mitp

Alistair Cockburn

Use Cases
effektiv erstellen

Das Fundament für gute Software-Entwicklung

Geschäftsprozesse modellieren mit Use Cases

Die Regeln für Use Cases sicher beherrschen

Einführung

- Wie sieht ein Use Case aus?
- Warum sind für verschiedene Projektteams unterschiedliche Schreibstile erforderlich?
- Wie fügen sich Use Cases in die Erhebung von Anforderungen ein?
- Wie können wir uns auf das Verfassen von Use Cases vorbereiten?

Bevor wir uns in die Details der Use Cases selbst vertiefen, sollten wir diese Fragen beantworten können.

1.1 Was ist ein Use Case? – Eine Annäherung

Ein Use Case (Anwendungsfall) erfasst eine Übereinkunft, die zwischen den Stakeholdern eines Systems über dessen Verhalten getroffen wird. Der Use Case beschreibt das Verhalten des Systems unter verschiedenen Bedingungen, während es auf eine Anfrage eines der Stakeholder, des sogenannten *Primärakteurs*, reagiert. Der Primärakteur löst eine Interaktion mit dem System aus, um ein Ziel zu erreichen. Das System reagiert darauf und wahrt dabei die Interessen aller Stakeholder. Je nachdem, welche bestimmte Anfrage erfolgt und unter welchen Bedingungen dies geschieht, können sich verschiedene Verhaltensabläufe oder Szenarien entfalten. In einem Use Case werden all diese verschiedenen Szenarien zusammengefasst.

Im Grunde handelt es sich bei Use Cases um eine textuelle Form, auch wenn sie mit Hilfe von Flussdiagrammen, Sequenzdiagrammen, Petri-Netzen oder Programmiersprachen dargestellt werden können. Im Normalfall dienen sie als Kommunikationsmittel zwischen einzelnen Personen, oftmals Personen ohne Spezialkenntnisse. Deshalb ist einfacher Text die beste Darstellungsform.

Als schriftliches Format kann ein Use Case die Diskussion innerhalb eines Teams über ein zu entwickelndes System anregen. Ein Team kann, wenn es will, die gegenwärtigen Systemanforderungen mit Hilfe von Use Cases dokumentieren. Andere Teams entscheiden sich vielleicht zur Dokumentation des endgültigen Designs durch Use Cases. All dies gilt sowohl für große Systeme wie beispielsweise einem gesamten Unternehmen als auch für kleine Systeme wie etwa einem Software-Anwendungsprogramm. Obwohl die Teams in all diesen Fällen mit einem

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

unterschiedlichem Genauigkeitsgrad und auf verschiedenen Ebenen der technischen Detaillierung formulieren, gelten interessanterweise für die Modellierung immer die gleichen Grundregeln.

Wenn Use Cases die Geschäftsprozesse eines Unternehmens dokumentieren, ist das zur Diskussion stehende System (system under discussion = SuD) das Unternehmen selbst. Stakeholder sind die Aktionäre, die Kunden, Anbieter und Regierungsbehörden. Zu den Primärakteuren gehören die Kunden und eventuell die Lieferanten des Unternehmens.

Wenn Use Cases die Verhaltensanforderungen für eine bestimmte Software verzeichnen, ist das SuD das Computerprogramm. Hier sind die Stakeholder die Programmanwender, die Firma, der das Programm gehört, Regierungsbehörden und andere Computerprogramme. Primärakteur ist der Anwender am Computerbildschirm oder ein anderes Computersystem.

Ein gut geschriebener Use Case ist leicht zu lesen. Die Sätze haben immer die gleiche grammatikalische Form und beschreiben einen einfachen Aktionsschritt, in dem ein Akteur ein Ziel erreicht oder Informationen an einen anderen Akteur weitergibt. Man sollte nach wenigen Minuten wissen, wie ein Use Case zu lesen ist.

Schwieriger ist es schon, einen guten Use Case zu verfassen. Der Autor muss drei Konzepte beherrschen, die jeden einzelnen Satz und den Use Case als Ganzes betreffen. Es mag auf den ersten Blick überraschen, dass die korrekte Befolgung dieser drei Konzepte nicht einfach ist. Die Schwierigkeiten beginnen schon, wenn Sie Ihren ersten Use Case verfassen. Die Rede ist von den folgenden drei Konzepten:

- *Umfang (Scope):* Was genau ist das zur Diskussion stehende System?
- *Primärakteur:* Wer verfolgt das Ziel?
- *Ebene:* Liegt das Ziel auf einer hohen oder niedrigen Ebene?

Es folgen einige Beispiele von Use Cases. Die Use-Case-Komponenten werden im nächsten Kapitel beschrieben. Im Moment sollten Sie sich diese zusammenfassenden Definitionen merken:

- *Akteur:* Jeder und alles, das ein Verhalten zeigt.
- *Stakeholder:* Jemand oder etwas mit einem ureigenen Interesse am Verhalten des zur Diskussion stehenden Systems (System under discussion = SuD)
- *Primärakteur:* Der Stakeholder, menschlich oder nicht-menschlich, der eine Interaktion mit dem SuD auslöst, um ein Ziel zu erreichen.
- *Use Case:* Eine Übereinkunft zum Verhalten des SuD.
- *Umfang:* Identifiziert das System, das zur Diskussion steht.

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

- *Vorbedingungen und Invarianten:* Bezeichnet, was vor und nach Ablauf des Use Case zutreffen muss.
- *Standardablauf:* Ein Fall, bei dem nichts schief geht.
- *Erweiterungen:* Andere Möglichkeiten für den Ablauf eines Szenarios.
- Die Nummerierungen bei den Erweiterungen beziehen sich auf die Nummern der Schrittabfolge im Standardablauf, für die eine Alternativmöglichkeit entdeckt wird (die Schritte 4a und 4b zeigen zum Beispiel zwei verschiedene Bedingungen, die für Schritt 4 gelten können).
- Bezieht sich ein Use Case auf einen anderen Use Case, wird der Referenzfall unterstrichen.

Der erste Use Case beschreibt eine Person, die im Internet Aktien einkaufen will. Um zu signalisieren, dass es um ein Ziel geht, das mit einem einzigen Arbeitsgang erreicht werden soll, markieren wir den Use Case mit dem *Wellensymbol*  , das ein *Ziel auf der Anwenderebene* bezeichnet. Der zweite Use Case beschreibt eine Person, die versucht, Schadenersatz für einen Autounfall zu erhalten. Dieses Ziel ist nicht in einem Arbeitsgang zu erreichen. Um dies zu verdeutlichen, zeichnen wir den Use Case mit dem *Drachensymbol*  für die *Überblicksebene* aus. Die Erklärung zu diesen Symbolen finden Sie in Kapitel 5, eine Zusammenfassung steht im Innendeckel des Buchs.

Der erste Use Case beschreibt die Interaktionen einer Person mit einem Programm (»PAF«), das auf einer mit dem Internet verbundenen Workstation läuft. Das Black-Box-Symbol  zeigt an, dass es sich bei dem zur Diskussion stehenden System um ein Computersystem handelt. Der zweite Use Case beschreibt die Interaktion einer Person mit einem Unternehmen, was durch das Gebäudesymbol  angezeigt wird. Die Verwendung dieser Symbole ist rein optional - die Auszeichnung von Umfang und Ebene allerdings nicht.

Es folgen die Use Cases 1 und 2.

