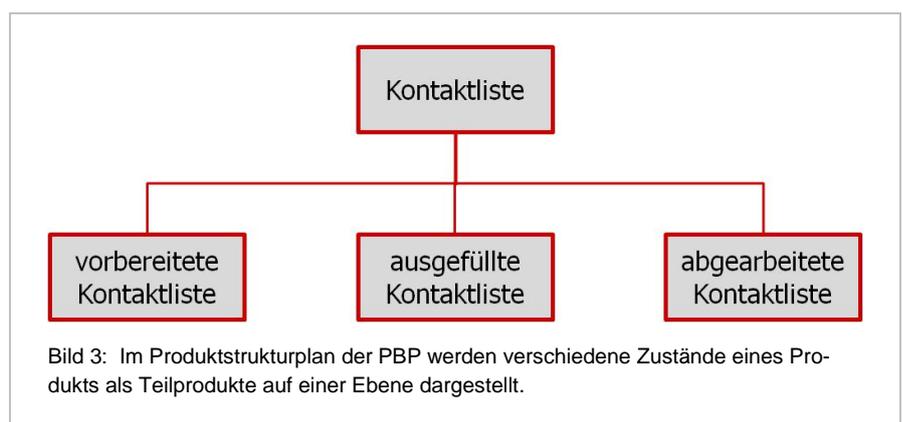


Bild 3: Im Produktstrukturplan der PBP werden verschiedene Zustände eines Produkts als Teilprodukte auf einer Ebene dargestellt.

Hilfreiche Spezialfälle: Externe Produkte und Gruppierungen

In der Praxis hat es sich als hilfreich erwiesen, die identifizierten Produkte danach zu unterscheiden, ob für ihre Erstellung Aufwände innerhalb des Projekts erforderlich sind oder nicht. Denn wenn der Planer sofort erkennen kann, dass ein Produkt keinen Aufwand im Projekt erzeugt, vereinfacht dies die Erstellung von Produktflussdiagramm und Ablaufplan.

Externe Produkte: Nicht innerhalb des Projekts erstellt

Im Beispiel geht Hans Meier davon aus, dass der Messestand qualitätsgesichert und transportbereit im Lager steht. Innerhalb des Projekts fallen deshalb erst mit dem Transport des Messestands nach Australien Aufwände an, er muss ihn lediglich reservieren. Das Produkt "reservierter Messestand" muss zwar trotzdem im Produktstrukturplan auftauchen, damit es nicht vergessen wird, aber im Projekt gibt es kein Arbeitspaket "Lagerung und Bereitstellung eines Messestands". Produkte dieser Art werden als "externe Produkte" (engl.: external products) bezeichnet. Das Wort "extern" bedeutet dabei, dass dieses Produkt außerhalb des Projekts erzeugt wird. Wenn hingegen ein anderes Unternehmen ein Produkt für das Projekt liefert, ist das dennoch ein internes Produkt für die PBP, da das Unternehmen Mitglied im Projektteam ist und somit im Projektplan mindestens ein Arbeitspaket für seine projektinterne (!) Erstellung und Qualitätssicherung enthalten ist. Die Kennzeichnung externer Produkte ist besonders empfehlenswert, wenn das Projekt Produkte von anderen Projekten übernimmt, z.B. weil es Teil eines Programms ist.

Gruppierungen: Hilfskonstruktionen im Produktstrukturplan

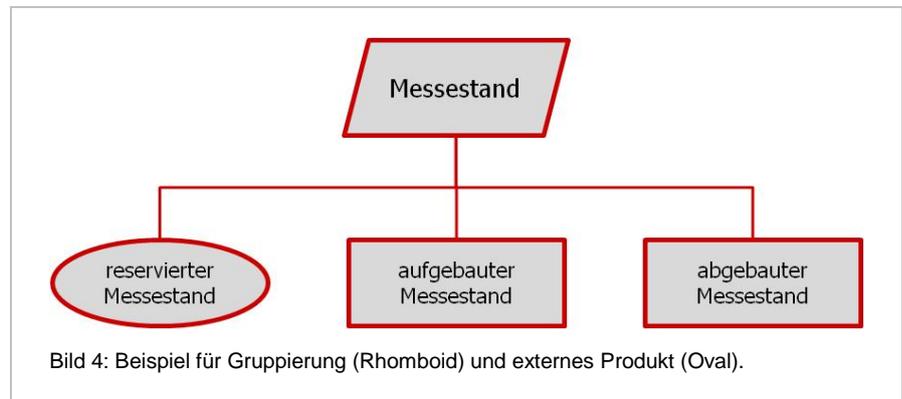
Produkte, die weiter aufgegliedert werden, werden als "Zwischenprodukte" (engl.: intermediate products) bezeichnet. Es gibt dabei grundsätzlich zwei Möglichkeiten, wie diese Strukturierung zu verstehen ist:

- Das Zwischenprodukt wird im Projekt aus seinen Bestandteilen erstellt, d.h. hierfür ist Arbeitsaufwand erforderlich. Diese Zwischenprodukte werden **Integrationsprodukte** (engl.: integration products) genannt.
- Sobald die untergeordneten Produkte fertig sind, ist automatisch auch das übergeordnete Zwischenprodukt fertig, d.h. es ist lediglich die Zusammenstellung seiner Teilprodukte ohne dass dafür etwas getan werden müsste. Diese Zwischenprodukte werden **Gruppierungen** (engl.: collective groupings) genannt.

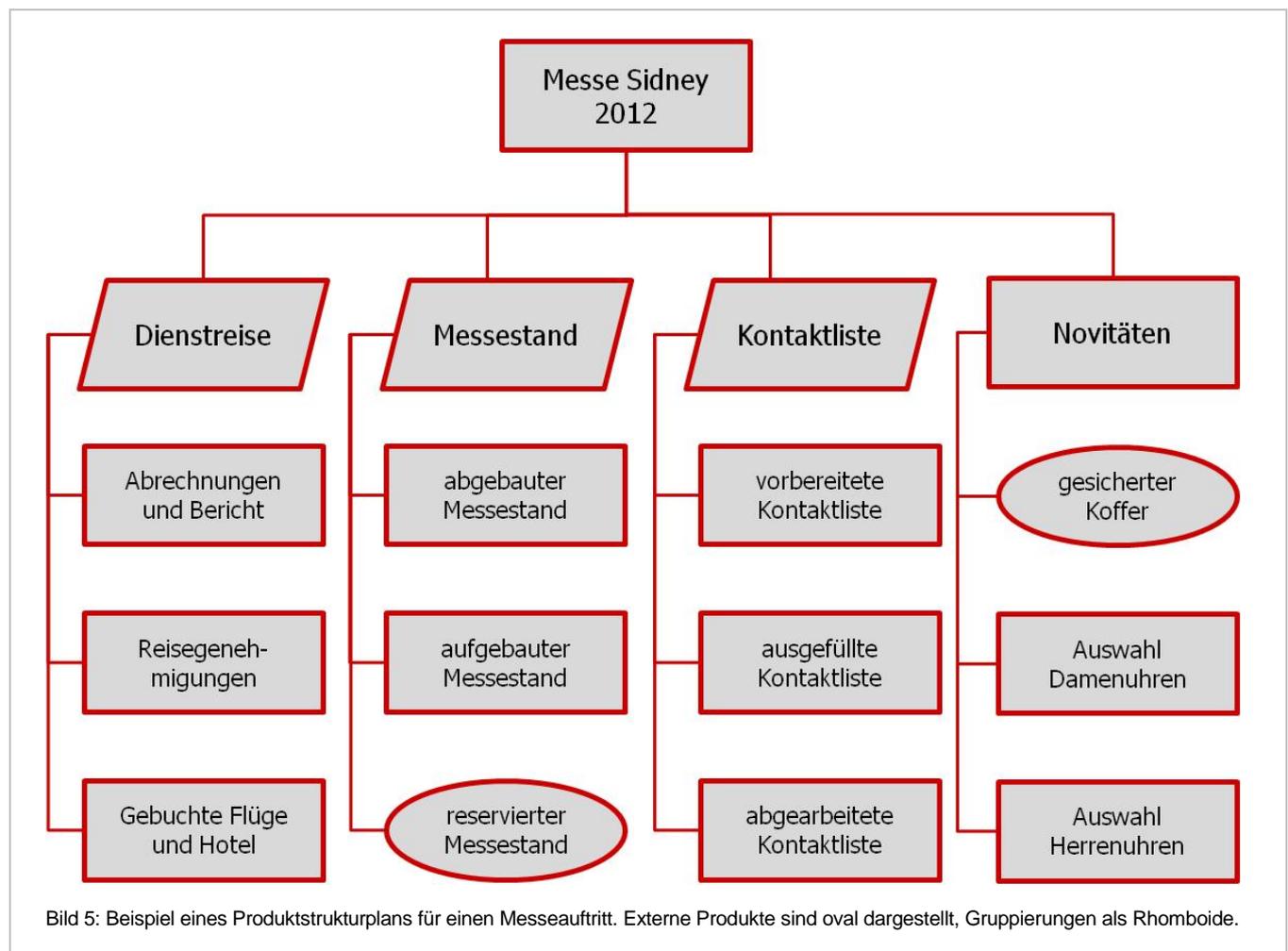
Diese Unterscheidung in Gruppierungen und Integrationsprodukte wird im PRINCE2-Manual von 2009 nicht mehr getroffen. Die Argumentation war, dass diese Unterscheidung die Anwender verwirrt hat. Meine persönliche Meinung ist allerdings, dass es für den Planungsprozess verwirrend ist, diese Unterscheidung nicht zu treffen, da dann bei der Erstellung des Projektplans Unklarheit darüber besteht, für welche Produkte Arbeitspakete zu erstellen sind. Ich empfehle Ihnen daher, die beiden folgenden Regeln einzuhalten, da sie Ihnen die weitere Schritte der PBP erleichtern:

- Kennzeichnen Sie bei Zwischenprodukten, ob es lediglich Gruppierungen sind, die keinen Arbeitsaufwand erfordern oder Integrationsprodukte, deren Erstellung mit Aufwand verbunden ist. Hierfür können Sie z.B. das Format des Produktsymbols im Strukturplan verändern oder vor den Produktnamen "Gruppierung:" schreiben.
- Kennzeichnen Sie externe Produkte, die von der Projektumgebung gestellt werden, ohne dass dafür innerhalb des Projekts Aufwände (auch nicht für Qualitätssicherung!) entstehen. Üblicherweise werden externe Produkte durch Ovale dargestellt. Alternativ können Sie auch "Extern:" vor den Produktnamen schreiben.

Hans Maier gliedert das Zwischenprodukt Messestand auf in "reservierter Messestand", "aufgebauter Messestand" und "abgebauter Messestand", da dies für ihn die drei wichtigen Zustände des Messestands sind. Um verschiedene Zustände eines Produkts regelkonform in den Produktstrukturplan einzubinden, hat er nur die Möglichkeit, sie mit einer Gruppierung zusammenzufassen.



Gleichzeitig macht er damit deutlich, dass für die Gruppierung "Messestand" keine Aufwände innerhalb des Projekts einzuplanen sind. Weiterhin kennzeichnet er den reservierten Messestand als externes Produkt, da er hierfür – wie oben beschrieben – keinen Arbeitsaufwand in seinem Projekt berücksichtigen muss. Der in Bild 4 dargestellte Ausschnitt des Produktstrukturplans symbolisiert, dass Aufwände innerhalb des Projekts ausschließlich für die Produkte "aufgebauter Messestand" und "abgebauter Messestand" entstehen.



Der fertige Produktstrukturplan

Als Routiniers haben Hans Maier und sein Team "Charly" den Produktstrukturplan in zehn Minuten fertig gestellt. Das in Bild 5 dargestellte Ergebnis ist nur eine von vielen Möglichkeiten, einen Produktstrukturplan für einen Messeauftritt zu erstellen. Entscheidend ist dass die Projektbeteiligten mit Hilfe des Produktstrukturplans ein gemeinsames Verständnis des Projektinhalts erzielen und er in ihren Augen vollständig den Leistungsumfang des Projekts darstellt. Z.B. haben sie für das Zwischenprodukt "Novitäten" entschieden, es als Integrationsprodukt darzustellen, da sie die Arbeitsaufwände für den persönlichen Transport und die tägliche Arbeit mit den wertvollen Ausstellungsstücken während der Messe eigens einplanen wollen.

Als nächsten wird Team "Charly" im dritten Schritt aus dem Produktstrukturplan das Produktflussdiagramm ableiten. Dies werden wir im zweiten Teil dieses Artikels genauer beleuchten, zuerst aber möchte ich Ihnen noch einige Tipps geben, wie Sie bei der Erstellung eines Produktstrukturplans vorgehen können.

Praktische Tipps für die Produktstrukturplanung

Die Schritte und die Regeln der PBP geben lediglich den Rahmen vor, Sie haben einen sehr großen Freiraum, wie Sie ihn methodisch und fachlich ausfüllen. Im Folgenden gebe ich Ihnen einige Tipps für den Produktstrukturplan und die Produktbeschreibungen. Diese sollen Sie in erster Linie anregen, selbst zu experimentieren und dadurch die PBP in Ihr Methodenrepertoire aufzunehmen.

Wie viele Produkte soll ein Produktstrukturplan umfassen?

Die am häufigsten geäußerte Kritik an der PBP ist: "Unser Projekt umfasst zu viele Produkte, als dass sie in einem Produktstrukturplan dargestellt werden könnten!"

Diese Aussage würde im Umkehrschluss bedeuten, dass das Projekt weder plan- noch steuerbar ist – egal ob mit oder ohne PBP! Zum Glück ist die Lösung hier sehr einfach: Die Regeln des Produktstrukturplans gewährleisten, dass es problemlos möglich ist, für jedes Einzelprodukt einen eigenen, ausgegliederten Produktstrukturplan zu erstellen. Ein Produktstrukturplan für sich genommen sollte deutlich weniger als 100 Produkte enthalten. Colin Bentley empfiehlt maximal 25 (Bentley, 2012). Ich bin da etwas großzügiger und halte auch noch rund 50 Produkte für gerechtfertigt. Entscheidend ist aber: Der Produktstrukturplan muss für alle Projektbeteiligten überschaubar bleiben. Sobald eine weitere Detaillierung erforderlich ist, muss die Ebene gewechselt werden – z.B. dadurch, dass für ein Produkt ein Teilprojekt oder ein Arbeitspaket definiert wird, in dem dann wiederum nach PBP geplant wird.

Verwenden Sie Kärtchen, keine Software!

Wohl jeder Trainer und Moderator wird meine Erfahrung bestätigen, dass es für einen kreativen Prozess nichts Besseres gibt, als mit greifbaren Materialien zu arbeiten, insbesondere wenn mehrere Personen daran beteiligt sind. Auf diese Weise arbeiten alle Beteiligten engagierter mit als wenn sie nur gebannt auf die Projektionsfläche starren und der Moderator die ganze Arbeit am Rechner macht.

Meine Empfehlung ist, mit einem Brainstorming zu beginnen und alle Produkte auf Kärtchen zu schreiben bevor Sie an die Konstruktion des Projektstrukturplans gehen. Diese Kärtchen können dann bei Bedarf zusammenge-

fasst werden, indem man z.B. eine neue Karte für das identifizierte Produkt schreibt und die anderen unter diese Karte legt oder ganz einfach zur Seite legt.

Sie können dieses Brainstorming ebenfalls mit einer Mindmap durchführen. Aber auch dann empfehle ich die Verwendung von Kärtchen, die an die Äste der Mindmap gepinnt werden. Kärtchen haben den großen Vorteil, dass sie leicht herumgeschoben werden können. Vor allem können sie anschließend mühelos für den nächsten Schritt der PBP, die Entwicklung des Produktflussdiagramms weiter verwendet werden.

Mit Hilfe einer Digitalkamera sollten Sie jeden wesentlichen Zwischenschritt festhalten – dadurch können Sie verschiedene Variationen des Produktstrukturplans dokumentieren und ggf. wieder herstellen.

Verwenden Sie vordefinierte Standardprodukte

Bei bestimmten Projektarten gibt es immer wieder die gleichen Produkte. Bei Konferenzen wird es z.B. immer ein Catering geben, bei Webprojekten ein ausgewähltes Content Management System, bei Produktentwicklungen ein genehmigtes Design usw. Sie beschleunigen die Planungsworkshops erheblich, wenn Sie für solche immer wiederkehrenden Produkte Standardkarten erstellen.

Literatur

- Bentley, Colin: Product-Based Planning, Video, Seahorse Production, 2005
- Bentley, Colin: Persönliche Mitteilung, Feb. 2012
- Burghardt, Manfred: Projektmanagement. Leitfaden für die Planung, Überwachung und Steuerung von Projekten, Erlangen, 2006
- International Project Management Association (IPMA): IPMA Competence Baseline Version 3.0, Nijkerk, Niederlande, 2006
- Office of Government Commerce (OGC): Managing Successful Projects with PRINCE2, 2005
- Office of Government Commerce (OGC): Managing Successful Projects with PRINCE2, 2009
- Project Management Institute (PMI): A Guide to the Project Management Body of Knowledge (PMBOK® Guide), Fourth Edition, Newton Square, Pennsylvania, USA, 2008

Fachbeitrag

Die Produktbasierte Planung nach PRINCE2

Teil 2: Produktflussdiagramm, Produktbeschreibungen und Projektplan

Im ersten Teil dieser Artikelserie erläuterte ich die Vorteile der produktorientierten Planung, gab einen Überblick über die produktbasierte Planungstechnik nach PRINCE2 und erklärte deren ersten beiden Schritte: Die Beschreibung des Projektendprodukts und die Erstellung des Produktstrukturplans. Dabei begleiteten uns in einem fiktiven Beispiel Hans Maier mit seinem Messeteam "Charly", die für die Edeluhr GmbH einen Messeauftritt auf einer Luxusmesse in Sydney planen ("Die Produktbasierte Planung nach PRINCE2. Teil 1: Produktbeschreibung und Produktstrukturplan", Projekt Magazin 4/2012).

