Dimensional Planning

der Universalhäcksler für User Storys, Iterationen und Releases

Von Stefan Roock

Kleine User Storys haben viele Vorteile. Leider ist nicht immer offensichtlich, wie man eine größere Anforderung in mehrere kleine User Storys zerlegt. Dieser Artikel stellt Dimensional Planning vor: eine relativ unbekannte, aber umso mächtigere Technik zum Zerlegen von User Storys, die sich auf Iterationen und Releases anwenden lässt – und sogar auf Organisationsentwicklung.

Vorteile kleiner User Storys

Die Erkenntnis, dass kleine User Storys wichtige Vorteile bieten, setzt sich immer mehr durch: Es ist viel einfacher, für eine kleine Story eine gemeinsame Vorstellung darüber zu etablieren, was realisiert werden soll. Dadurch lassen sich kleine Storys schneller und genauer abschätzen als große. Eine Sammlung kleiner Storys macht es außerdem im Team einfacher, die Arbeit zu organisieren. Weiterhin führen kleine Storys eher dazu, dass man Möglichkeiten für kleinere Releases findet. Und nicht zuletzt bedeuten kleine Storys kürzere Durchlaufzeiten und verkürzte Feedbackzyklen. Dies wiederum beschleunigt unser Lernen, was zu einem relevanten Wettbewerbsvorteil werden kann.

Schwierigkeiten beim Zerlegen von Storys

Während die Vorteile kleiner User Storys auf der Hand liegen, ist es nicht immer trivial, große Anforderungen in kleine User Storys zu zerlegen. Nehmen wir an, wir wollen ein Fantasy-Adventure-Spiel entwickeln. Dort gibt es eine Geografie mit Städten, Dörfern, Wegen, Flüssen, Wäldern und Bergen sowie Gegenstände und Akteure. All diese Aspekte könnten Kandidaten für User Storys sein. Aber die genannten Aspekte sind jeweils wiederum zu groß, und wir müssen sie erneut zerlegen. Für Gegenstände könnten wir einzelne User Storys erstellen, um die verschiedenen Aktionen zu ermöglichen: erstellen, platzieren, untersuchen, aufnehmen, ablegen, benutzen, verkaufen, kaufen. Leider benötigen unsere Testspieler die meisten dieser Aktionen, bevor sie das Spiel spielen können. Dadurch bekommen wir erst spät Feedback zum Spiel. Dimensional Planning ist eine alternative Zerlegungstechnik, die häufig zu früherem Feedback und kürzeren Releasezyklen führt.

User Storys

Unter einer User Story versteht man eine Anforderung, die aus Nutzersicht beschrieben ist. Jede Story soll klar machen, welches Bedürfnis für welche Person befriedigt werden soll. Eine mögliche Story könnte lauten: „Als Käufer im Web-Shop möchte ich meine Zahlungsdaten speichern können, um sie beim nächsten Einkauf nicht erneut eingeben zu müssen." Welche technische Lösung verwendet wird, um dieses Bedürfnis zu befriedigen, bleibt vorerst weit gehend ausgeklammert. Nach Ron Jeffries ist eine User Story ein „Versprechen auf ein Gespräch". Es steht nicht so sehr die möglichst genaue Spezifizierung der Anforderung im Vordergrund, sondern die Story soll dazu anregen, dass verschiedene Personen (z. B. Produktmanager und Entwicklungsteam) gemeinsam daran arbeiten, dass das beschriebene Bedürfnis möglichst gut und effizient erfüllt wird. Mehr zu User Storys finden sie im Buch „User Stories Applied" von Mike Cohn (vgl. [1]).

Dimensional Planning

Dimensional Planning (vgl. [2]) zergliedert User Storys entlang von Ausbaustufen bzw. Benutzungsqualität. Das Verfahren lässt sich gut mit der Straßenmetapher erläutern, in der User Storys in vier Ausbaustufen zerteilt werden:

- Feldweg
- Kopfsteinpflaster
- Asphaltstraße
- Autobahn

Mit der Feldweg-Implementierung ist das Ziel der User Story mit minimaler Unterstützung erreichbar. ▶