Use Case 1 **Aktien im Internet Kaufen**

Primärakteur: Käufer

Umfang: Personal Advisor (Persönlicher Berater)/Finanzpaket (PAF)

Ebene: Ziel auf Anwenderebene

Stakeholder und Interessen:

Käufer – möchte Aktien kaufen und sie automatisch seinem PAF-Konto hinzufügen.

Aktienhändler – braucht die vollständigen Kaufinformationen.

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

Vorbedingung: Der Anwender hat PAF bereits geöffnet.

Invariante: Es sind genügend Informationen für das Protokoll vorhanden, damit PAF einen Fehler entdecken kann und den Anwender auffordert, die Details anzugeben.

Nachbedingungen: Die externe Website bestätigt den Kauf; Protokoll und Anwenderkonto werden aktualisiert.

Standardablauf:

1. Der Käufer wählt die Option, Aktien im Internet zu kaufen.
2. Der Anwender gibt den Namen der Website, die genutzt werden soll, in PAF ein (DAB, Consors o.ä.).
3. PAF stellt eine Internetverbindung zu der Site her, behält dabei die Steuerung.
4. Der Käufer durchsucht die Website und kauft dort Aktien ein.
5. PAF empfängt die Antworten von der Website und aktualisiert das Anwenderkonto.
6. PAF zeigt dem Anwender den neuen Kontostand.

Erweiterungen:

2a. Der Käufer ruft eine Website auf, die von PAF nicht unterstützt wird.

2a1. Das System erhält vom Käufer einen anderen Vorschlag und die Option, den Use Case abzubrechen.

3a. Beliebiger Internetfehler während des Setup:

3a1. Das System meldet dem Käufer einen Fehler mit einem entsprechenden Hinweis und kehrt zum vorherigen Schritt zurück.

3a2. Der Käufer bricht den Use Case entweder ab oder versucht es erneut.

4a. Der Computer stürzt ab oder wird während der Transaktion zum Kauf abgeschaltet.

4a1. (Was tun wir in diesem Fall?)

4b. Die Website bestätigt den Kauf nicht, sondern setzt ihn auf die Warteliste:

4b1. PAF protokolliert die Verzögerung, setzt einen Timer, um den Käufer nach dem Ergebnis zu fragen.

5a. Die Website gibt die benötigten Kaufinformationen nicht zurück:

5a1. PAF protokolliert den Informationsmangel und lässt den Käufer den fraglichen Einkauf aktualisieren.

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

Use Case 2 **Bezahlung für Autounfall erhalten**

Primärakteur: Antragsteller

Umfang: Versicherungsgesellschaft (»MeineVersGes«)

Ebene: Überblick

Stakeholder und Interessen:

Antragsteller – so viel Geld wie möglich erhalten.

MeineVersGes – den geringstmöglichen Betrag auszahlen.

Versicherungskammer – aufpassen, dass alle Richtlinien befolgt werden.

Vorbedingung: Keine.

Invarianten: MeineVersGes nimmt das Protokoll zum Antrag und allen Aktivitäten auf.

Nachbedingungen: Antragsteller und MeineVersGes einigen sich auf die zu zahlende Summe; Antragsteller erhält diese ausbezahlt.

Trigger: Antragsteller reicht eine Forderung ein.

Standardablauf:

1. Antragsteller reicht Forderung mit Daten ein, die sie untermauern.
2. Die Versicherungsgesellschaft verifiziert, dass der Antragsteller über eine gültige Versicherungspolice verfügt.
3. Die Versicherungsgesellschaft weist den Fall einem Agenten zu.
4. Die Versicherungsgesellschaft verifiziert, dass alle Details im Einklang mit den Richtlinien der Versicherungspolice stehen.
5. Die Versicherungsgesellschaft zahlt die Summe an den Antragsteller aus und schließt die Akten.

Erweiterungen:

1a. Die eingereichten Daten sind unvollständig:

1a1. Die Versicherungsgesellschaft fordert die fehlenden Informationen an.

1a2. Der Antragsteller reicht die fehlenden Informationen ein.

2a. Der Antragsteller hat keine gültige Versicherungspolice:

2a1. Die Versicherungsgesellschaft lehnt den Antrag ab, benachrichtigt den Antragsteller, protokolliert alles und schließt den Vorgang ab.

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

3a. Im Moment sind keine Versicherungsagenten verfügbar:

3a1. (Was macht die Versicherungsgesellschaft in diesem Fall?)

4a. Der Unfall steht nicht im Einklang mit den elementaren Richtlinien der Police:

4a1. Die Versicherungsgesellschaft lehnt den Antrag ab, benachrichtigt den Antragsteller, protokolliert alles und schließt den Vorgang ab.

4b. Der Unfall steht nicht im Einklang mit einigen weniger bedeutenden Richtlinien der Police:

4b1. Die Versicherungsgesellschaft beginnt, mit dem Antragsteller über den auszuzahlenden Betrag zu verhandeln.

Die meisten Use Cases im vorliegenden Buch stammen aus realen Projekten und ich habe Wert darauf gelegt, sie nicht nachzubessern (abgesehen vom Hinzufügen der Einträge für Umfang und Ebene, soweit diese nicht vorhanden waren). Sie sollen Beispiele kennen lernen, die in der Praxis funktionieren und nicht nur für den Unterricht attraktiv sind. Selten hat jemand die Zeit, Use Cases formal, vollständig und rund zu gestalten. Normalerweise reicht die Zeit gerade dazu aus, sie »passabel« zu gestalten und mehr ist auch nicht notwendig. Ich führe Ihnen diese Beispiele aus der Praxis vor, weil auch Sie trotz meiner Hilfestellungen selten die Gelegenheit haben werden, perfekte Use Cases zu modellieren. Ich kann die meiste Zeit über selbst keine perfekten Use Cases gestalten.

Use Case 3 wurde von Torfinn Aas von der Norwegischen Zentralbank für einen Kollegen, seinen Anwenderrepräsentanten, und für sich selbst verfasst. Er zeigt, wie die Form verändert werden kann, ohne dass der Wert dadurch geringer würde. Der Autor hat zusätzlich geschäftsbezogene Informationen in die Beschreibung aufgenommen, die zeigen, wie die Computerapplikation im Verlauf eines Arbeitstages funktioniert. Dies war nützlich, da so vermieden werden konnte, den Geschäftsprozess in einem gesonderten Dokument zu beschreiben. Niemand wurde verwirrt und für die Betroffenen war dies informativ.

Use Case 3  **Eingang einer Kiste Registrieren** 

EA – »Entgegennehmender Angestellter«

RV – »Registrierungsverwalter«

Primärakteur: EA

Umfang: Nachtschicht empfängt Registrier-Software

Ebene: Ziel auf Anwenderebene

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

Standardablauf:

1. EA nimmt die Kiste vom Transportunternehmen (TU) entgegen und öffnet sie (Kisten-ID, Tüten mit Tüten-IDs).
2. EA überprüft ID der Kiste anhand der vom TU registrierten IDs.
3. EA unterschreibt eventuell einen Beleg für den Lieferanten.
4. EA registriert die Ankunft der Kiste im System, das folgendes speichert:
   ID des EA
   Datum, Zeit
   ID der Kiste
   Transportunternehmen
   <Name Person?>
   # Tüten (Mit Tüten-IDs?)
   <Geschätzter Wert?>
5. EA nimmt die Tüten aus der Kiste, legt sie auf einen Wagen und bringt sie zum RV.

Erweiterungen:

2a. Kisten-ID stimmt nicht mit der ID beim Transportunternehmen überein.

4a. Ein Feueralarm wird ausgelöst und unterbricht die Registrierung.