Nachdem die Mitglieder des Projektteams das Projektendprodukt beschrieben und den Produktstrukturplan erstellt haben, bringen sie jetzt die identifizierten Produkte in die Reihenfolge, in der sie nacheinander benötigt werden. Hierfür erstellen sie das sog. Produktflussdiagramm.

! Anmerkung: Ich weiche hier in der Reihenfolge der Schritte bewusst vom PRINCE2-Handbuch ab. Dort werden zunächst die Produktbeschreibungen (s. Schritt 4) erstellt und dann erst das Produktflussdiagramm. Der Grund für die von mir gewählte Reihenfolge ist, dass ich das unmittelbare Zusammenspiel aus Produktstrukturplan und Produktflussdiagramm in der Praxis als besonders effizient erlebt habe. Das PRINCE2-Handbuch weist in den Erläuterungen zum Produktflussdiagramm auch explizit auf diese alternative Reihenfolge hin – meine Darstellung ist also durchaus konform mit PRINCE2 (OGC, 2009, Abschnitt 7.3.3.4).

Schritt 3: Übertragen Sie die identifizierten Produkte in ein Produktflussdiagramm!

Ein Produktflussdiagramm ist formal gesehen ein Meilenstein-Netzplan. Dies bedeutet, dass die Produkte im Produktflussdiagramm mit Pfeilen so miteinander verbunden werden, wie es der Projektablauf erfordert (Bild 1).

Beim Produktflussdiagramm ist die Leitfrage: Welche Produkte müssen vorliegen, damit das nächste Produkt hergestellt werden kann? Dies ist eine andere Fragestellung als beim Produktstrukturplan, der ausschließlich die Produkte nach ihrer Zusammensetzung aufgliedert. Bei einer zeitlichen Abfolge gehen hingegen auch Managementaspekte ein, wie z.B. Phasenfreigaben, die erst erfolgen, sobald bestimmte Produkte abgenommen worden sind. Zudem

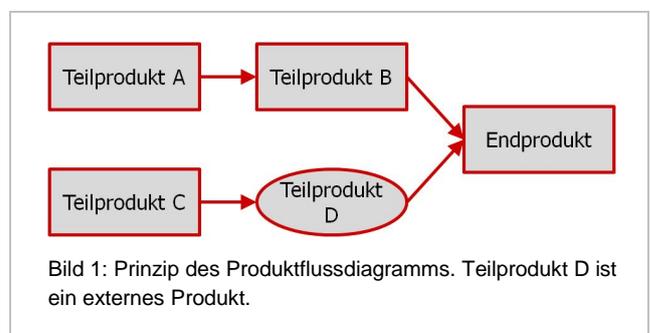
Autor



Dr. Georg Angermeier
Selbstständiger Berater
und Trainer für PM,
PRINCE2 Practitioner,
Autor Glossar Projekt Magazin.

Kontakt:
Georg.Angermeier@Ask-ASC.de

Mehr Informationen unter:
projektmagazin.de/autoren



benötigt man für die Herstellung eines Produkts z.B. Maschinen oder Werkzeuge, die aber nicht Bestandteil des Produkts sind und somit im Produktstrukturplan in einem getrennten Ast aufgeführt werden – im Produktflussdiagramm werden sie vor dem zu erstellenden Produkt eingefügt.

Für die Erstellung des Produktflussdiagramms gibt es wiederum ganz einfache Regeln, die allerdings nicht explizit in dieser Form im PRINCE2-Handbuch stehen. Vielmehr habe ich sie aus den Anleitungen und der inhärenten Logik des Planungsvorgangs abgeleitet:

1. Alle Produkte des Produktstrukturplans mit Ausnahme der Gruppierungen müssen auch im Produktflussdiagramm enthalten sein.
2. Alle Produkte des Produktflussdiagramms müssen auch im Produktstrukturplan enthalten sein.
3. Die Namen der Produkte müssen in Produktstrukturplan und Produktflussdiagramm identisch sein.
4. Vom Projektendprodukt geht kein Pfeil aus, von jedem anderen Produkt geht mindestens ein Pfeil aus.
5. Es dürfen keine Zirkelschlüsse entstehen.

Diese Regeln gewährleisten zum einen, dass der gesamte Leistungsumfang des Projekts vollständig erfasst wird und zum anderen, dass das Produktflussdiagramm als Basis für den Terminplan geeignet ist.

Die Gruppierungen (d.h. Zwischenprodukte ohne Arbeitsaufwand, vgl. Teil 1) werden deswegen nicht aus dem Produktstrukturplan in das Produktflussdiagramm übernommen, da sie kein mit Arbeitsaufwand zu erstellendes Produkt darstellen. Sobald alle Teilprodukte einer Gruppierung im Flussdiagramm enthalten sind, liegt damit automatisch auch die Gruppierung selbst vor.

Fehler im Ablauf erkennen

Wenn ein Produkt nicht korrekt in das Produktflussdiagramm eingebunden werden kann (d.h. Regel 1 oder 4 ist verletzt), dann bedeutet dies, dass entweder das Produkt für dieses Projekt gar nicht erforderlich ist, oder dass ein weiteres Produkt vergessen wurde, das die Verbindung mit einem anderen Produkt herstellt.

Beispiel: Genau dieser Fehler ist dem Messteam unter Leitung von Hans Maier in der Eile der Planung unterlaufen. Als sie die Kärtchen für den Messestand in das Produktflussdiagramm einbinden wollen, stellen sie fest, dass sie das Projekt gar nicht abschließen können, solange der Messestand noch abgebaut und transportbereit in Sydney steht. Die Bestätigung eines ordnungsgemäßen Projektabschlusses erhalten sie nur, wenn der Messestand wieder eingelagert worden ist. Sie ergänzen daher das Produkt "eingelagerter Messestand" und fügen dieses zwischen "abgebauter Messestand" und dem Projektendprodukt "Messe Sydney 2012" ein (vgl. Bild 2).



Bild 2: Der Perspektivenwechsel von Produktstruktur zum Produktfluss lässt Fehler schnell erkennen.

Wenn Sie ein neues Produkt identifizieren, tragen Sie dies unbedingt auch in den Produktstrukturplan ein! Auf diese Weise überprüfen Sie nochmals, ob der Produktstrukturplan wirklich vollständig ist.

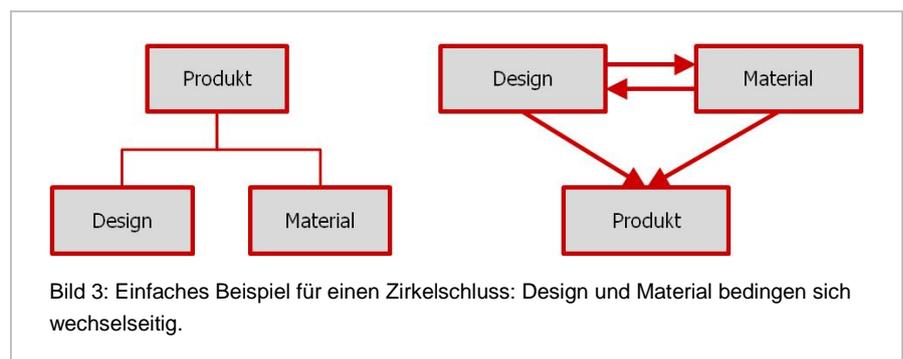
Identische Produktnamen verwenden

! Dass die Produktnamen in Produktstrukturplan und Produktflussdiagramm identisch sein müssen (Regel 3) erscheint auf den ersten Blick trivial, ist aber die am häufigsten verletzte Regel. Wenn die Planer beim Perspektivenwechsel intuitiv einen anderen Produktnamen verwenden, dann ist dies ein deutlicher Hinweis darauf, dass noch kein klares Bild über dieses Produkt besteht oder noch keine einheitliche Nomenklatur vorliegt.

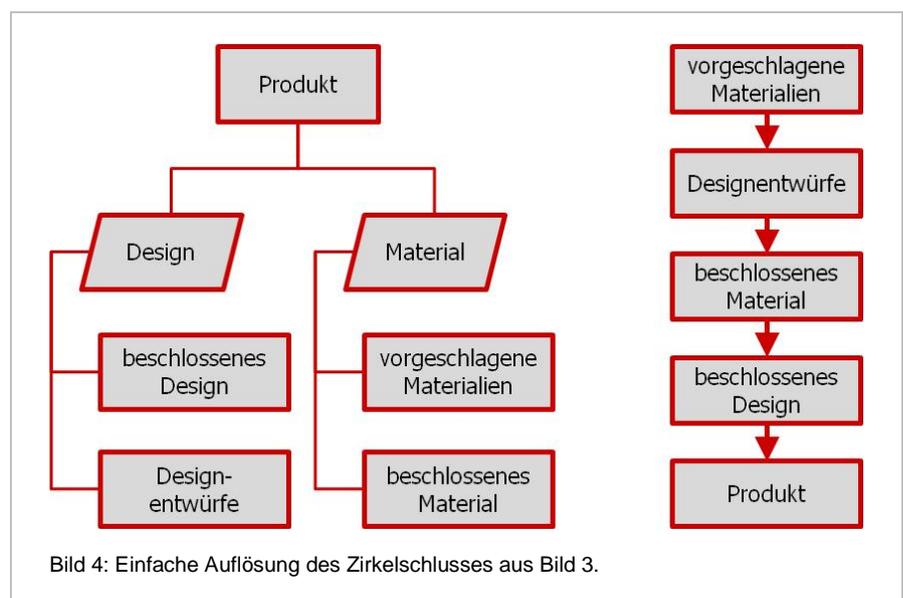
Beispiel: Bei Erstellung des Produktflussdiagramms stellt ein Teammitglied in Frage, ob es tatsächlich "abgearbeitete Kontaktliste" oder nicht besser "ins CRM übertragene Kontaktliste" heißen müsse. Hans Maier klärt deshalb mit der Vertriebsleiterin kurzfristig die Schnittstelle zwischen Messteam und Vertrieb ab: Die Übertragung ins CRM ist nicht mehr Aufgabe des Messteams, die Produktbezeichnung ist also korrekt.

Zirkelschlüsse auflösen

Wenn im Produktflussdiagramm Zirkelschlüsse (Regel 5) entstehen, dann muss mindestens eins der darin enthaltenen Produkte weiter zergliedert werden. Bild 3 zeigt ein typisches Beispiel für einen solchen Zirkelschluss: Das Design und die Materialauswahl für ein Produkt bedingen sich wechselseitig und müssen deshalb feiner strukturiert werden.



Um Zirkelschlüsse im Produktflussdiagramm aufzulösen, können Sie die Produkte in ihre Zustände während des Projektablaufs gliedern. Bild 4 zeigt, wie Design und Material als Gruppierungen in die Zustände "vorgeschlagene Materialien" und "beschlossenes Material" sowie "Designentwürfe" und "beschlossenes Design" strukturiert werden. Mit dieser Strukturierung ist dann auch ein regelkonformes Produktflussdiagramm möglich (s. Bild 4, rechts).



Zeitliche Abhängigkeiten darstellen

Für das in Bild 5 dargestellte Produktflussdiagramm braucht das Projektteam deutlich länger als für den Produktstrukturplan, da kontroverse Diskussionen darüber geführt werden, wann welches Produkt benötigt wird. Die vorbereitete Kontaktliste wurde z.B. zuerst nicht als notwendiges Produkt für den aufgebauten Messestand angesehen, schließlich, so ein Teammitglied ironisch, "würde der Messestand ohne Kontaktliste ja nicht in sich zusammenfallen".

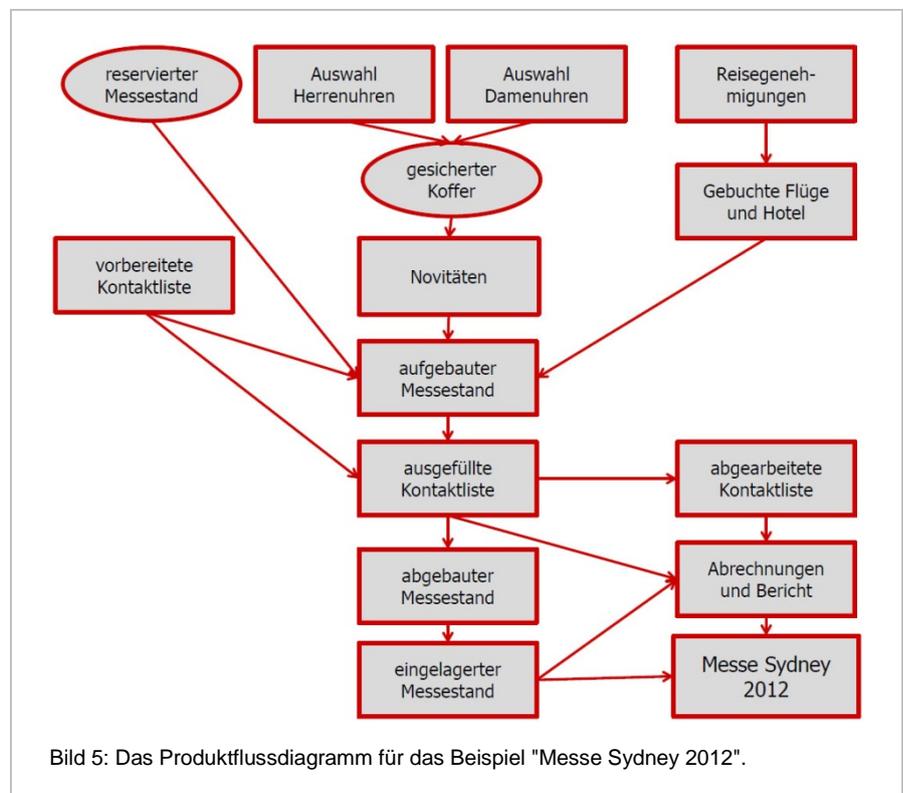
Bei diesem Argument handelt es sich um ein typisches Missverständnis bei der Erstellung des Produktflussdiagramms: Nicht die Zusammensetzung des Produkts, sondern die zeitliche Abhängigkeiten sollen dargestellt werden! Da die vorbereitete Kontaktliste, die Reisebuchungen und die Novitäten alle fertiggestellt sein müssen, bevor das Team nach Sydney fliegt, müssen diese Produkte auch alle vor dem aufgebauten Messestand im Produktflussdiagramm einsortiert werden.

Keine Aktivitäten einfügen

An diesem Beispiel wird auch eine Grenze des Produktflussdiagramms deutlich: Intuitiver wäre es natürlich, vor dem Produkt "aufgebauter Messestand" die Aktivität "Durchführung der Dienstreise" einzufügen. Dann wären die zeitlichen Zusammenhänge leichter aus dem Diagramm selbst heraus verständlich. Aktivitäten sind aber bei der PBP streng verboten, um nicht bereits im Vorhinein den Lösungsweg einzuschränken. Viele Planer führen deshalb formale Produkte ein, die Aktivitäten unter einem Produktdeckmantel einführen, z.B. könnte man ein Produkt "durchgeführte Dienstreise" einfügen, um den formalen Anforderungen der PBP gerecht zu werden.

Ich rate von diesem Trick aus zwei Gründen ab: Zum einen konterkariert dieses Vorgehen meiner Meinung nach die eigentliche Idee der PBP und bringt unnötige Komplexität in die Methode. Zum anderen behindert es die anschließende Erstellung des Projektplans, bei dem es explizit um die Definition von Arbeitspaketen bzw. Vorgängen geht (s.u.).

Die vier Schritte der PBP hängen sehr eng voneinander ab. Von daher kann das Planungsteam letztlich die Reihenfolge, in der sie durchgeführt werden, so wählen, wie es der Aufgabe am angemessensten erscheint. Z.B. benötigt man für die Erstellung der Produktbeschreibungen auch bereits das Produktflussdiagramm und umgekehrt helfen die Produktbeschrei-



bungen bei der Erstellung des Produktflussdiagramms. Zudem sind Produktstrukturplan und Produktflussdiagramm gleichberechtigte Darstellungen des Projektleistungsumfangs aus zwei verschiedenen Perspektiven. Die produktbasierte Planung zwingt die Planer dazu, einen Perspektivenwechsel vorzunehmen und sorgt auf diese Weise dafür, dass die Planung so vollständig wie möglich erfolgt.

! Betrachten Sie die vier Schritte der PBP nicht als voneinander isolierte Aufgaben! Es ist zu empfehlen, sie mindestens zweimal hintereinander zu durchlaufen, um sicherzustellen, dass es keine inneren Widersprüche gibt.

Schritt 4: Erstellen Sie mindestens für alle Einzelprodukte (d.h. nicht weiter zerlegten Produkte) eine Produktbeschreibung!

Der vierte Schritt ist sehr leicht erklärt, erfordert aber in der Regel den höchsten Arbeitsaufwand. Sie müssen für alle Produkte mit Ausnahme der Gruppierungen eine Produktbeschreibung erstellen, so wie ich dies im ersten Teil dieser Artikelserie bereits für das Projektendprodukt erläutert habe.