4b. Der Computer stürzt ab.

Geduldig abwarten, bis der Computer wieder betriebsbereit ist.

Variationen:

4‘. Mit und ohne Personen-ID.

4‘‘. Mit und ohne geschätzten Wert.

5‘. EA lässt die Tüten in der Kiste.

1.2 Ein Use Case ist nicht wie der andere

Use Cases sind eine Textform, die in verschiedenen Situationen verwendet werden kann, etwa den folgenden:

- Zur Beschreibung von Geschäftsprozessen.
- Zur Fokussierung der Diskussion *auf* die erwarteten Anforderungen eines zu erzeugenden Softwaresystems. Der Use Case sollte jedoch nicht selbst die Beschreibung der Anforderungen darstellen.

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

- Zur Darstellung der funktionalen Anforderungen an ein System.
- Zur Dokumentation des Systemdesigns.
- Zur Modellierung durch eine kleine, eng zusammenarbeitende oder durch eine große bzw. verteilte Gruppe.

Jede Situation erfordert einen etwas anderen Schreibstil. Es folgen die wichtigsten Unterformen von Use Cases, unterteilt nach ihrem *Zweck*.

- Eine eng zusammenarbeitende Gruppe, die Anforderungen sammelt oder eine größere Gruppe, die über künftige Anforderungen diskutiert, verfasst *formlose* Use Cases, anders als große und örtlich verteilte oder formal arbeitende Gruppen, die *vollständig ausgearbeitete* Use Cases modellieren. Die formlose Variante kürzt die Use-Case-Schablone ab, damit die Use Cases schneller zu schreiben sind (mehr dazu später). Die Use Cases 1 bis 3 sind vollständig ausgearbeitet und folgen der kompletten Use-Case-Schablone und dem Nummerierungs-Schema. Ein Beispiel für eine formlose Variante sehen Sie in Use Case 4.
- Wer Geschäftsprozesse erstellt, verfasst *Geschäfts*-Use-Cases, welche die Geschäftsoperationen beschreiben. Ein Entwicklungsteam für Hardware oder Software dagegen verfasst *System*-Use-Cases zur Beschreibung der Systemanforderungen. Ein Designteam kann ebenfalls *System*-Use-Cases schreiben, um das Design zu dokumentieren oder die Anforderungen für kleine Subsysteme aufzuschlüsseln.
- Es hängt von der jeweils erforderlichen Sichtweise ab, ob der Autor ein Ziel wählt, das in mehreren Schritten zu erreichen ist oder eines, das in einem Schritt zu erreichen ist – also ein Ziel auf der *Überblicks-* oder der *Anwender*ebene oder aber den Teil eines Ziels auf Anwenderebene, auch *Subfunktion* genannt. Es ist äußerst wichtig zu vermitteln, auf welcher Ebene die Beschreibung erfolgt. Deshalb haben meine Studenten zwei verschiedene Maßstäbe eingeführt, um sie zu definieren: Die Höhe im Verhältnis zum Meeresspiegel (Drachenebene, Wellenebene und Unterwasserebene) und die Farbe (weiß, blau, indigofarben).
- Jeder, der Anforderungen für ein neu zu gestaltendes System formuliert, egal ob es sich um einen Geschäftsprozess oder ein Computersystem handelt, verfasst *Black-Box*-Use-Cases – das sind diejenigen, die sich nicht mit dem Innenleben des Systems beschäftigen. Die Designer der Geschäftsprozesse verfassen *White-Box*-Use-Cases, die zeigen, wie die internen Prozesse in einem Unternehmen oder einer Organisation ablaufen. Auch ein technisches Entwicklungsteam kann den Ansatz der Black-Box wählen, um den operativen Zusammenhang des von ihnen zu entwerfenden Systems zu dokumentieren, und den White-Box-Ansatz wählen, um dann die Arbeitsweise des Systems, das sie gerade entworfen haben, zu dokumentieren..

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

Es ist sehr vorteilhaft, dass das schriftliche Format für Use Cases in solch vielfältigen Situationen verwendet werden kann, aber es verwirrt einen auch. Wahrscheinlich wird so mancher unter Ihnen erleben, dass Uneinigkeit über eine bestimmte Frage der Modellierung besteht, einfach weil die jeweiligen Use Cases für unterschiedliche Zwecke gedacht sind. Und Sie werden im Laufe der Zeit wahrscheinlich unterschiedliche Kombinationen der eben genannten charakteristischen Varianten antreffen.

Die Frage, wie man über Use Cases im Allgemeinen spricht und dabei all diese Varianten berücksichtigen kann, wird uns das ganze Buch hindurch beschäftigen. Am besten ist es, dieses Thema an dieser Stelle anzureißen und dann die Beispiele für sich selbst sprechen zu lassen.

Vielleicht möchten Sie anhand der Use Cases dieses Kapitels überprüfen, was Sie über die Use-Case-Einteilung gelernt haben. Die Use Cases 1, 3 und 5 wurden für den Zweck der Systemanforderungen geschrieben, sind also vollständig ausgearbeitet, betrachten das System als Black Box, sind vom Systemtyp und behandeln ein Ziel auf der Anwenderebene. Das gleiche gilt für Use Case 4, nur dass er nicht vollständig ausgearbeitet sondern formlos ist. Use Case 2, der als kontextsetzende Dokumentation für einen Geschäftsprozess geschrieben ist, ist vollständig ausgearbeitet, betrachtet das System als Black Box und auf der Überblicksebene.

Am stärksten unterscheiden sich Use-Case-Formate durch den Grad ihrer Ausarbeitung. Schauen Sie sich folgende unterschiedliche Situationen an:

- Ein Team arbeitet an einer Software für ein großes Projekt von entscheidender Bedeutung. Es beschließt, dass sich die Extrakosten für einen Extraaufwand lohnen und dass (a) die Use-Case-Schablone ausführlicher und detaillierter sein soll, (b) das Autorenteam im gleichen Stil schreiben soll, um Mehrdeutigkeiten und Missverständnisse zu vermeiden sowie (c) genauere Überarbeitungen nötig sind, um die Use Cases exakter auf Auslassungen und Mehrdeutigkeiten zu untersuchen. Da nur eine geringe Fehlertoleranz akzeptiert werden kann, muss auch die Toleranzgrenze beim Schreiben der Use Cases (Abweichungen bei den Beteiligten) reduziert werden.
- Ein Team von drei bis fünf Personen arbeitet am Aufbau eines Systems, das schlimmstenfalls ein weniger an Bequemlichkeit aufweist, was sich durch einen Anruf leicht beheben ließe. Es betrachtet den ganzen Aufwand als Verschwendung von Zeit, Energie und Geld. Daher einigt man sich auf (a) eine einfachere Schablone, (b) das Tolerieren von stärkeren Abweichungen im Schreibstil und (c) nicht so viele und weniger genaue Überarbeitungen. Die Fehler und Auslassungen in der Modellierung sollen während anderer Projektabläufe korrigiert werden, vor allem durch Besprechungen unter den Teammitgliedern und mit den Anwendern. Im schriftlichen Format kann eine höhere Fehlertoleranz akzeptiert werden, also genügt ein formloser Stil und Abweichungen unter den Beteiligten fallen nicht ins Gewicht.

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

Beide Positionen sind berechtigt. Entscheidungen dieser Art werden auf der Grundlage der einzelnen Projekte getroffen. Das ist die wichtigste Erkenntnis, die ich in methodologischer Hinsicht in den letzten 5 Jahren gewinnen konnte. Natürlich haben wir immer schon gesagt, dass nicht alle Fälle gleich sind. Aber wie man das in konkrete Ratschläge umsetzt, lässt sich nicht genau sagen.