! Das Erstellen der Produktbeschreibungen ist meist ein kontinuierlicher, parallel ablaufender Prozess in der PBP, auch wenn er hier als vierter Schritt erklärt wird.

Es kann sein, dass Sie diese Produktbeschreibungen jetzt noch nicht vollständig erstellen können, z.B. weil Sie die Qualitätsverantwortlichkeiten erst später festlegen können. Es ist aber wichtig, sie jetzt bereits so vollständig wie möglich zu erstellen, da sie zentrale Informationen des Planungsprozesses mitdokumentieren.

Beschreiben Sie alle Produkte

Bei der Spezifikation der Produkte im Formalismus der Produktbeschreibung nach PRINCE2 führen Sie zugleich eine Qualitätssicherung des Produktstrukturplans und des Produktflussdiagramms durch. Da Sie u.a. den Zweck, die Herleitung, die Zusammensetzung und die Abnahme des Produkts beschreiben, überprüfen Sie damit zugleich, ob das Produkt korrekt strukturiert und richtig in der zeitlichen Reihenfolge eingeordnet ist.

Im PRINCE2-Handbuch wird angemerkt, dass es für ein kleines Projekt ausreichend sein kann, nur für das Projektendprodukt eine Produktbeschreibung zu erstellen (OGC, 2009, Abschnitt 7.3.3.3). Dies halte ich persönlich für sehr riskant und rate dazu, selbst bei einem Kleinprojekt mindestens die nicht weiter aufgegliederten Einzelprodukte (engl.: "single product") zu beschreiben.

Auch dies ist bereits eine Kompromisslösung: In den nächsten Planungsschritten müssen schließlich die Arbeitspakete erstellt werden. Für diese sind Produktbeschreibungen erforderlich, da sonst kein klarer Arbeitsauftrag formuliert werden kann. Auf eine Produktbeschreibung eines Integrationsprodukts (ein aus mehreren Produkten zu erstellendes Produkt, vgl. Teil 1) zu verzichten, halte ich nur dann für möglich, wenn die Beschreibung unmittelbar aus der Produktbeschreibung des Endprodukts hervorgeht.

Anmerkung: Das deutsche PRINCE2-Handbuch verwendet für "single product" die missverständliche Übersetzung "einfaches Produkt". Damit ist nicht gemeint, dass die nicht weiter zerlegten Produkte einfach zu erstellen seien, sondern dass sie eben nicht aus mehreren anderen Produkten zusammengesetzt sind. Deshalb verwende ich die treffendere Übersetzung "Einzelprodukt".

Produktbeschreibung für kleine Projekte als Tabelle

Statt auf Produktbeschreibungen zu verzichten, empfehle ich als wesentliche Arbeitserleichterung, bei Kleinprojekten die Produktbeschreibungen in Form einer komprimierten Tabelle zu erstellen. Z.B. können Sie alle das Qualitätsmanagement betreffenden Punkte der ausführlichen Produktbeschreibung (vgl. Teil 1) in einer Spalte unter der Überschrift "Qualität" zusammenfassen. Tabelle 1 zeigt, wie für ein Kleinprojekt die stark vereinfachten Produktbeschreibungen in einer Tabelle schnell und übersichtlich dokumentiert werden können.

Bezeichnung	Zweck	Spezifikation	Ersteller	Qualität
Vorbereitete Kontaktliste	Messevorbereitung	Kunden und Geschäftspartner aus Asia-Pacific, eingeladene Messegäste	Vertrieb	siehe Messehandbuch, Prüfung vor Abreise durch Leitung Messestand
Ausgefüllte Kontaktliste	Messedokumentation für Nachbereitung	Ergebnisse der Kontaktgespräche, neue Prospects	Messteam	Leitung des Messestands überprüft auf Vollständigkeit und Konsistenz
Abgearbeitete Kontaktliste	Dokumentation Vertriebskontakte, Übergabe an CRM	siehe Vertriebshandbuch	Messteam	Leiter Vertrieb überprüft Liste vor Übergabe an CRM
...

Tabelle 1: Komprimierte Produktbeschreibungen in Form einer Tabelle für Kleinprojekte.

Vom Produktflussdiagramm zum Terminplan

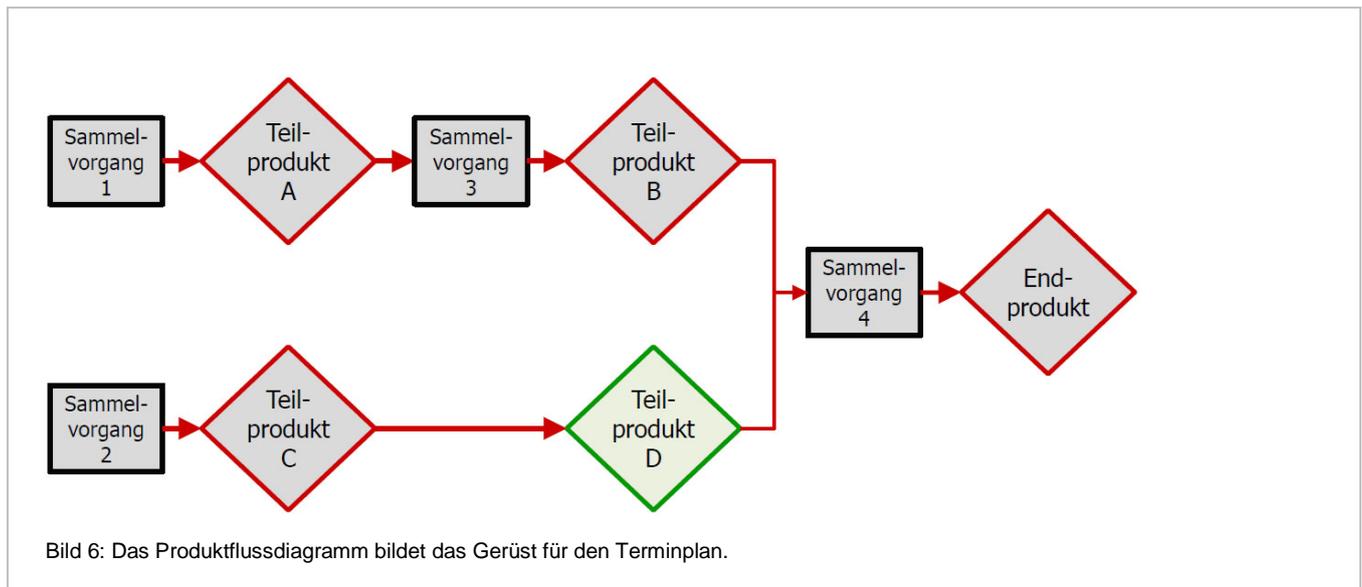
Aus dem Produktflussdiagramm lässt sich sehr einfach der Terminplan des Projekts ableiten. Leider zeigt das PRINCE2-Handbuch diesen Weg nicht mehr konsequent im Sinne der Produktorientierung auf und beschreibt die Erstellung des Terminplans nur sehr pauschal. Ganz offensichtlich bemühten sich die Autoren des Handbuchs, bei der Aktivitätenplanung möglichst unverbindlich zu bleiben, vielleicht um keine Reibungspunkte mit anderen Richtlinien entstehen zu lassen. Z.B. führt das PRINCE2-Handbuch die Möglichkeit auf, ergänzend zum Produktstrukturplan einen Projektstrukturplan zu erstellen, um daraus die Aktivitäten herzuleiten (OGC, 2009, Abschnitt 7.3.4.1). Diesen Vorschlag halte ich für wenig effizient und eine unnötige Erhöhung des Planungsaufwands.

Ich schlage als pragmatische Vorgehensweise bei der Terminplanung vor, das Produktflussdiagramm ganz einfach als Meilensteinplan zu verwenden und in diesen dann die Arbeitspakete bzw. Vorgänge einzufügen. Ein weiterer Strukturplan ist dann überflüssig.

Formell ist das Vorgehen sehr einfach:

- Übertragen Sie die Produkte aus Produktflussdiagramm als Meilensteine in einen Netzplan.
- Setzen Sie vor jedes interne Produkt einen Sammelvorgang.
- Identifizieren Sie, welche Aktivitäten notwendig sind, um die Produkte jeweils zu erstellen.
- Zerlegen Sie dementsprechend die Sammelvorgänge und ergänzen Sie bei Bedarf weitere Vorgänge.

- Verbinden Sie die neuen Vorgänge mit den erforderlichen Anordnungsbeziehungen. Diese können auch über die Meilensteine hinweggehen.
- Falls Sie dabei neue Produkte identifizieren, ergänzen Sie unbedingt auch Produktstrukturplan und Produktflussdiagramm.



Auf diese Weise erhalten Sie einen Terminplan, der eng mit der Spezifikation des Leistungsumfangs verflochten ist. Nun haben Sie eine verlässliche Basis für Aufwandsschätzungen und Ressourcenplanungen. Bild 6 illustriert das prinzipielle Vorgehen. Produkt D (grün dargestellt) benötigt keinen vorgeschalteten Sammelvorgang, da es ein externes Produkt ist.

Praktische Tipps für die produktorientierte Planung

Das hier beschriebene Vorgehen ist nur ein Teil der Projektplanung. Die folgenden Tipps sollen Ihnen helfen, die PBP in den bei Ihnen üblicherweise verwendeten Planungsprozess einzubinden.

Abbildung von Phasen in der PBP

Projekte laufen typischerweise in mehreren Phasen ab. Diese können auch mit der PBP abgebildet werden, indem Sie die Phasenfreigaben als Produkte behandeln. PRINCE2 nennt diese Art von Produkten "Managementprodukte", z.B. gehören auch der Projektplan oder ein Statusbericht zu den Managementprodukten.

Wenn Sie im Produktflussdiagramm Managementprodukte aufnehmen, dann empfiehlt es sich, diese bei der Übertragung in den Netzplan auf der obersten Planungsebene und die anderen Produkte auf den darunter liegenden Ebenen einzubinden.

Risiken identifizieren

Während der Projektplanung lassen sich Risiken am leichtesten identifizieren, da hier die Zusammenhänge zwischen den einzelnen Elementen des Plans analysiert werden. Besondere Aufmerksamkeit sollten Sie folgenden Punkten widmen, da diese typische Frühindikatoren für Projektrisiken sind:

- Externe Produkte sind nicht durch das Projekt beeinflussbar. Ihre mögliche Nicht-Verfügbarkeit stellt somit automatisch ein Risiko dar.
- Je mehr Produkte im Produktflussdiagramm für die Erstellung eines Produkts erforderlich sind, desto höher sind für dieses Produkt die Risiken für Zeit, Qualität und Kosten.
- Können bei einer Produktbeschreibung qualitätsrelevante Punkte noch nicht ausgefüllt werden, ist dies als Ursache für möglicherweise projektgefährdende Risiken zu behandeln, da damit weder Zeit noch Kosten verlässlich abgeschätzt werden können und somit auch der Nutzen des Projekts gefährdet ist.

Treffende Kurzbezeichnungen für die Produkte verwenden

Verdeutlichen Sie den abnahmerelevanten Zustand der Produkte durch Adjektive oder Partizipien in der Produktbezeichnung so klar wie möglich. Die Qualifizierung von Leistungen und Teilleistungen durch Attribute sorgt dafür, dass über den Zustand Klarheit herrscht, der durch das Projekt herbeigeführt werden soll. Vermeiden Sie triviale, nichts besagende Adjektive wie z.B. "erstellt" oder "fertig". Besonders gefährlich sind scheinbar qualitätsbeschreibende Adjektive wie "hochwertig" oder "bestmöglich", da sie eine nicht vorhandene Präzision vorgaukeln. Auch wenn es sprachlich langweilig wirkt, sollten Sie statt dessen besser eindeutige Bezeichnungen wie z.B. "übergeben", "geprüft" oder "abgenommen" verwenden, die klar den projektrelevanten Status des Produkts definieren.

Anforderungen an Planungs-Software für PBP

Grundsätzlich ist die PBP eine Papier-und-Stift-Methode, d.h. es ist für sie keine Software notwendig. Für den Projektplan hingegen kann eine Planungssoftware sehr sinnvoll sein, insbesondere dann, wenn der Plan weiter detailliert wird und im Laufe des Projekts mehrmals aktualisiert werden muss. Unverzichtbar wird die Software-Unterstützung, sobald Ressourcenmanagement und Aufwandserfassung mit dem Projektplan synchronisiert ablaufen sollen.

Einfache Terminplanungssoftware kann bei der PBP nur eingeschränkt verwendet werden, da sie erst ab dem Produktflussdiagramm die benötigten Funktionen bietet. Leistungsfähigere PM-Software-Produkte haben auch die Funktionalität eines vom Terminplan unabhängigen Projektstrukturplans, manche sogar einen expliziten Produktstrukturplan. Eine vollständige Abbildung der PBP in einer PM-Software ist möglich, sobald diese darüber hinaus über ein Dokumentenmanagement verfügt, das eine Zuordnung von versionierten Dokumenten zu einzelnen Elementen von Strukturplan und Ablaufplan erlaubt.

Dies sind objektiv gesehen keine großen Ansprüche, leider erfüllen sie nur wenige Produkte auf dem Markt. Wenn Sie nach einer PM-Software suchen, sollten Sie deshalb die von Ihnen in Betracht gezogenen Produkte kritisch auf diese Anforderungen hin überprüfen. Nur deswegen, weil ein Software-Anbieter behauptet, dass sein

Produkt PRINCE2-konform ist, heißt dies noch lange nicht, dass es tatsächlich in der Lage ist, die Prozesse und Methoden von PRINCE2 abzubilden.

Fazit: Produkte vereinfachen vieles!

Der Ansatz der produktbasierten Planung entspricht in vielen Dingen der traditionellen Herangehensweise der Auftragsklärung mit Lastenheft und Pflichtenheft. Alle Produktbeschreibungen zusammengenommen sind ein ausführliches und mit Managementinformationen erweitertes Leistungsverzeichnis. Gemeinsam mit dem Produktflussdiagramm enthalten sie die gleichen Informationen wie ein Lastenheft. Der vollständige Projektplan mit Anmerkungen über Annahmen und Voraussetzungen ist das Pendant zum Pflichtenheft.

Die produktbasierte Planung ist damit letztlich nur eine systematische und risikominimierende Methode der Lastenheftentwicklung, weswegen sie auch problemlos unabhängig von PRINCE2 verwendet werden kann.

Das Vorgehen der PBP widerspricht zwar zunächst dem intuitiven "Ärmel aufkrempeln und anpacken". Sie liefert aber als Ergebnis einen stabilen Projektplan, der sich durch folgende Eigenschaften auszeichnet:

- Er kann, ohne Widersprüche zu erzeugen, weiter detailliert werden.
- Er ist für alle Projektbeteiligten intuitiv nachvollziehbar.
- Er ist unmittelbar mit dem Leistungsumfang des Projekts verknüpft.
- Er verbindet Terminplanung und Kostenmanagement mit Qualitätsmanagement, Risikomanagement und Ressourcenmanagement.

Literatur

- Angermeier, Georg: **Die Produktbasierte Planung nach PRINCE2. Teil 1: Produktbeschreibung und Produktstrukturplan**, Projekt Magazin 4/2012
- Office of Government Commerce (OGC): **Erfolgreiche Projekte mit PRINCE2**, 2008, deutsche Übersetzung der vierten Ausgabe des Handbuchs "Managing Successful Projects with PRINCE2", 2005
- Office of Government Commerce (OGC): **Erfolgreiche Projekte managen mit PRINCE2™**, 2009, deutsche Übersetzung der fünften englischen Ausgabe von "Managing Successful Projects with PRINCE2", 2009

Methode

Klar vereinbarte Leistungen – zuverlässige Pläne

Produktorientierte Planung von IT-Projekten

Eine der wichtigsten und bekanntesten Ursachen für Schwierigkeiten in IT-Projekten ist, dass Auftraggeber und Auftragnehmer vor dem Projektstart nicht ausreichend Klarheit über die zu erstellenden Ergebnisse geschaffen haben. Es ist relativ einfach, ein Budget (z.B. 0,5 Mio €) oder einen Fertigstellungszeitpunkt (z.B. 2015-01-31) genau anzugeben. Wesentlich schwieriger gestaltet es sich, alle Komponenten des Produkts genau zu benennen.

Wenn z.B. der Abteilungsleiter eines Fachbereichs eine Software kennenlernt, von der er sich eine deutliche Effizienzsteigerung verspricht, stellt er schnell entschlossen einen Projektantrag an die hauseigene IT-Abteilung. Im diesem Antrag beschreibt er, wie er diese Software zur Unterstützung eines Geschäftsprozesses einsetzen möchte, er definiert ein Budget und einen Zeitrahmen. Eventuell ergänzt er noch einen Business Case, in dem er die Nutzeffekte überaus optimistisch darstellt. Weitere Fachexperten oder Vertreter der betreuenden IT-Abteilung zieht er nur selten hinzu. Die Folge davon ist, dass z.B. die Migration von Datenaltbeständen, der Aufbau und die Anpassung von Schnittstellen zu anderen Systemen oder Einführungs- und Trainingsaufwände völlig außer Acht gelassen werden.

Im Laufe des Projekts werden diese Themen unweigerlich hochkommen und zu Diskussionen zwischen Antragsteller und dem Projektleiter über benötigtes zusätzliches Budget und eine Ausdehnung des geplanten Zeitrahmens führen: Das Projekt gerät in Schieflage oder droht sogar zu scheitern. Der "schwarze Peter" liegt dann meist beim Projektleiter, der sich rechtfertigen muss, warum er nicht zeit- oder kostengerecht alle nun gewünschten Produkte und Eigenschaften liefern kann (Enderle da Silva, 2013).

Als verantwortlicher Manager für Prozesse und Tools im PMO der Abteilung Information Management der Airbus Defence and Space (vorher: Cassidian als Tochter der EADS) suche ich beständig nach Verbesserungsmöglichkeiten für das Projektmanagement des Bereichs Information Management (IM). Deshalb analysierte ich einzelne in Turbulenzen geratene Projekte und stellte dabei fest, dass genau die oben beschriebene Problematik in der Vergangenheit zu den Schwierigkeiten führte. Hierzu zwei weitere Beispiele:

- Es wird vergessen oder nur halbherzig berücksichtigt, dass bei einem internationalen Roll-Out in den einzelnen Ländern genügend Zeit und Ressourcen für die Vorbereitung und Durchführung der Anwendertrainings in der jeweiligen Landessprache zur Verfügung stehen, was entscheidend zur Akzeptanz beiträgt.
- Die Kosten und der Aufwand zum Aufbau der späteren operativen Services (Kalkulation der Servicekosten, Aufnahme in den Servicekatalog, Berücksichtigung der Kosten in das Fachbereichsbudget, Einweisung des Help Desks, Bereitstellung der benötigten Experten im Fehlerfall, usw.) werden nicht oder nur unzureichend berücksichtigt.

Autor



Winfried Vogel

Dipl.-Kaufmann, zert. als
Organisator, Berater und
Black Belt.

Kontakt: winfried.vogel@gmx.de

Mehr Informationen unter:

projektmagazin.de/autoren

Lösungsansatz: Produktorientierte Planung

Um alle Projektmanagementaktivitäten der Abteilung Information Management (IM) bei Airbus Defence & Space umfassender zu koordinieren und im Sinne eines kontinuierlichen Verbesserungsprozesse zu optimieren ist ein eigenes PMO (IM PMO) installiert worden, das hinsichtlich Kosten, Termine und Ergebnisse für alle Projekte des Bereiches IM (Information Management) verantwortlich ist und auch die Projektmanager disziplinarisch führt. Das IM PMO verfügt über ca. 35 Projektleiter. Die Projekte werden über einen definierten Projektportfolioprozess (s.u.) beantragt und gestartet.

Bei der Analyse erkannte ich, dass genau durch diesen Prozess die Projektleiter einem hohen Erwartungsdruck durch die Auftraggeber ausgesetzt sind, vor allem dann, wenn sie unvorbereitet ein sofort umzusetzendes Projekt übernehmen sollen. Die Position der Auftraggeber lässt sich pointiert so formulieren: "Was diskutiert ihr noch lange herum, die Software ist doch bereits gekauft! Macht doch endlich vorwärts, damit wir mit unserer Anwendung arbeiten können!"

In der Folge konzentrierten sich die Projektleiter in bestimmten Fällen darauf, wie sie die unmittelbaren technischen Aspekte umsetzen konnten. Bei der Planung stand die Work Breakdown Structure (WBS) mit ihren Arbeitspaketen, also die Umsetzungsbausteine, im Mittelpunkt. Dabei vergaßen sie teilweise, die Gesamtheit aller Lieferbausteine vorher zusammen zu tragen und mit dem Kunden klar und einvernehmlich abzustimmen.

Den Lösungsansatz für dieses Dilemma fand ich in der produktbasierten Planungsmethode des britischen Projektmanagementsystems PRINCE2® (OGC, 2009). Angeregt durch den Artikel "**Produktbasierte Planung nach PRINCE2**" im Projekt Magazin (Angermeier, Projekt Magazin 4/2012 und 5/2012) entwickelte ich eine standardisierte Struktur für die Liefergegenstände eines IT-Projektes und führte sie beim IM PMO ein.

Produktstruktur eines IT-Projekts

Im Folgenden beschreibe ich eine systematische und strukturierte Vorgehensweise beim Projektinhaltsmanagement (im Folgenden "Scope-Management"), die ich bei IT-Projekten erfolgreich eingesetzt habe, um die oben beschriebenen Störungen im Projekt zu verhindern oder zumindest weitestgehend zu vermindern.

Ein IT-Projekt hat typischerweise die folgenden Ergebnis-Komponenten:

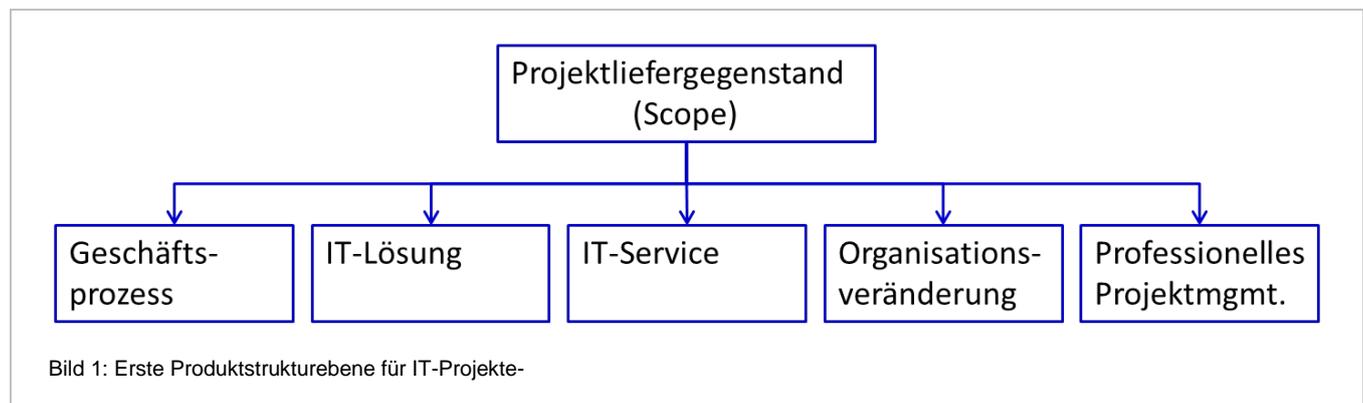
- die eigentliche IT-Lösung, z.B. eine neu installierte Software für das Produktmanagement
- der dazu gehörende IT-Service (Wartung und Support)
- der Geschäftsprozess, den die IT-Lösung unterstützen soll, z.B. die Dokumentation neuer Produktversionen. Dieser Prozess muss als verbindlich anzuwendender Standard (neu) definiert und beschrieben werden.
- die Einführung des neuen bzw. veränderten Geschäftsprozesses sowie der dazu gehörenden IT-Lösung in den Unternehmensalltag. Diese Einführung wird auch "Transformation Management" oder "Organizational Change Management" genannt.

Weiterhin soll das IT-Projekt durch ein professionelles und transparentes Projektmanagement gesteuert werden, so dass diese Ergebnisse erfolgreich erzielt werden. Das Projektmanagement hat zudem sicherzustellen, dass alle formalen Erfordernisse wie z.B. Unternehmensrichtlinien, technische Standards sowie erforderliche Geneh-

migungen wie Betriebsvereinbarungen mit dem Betriebsrat, Genehmigung der IT-Architektur und Abstimmungen mit den Infrastruktur- und Service-Verantwortlichen inhaltlich und formgerecht durchgeführt werden.

Als erste Strukturebene für die Liefergegenstände eines IT-Projekts unterhalb des zu spezifizierenden Projektziels bzw. Projektgegenstands habe ich deshalb für den Produktstrukturplan (Product Breakdown Structure = PBS) folgende fünf Teilprodukte definiert (vgl. Bild 1):

1. Geschäftsprozess
2. IT-Lösung
3. IT-Service
4. Organisationsveränderung
5. Professionelles Projektmanagement



Meiner bisherigen Erfahrung nach dürfte sie auf fast alle Arten von IT-Projekten passen. Sie können diese Struktur natürlich an Ihr eigenes Projektumfeld anpassen. Wichtig ist, dass Sie bei einer Anpassung alle Elemente vorurteilsfrei prüfen und dann begründet ein Element ausschließen oder weitere Elemente hinzufügen. Dokumentieren Sie diese Begründungen, damit Sie bei einem späteren Review nachvollziehen können, warum der Produktstrukturplan diese Form hatte.

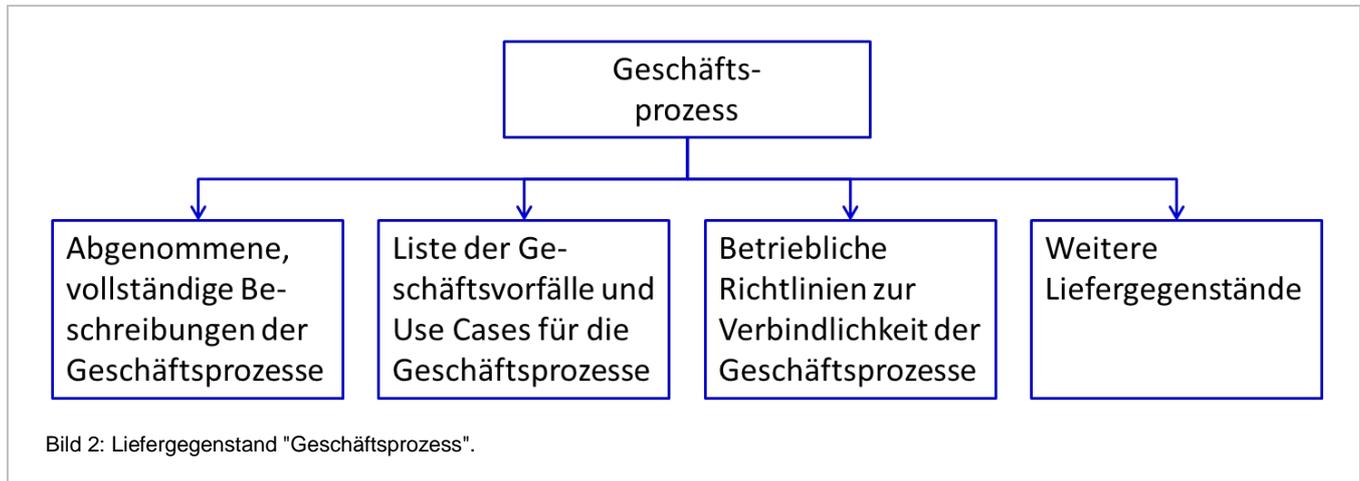
Diese fünf Teilprodukte sind bei der Projektplanung weiter zu strukturieren. Im Folgenden beschreibe ich aus meiner Erfahrung kurz ihre wesentlichen Inhalte. Diese können Sie als eine Art Checkliste für die Definition Ihrer eigenen Liefergegenstände verwenden.

Geschäftsprozess

Der Liefergegenstand "Geschäftsprozess" umfasst:

- abgenommene Geschäftsprozessbeschreibung in Form eines Flussdiagramms mit Rollenbeschreibungen und definierten Verantwortlichkeiten
- Liste der Geschäftsvorfälle und Use Cases, die mit diesem Geschäftsprozess abzudecken sind

- betriebliche Richtlinien und Anweisungen, die den Geschäftsprozess zu einem verbindlichen Bestandteil der Betriebsführung machen
- weitere Liefergegenstände wie z.B. Betriebsvereinbarung, Dateneinstufung, ...

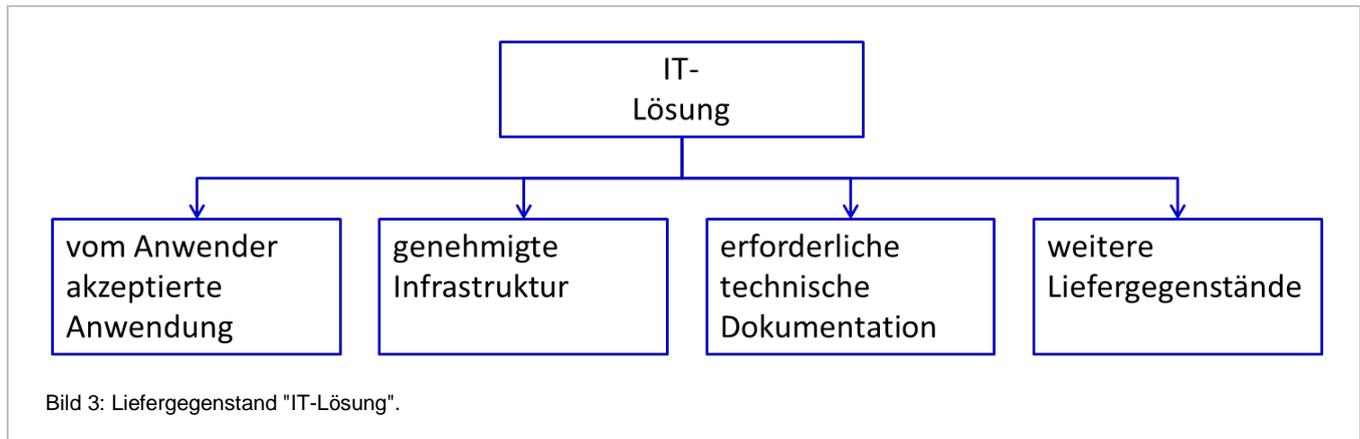


Bei reinen Infrastrukturprojekten entfällt häufig die Beschreibung des Geschäftsprozesses. Jedoch sind auch bei Infrastrukturprojekten zumindest IT-interne Prozesse zu beachten, die je nach Aufgabenstellung verändert, abgeschafft bzw. neu erstellt werden. Man kann die Behandlung dieser Prozesse sowohl beim Teilprodukt "Geschäftsprozess" als auch als weiteren Punkt unter das Teilprodukt "IT-Service" (s.u.) einordnen. Ausschlaggebend dabei ist, dass bei der Definition des Projektauftrags diese Liefergegenstände und die mit ihnen verbundenen Veränderungen berücksichtigt werden.

IT-Lösung

Der Liefergegenstand "IT-Lösung" besteht aus der eigentlichen informationstechnischen Lösung. Diese umfasst unter anderem:

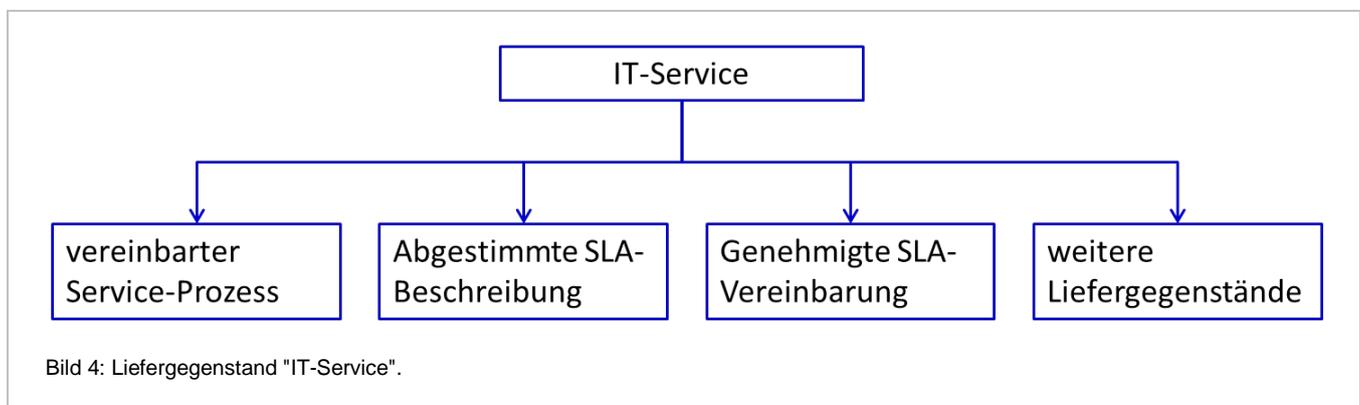
- die vom Anwender akzeptierte Applikation / Anwendung
- die vom entsprechenden Verantwortlichen genehmigte Infrastruktur
- die notwendige technische Dokumentation bzw. Handbücher
- weitere Liefergegenstände



IT-Service

Der Liefergegenstand "IT-Service" beschreibt die notwendigen IT-Services und umfasst in der Regel:

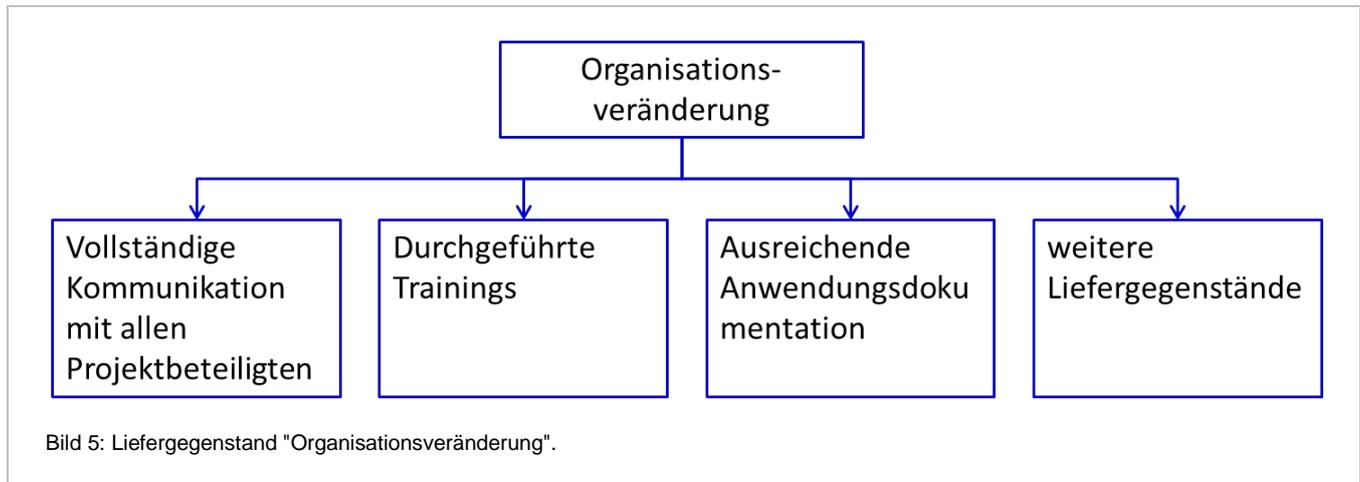
- die mit den Anwendern und der für den IT-Betrieb zuständigen Abteilung vereinbarten Service-Prozesse
- die mit den Beteiligten abgestimmte Beschreibung der Service Level Agreements (SLA)
- die genehmigte und mit Preisen versehene Vereinbarung über die SLA sowie die SLA-Verrechnungsregeln
- weitere Liefergegenstände