Am schlimmsten ist es, wenn man sich zu stark in Präzision und Striktheit verfängt, wo dies gar nicht nötig ist. Das kostet ein Projekt viel Zeit und Energie. Jim Sawyer hat dies in einer E-Mail-Diskussion einmal so formuliert: »...Hauptsache, die Schablonen sind nicht so formelhaft, dass man sich wieder und wieder in Einzelfragen verbeißt, die die Zuständigkeit der Designer betreffen. Wenn das eintritt, sage ich immer, schaut Euch an, was Sache ist, erzählt, was am wichtigsten ist oder kritzelt es auf eine Serviette.«

Ich habe die Erfahrung gemacht, dass eine einzige Schablone nicht genügt. Es sollten mindestens zwei sein: eine formlose für Projekte ohne großen Aufwand und eine vollständig ausgearbeitete für Projekte, die einen größeren Aufwand erfordern. Eines der beiden Formate lässt sich auf alle Situationen anwenden, die im Laufe eines Projekts auftreten können. Die nächsten beiden Use Cases sind identisch und einmal im formlosen, einmal im vollständig ausgearbeiteten Stil verfasst.

Use Case 4 Etwas Kaufen (Formlose Version)

Der Besteller initiiert eine Warenanforderung und schickt sie an den Bearbeiter. Der Bearbeiter überprüft, ob Geld im Budget vorhanden ist, ob der Preis der Waren stimmt, macht die Anforderung fertig für die Einreichung und schickt sie an den Einkäufer. Der Einkäufer prüft die Lagerbestände und sucht nach dem besten Händler für die Ware. Ein Bevollmächtigter validiert die Unterschrift des Bearbeiters. Der Einkäufer macht aus der Anforderung eine fertige Bestellung und initiiert die Bestellung beim Händler. Der Händler liefert die Ware an den Empfänger, erhält eine Empfangsbescheinigung für die Lieferung (nicht im Umfang des zu untersuchenden Systems). Der Empfänger registriert die Lieferung und schickt die Ware an den Besteller. Der Besteller trägt die Bestellung als ausgeliefert aus.

Der Besteller kann zu jeder Zeit vor dem Empfang der Ware seine Anforderung ändern oder stornieren. Eine Stornierung stoppt alle aktiven Bearbeitungen der Anforderung (löscht sie aus dem System?). Bei einer Preisreduzierung läuft die Bearbeitung weiter. Bei einer Preiserhöhung erfolgt eine Rücksendung an den Bearbeiter.

Use Case 5 Etwas Kaufen (Vollständig Ausgearbeitete Version)

Primärakteur: Besteller

Ziel im Rahmen des Gesamtsystems: Besteller kauft etwas über das System und bekommt es. Bezahlung dafür nicht eingeschlossen

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

Umfang: Geschäft – der generelle Kaufmechanismus, elektronisch oder nicht-elektronisch, wie ihn die Firmenmitarbeiter sehen

Ebene: Überblick

Stakeholder und Interessen:

Besteller: Will das Bestellte erhalten, auf einfache Weise.

Firma: Will Ausgaben beschränken, nötige Käufe aber zulassen.

Händler: Will Bezahlung für alle gelieferten Waren erhalten.

Vorbedingung: keine

Invarianten: Alle abgeschickten Bestellungen wurden von einer zur Autorisierung berechtigten Person bestätigt. Die Bestellung wurde ausreichend protokolliert, so dass der Firma nur ordnungsgemäß gelieferte Ware in Rechnung gestellt wird.

Nachbedingung: Der Besteller hat die Ware erhalten, das entsprechende Budget ist bereit, belastet zu werden.

Trigger: Besteller entschließt sich, etwas zu kaufen.

Standardablauf:

1. *Besteller:* eine Anforderung eröffnen.
2. *Bearbeiter:* Geld im Budget überprüfen, Warenpreis überprüfen, Anforderung zur Einreichung vervollständigen.
3. *Einkäufer:* Lagerbestände prüfen, besten Händler für Ware suchen.
4. *Bevollmächtigter:* Unterschrift des Bearbeiters validieren.
5. *Einkäufer:* Anforderung für Bestellung vervollständigen, Bestellung bei Händler eröffnen.
6. *Händler:* Ware an Empfänger ausliefern, Empfangsbescheinigung für Lieferung erhalten (nicht im Umfang des zu entwerfenden Systems).
7. *Empfänger:* Lieferung registrieren; Ware an Besteller senden.
8. *Besteller:* Anforderung als geliefert auszeichnen.

Erweiterungen:

1a. Besteller kennt Händler oder Preis nicht: Diese Abschnitte nicht ausfüllen und fortfahren.

1b. Zu jeder Zeit vor Empfang der Ware kann Besteller die Anforderung ändern oder stornieren:

Eine Stornierung stoppt alle aktiven Bearbeitungen der Anforderung (löscht sie aus dem System?).

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

Bei einer Preisreduzierung läuft die Bearbeitung weiter.

Bei einer Preiserhöhung wird sie an den Bearbeiter zurück geschickt.

2a. Bearbeiter kennt Händler oder Preis nicht: Nicht ausfüllen und vom Einkäufer eintragen lassen oder telefonisch abklären.

2b. Bearbeiter ist nicht Manager des Bestellers: Auch in Ordnung, wenn der Bearbeiter unterzeichnet.

2c. Bearbeiter lehnt ab: An Besteller wegen Änderung oder Stornierung zurück schicken.

3a. Einkäufer findet Ware im Lager: Ware herauf schicken, Anforderung um diesen Artikel verringern und weiter bearbeiten.

3b. Einkäufer trägt fehlenden Händler und Preis ein: Anforderung wird an Bearbeiter zurück geschickt.

4a. Bevollmächtigter weist Bearbeiter zurück: An Besteller zurück schicken und aktive Bearbeitungen stoppen. (Was folgt daraus?)

5a. Anforderung betrifft mehrere Händler: Einkäufer erstellt mehrere Bestellungen.

5b. Einkäufer fasst verschiedene Anforderungen zusammen: Gleicher Prozess, aber zusammengefasste Bestellungen auf dem Konto vermerken.

6a. Händler liefert nicht rechtzeitig: System soll Nicht-Auslieferung melden.

7a. Teilweise Auslieferung: Empfänger verzeichnet Teillieferung auf der Bestellung und fährt fort.

7b. Teilweise Auslieferung für ein Konto mit mehreren Bestellungen: Empfänger ordnet die Teilmenge der Anforderung zu und fährt fort.

8a. Ware ist nicht korrekt oder von schlechter Qualität: Besteller weist gelieferte Ware zurück. (Was folgt daraus?)

8b. Besteller hat die Firma verlassen: Einkäufer hält Rücksprache mit dem Manager des Bestellers: entweder Besteller neu zuweisen oder Ware zurückschicken und Anforderung stornieren.

Liste der Technik- und Datenvariationen: Keine verschiedenen

Priorität: Unterschiedlich

Versionen: Mehrere

Reaktionszeit: Unterschiedlich

Häufigkeit des Auftretens: 3x täglich

Kanal zum Primärakteur: Internet-Browser, E-Mail-System oder gleichwertig

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

Sekundärakteure: Händler

Kanal zu Sekundärakteuren: Fax, Telefon, Auto

Offene Fragen:

Wann wird eine stornierte Anforderung aus dem System gelöscht?