Organisationsveränderung

Der Liefergegenstand "Organisationsveränderung" berücksichtigt die notwendige Kommunikation an die Betroffenen und die erforderlichen Trainings:

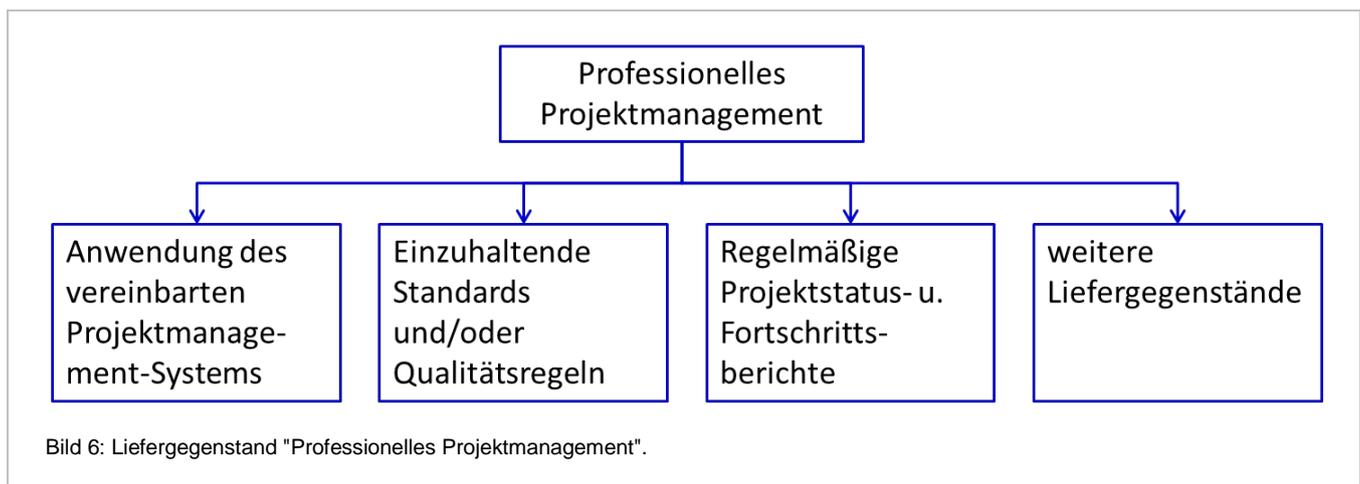
- regelmäßige und umfassende Kommunikation an alle / mit allen Projektbeteiligten
- durchgeführte Trainings/Schulungen mit entsprechenden Trainings-/Schulungsmaterialien
- ausreichende Anwendungsdokumentation für die Benutzer
- weitere Liefergegenstände



Professionelles Projektmanagement

Der Liefergegenstand "Professionelles Projektmanagement" beschreibt z.B. die zu verwendende Methodik und/oder deren Anpassung sowie die ggfs. einzuhaltenden (technischen) Standards oder Qualitätsregeln:

- Anwendung des vereinbarten bzw. vorhandenen Projektmanagement-Systems und seine Anpassung an das Projekt
- einzuhaltende Standards und/oder Qualitätsregeln
- regelmäßige Berichterstattung über Projektstatus und Fortschritt
- weitere Liefergegenstände



Mit Hilfe der oben aufgeführten Gruppen von Liefergegenständen lassen sich einfach, umfassend und frühzeitig alle Liefergegenstände identifizieren, diskutieren und dokumentieren. Damit erreichen wir schon bei der Projektdefinition eine hohe Vollständigkeit der Liefergegenstände und damit eine zuverlässige Grundlage, um die Größe und den Umfang eines Projekts abschätzen zu können. Dies ist ein großer Vorteil bei der Formulierung der Projektanfrage im Projektportfolioprozess.

Einbindung der PBS in das Projektportfoliomanagement

Abhängig von Reifegrad und Projektkultur des Unternehmens durchlaufen Projektanträge ein Genehmigungsverfahren. Bei uns gibt es hierfür einen definierten Prozess, der von einer divisionsweiten Software unterstützt wird und für den der Projektportfoliomanager verantwortlich ist.

Der typische Entstehungsweg eines Projekts beginnt damit, dass ein Fachbereich beim IM PMO nach einer "projektähnlichen Leistung" anfragt. Der zuständige Kundenbereichsbetreuer nimmt diese Anfrage auf. Zusammen mit dem Kunden aus dem Fachbereich, dem Bedarfsträger, formuliert er sie aus und bewertet sie finanziell. Anschließend wird der Antrag nach der sog. "100/100/Komplexitätsregel" beurteilt: Werden die Kosten für das Vorhaben auf mehr als 100 k€, sein Aufwand auf mehr als 100 Manntage oder seine "Komplexität" als hoch beurteilt, dann ist es ein Kandidat für ein Projekt, ansonsten zählt es nur als "Erweiterung" und wird nach einem vereinfachten Verfahren genehmigt. "Komplex" bedeutet in diesem Zusammenhang, dass mehrere Fachbereiche involviert sind, viele Applikationen gleichzeitig betroffen sind und/oder innovative Inhalte nachgefragt werden.

Der Projektportfoliomanager ist für die als "Projekt" eingestuftten Vorhaben verantwortlich (s. Bild 7). Er validiert die Projektanfrage und vervollständigt sie in Zusammenarbeit mit dem verantwortlichen Kundenbereichsbetreuer.

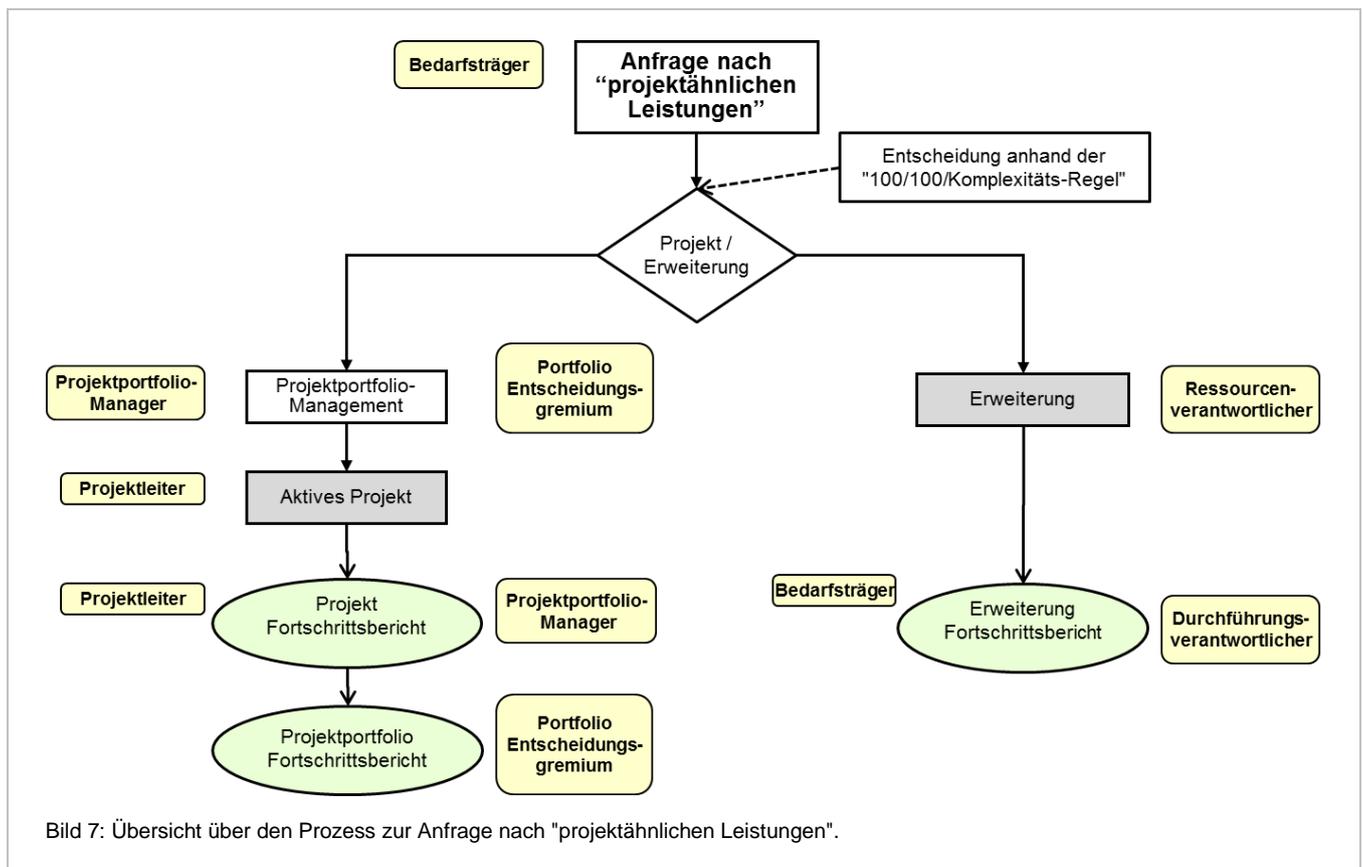


Bild 7: Übersicht über den Prozess zur Anfrage nach "projektähnlichen Leistungen".

Bei Bedarf zieht er weitere Experten hinzu. Zentrales Qualitätskriterium für die Vollständigkeit der Projektanfrage ist, dass die Produktstruktur vollständig ausgefüllt ist. Dadurch wird jede Anfrage auf ein vergleichbares Qualitäts-

niveau gehoben und es wird sichergestellt, dass bei einer ersten Kostenschätzung und Budgetierung alle wichtigen Kostenbereiche und -elemente berücksichtigt werden.

Auf diese Weise sind im Projektportfolio-Prozess bereits die ersten Planungsansätze weitgehend vollständig und aussagekräftig hinsichtlich der Arbeitsaufwände (benötigte Human Resources), des Investments, des Budgets sowie der Zeitplanung und der sich ggfs. daraus ergebenden Abhängigkeiten und Einflussgrößen des Portfolios.

Im anschließenden Genehmigungsprozess wird die aufbereitete Projektanfrage auf technische Umsetzbarkeit geprüft sowie nach weiteren Kriterien bewertet und dem Portfolio-Entscheidungsgremium vorgelegt. Dieses kann eine Anfrage ablehnen, genehmigen oder auf die Warteliste setzen, z.B. wenn Engpässe bei den benötigten Ressourcen oder dem Portfolio-Budget bestehen.

Liegt ein genehmigtes Projekt vor, dann wird zum definierten Startzeitpunkt ein Projektmanager benannt, der dieses Projekt verantwortlich übernimmt. Eine seiner ersten und wichtigsten Aufgaben ist es, den Projektauftrag auf Basis der bereits erstellten Beschreibung inklusive der PBS für eine Umsetzung weiter zu verfeinern und auch eine geeignete erste Projektplanung als Referenzdokument zu erstellen und zu vereinbaren.

Planungsprozess mit der PBS

Die Kunden-PBS dient während des Projekts als Baseline für den vereinbarten Leistungsumfang sowie als Abnahme-Checkliste für Projektleiter und Auftraggeber am Ende des Projekts. Änderungsanträge während des Projekts (Project Change Requests) werden mit dieser PBS als Scope-Baseline verglichen.

Die PBS dient in den weiteren Schritten (s. Bild 8) zur Ableitung der Work Breakdown Structure (WBS), die zusammen mit der Terminplanung beschreibt, wie, mit wem und zu welchen Kosten das Projekt durchgeführt wird.

Die Ableitung einer WBS aus der PBS und die Projektzeitplanung sind lediglich die logischen nächsten Schritte einer traditionellen Planung der Projektdurchführung. Diese Projektdurchführung nach den in Bild 8 dargestellten Schritten werde ich im Folgenden beschreiben.



Erster Schritt: Kunden-PBS

Der benannte Projektleiter erstellt mit der aus dem Portfolioprozess kommenden ersten PBS-Definition die sog. Kunden-PBS. Hierzu stimmt er sich mit dem Auftraggeber und den wichtigsten Beteiligten bzw. Betroffenen ab.

Die Kunden-PBS ist in der Sprache des Kunden formuliert, dennoch beschreibt sie ausreichend präzise die zu erstellenden Liefergegenstände, so dass Auftraggeber, Projektleiter und ggf. weitere Beteiligte ein gemeinsames Verständnis von ihnen erhalten. Die Kunden-PBS benennt die abzuliefernden Leistungen auf einer relativ hohen Integrationsebene und enthält keine Zwischenprodukte wie z.B. eine technische Serverspezifikation oder ähnliches. Typische Hilfsmittel für die Erstellung der Kunden-PBS sind eine Mindmap oder ein hierarchisches Baumdiagramm (z.B. Organigramm in Powerpoint). Diese Darstellungen eignen sich sehr gut für das gemeinsame Erstellen, da sie sich leicht anpassen lassen.

Größte Bedeutung bei der Kunden-PBS hat die Vollständigkeit, da wir uns am Anfang eines Projekts befinden und den Lieferrahmen klar abstimmen wollen.

Die Kunden-PBS:

- ist in einer an den Anforderungen des Kunden orientierten und für ihn verständlichen Form abgefasst
- enthält zielorientiert nur finale Liefer-/Leistungsgegenstände und deren Beschreibung
- definiert vollständig die Baseline für den Scope und dessen Gliederungsstruktur

Das nachfolgende Beispiel einer Kunden-PBS gliedert einen neuen Geschäftsprozess im Bereich Kundenmanagement (CRM, Customer Relation Management), der verbindlich eingeführt werden soll. Dazu soll eine Applikation mit den notwendigen IT-Services bereitgestellt werden, die diesen Geschäftsprozess unterstützen. Ferner soll ein entsprechendes Training mit den Benutzern in allen Ländern durchgeführt werden. Das Projekt soll durch ein professionelles Projektmanagement gemäß den gegebenen Regeln zum Erfolg geführt werden (s. Bild 9).

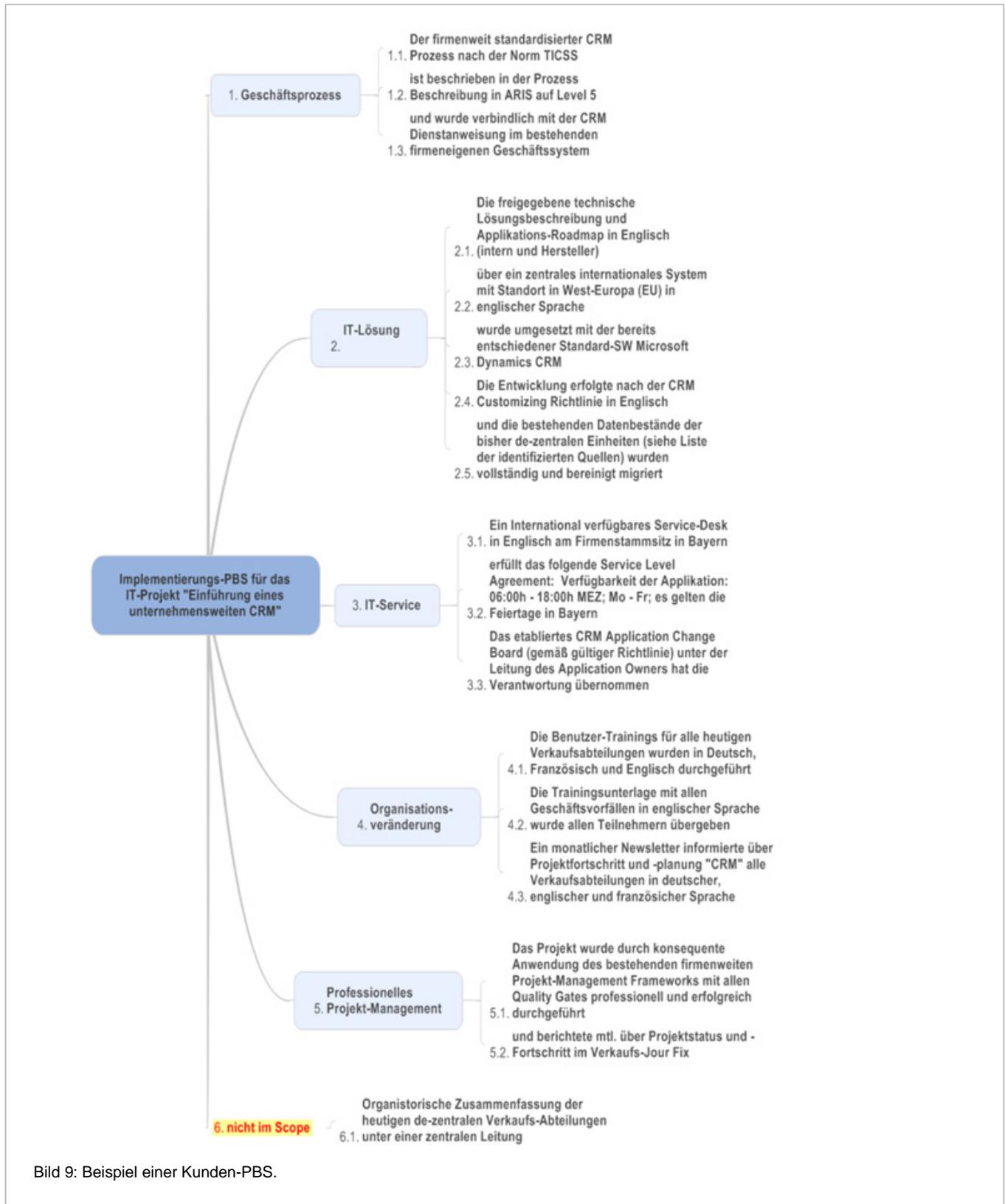


Bild 9: Beispiel einer Kunden-PBS.