Welche Autorisierung ist erforderlich, um eine Anforderung zu stornieren?

Wer darf den Inhalt einer Anforderung verändern?

Welche Änderungshistorie muß bei Anforderungen geführt werden?

Was geschieht, wenn der Besteller gelieferte Ware zurückweist?

Wie unterscheidet sich eine Anforderung von einer Bestellung?

Wie referenziert und nutzt eine Bestellung das interne Lager?

Hoffentlich ist klar geworden, dass man nicht einfach sagen kann, »wir schreiben Use Cases für dieses Projekt«. Das besagt nicht viel und alle Empfehlungen oder Prozessdefinitionen, die nur das Schreiben von Use Cases einfordern, sind unvollständig. Ein Use Case, der für ein Projekt Gültigkeit hat, muss nicht automatisch für ein anderes gelten. Es bedarf der Klärung, ob vollständig ausgearbeitete oder formlose Use Cases verwendet werden sollen, welche Teile der Schablone und welche Formate vorgeschrieben sind und wie groß die erlaubte Toleranzschwelle unter den Autoren ist.

Eine vollständige Besprechung zu den Fragen der Toleranz und der möglichen Variationen bei verschiedenen Projekten findet sich in *Software Development as a Cooperative Game* (Cockburn 2001). Wir müssen diese Dinge nicht ausführlich besprechen, um zu lernen, wie man Use Cases verfasst. Aber wir müssen die *Schreibtechnik* von der *Qualität der Use Cases* und den *Projektstandards* unterscheiden.

»Techniken« sind momentane Vorstellungen oder Aktionen, derer sich jemand bedient, wenn er Use Cases modelliert. Dieses Buch befasst sich hauptsächlich mit Techniken: Welche Vorstellungen sind am geeignetsten, wie soll man Sätze formulieren und in welcher Reihenfolge soll man arbeiten. Glücklicherweise sind die Techniken weitgehend unabhängig von der Größe eines Projekts. Ein Fachmann für eine bestimmte Technik kann sie sowohl auf große wie auf kleine Projekte anwenden.

»Qualität« sagt etwas darüber aus, wie geeignet die modellierten Use Cases für den jeweiligen Zweck sind. Im vorliegenden Buch beschreibe ich die am besten für unterschiedliche Zwecke verfassten Use Cases und Use-Case-Teile, die ich bisher gesehen habe. Aber am Ende hängt die Bewertung der Qualität Ihrer Use Cases vom Zweck, der Toleranzschwelle und dem Grad an Aufwand ab, für die Sie sich entscheiden.

Mit »Standards« bezeichnen wir die Vereinbarungen, die Projektteilnehmer treffen, wenn sie ihre Use Cases verfassen. Dieses Buch diskutiert verschiedene ver-

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

nünftige Standards und führt dabei mehrere Schablonen und verschiedene Stile für Sätze und Überschriften vor. Am Ende geben wir einige spezielle Empfehlungen, aber letztlich bleibt die Entscheidung, welche Standards gesetzt oder angepasst werden sollen und wie restriktiv dies gehandhabt wird, der Organisation oder dem Projekt überlassen.

Der größte Teil dieses Buchs beschäftigt sich mit dem kompliziertesten Problem – der Modellierung von präzisen Anforderungen. In folgendem Erlebnisbericht beschreibt der Unternehmensberater Steve Adolph, wie man Use Cases dazu verwendet, Anforderungen zu *entdecken*, statt sie zu dokumentieren.

Steve Adolph: Anforderungen auf neuen Gebieten »entdecken«

Use Cases gelten normalerweise als eine Möglichkeit, bekannte funktionale Anforderungen zu erfassen und zu modellieren. Die Leute halten das an Erzählungen angelehnte Format für verständlicher als die langen Auflistungen der traditionellen Anforderungen. Mit seiner Hilfe können sie wirklich begreifen, was das System tun soll.

Aber was, wenn niemand weiß, was das System tun soll? Die Automatisierung eines Prozesses verändert diesen in der Regel. Die Druckindustrie war vor kurzem von einer der größten Veränderungen seit der Erfindung des Offset-Drucks betroffen: der Entwicklung der Computer-To-Plate-Technik (CTP, direct-to-plate) und des Drucks direkt auf die Druckmaschine (direct-to-press). Früher war das Setzen einer Druckerpresse ein arbeitsintensiver Prozess in mehreren Schritten. Die neuen Druckverfahren machen das Drucken im industriellen Ausmaß genauso einfach wie den Ausdruck eines Textverarbeitungs-Dokuments.

Was würden Sie tun, um die Anforderungen für etwas vollkommen neues zu sammeln, wenn Sie der Analytiker wären, der für das Workflow-Management eines solchen CTP-Drucksystems verantwortlich ist? Zunächst könnten Sie nach den Use Cases des vorhandenen Systems suchen und Akteure und Dienste identifizieren. Diese jedoch gelten nur für das vorhandene System. Noch hat niemand an der neuen Herausforderung gearbeitet, so dass sämtliche Fachexperten das System genau wie Sie erst kennen lernen müssen. Sie gestalten zur gleichen Zeit einen neuen Prozess und eine neue Software. Viel Vergnügen! Wie soll man Spuren im Schnee finden, wenn der gerade frisch gefallen ist? Nehmen Sie sich das vorhandene Modell vor und stellen Sie sich die Frage: »Was verändert sich?« Die Antwort könnte sehr wohl lauten: »Alles«.

Wenn Sie Use Cases verfassen, um Anforderungen zu dokumentieren, gab es bereits jemanden, der eine Vision für das System entwickelt hat. Sie drücken diese Vision lediglich so aus, dass sie jeder verstehen kann. Wenn Sie die Anforderungen jedoch entdecken sollen, sind Sie derjenige, der die Vision entstehen lässt.

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

Verwenden Sie die Use Cases als Brainstorming-Werkzeug. Fragen Sie, welche Schritte des Use Case angesichts der neuen Technologie nicht mehr zur Wertschöpfung hinsichtlich des Ziels beitragen. Überlegen Sie sich eine neue Geschichte, die erzählt, wie die Akteure ihre Ziele erreichen. Die Ziele sind nach wie vor die gleichen, aber einige der unterstützenden Akteure sind verschwunden oder haben sich verändert.

Verwenden Sie den Tiefen- und Oberflächenansatz. Erstellen Sie ein überblickhaftes Modell auf einer hohen Ebene, das Ihre Vorstellungen darüber ausdrückt, wie das neue System funktionieren könnte. Halten Sie die Dinge möglichst einfach, denn es handelt sich um Neuland. Finden Sie heraus, wie der Standardablauf aussehen könnte. Gehen Sie alles mit den Fachexperten durch.

Tauchen Sie anschließend in die Details eines einzelnen Use Case ein. Überlegen Sie, welche Alternativen es gibt. Machen Sie sich die Tatsache zu Nutze, dass Geschichten leicht zu verstehen sind, um die fehlenden Anforderungen zu ermitteln . Lesen Sie sich einen der Use-Case-Schritte durch und fragen Sie sich, »Was passiert, wenn der Kunde einen Hardcopy-Probedruck haben will, keinen digitalen?« Das ist einfacher, als die Ausarbeitung eines vollständigen gedanklichen Modells der Systemfunktionalität zu versuchen.

Kehren Sie danach an die Oberfläche zurück. Was hat sich verändert, nachdem Sie nun in die Details abgetaucht waren? Passen Sie das Modell an und gehen Sie bei einem anderen Use Case erneut auf Tauchstation.

Ich habe die Erfahrung gemacht, dass die Verwendung von Use Cases zum Entdecken der Anforderungen zu einer besseren Qualität bei den funktionalen Anforderungen führt. Sie sind besser aufgebaut und vollständiger.

1.3 Anforderungen und Use Cases

Wenn Sie in Ihren Use Cases Anforderungen beschreiben, sollten Sie diese beiden Dinge bedenken:

- *Es handelt sich tatsächlich um Anforderungen:* Es sollte nicht notwendig sein, sie in eine andere Form von Verhaltensanforderungen umwandeln zu müssen. Gut geschriebene Use Cases führen detailgetreu aus, was das System leisten muss.
- *Sie repräsentieren nicht alle Anforderungen:* Es werden keine Details über externe Schnittstellen, Datenformate, Geschäftsregeln und komplizierte Anweisungen angegeben. Use Cases machen lediglich einen Teil der zu sammelnden Anforderungen aus (etwa ein Drittel) – einen sehr bedeutenden, aber eben nur einen Teil.

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

Jede Organisation sammelt Anforderungen, die ihren Bedürfnissen entsprechen. Es stehen sogar Standards für die Beschreibung von Anforderungen zur Verfügung. In jeder Anforderungsbeschreibung machen Use Cases nur einen Teil der insgesamt dokumentierten Anforderungen aus.

Die folgende grobe Skizzierung der Anforderungen ist äußerst gewinnbringend. Ich habe sie auf der Basis der Schablone erstellt, die Suzanne Robertson und die Atlantic Systems Guild auf ihrer Website und in dem Buch *Managing Requirements* (Robertson und Robertson 1999) veröffentlicht haben. Ihre Schablone wirkt wegen ihrer umfassenden Darstellung abschreckend, deshalb habe ich sie auf die Form reduziert, die ich in der folgenden Grobskizze als Richtlinie aufstelle. Für die meisten realen Projekte ist sie immer noch zu umfangreich, weshalb ich sie bei Bedarf eher noch weiter beschränke. Aber wie umfangreich sie auch immer ist, sie fragt nach interessanten Dingen, die ansonsten unbeachtet bleiben, etwa: »Welche Sicherung durch den Menschen gibt es gegen einen Systemfehler?« und »Gibt es Anforderungen, die von taktischen Überlegungen angeregt wurden?«

Dieses Buch versucht nicht, eine Norm für das Arbeitsergebnis bezüglich Anforderungsanalyse aufzustellen, aber ich bin vielen Leuten begegnet, die noch nie eine Anforderungsskizze gesehen haben. Deshalb soll Ihnen hier eine vorgestellt werden, die Sie in Ihre Überlegungen einbeziehen können. Ihr Hauptzweck ist, die Use Cases in die Gesamtheit der Anforderungen einzuordnen und zu verdeutlichen, dass Use Cases nicht alle Anforderungen umfassen, sondern nur den Teil beschreiben, der sich mit dem Verhalten beschäftigt, der erforderlichen Funktionalität.

Eine plausible Anforderungsskizze

Kapitel 1. Zweck und Umfang

- 1a. Was ist der Gesamtumfang und das allgemeine Ziel?
- 1b. Stakeholder (Wen geht es an?)
- 1c. Was wird betrachtet, was nicht?

Kapitel 2. Verwendete Begriffe/Glossar

Kapitel 3. Use Cases

- 3a. Primärakteure und ihre allgemeinen Ziele
- 3b. Geschäfts-Use-Cases (operative Konzepte)
- 3c. System-Use-Cases

Kapitel 4. Verwendete Techniken

- 4a. Welche Technik-Anforderungen gelten für das System?
- 4b. Mit welchen Systemen und mit welchen Anforderungen hat das System eine gemeinsame Schnittstelle

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

Kapitel 5. Andere Anforderungen

5a. Entwicklungsprozess

Erste Frage: Wer ist am Projekt beteiligt?

Zweite Frage: Welche Werte gehen in die Überlegungen ein (Einfachheit, rasches Umsetzen, Schnelligkeit oder Flexibilität)?

Dritte Frage: Wieviel Feed-Back und Sichtbarkeit wünschen Anwender und Sponsoren für das Projekt?

Vierte Frage: Was kann man kaufen, was muss man aufbauen, wer sind die Konkurrenten?

Fünfte Frage: Gibt es andere Prozessanforderungen (zu Tests, Installation usw.)?

Sechste Frage: Welche Abhängigkeiten gelten für das Projekt?

5b. Geschäftsregeln

5c. Performanz

5d. Operationen, Sicherheit, Dokumentation

5e. Betrieb und Anwendungsfreundlichkeit

5f. Wartung und Portabilität

5g. Ungelöstes oder Aufgeschobenes

Kapitel 6. Sicherung durch Menschen, Juristisches, Politisches und Organisatorisches

Erste Frage: Welche Sicherung für den Systembetrieb gewährleistet der Mensch?

Zweite Frage: Welche juristischen und politischen Anforderungen bestehen?

Dritte Frage: Welche Konsequenzen hat die Systemfertigstellung für den Menschen?

Vierte Frage: Welche Schulungsanforderungen gibt es?

Fünfte Frage: Welche Voraussetzungen und Abhängigkeiten gründen in der Personenumgebung?

Beachten Sie, dass Use Cases nur das dritte Kapitel der Anforderungen ausmachen. Sie beschreiben nicht alle Anforderungen sondern *lediglich* die Verhaltensanforderungen, diese jedoch vollständig. Geschäftsregeln, Glossar, Ziele hinsichtlich der Performanz, Prozessanforderungen und vieles andere fällt nicht in die Kategorie Verhalten. Dafür werden eigene Kapitel benötigt (siehe Abbildung 1.1)

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

1.3.1 Use Cases als Struktur zur Projektverknüpfung

Use Cases verbinden viele Anforderungsdetails miteinander. Sie bilden ein Gerüst, das Informationen aus verschiedenen Anforderungsteilen verknüpft und sie erleichtern die Querverbindung zu den Anwenderprofil-Informationen, zu den Geschäftsregeln und den Anforderungen an das Datenformat.

Über die Dokumentation der Anforderungen hinaus erleichtern Use Cases die Strukturierung von Informationen zur Projektplanung wie Freigabetermin, Teams, Prioritäten und Entwicklungsstand. Außerdem helfen sie dem Designteam bei der Ermittlung bestimmter Ergebnisse, vor allem zum Design der Benutzeroberfläche und der Systemtests.

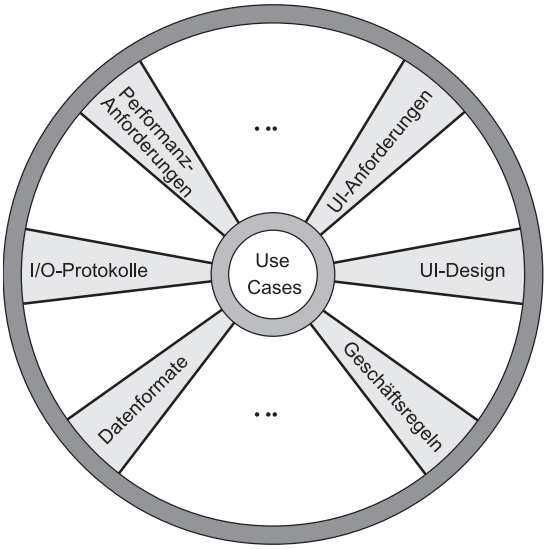

Abb. 1.1: Das »Nabe-Speiche«-Modell der Anforderungen

Auch wenn sie nicht Bestandteil der Use Cases sind, sind all diese Anforderungen mit ihnen verbunden. Use Cases verhalten sich wie die Nabe des Rads in Abbildung 1.1, während die anderen Informationen als Speichen fungieren, die in verschiedene Richtungen führen. Deshalb werden Use Cases gern als zentrales Element der Anforderungen betrachtet, ja sogar als Zentralelement im Entwicklungsprozess.