Zweiter Schritt: Implementierungs-PBS

Nachdem er mit dem verantwortlichen Auftraggeber die Kunden-PBS verabschiedet hat, erstellt der Projektleiter im nächsten Schritt die Implementierungs-PBS. Dabei gliedert er die in der Kunden-PBS definierten Liefergegenstände in ihre technischen und formal notwendigen Elemente. Hierzu holt er sich gezielt qualifizierte und erfahrene Experten aus allen benötigten Fachbereichen in das Kernteam seines Projekts. Zusammen mit ihnen erarbeitet der Projektleiter eine vollständige "Stückliste", die alle notwendigen Komponenten und Zwischenprodukte enthält, um die Gesamtleistung erbringen zu können. Dabei sind insbesondere alle formalen Aspekte wie z.B. die Genehmigung der IT Architektur, sowie die Einhaltung bestehender Normen und Standards oder auch das Einbeziehen weiterer Verantwortungsbereiche wie z.B. der Datenschutzbeauftragte oder der Betriebsrat zu berücksichtigen. Bild 10 zeigt den Ausschnitt einer auf diese Weise entstandenen Implementierungs-PBS.

Liegt die Kunden-PBS als elektronische Mindmap vor, lässt sie sich gut in ein Tabellenkalkulationsprogramm überführen und dort um die Details der Stücklistenelemente erweitern. Dazu entwickelten wir in Microsoft Excel ein ausführliches, internes Template, welches bereits alle typischen Stücklistenelemente enthält. Nach dem gleichen Prinzip funktioniert Accelerated SAP, ein Werkzeug der SAP AG für die standardisierte Einführung von SAP (SAP, 2014 und Knauer, 2014).

Die Implementierungs-PBS

- ist erstellungsprozessorientiert
- enthält alle Elemente der Kunden-PBS
- gliedert die benötigten Elemente aus technischer Sicht mit ihren Spezifikationen
- stellt alle benötigten Elemente vollständig zusammen, inklusive aller Zwischenprodukte und formalen Genehmigungen, die entweder aus dem organisatorischen Rahmen für das Projektmanagement kommen (z.B. architekturelle oder betriebliche Compliance Freigaben) oder sich aus den Unternehmensgegebenheiten (Betriebsrat, Produktmanagement, ...) und/oder aus technischen bzw. gesetzlichen Erfordernissen (Normen, Standards, ..) ergeben
- kennzeichnet alle beabsichtigten Zukäufe (Hardware, Software, Service-Leistungen)
- zeigt bestehende Abhängigkeiten im Sinne einer notwendigen (technischen) Erstellungsreihenfolge auf

Dritter Schritt: Work-Breakdown-Structure (WBS)

Im nächsten Schritt geht es darum, die dazu notwendige "Arbeit" und die dazu erforderlichen Qualifikationen zu ermitteln, um mit den Umsetzungsverantwortlichen über die benötigten Ressourcen zu verhandeln. Die Kunden-PBS führt alle Elemente auf, die das spätere Endprodukt ausmachen und die Implementierungs-PBS listet in einer Art Stückliste alle Bausteine auf, aus denen das Produkt "zusammengebaut" werden soll. Daraus leitet der Projektleiter nun die Work-Breakdown-Structure ab.

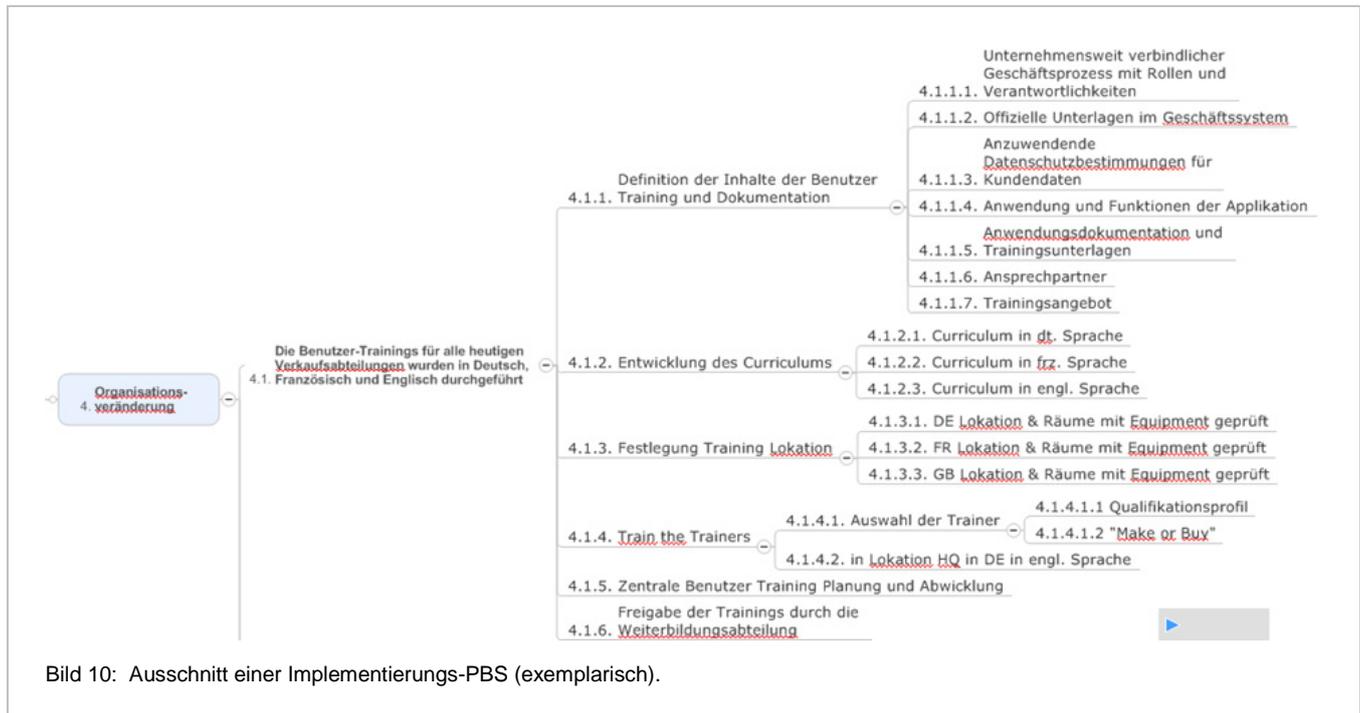


Bild 10: Ausschnitt einer Implementierungs-PBS (exemplarisch).

Gemeinsam mit seinem Projektteam untersucht der Projektleiter jedes einzelne Element der Implementierungs-PBS und definiert die zur Erstellung notwendigen Schritte und Maßnahmen. Für jedes Element der Implementierung-PBS werden so Aufgaben und Arbeitspakete definiert. Zu diesen Arbeitspaketen ermittelt der Projektleiter die notwendigen Skills und die organisatorischen Zuständigkeitsbereiche für die Umsetzung bzw. Erstellung. Der Projektleiter ist damit in der Lage, auf die Verantwortlichen dieser Zuständigkeitsbereiche zuzugehen und den benötigten Ressourcenbedarf für das Projekt und die Beteiligung an der Erstellung der Arbeitspakete qualifiziert zu diskutieren und grundsätzlich zu vereinbaren.

Geeignete Werkzeuge sind hierfür die typischen Software-Tools für die Projektplanung. Wir verwenden Microsoft Project, in das wir die Implementierungs-PBS als Excel-Datei importieren. Die traditionellen Methoden der Projektarbeit (z.B. gemäß PMBOK® Guide) zur Erarbeitung einer WBS kommen hier zum Einsatz.

Die Work Breakdown Structure (WBS):

- ist erstellungsorientiert
- definiert grob die benötigten Erstellungsschritte für die einzelnen Arbeitspakete
- ordnet die Verantwortlichkeitsbereiche (für die Erstellung) im Unternehmen zu
- ermittelt das erforderliche Know-how sowie den notwendigen Qualifikations- und Erfahrungslevel der Ressourcen
- enthält Aufwandsschätzungen
- ist in zeitliche Lieferabschnitte gegliedert, enthält jedoch noch keine konkreten Termine wie Erstellungs- und Lieferzeitpunkte
- identifiziert technische Risiken und Ressourcenengpässe

Nehmen wir den Punkt 4.1.4. "Train the Trainer" aus dem Beispiel in Bild 10 heraus, so hat das Projektkernteam dementsprechend unter anderem folgende Punkte zu erarbeiten:

- Welche Organisationseinheit soll den/die Trainer (Fachbereich, HR Weiterbildung, ...) stellen oder sollen externe Trainer beauftragt werden?
- Anforderungsprofile eines Trainers, wie z.B. Trainer-Ausbildung, praktische Erfahrung mit dem Thema, Akkreditierung im Unternehmen usw.
- Aufwandsschätzungen der mitwirkenden Experten im Projektkernteam für die Bereitstellung der Trainer
- Zielkorridore für den Liefertermin (das Arbeitspaket selbst enthält dabei jedoch noch keine konkreten Termine wie Erstellungs- und Lieferzeitpunkte)
- In der Diskussion mit den Ressourcenverantwortlichen wird die Umsetzbarkeit vor dem Hintergrund der vorhanden und verfügbaren Mitarbeiter und deren Qualifikation abgestimmt. Ggfs. werden notwendige Maßnahmen wie Qualifikationserweiterungen oder die Zuhilfenahme externer Lieferanten abgestimmt.

Vierter Schritt: Ablaufplanung

Die Kunst für den Projektleiter mit seinem Kernteam besteht nun darin, die aus der Implementierungs-PBS abgeleiteten Arbeitspakete in der WBS mit dem vorgegebenen zeitlichen Rahmen in Einklang zu bringen. Wir verwenden ein wasserfallartiges Phasenmodell mit Quality Gates, in das die ermittelten PBS-Elemente und Arbeitspakete einzuordnen sind, so dass ein durchgängiger Ablaufplan entsteht. Jetzt führt der Projektleiter auch die Ressourcenplanung durch, d.h. er handelt mit den einzelnen Fachbereichen die konkreten Einsatztermine für die benötigten und bereits vorbesprochenen Ressourcen aus. Der Plan ist fertig, sobald jedem Arbeitspaket die benannten Ressourcen verbindlich zugeordnet sind.

Der Vorteil der dargestellten Vorgehensweise ist, dass bei der Projektplanung nicht mehr über die Lieferinhalte sowie deren technische Notwendigkeiten diskutiert wird – dies ist bereits vorher endgültig beschlossen. Es geht nur noch darum, die benötigten "Skills" in der richtigen Menge in einem bestimmten zeitlichen Zielkorridor zu bekommen. Wenn dies intern nicht möglich ist, kann mit einer klaren Beschreibung (Statement of Work, SOW) die benötigte Leistung am Markt für externe Dienstleister ausgeschrieben und fristgerecht besorgt werden.

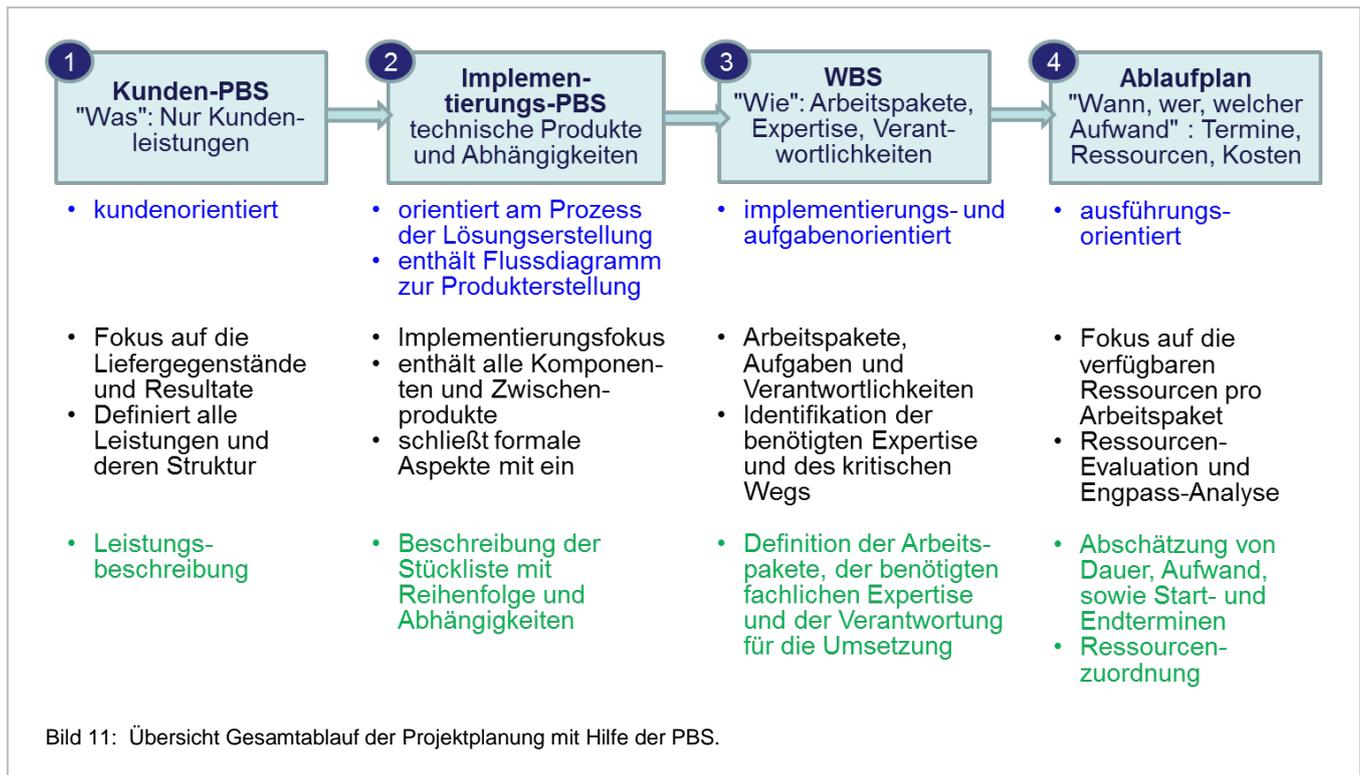
Die so erstellten Struktur-, Ablauf- und Ressourcenpläne liefern eine verlässliche und solide Basis, um das gesamte Liefervorhaben zu überwachen und zu steuern. Notwendige Anpassungen aller Art (Terminverschiebungen, Kostenaspekte, Ressourcenverfügbarkeit usw.) können in ihrer gesamten strukturellen Wirkungskette (von der Kunden-PBS zur WBS und zurück) betrachtet und optimiert werden.

Die Ablaufplanung

- ist ausführungsorientiert und konzentriert sich darauf, die konkreten Ressourcen und ihre Einsatzzeiten für die Projektdurchführung zu planen
- optimiert kritische Wege und Abhängigkeiten
- ordnet Arbeitspaketen konkrete Verantwortliche und umsetzende Ressourcen zu

- schätzt detailliert Aufwand, Dauer sowie Anfang und Ende jeder Aktivität ab und vereinbart diese mit den jeweiligen Arbeitspaketverantwortlichen
- sorgt für ein aktives Risiko- und Chancen-Management

Bild 11 zeigt im Überblick die gesamte dargestellte Vorgehensweise.



Vorteile dieses Ansatzes

Die Vorteile der an den Liefergegenständen orientierten Vorgehensweise liegen auf der Hand: Alle Beteiligten – vom Kunden über die Entscheider bis hin zu den Ressourcenverantwortlichen – sind eingebunden und arbeiten an derselben durchgängigen und einfach zu verstehenden Struktur. Der gesamte Planungsprozess bildet eine Wirkungskette, entlang der alle Veränderungen hinsichtlich ihrer Auswirkungen analysiert, diskutiert und entschieden werden können.

Es gehört Disziplin und Erfahrung dazu, ein Projekt auf diese Art und Weise anzugehen, z.B. wenn ungeduldige Entscheider schnelle Ergebnisse im Sinne einer sofortigen Implementierung sehen wollen. Die vorgestellte Vorgehensweise ermöglicht jedoch die schrittweise und fokussierte Ausarbeitung aller notwendigen Maßnahmen, um das gewünschte Produkt in der definierten Qualität und zur Zufriedenheit des Kunden zu erstellen. Dabei ist auch eine agile Vorgehensweise für die Umsetzung möglich, wenn man den Ausführungen von Hüsselmann zur Agilität in den Implementierungsphasen folgt (Hüsselmann, 2014).