1.4 Die Vorzüge der Use Cases

Use Cases sind hauptsächlich deswegen so beliebt, weil sie in zusammenhängenden Geschichten erzählen, wie sich das System im Betrieb verhalten wird. Die Systemanwender bekommen eine Vorstellung davon, wie ein neues System aussehen

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

wird. Sie können frühzeitig ihre Konsequenzen ziehen und die Geschichten verfeinern oder ablehnen (»*Das* sollen wir tun?«). Das ist aber nur ein Vorzug, den Use Cases haben, und wahrscheinlich nicht einmal der wichtigste.

Der erste Vorzug von Use Cases wird deutlich, wenn sie nach Anwenderzielen benannt werden, die das System unterstützen soll und deren Auflistung zusammengestellt wird. Diese Liste besagt, was das System leisten wird und enthüllt Systemumfang und –zweck. Sie wird für die unterschiedlichen Stakeholder eines Projekts zum Mittel der Kommunikation.

Die Liste der Ziele wird von Anwenderrepräsentanten, leitenden Angestellten, professionellen Entwicklern und Projektmanagern untersucht, die auf ihrer Grundlage Kosten und Komplexität des Systems einstufen. Sie handeln untereinander aus, welche der Funktionen zuerst aufgebaut und wie die Teams zusammengestellt werden. Die Liste ist ein Gerüst, an dem Komplexität, Kosten, Zeitplan und Statusmessungen festgemacht werden. Mit ihr werden im Lauf der Lebensspanne eines Projekts verschiedene Informationen gesammelt.

Zweitens erweisen sich Use Cases als besonders wertvoll, wenn ihre Autoren ein Brainstorming über all die Dinge durchführen, die bei einem Standardablauf schiefgehen können, diese auflisten und anfangen zu dokumentieren, wie das System darauf reagieren soll. An diesem Punkt wird das Team vermutlich Überraschendes finden, Dinge, an die niemand im Team oder unter den Erstellern der Anforderungen gedacht hatte.

Wenn mir das Verfassen eines Use Case zuviel wird, versuche ich durchzuhalten, bis ich zu den Fehlerbedingungen komme. Wenn ich dann die Fehlerbehandlung dokumentiere, finde ich in der Regel weitere Stakeholder, Systeme, Ziele oder Geschäftsregeln. Wenn wir an der Frage arbeiten, wie mit diesen Bedingungen umzugehen ist, kommt es häufig vor, dass die Geschäftsexperten sich zusammensetzen oder miteinander telefonieren, um herauszufinden, wie das System dann reagieren soll.

Ohne wohlüberlegte Use-Case-Einzelschritte und ohne ein Brainstorming zu den Fehlerfällen werden viele Fehlerbedingungen nicht entdeckt, bis ein Programmierer sie beim Eingeben eines Codefragments findet. Neue Funktionen und Geschäftsregeln sollten jedoch früher gefunden werden. Die Fachexperten sind dann in der Regel nicht länger beteiligt und die Zeit drängt, sodass die Programmierer eine Lösung improvisieren müssen, anstatt das erwünschte Verhalten zu untersuchen.

Wer Use Cases lediglich in einem Absatz festhält, spart durch den geringeren Schreibaufwand eine Menge Zeit und profitiert doch bereits von einem ihrer Vorzüge. Wer die Fehlerbehandlung sorgfältig ausarbeitet, spart eine Menge Zeit, weil er nicht auf den ersten Blick erkennbare Anforderungen frühzeitig entdeckt.

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

1.5 Teilen Sie Ihre Kräfte ein

Sparen Sie Energie oder teilen sie sich Ihre Kräfte wenigstens ein. Wenn Sie alle Details gleich bei der ersten Sitzung niederschreiben wollen, können Sie den einzelnen Themen nicht genügend Zeit widmen. Wenn Sie am Anfang lediglich eine Skizze entwerfen und die einzelnen Use Cases dann auf ihre Essenz bringen,

- geben Sie den Stakeholdern die Gelegenheit, Verbesserungsvorschläge zu machen und die Prioritäten frühzeitig zu erkennen.
- ermöglichen Sie die Aufteilung der Arbeit auf mehrere Gruppen, ein stärker paralleles Vorgehen und eine höhere Produktivität.

Man hört oft, »Ich will einen Überblick aus der Vogelperspektive haben«, »Ich brauche nur eine Skizze« oder »Die Details tragen wir später ein«. Damit soll gesagt werden, dass es für den Moment genügt, mit einem niedrigen Präzisionsgrad zu arbeiten und dass die Feinheiten später hinzugefügt werden können.

Präzision bezeichnet den Grad der Verdeutlichung, den Sie wählen. Mit der Aussage »Ein ‚Kunde' will ein Video ausleihen« machen Sie nicht viele Worte, sagen Ihren Lesern aber eine Menge. Wenn Sie Ihren Stakeholdern eine Liste mit allen Zielen vorlegen können, die das von Ihnen vorgeschlagene System unterstützen wird, geben Sie ihnen mit ein paar Worten eine enorme Menge an Informationen.

Präzision ist nicht das gleiche wie Korrektheit. Wenn Ihnen jemand sagt, »π ist 4,141592«, ist der Präzisionsgrad hoch. Es stimmt aber nicht, und zwar bei weitem nicht. Die Aussage »π ist ungefähr 3« hat keinen hohen Präzisionsgrad (die Nachkommastellen fehlen), aber was gesagt wurde, ist korrekt. Das gleiche gilt für Use Cases.

Gelegentlich fügen Sie den einzelnen Use Cases Details hinzu und erhöhen damit den Präzisionsgrad. Wenn Ihre ursprüngliche Zielaussage mit einem niedrigen Präzisionsgrad nicht stimmt (nicht korrekt ist), vergeuden Sie Ihre Energie, wenn Sie darauf basierend eine Beschreibung von hohem Präzisionsgrad erstellen. Es ist besser, am Anfang eine korrekte Zielliste aufzustellen, bevor man monatelang seine Energie in die vollständige Ausarbeitung der Use-Case-Reihe steckt.

Ich verteile meine Kraft beim Verfassen von Use Cases auf vier Stufen mit unterschiedlichem Präzisionsgrad, gestaffelt nach der Menge an notwendigem Energieaufwand und dem Wert einer Ruhepause nach den einzelnen Phasen:

1. *Akteure und Ziele:* Verfassen Sie eine Liste, die aufzählt, welche Akteure und welche ihrer Ziele vom System unterstützt werden. Kontrollieren Sie Korrektheit und Vollständigkeit dieser Liste. Setzen Sie Prioritäten und ordnen Sie die Ziele einzelnen Teams und verschiedenen Software-Versionen zu. Dann haben Sie die funktionalen Anforderungen auf der ersten Präzisionsebene.

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

2. *Kurzbeschreibung des Use Case oder Standardablauf:* Skizzieren Sie den Standardablauf für die Use Cases, die Sie für die Ausarbeitung ausgewählt haben. Kontrollieren Sie den Rohentwurf, um sicherzustellen, dass das System den Interessen der für Sie wichtigen Stakeholder wirklich dient. Das ist die zweite Präzisionsebene der funktionalen Anforderungen. Anders als bei den nächsten beiden Ebenen, ist das Material hier sehr leicht zu entwerfen.

3. *Fehlerbedingungen:* Vervollständigen Sie den Standardablauf und führen Sie ein Brainstorming nach allen Fehlerfällen, die auftreten könnten, durch. Arbeiten Sie einen vollständigen Entwurf dieser Liste aus, bevor Sie die Reaktion des Systems darauf beschreiben. Das »Ausdetaillieren« der Fehlerbehandlung erfordert wesentlich mehr Energieaufwand, als die Auflistung der Fehlerfälle. Wer gleich mit der Beschreibung der Fehlerbehandlung beginnt, verliert häufig den Mut, bevor er alle Fehlerbedingungen erkannt hat.

4. *Fehlerbehandlung:* Beschreiben Sie, wie das System auf die einzelnen Fehler reagieren soll. Das ist oft eine verzwickte und ermüdende Tätigkeit, die voller Überraschungen steckt. Denn häufig werden im Zuge dieser Beschreibung unklare Geschäftsregeln enthüllt oder neue Akteure und Ziele entdeckt, die unterstützt werden müssen.

Den meisten Projekten stehen zu wenig Zeit und Energie zur Verfügung. Es sollte daher eine der Prioriäten des Projekts sein, dass Sie den Präzisionsgrad festlegen, auf dem Sie arbeiten. Ich empfehle mit Nachdruck die hier dargestellte Reihenfolge.

1.6 Die Aufwärmphase: Eine Anwendungserzählung

Eine Anwendungs*erzählung* ist ein konkretes Beispiel für einen Use Case in Aktion – ein einmaliges, äußerst spezifisches Beispiel dafür, wie ein Akteur das System nutzt. Es handelt sich aber nicht um den Use Case selbst: Die Anwendungserzählung geht in den meisten Projekten nicht in das offizielle Anforderungsdokument ein. Sie ist aber ein nützliches Mittel, das beschrieben und verfasst werden sollte.

Am Anfang eines neuen Projekts haben Sie oder die Geschäftsexperten vielleicht wenig Erfahrung im Verfassen von Use Cases oder kaum eine Vorstellung von der genauen Operation des Systems. Fertigen Sie eine *Skizze* an, die einige Momente aus dem Leben eines der Akteure beleuchtet, um sich mit der Materie vertraut zu machen.

Erfinden Sie für diese Erzählung einen fiktiven aber spezifischen Akteur und geben Sie einen kurzen Einblick in seine Gemütsverfassung – warum will er das erreichen, was er erreichen will, welche Umstände veranlassen ihn, so zu handeln, wie er es tut. Wie auch sonst im Zusammenhang mit Use Cases üblich, müssen Sie

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

nicht viel schreiben. Es ist erstaunlich, wie viele Informationen man mit wenigen Worten übermitteln kann. Halten Sie fest, wie in diesem bestimmten Fall die Dinge vom Anfang bis zum Ende einer Situation ablaufen.

Wichtig ist die Kürze, damit der Leser die Geschichte mit einem Blick erfassen kann. Details, Motive und emotionaler Gehalt sind von Bedeutung, damit alle Leser vom Anforderungsprüfer zum Softwaredesigner und vom Testgestalter zum Autor des Schulungsmaterials verstehen, wie das System optimiert werden kann, um dem Anwender zu nützen.

Eine Anwendungserzählung zu schreiben kostet wenig Energie und Platz, führt aber den Leser sachte zum eigentlichen Use Case hin. Sehen Sie sich ein Beispiel an:

Anwendungserzählung: »Schnellabhebung« machen

Mary bringt auf dem Weg zur Arbeit ihre beiden Töchter in den Kinderhort, fährt dabei an ihren Geldautomat, zieht die Karte durch das Kartenlesegerät, gibt ihre PIN-Nummer ein, wählt SCHNELLABHEBUNG und gibt den Betrag von $ 35 ein. Der Bankomat gibt eine 20-$-Note und drei 5-$-Noten aus sowie eine Quittung, die ihren Kontostand nach der Belastung mit den $ 35 anzeigt. Der Bankomat setzt den Bildschirm nach jeder SCHNELLABHEBUNG zurück, so dass Mary wegfahren kann, ohne befürchten zu müssen, dass der nächste Kunde Zugang zu ihrem Konto hat. Mary mag SCHNELLABHEBUNGEN, weil dabei nicht die vielen Fragen gestellt werden, die die Transaktion verlangsamen. Sie benutzt diesen speziellen Bankomat, weil er 5-$-Noten ausgibt, die sie benötigt, um den Kinderhort zu bezahlen und weil sie ihn vom Auto aus bedienen kann.

Man schreibt Anwendungserzählungen, um sich die Systemanwendung besser vorstellen zu können. Sie sind auch als Fingerübung zum Verfassen von Use Cases geeignet, bei der die Details durchgegangen werden. Gelegentlich kommt es vor, dass ein Team die Erzählung veröffentlicht und sie allen Use Cases oder dem speziellen Use Case, den sie veranschaulicht, voranstellt. Eine unserer Gruppen berichtet, dass sie immer einen Anwender, einen Analytiker und einen Anforderungsautor zusammenbringt, um die Erzählung mit Leben zu füllen, damit der Systemumfang klarer wird und eine gemeinsame Vision von der Systemanwendung entwickelt wird.

Die Erzählung ist mit den Anforderungen nicht identisch. Sie legt vielmehr einen Rahmen für detailliertere und allgemeinere Beschreibungen der Anforderungen fest. Sie dient als Verankerung für den Use Case. Der Use Case selbst ist eine abgespeckte Form der Erzählung – eine Formel – mit einem generischen Akteur anstelle der namentlich benannten Person in der Anwendungserzählung.

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796

1.7 Übungen

1.7.1 Anforderungsdokument

1. Welche Abschnitte der Skizze des Anforderungsdokuments sind relevant für Use Cases, welche nicht? Diskutieren Sie dies mit jemand anderem und überlegen Sie sich, warum Sie diese Frage unterschiedlich beantworten.
2. Erstellen Sie eine alternative, plausible Anforderungsskizze, die als HTML-Dokument ins Intranet gestellt werden kann. Achten Sie auf die Struktur Ihres Unterverzeichnisses und auf Ihre Konventionen bei der Datumseingabe (wozu brauchen Sie Datumskonventionen?)

1.7.2 Anwendungserzählung

1. Schreiben Sie zwei Anwendungserzählungen über Ihren Bankomat. Wie unterscheiden sie sich von unserem Erzählbeispiel und warum? Wie bedeutend sind diese Unterschiede für die Systemdesigner?
2. Schreiben Sie eine Anwendungserzählung über eine Person, die in einer neu eröffneten Videothek die Originalversion von *Ein Zwilling kommt selten allein* ausleihen will.
3. Schreiben Sie eine Anwendungserzählung für Ihr aktuelles Projekt. Lassen Sie eine andere Person eine Anwendungserzählung über die gleiche Situation verfassen. Vergleichen Sie beide Fassungen und diskutieren Sie darüber. Warum unterscheiden sie sich, was wollen Sie wegen dieser Unterschiede unternehmen – sind sie im Toleranzrahmen oder von Bedeutung?

© des Titels »Use Cases effektiv erstellen« (ISBN 978-3-8266-1796-6) 2007 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg.
Nähere Informationen unter: http://www.mitp.de/1796