Zusammenfassend die wichtigsten Punkte und Vorteile der hier beschriebenen Vorgehensweise:

1. Die Kunden-PBS sorgt bereits bei der Erstellung der Projektanfrage dafür, dass alle typischen Lieferelemente eines IT-Projekts (Geschäftsprozess, IT-Lösung, IT-Service, Transformation) abgefragt werden und bildet somit eine solide Basis für die weiteren Schritte. Erfahrungsgemäß sind dazu zwar etliche Arbeitssitzungen notwendig, es erwächst dadurch aber auch ein viel besseres Verständnis für die "Klappen" bzw. Risiken des Projekts.
2. Die auf dieser Basis eingestellten finanziellen Planungstitel bzw. die grundsätzlich bereitgestellten Budgets sind meiner Erfahrung nach zwar meist höher, aber weit realistischer und qualifizierter als ohne die durch die Kunden-PBS geschaffene Klarheit.
3. Der Projektleiter erhält bei der Übernahme der Verantwortung ein qualifiziertes Pre-Mandat und nachvollziehbare Rahmenbedingungen für die Kosten- und Terminplanung.
4. Kunde und Projektleiter treffen durch die Kunden-PBS eine klare Abmachung über die Liefergegenstände, die sie jederzeit zur Klärung der Projektinhalte (Scope) heranziehen können.
5. Der Projektleiter kann dem Projektteam sowie den Ressourcenverantwortlichen jederzeit die "Wirkungskette" vom geplanten Leistungsumfang (Kunden-PBS) zu den geplanten und notwendigen Arbeitspaketen (Implementierungs-PBS, WBS und Projektzeitplanung) nachvollziehbar darstellen.
6. Die Auswirkungen von Änderungsanträgen (Project Change Requests) auf die drei "magischen" Elemente des Projektmanagements (Leistung, Zeit, Kosten) können leicht analysiert und deutlich gemacht werden.

Erfolgsfaktoren: Kompetenz, Lernbereitschaft und Entschlossenheit

Für die Erstellung der Implementierungs-PBS und deren Überführung in eine WBS sind hochqualifizierte Experten erforderlich, die mit ihrer Erfahrung und ihrem methodischen Fachwissen eine vollständige und realisierbare Umsetzungsplanung (Schritte 2 und 3 unserer Vorgehensweise!) aufsetzen können.

Sowohl die Kundenverantwortlichen als auch das IT-Management müssen dazu bereit sein, aus den Fehlern der Vergangenheit zu lernen, sich auf diese Vorgehensweise einzulassen und sie weiterzuentwickeln.

Die größte Herausforderung besteht darin, dass die Beteiligten die vereinbarte Struktur und Vorgehensweise erlernen und diszipliniert anwenden müssen. Sie haben dadurch am Anfang zwar einen etwas höheren Aufwand als gewohnt, dieser zahlt sich jedoch schnell aus durch die gewonnene Reibungsarmut und die höhere Qualität der Projektergebnisse.

Meine bisherigen Erfahrungen mit der beschriebenen Vorgehensweise waren sehr positiv. Trotz anfänglicher Skepsis empfanden es die Projektteams, die Bedarfsträger und die Auftraggeber als wohlthuend transparent und hilfreich, stets Klarheit über den vereinbarten Scope in allen Details zu haben. Das Topmanagement schätzte die dadurch erreichte Transparenz, vor allem aber die erzielte Erfolgssteigerung des Projektportfolios.

Literatur:

- Enderle da Silva, Kristina: Aus der Schusslinie nehmen, Interview mit Winfried Vogel, personalmagazin 11/2013, S. 18, Verlag Haufe-Lexware
- Office of Government Commerce (OGC): Erfolgreiche Projekte managen mit PRINCE2, 2009, Anhang D: Produktbasierte Planung
- Angermeier, Georg: **Die Produktbasierte Planung nach PRINCE2. Teil 1: Produktbeschreibung und Produktstrukturplan**, Projekt Magazin 04/2012
- Angermeier, Georg: **Die Produktbasierte Planung nach PRINCE2. Teil 2: Produktflussdiagramm, Produktbeschreibungen und Projektplan**, Projekt Magazin 05/2012
- SAP: Accelerated SAP, SAP Help Portal, http://help.sap.com/saphelp_46c/helpdata/de/fc/5dc13690581b20e10000009b38f839/content.htm, zuletzt besucht am 22.04.2014
- Knauer, Stefan; Pape, Christian u. Spiekermann, Mathias: SAP-Implementierung auf Basis des ASAP-Modells, WinfWiki, http://winfwiki.wi-fom.de/index.php/SAP-Implementierung_auf_Basis_des_ASAP-Modells, zuletzt besucht am 22.04.2014
- Hüsselmann, Claus: Agilität im Auftraggeber-Auftragnehmer-Spannungsfeld, projektMANAGEMENT aktuell 1/2014, S. 38-42

Vollständig und schnell: standardisierte Auftragsklärung im Sondermaschinenbau

Im Sondermaschinenbau ist eine vollständige Auftragsklärung von besonderer Bedeutung. Werden Anforderungen übersehen oder falsch formuliert, sind nachträgliche Änderungen extrem teuer. Im Rahmen des Projekts eLOP (electronic Lean Order Processing) haben die TISORA Sondermaschinen GmbH, der IT-Dienstleister Community4you und die Professur Fabrikplanung und Fabrikbetrieb der TU Chemnitz eine Vorgehensweise entwickelt, die eine frühzeitige und umfassende Auftragsklärung ermöglicht und somit Folgekosten verhindert. Diese Vorteile konnten in der Praxis bei der TISORA Sondermaschinen GmbH verifiziert werden.

Die Vorgehensweise ist für Unternehmen interessant, die im Sondermaschinenbau tätig sind oder mit diesem Bereich zusammenarbeiten. Außerdem kann sie in der IT-Branche angewendet werden. Darüber hinaus ist geplant, sie für andere Bereiche zu adaptieren (z.B. Apparatebau, Elektro- und Elektronikindustrie, Baugewerbe).

Voraussetzung für Effizienz: vollständige Auftragsklärung

Die betriebliche Praxis bei der Projektabwicklung in kleinen und mittelständischen Unternehmen (KMU), speziell im Bereich des kundenspezifischen Maschinen- und Anlagenbaus, zeigt große Optimierungspotenziale auf. Dies bestätigten Kontakte und Projektpartner der Professur Fabrikplanung und Fabrikbetrieb der TU Chemnitz sowie Literaturquellen (siehe Müller 2006, Gizanis 2006, Schuh 2006, Ulrich 2007, Wagner 2007). Die Auftragsabwicklung ist noch zu wenig automatisiert. So erfolgen z.B. die Umwandlung von Lastenhefte in Pflichtenhefte sowie die Angebotserstellung in vielen Unternehmen manuell. Außerdem sind die Prozesse noch nicht transparent genug, Auftragnehmer und -geber sprechen sich zu wenig ab und es fehlt eine gemeinsame Datenbasis, die eine Wiederverwendung von bereits vorhandenem Wissen ermöglichen würde. Die Folge sind Informationsverluste während der Auftragsabwicklung, eine erhöhte Fehleranfälligkeit und ressourcenraubende Mehrfacharbeiten. Um diese Probleme zu lösen, ist es notwendig, die Auftragsabwicklungsprozesse zu standardisieren und zu automatisieren. (Siehe auch Reinhold 2008, Hagen 2008, Müller 2007, Gansauge 2007, Birn 2006).

Autor



Prof. Dr.-Ing. Egon Müller

Direktor des Instituts für Betriebswissenschaften und Fabriksysteme,

Inhaber der Professur Fabrikplanung und Fabrikbetrieb an der TU Chemnitz

Kontakt:

egon.mueller@mb.tu-chemnitz.de

Mehr Informationen unter:

projektmagazin.de/autoren



Ulf Wagner

Dipl.-Ing., wissenschaftl. Mitarbeiter an der Professur Fabrikplanung und Fabrik-

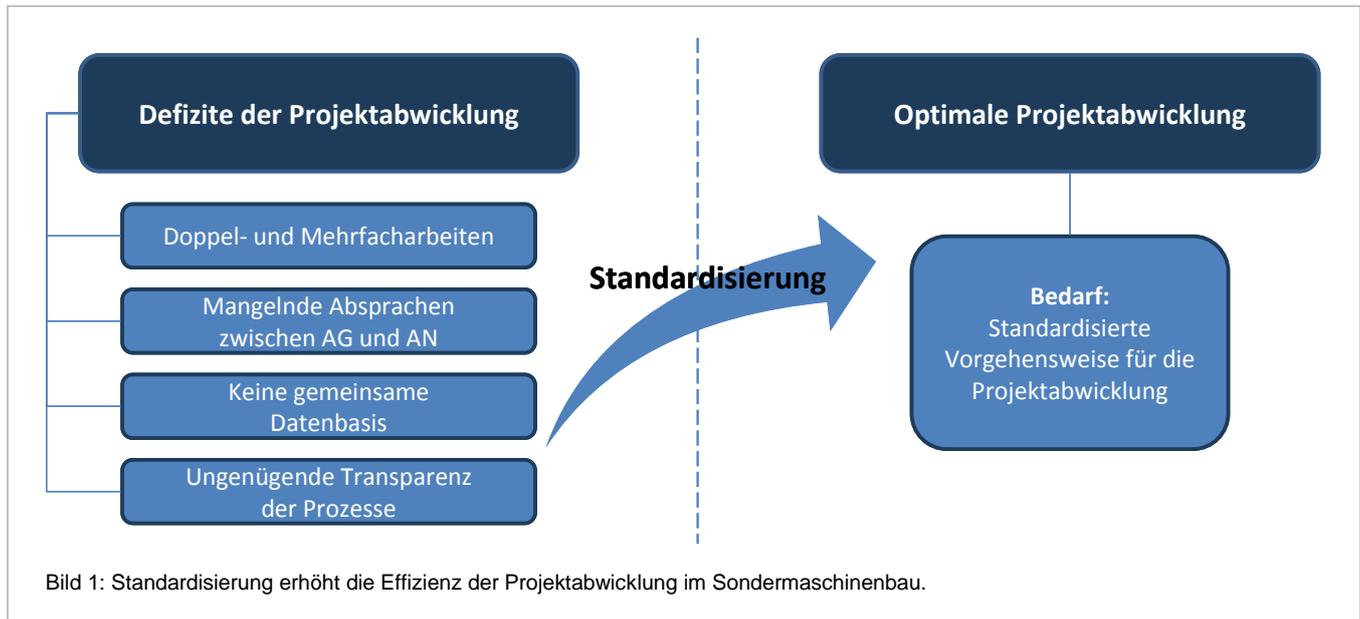
betrieb am Institut für Betriebswissenschaften und Fabriksysteme der TU Chemnitz

Kontakt:

ulf.wagner@mb.tu-chemnitz.de

Mehr Informationen unter:

projektmagazin.de/autoren



In der betrieblichen Praxis von KMU im Sondermaschinenbau gibt es derzeit noch keine informationstechnischen Lösungen, die eine Standardisierung der Auftragsabwicklung unterstützen. Dies ist darauf zurückzuführen, dass im Sondermaschinenbau Methoden fehlen, auf die solche Lösungen aufbauen könnten. Bestehende Methoden wie PRINCE 2 oder das V-Modell (vgl. Schelle 2007) enthalten keine detaillierte Anleitung, um eine möglichst lückenlose Auftragsklärung zu gewährleisten. Somit sind sie zu allgemein gehalten und nicht auf die spezifischen Anforderungen der Auftragsklärung im Sondermaschinenbau ausgerichtet.

Besonderes Kennzeichen des Sondermaschinenbaus ist u.a., dass die Losgröße zwischen eins und drei liegt und die Maschinen nach sehr speziellen Kundenwünschen geplant, konstruiert und hergestellt werden. Die kundenspezifische Herstellung geht dabei weit über die Auswahl von Ausprägungen, Zusatzfunktionen oder Anpassungen hinaus. Sondermaschinen sind deshalb durch einen sehr hohen technologischen Neuheitsgrad charakterisiert.

Im Sondermaschinenbau ist es sehr wichtig, dass der Auftrag vor Beginn von Entwicklung, Fertigung und Montage vollständig definiert ist. Wurden Anforderungen übersehen oder erfüllt ein Bauteil seine Funktion nicht, sind die erforderlichen Nacharbeiten sehr kostenintensiv. Falsch konzipierte Bauteile muss man erneut konzipieren und fertigen, so dass Doppelarbeiten anfallen. Außerdem streiten sich Auftraggeber und Auftragnehmer im Fall von Nacharbeiten oft darüber, wer diese zusätzlichen Kosten tragen soll.

Vorteile einer standardisierten Auftragsklärung

Um solche Probleme zu vermeiden, müssen Lasten- und Pflichtenheft vollständig sein und nachträgliche Änderungen soweit wie möglich ausgeschlossen werden. Die im Rahmen von eLOP entwickelte, standardisierte Vorgehensweise für die Auftragsabwicklung im Sondermaschinenbau ermöglicht es, Informationslücken bereits während der Erstellung des Pflichtenhefts zu entdecken und zu schließen und somit sämtliche Anforderungen frühzeitig zu erfassen.

Da Sondermaschinen nach speziellen Kundenwünschen gefertigt werden, ist es sehr arbeitsaufwändig, die Lastenhefte zu analysieren und sie in Pflichtenhefte zu übertragen. Der Arbeitsaufwand variiert stark. Er hängt u.a.

von den betrieblichen Rahmenbedingungen sowie der Erfahrung und Kreativität der Projektmitarbeiter ab und kann zwischen zwei und vier Wochen betragen. Somit variiert auch das Einsparpotenzial durch die Standardisierung. Das Projektteam von eLOP geht davon aus, dass sich der Zeitaufwand für die Auftragsklärung mit einer standardisierten Vorgehensweise um die Hälfte reduzieren lässt.

Ein speziell auf die Anforderungen im Sondermaschinenbau angepasstes Vorgehensmodell hat folgende Vorteile (vgl. Wagner 2008):

- Schnellere und umfassendere Spezifikation des Projektergebnisses.
- Bei der Erstellung eines Angebots für den potenziellen Kunden können alle preisbildenden Faktoren berücksichtigt werden.
- Schnellere Projektabwicklung durch die Eliminierung von Mehrfacharbeiten.
- Steigerung der Produktqualität durch eine effiziente Analyse und Umsetzung der Kundenanforderungen.
- Geringere Fehlerquote der Anlagen bei der Inbetriebnahme.
- Sichere Beherrschung des Prozesses der Auftragsklärung, da die einzelnen Prozessschritte detailliert festgelegt sind.
- Einheitliche Datenbasis zwischen den Projektbeteiligten des Anlagenherstellers sowie dem Auftraggeber durch zentrale Dokumente (Pflichtenheft usw.).
- Erhöhter Erkenntnistransfer zwischen den Projekten durch Wiederverwendung von Erfahrungswissen, welches z.B. in den Pflichtenheften dokumentiert ist.

eLOP: Entwicklung einer neuen Vorgehensweise

An der Realisierung des Projekts eLOP beteiligten sich die Professur Fabrikplanung und Fabrikbetrieb der TU Chemnitz, der IT-Dienstleister Community4you und die TISORA Sondermaschinen GmbH. Gemeinsam entwickelten sie eine Vorgehensweise, die das Methoden-Defizit im Bereich des Sondermaschinenbaus in Forschung und Implementierung ausfüllen und ein erster Schritt hin zu einer optimalen Projektabwicklung ist. Das Projektteam war interdisziplinär aufgestellt und bestand aus Informatikern und Ingenieuren aus Wissenschaft und Praxis.

Das Projekt eLOP gliederte sich in drei Teilprojekte. Im ersten Teilprojekt erarbeiteten die TISORA-Mitarbeiter eine Vorgehensweise für die Auftragsabwicklung, die auf die Rahmenbedingungen ihres Unternehmens abgestimmt war. Auf der Basis der von der TISORA erarbeiteten Vorgehensweise konnte die Professur Fabrikplanung und Fabrikbetrieb der TU Chemnitz im zweiten Teilprojekt eine allgemein gehaltene, standardisierte Vorgehensweise konzipieren, die sich auch für den Einsatz in anderen Unternehmen eignet. Grundlage für dieses Konzept waren Praxisdaten von TISORA, z.B. Dokumente aus Projektordnern, Prozesse aus dem Qualitätsmanagement-Handbuch und Informationen zu Fehlern aus Kundeninterviews zur Auftragsabwicklung. Bei TISORA erfolgte die Pilotanwendung der Ergebnisse. Das dritte Teilprojekt, die Entwicklung einer informationstechnischen Lösung, lag in der Verantwortung der Community4you GmbH. Die Mitarbeiter erstellten ein Funktionsmuster und einen Prototypen.

eLOP – electronic Lean Order Processing	
Ziel	Entwicklung und Verifizierung eines Vorgehensmodells für die Projektabwicklung im Sondermaschinenbau.
Projektträger	Sächsische Aufbaubank (SAB)
Beteiligte	<ul style="list-style-type: none">• TISORA Sondermaschinen GmbH (verantwortlich für Teilprojekt 1: Entwicklung einer unternehmensspezifischen Vorgehensweise für die Auftragsabwicklung; Pilotanwendung der allgemeinen, von der Professur Fabrikplanung und Fabrikbetrieb der TU Chemnitz entwickelten Vorgehensweise)• Professur Fabrikplanung und Fabrikbetrieb der TU Chemnitz (verantwortlich für Teilprojekt 2: Entwicklung einer allgemeinen, standardisierten Vorgehensweise für die Auftragsabwicklung)• IT-Dienstleister Community4you GmbH (verantwortlich für Teilprojekt 3: Entwicklung einer informationstechnischen Lösung)
Laufzeit	06/2007 – 11/2008

Tabelle 1: eLOP – Projektbeteiligte.

Einzelne Schritte

Die in eLOP entwickelte Vorgehensweise umfasst u.a folgende Schritte:

- Lastenheftanalyse
- Erstellung des Pflichtenhefts und der Angebotsunterlagen
- Erstellung und Kategorisierung weiterer Dokumente

Außerdem umfasst die Vorgehensweise Referenzprozesse zum Zweck der Standardisierung der Auftragsabwicklung im kundenspezifischen Maschinen- und Anlagenbau. Zu diesen Prozessen gehören beispielsweise die Analyse von Lastenheften bzgl. Angebotserstellung, die Angebotserstellung und die Vertragsverhandlung. Diese Prozesse werden u.a. durch die oben aufgeführten Dokumente und deren Bestandteile unterlegt. Diese Bestandteile werden "Informationsbausteine" genannt. Ein Informationsbaustein ist eine logische Gruppierung von Informationen, wie zum Beispiel "Aufgabenstellung des Auftraggebers" oder "Spezifizierung (Ergänzung / Umsetzung) der Aufgabenstellung".

Analyse des Lastenhefts

Zentral bei der Auftragsabwicklung ist die Umwandlung eines Lastenhefts in ein Pflichtenheft. Hierbei überführt der Auftragnehmer die Anforderungen des Kunden in konkrete Maßnahmen und hält fest, wie bzw. mit welchen technischen Mitteln er die Kundenanforderungen erfüllen wird (Tabelle 2).

Lastenheft	Pflichtenheft
Beschreibung der geforderten Leistung und der Rahmenfaktoren des Auftraggebers	Entwickelte Lösung des Auftragnehmers zur geforderten Leistung des Auftraggebers
Abgrenzung der Anforderungen des Benutzers bzw. des Verwertenden der Produktlösung	Realisierung der Benutzeranforderungen durch ein technologisches Lösungskonzept
Kommunikationsmedium	Planung, Bewertung und Kontrolle der Projektabwicklung (Zielsystem)
Grundlage für Ausschreibungen und Angebots- bzw. Vertragserstellung	Grundlage für die Vorbereitung der Abnahmen und des Betriebs

Tabelle 2: Inhalte von Lasten- und Pflichtenheft.

Das in eLOP entwickelte Vorgehensmodell ermöglicht die methodisch gestützte Erstellung eines Pflichtenhefts. Als Informationsgrundlage dient hierfür ein beliebig strukturiertes Lastenheft. Im Lastenheft erläutert der Auftraggeber – hier der Anlagenbetreiber – die prinzipielle Aufgabenstellung und beschreibt das benötigte Leistungsspektrum der Sondermaschine. Es kann immer sein, dass das Lastenheft Lücken und zweideutige Informationen enthält. Diese müssen aufgedeckt werden, bevor das Pflichtenheft erstellt wird. Andernfalls kann es geschehen, dass der Anlagenhersteller bestimmte Produktanforderungen nicht oder falsch erfüllt oder wichtige Rahmenbedingungen übersieht.

Jeder Kunde strukturiert das Lastenheft unterschiedlich, was es erschwert, fehlende Informationen zu entdecken. Die wesentliche Herausforderung bei der Erstellung eines vollständigen Pflichtenhefts ist aber die Komplexität des Anwendungsbereichs, in diesem Fall des Sondermaschinenbaus. Der Auftragnehmer muss eine kundenspezifische Produktlösung vollkommen neu planen und entwickeln. Somit liegt die Komplexität zum einen im Produkt selbst bzw. in dessen technologischen Neuheitsgrad und zum anderen in den Prozessen der Projektabwicklung, die abhängig von der Aufgabenstellung sehr unterschiedlich verlaufen können.

Erstellung des Pflichtenhefts

Im ersten Schritt analysiert der Anlagenhersteller die Daten des Lastenhefts, z.B. technische Kennwerte oder Anlagenfunktionen. Die Analyse erfolgt anhand eines Fragenkatalogs, der in eLOP am Beispiel der TISORA Sondermaschinen GmbH entwickelt und ausgehend von Microsoft Excel später in den Prototypen übertragen wurde. Mit Hilfe des Fragenkatalogs wird die Vollständigkeit und Eindeutigkeit der jeweiligen Aufgabenstellung geprüft. Mittels des Fragebogens soll sichergestellt werden, dass sämtliche Fragen, welche die Anforderungen betreffen, geklärt werden und keine Aspekte vergessen oder übersehen werden. Die Qualität der Auftragsklärung soll somit unabhängig werden von einzelnen Mitarbeitern, ihrer Erfahrung und ihren Vorgehensweisen. Der Fragenkatalog enthält Fragen zu vier verschiedenen Kategorien. In Tabelle 3 sind die Kategorien sowie Beispielfragen aufgeführt.

Kategorien	Beispielfragen
Grundinformationen zum Projekt (Auftraggeber, Kurzbeschreibung des Leistungsspektrums, Ecktermine, Projektrahmen, Arbeitsgut usw.)	Wann ist der angestrebte Liefertermin? Wann ist die angestrebte Endabnahme? Gibt es Zeichnungen zum Arbeitsgut? Gibt es Varianten bzgl. des Arbeitsgutes? Usw.
Produktforderungen (technische Kennwerte, Vorgaben zu Funktionen, Bestandteilen, Steuerung, Elektrik, Arbeitsschutz und Ergonomie)	Wie ist die geforderte Taktzeit? Wie ist die geforderte Maschinenverfügbarkeit? Gibt es Vorgaben zu Dauerschalldruckpegel? Gibt es Vorgaben zu Hilfs- und Betriebsstoffverbrauch? Usw.
Aufstell- und Betreiberbedingungen (vor- und nachgelagerte Prozesse, Umwelteinflüsse, räumliche Gegebenheiten usw.)	Müssen Schnittstellen zu vor- und nachgelagerten Prozessen beachtet werden? Wie ist die Umgebungstemperatur beim Betreiber? Wie ist die relative Luftfeuchtigkeit beim Betreiber? Wie sind die Abmessungen des Eingangstors? Wie ist die Transportwegbreite? Wie ist die Bodenbeschaffenheit? (Fundamentbeschreibung, Tragfähigkeit) Usw.
Spezifikation der Projektabwicklung	Gibt es Vorgaben zu Zahlungs- und Lieferbedingungen? Gibt es Vorgaben des Kunden bzgl. einzusetzender Software? Usw.

Tabelle 3: Kategorien und Beispielfragen des Fragenkatalogs.

Es gibt verschiedene Varianten des Fragekatalogs, die jeweils unterschiedlich viele Fragen enthalten. Welcher Katalog herangezogen wird, ist zum einen von der Art des vom Kunden geforderten Angebots abhängig. Ein Richtpreisangebot beispielsweise enthält lediglich eine Kostenschätzung und eine grobe Angabe des zeitlichen Rahmens, sodass der Erstellungsaufwand für das Dokument geringer ist als bei einem verbindlichen Angebot. Folglich sind für ein Richtpreisangebot weniger Fragen erforderlich, da z.B. Detailinformationen zu mechanischen Baugruppen oder zur Steuerung und Elektrik sowie entsprechende Lieferantenvorgaben entfallen. Zum anderen hängt die Wahl des Fragenkatalogs von der Projektabwicklungsphase ab, in der das Lastenheft geprüft wird, z.B. nach Auftragsvergabe. In dieser Phase müssen Fragen, wie "Gibt es Vorgaben zum Transport oder Informationsaustausch (Datenformate)?" geklärt werden. Die Anzahl der Fragen, hängt von der Anlagenklassifikation ab (Handlinganlage, Werkzeugmaschinen, Verarbeitungsmaschine usw.). Grundsätzlich enthält ein Fragenkatalog etwa 30 bis 40 Fragen. Der Zeitaufwand, um einen Fragenkatalog abzuarbeiten, variiert stark und kann wenige Tage bis hin zu drei oder vier Wochen betragen.

Falls Informationen fehlen oder nicht eindeutig sind, werden sie in einem Abstimmungsprotokoll systematisch erfasst. So kann der Anlagenhersteller fehlende Daten gezielt beim Kunden ermitteln. Ein weiterer Vorteil ist, dass aus dem Lastenheft nur die Informationen herausgefiltert werden, die für die Projektabwicklung tatsächlich notwendig sind. Oft sind Lastenhefte aufgebläht und enthalten viele unwichtige Daten, weil der Kunde nicht genau weiß, welche Informationen im Hinblick auf die Aufgabenstellung von Bedeutung sind und welche nicht. Dies ist nicht verwunderlich, da Projekte im Sondermaschinenbau einen sehr hohen Neuheitsgrad haben.

Die bei der Analyse ermittelten Daten werden in eine Dokumentenvorlage für das Pflichtenheft transferiert. Damit enthält das Pflichtenheft die vollständige Aufgabenstellung des Auftraggebers. Ausgehend von der Aufgabenstellung und des geforderten Leistungsspektrums müssen Lösungskonzepte entwickelt und im Pflichtenheft festgehalten werden. Welche Daten dafür konkret zu erarbeiten sind, wird in der Dokumentenvorlage durch einen offenen, noch auszufüllenden Teil vorgegeben. Für jede Anforderung wird ein Lösungskonzept bzw. eine Inhaltsbeschreibung erstellt, ggf. mit näheren Erläuterungen. Hier werden u.a. die Hauptbestandteile, Funktionsabläufe sowie allgemeine und mechanische Gestaltungsmerkmale zu einer Maschine beschrieben. Mit Hilfe dieser Systematisierung können Arbeitsaufwände bei der Dokumentenerstellung reduziert werden, was wiederum Zeiteinsparungen zur Folge hat.

Ausgehend vom Pflichtenheft kann anschließend ein Angebot erstellt werden. Dank der Lastenheftanalyse und dem vollständigen Pflichtenheft erfolgt die Angebotserstellung unter Berücksichtigung aller preisbildenden Faktoren.

Rechnergestützte Variante

Es existiert bereits ein Prototyp bzw. Funktionsmuster für eine rechnergestützten Variante des Pflichtenhefts. In dieser können den Mitarbeitern, die das Pflichtenheft erstellen, an geeigneter Stelle standardisierte Angaben zur Verfügung gestellt werden, z.B. mittels Dropdown-Auswahllisten. Geeignet sind Angaben, die durch einen definierten Wertebereich beschrieben werden können, d.h. anhand überschaubarer Alternativen von Angaben. Bei einer Handling-Anlage sind das z.B. mögliche Verunreinigungen des Arbeitsgutes wie "Späne", "Öl" oder "Staub". Bei den Anforderungen für die Form der Dokumentation kann man beispielsweise die Auswahlmöglichkeiten "Papier" und "Datenträger" einrichten. Wichtig ist, dass auch mehrere Angaben gleichzeitig ausgewählt werden können. Wer die Vorlagen eingibt und pflegt, hängt von der jeweiligen Aufbauorganisation des Unternehmens ab. Diese Aufgabe kann z.B. dem Projektleiter zufallen.

Andere Angaben wie beispielsweise technische Daten (Taktzeiten, Verfügbarkeiten) oder Informationen zur Prozesssicherheit können nicht als Vorlagen eingespeichert werden, da sie sich bei jeder Sondermaschine unterscheiden.

Weitere Dokumente

Aus den Inhalten von Pflichtenheft und Angebot können weitere Dokumente generiert werden. Dies spart Mehrfacharbeiten und Zeit. Dokumente, die auf Basis des Pflichtenhefts erstellt werden können, sind beispielsweise:

- standardisierte Lastenhefte für Unterauftragnehmer.
- Dokumente bzw. Checklisten, anhand derer die Bedingungen für Aufstellung und Montage der Maschinen systematisch geprüft werden kann, z.B. die Bedingungen hinsichtlich Transport, Installationshilfsmittel, Schnittstellen, Personal usw.
- Abnahmeprotokolle, die anhand der auftragspezifischen Abnahmekriterien, erstellt werden.

Pilotanwendung

Die in eLOP entwickelte Vorgehensweise wurde durch TISORA in einer Pilotanwendung getestet. TISORAs Prozesse für die Auftragsabwicklung waren vorher nicht ausreichend standardisiert, automatisiert und dokumentiert. Durch

die Einführung der in eLOP entwickelten Vorgehensweise können bei TISORA die Optimierungspotenziale ausgeschöpft und erhebliche Verbesserungen in Form von kürzeren Projektabwicklungszeiten erzielt werden. Da die einzelnen Prozessschritte der Auftragsabwicklung vorgeschrieben werden, der Fragenkatalog vorliegt und in den Dokumentenvorlagen Hilfetexte zur Verfügung stehen, können die Mitarbeiter Fehler und Änderungen verringern.

Informationen, die für die Planung und Realisierung des Maschinenbaus erforderlich sind, lassen sich nun bereits in frühen Phasen des Projekts eindeutig und vollständig ermitteln. Dadurch können Fehler in späteren Projektphasen (z.B. Entwicklung, Konstruktion und Arbeitsvorbereitung) sowie kostspielige Änderungen vermieden werden. Die Reduzierung von Fehlerquellen resultiert auch in einer verbesserten Produktqualität. In der Pilotanwendung zeigte sich außerdem, dass die Pflichtenhefterstellung auf Basis der Lastenheftanalyse und der Dokumentenvorlagen die Erarbeitung technisch-technologischer Lösungen vereinfacht und verkürzt.

Fazit

Den Defiziten in der kundenbezogenen Einzel- und Kleinserienfertigung muss mit optimierenden Standardisierungssystematiken entgegengetreten werden. Bisher existierten kaum praxistaugliche Lösungen für dieses Problem.

Die in eLOP entwickelte Vorgehensweise ermöglicht nun die standardisierte Realisierung der Auftragsabwicklung. Die Vorgehensweise wurde mit Hilfe der Projektpartner rechentechnisch umgesetzt und in der Praxis erprobt. Dafür wurden ein Funktionsmuster und ein Prototyp entwickelt. eLOP stellt Unternehmen somit ein zweckmäßiges Hilfsmittel zur Optimierung der Auftragsabwicklung der Verfügung.



Das Vorhaben wurde im Rahmen des Europäischen Fonds für regionale Entwicklung (EFRE) und mit Mitteln des Freistaates Sachsen gefördert.

Literatur

- Birn, L.; Schweicher, B.; Walber, B.: Erhöhung der Liefertreue durch den neuen Qua-si- Standard myOpenFactory. psipenta Software Systems GmbH, Berlin, 2006
- Gansauge, L.; Riedel, R.; Müller, E.: Mit Standards fit für die Globalisierung (Teil 1), In: Werkzeug&formenbau, Ausgabe April 2007, S. 58-60
- Gansauge, L.; Riedel, R.; Müller, E.: Prozessstandardisierung in der Einzelfertigung - Optimierung der Auftragsabwicklung durch Prozessmanagement und konsequente Standardisierung am Beispiel des Formenbaus. In: wt Werkstattstechnik online, Jahrgang 97 (2007) H. 4, S. 279-283
- Gizanis, D.: Kooperative Auftragsabwicklung – Architektur, Praxisbeispiele und Nutzenpotenziale. Dissertation, Difo-Druck GmbH, Bamberg, 2006

- Hagen, M.; Korn, G. H.; Brinkop, M.: Informationsstrom-Design. In: Werkstattstechnik Online 98 (2008) 4, S. 274-279
- Müller, E.; Riedel, R.; Gansauge, L.: Prozessmanagement im Werkzeug- und Formenbau. In: VDI-Z Integrierte Produktion. Special Werkzeug-/ Formenbau November 2006, S. 20-23
- Müller, E.; Riedel, R.; Gansauge, L.: Die Störgrößen fest im Griff (Teil 2), In: Werkzeug&formenbau, Ausgabe Juni 2007, S. 42-44
- Reinhold, M.: Mehr Transparenz – Vertrieb und Entwicklung einbinden bei Produkt- und Auftragskonfiguration im Anlagenbau. In: Chemie Technik (2008) 9, S. 18-20
- Schuh, G.; Westkämper, E.: Liefertreue im Maschinen- und Anlagenbau. Fraunhofer IPA, Aachen, 2006
- Schelle H.; Ottmann R.; Pfeiffer A.: ProjektManager. 2. Aufl., Nachdr., GPM – Deutsche Gesellschaft für Projektmanagement, Nürnberg, 2007
- Ulrich, T.: Produktdatenmanagement im Maschinen- und Anlagenbau – Aktivierung signifikanter Kosten durch die Integration aller Informationen zum Projekt, Produkt und Prozess. In: Zeitschrift für wirtschaftlichen Fabrikbetrieb 102 (2007) 4, S. 239-242
- Wagner, U.; Näser, P.; Müller, E.: Optimierung betrieblicher Prozesse – Standardisierte Auftragsabwicklung führt zum Unternehmenserfolg. In: Werkstattstechnik online 97 (2007) 7/8, S. 586-591
- Wagner, U.: Homepage eLOP – electronic Lean Order Processing, Standardisierung von Auftragsabwicklungsprozessen entlang der Lebenszyklusphasen von kundenspezifischen Produktlösungen., www.e-lop.de, 16.10.2008