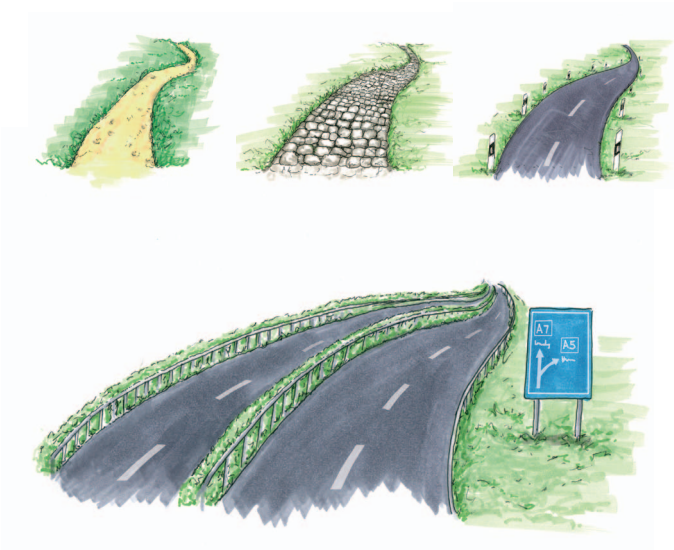

Abb. 1: Verschiedene Ausbaustufen am Beispiel einer Straße

Dadurch ist der Weg umständlich und fehleranfällig, aber machbar. Bei einem System zur Rechnungserstellung könnte dies so aussehen, dass die Anwender eine vorgefertigte SQL-Abfrage direkt an die Datenbank senden, aus dem Abfrageergebnis die relevanten Daten in eine Word-Vorlage kopieren und die Rechnung aus Word heraus ausdrucken.

Bei der Kopfsteinpflaster-Implementierung gibt es bereits einen deutlich höheren Automatisierungsgrad, und die Fehleranfälligkeit ist ebenfalls reduziert. Man nimmt aber weiterhin spürbare Abstriche beim Komfort in Kauf. In unserem Rechnungssystembeispiel könnte eine Kopfsteinpflaster-Implementierung die Rechnungen automatisch aus der Datenbank erzeugen, jedoch nur mit den rechtlich notwendigen Informationen wie Adressat, Datum, Gesamtsumme und Umsatzsteuer. Das System unterstützt jedoch noch keine Einzelpersonen; auch Rabatt und Skonto gibt es noch nicht.

Die Asphaltstraßen-Implementierung entspricht dem „Stand der Technik“ und enthält die Funktionen und den Komfort, die vom Anwender erwartet werden. Das System erzeugt Rechnungen mit Einzelpositionen, Rabatt und Skonto.

Die Autobahn-Implementierung geht über den „Stand der Technik“ hinaus. Häufig finden sich hier Begeisterungsfeatures, die die Anwender und Kunden nicht erwarten und die Alleinstellungsmerkmale gegenüber der Konkurrenz darstellen. Das Rechnungssystem könnte anbieten, dass Kunden das Rechnungslayout selbst festlegen können – z. B. so, dass sie die Rechnungen leichter weiterverarbeiten können.

Wenn wir Dimensional Planning auf unser Fantasy-Adventure-Spiel anwenden, erhalten wir eine gänzlich andere Zerlegung des Gegenstand-Features. Die so entstandenen User Storys könnten wie folgt aussehen (zur Erinnerung: Die gewünschten Aktionen für den Umgang mit Gegenständen waren erstellen, platzieren, untersuchen, aufnehmen, ablegen, benutzen, verkaufen, kaufen):

Feldweg-Story für Gegenstand

- Darstellung des Gegenstandes nur als Text in der Konsole.
- Gegenstände werden eindeutig über eine ID identifiziert.
- Interaktion mit dem Gegenstand ist nur per textuellen Befehlen möglich (keine grafische Darstellung des Gegenstandes, keine Maus-Interaktion mit dem Gegenstand).
- Wenn der Spieler den Gegenstand aufgenommen hat, muss er sich selbst auf einem Zettel merken, dass er diesen Gegenstand besitzt. (Aktion aufnehmen wird als schlichtes Entfernen aus der Landschaft implementiert.)
- Gegenstände lassen sich noch nicht im System untersuchen. Nachdem der Spieler einen Gegenstand aufgenommen hat, kann er in einem Katalog (Text-Datei) nachsehen,

welche Eigenschaften dieser Gegenstand hat und was man damit tun kann.

- Verwenden des Gegenstandes wird implementiert – per Konsolenbefehl.
- Ablegen, kaufen und verkaufen wird in dieser Story gar nicht implementiert.

Kopfsteinpflaster-Story für Gegenstand

- Es gibt eine einfache grafische Darstellung des Gegenstandes.
- Die Interaktion mit dem Gegenstand erfolgt über ein Standard-Kontextmenü (noch nicht über Menüs, deren Look&Feel spezifisch für das Spiel ist).
- Gegenstände können tatsächlich aufgenommen und wieder abgelegt werden. (Der Spieler muss sich nicht mehr die IDs der Gegenstände notieren, die er besitzt.)
- Über die ID können Gegenstände benutzt werden.

Asphalt-Story für Gegenstand

- Es gibt eine ansprechende grafische Darstellung des Gegenstandes.
- Das Menü für die Interaktion mit dem Gegenstand hat ein Look&Feel passend zum Fantasy-Charakter des Spiels.
- Alle Aktionen (erstellen, platzieren, untersuchen, aufnehmen, ablegen, anwenden, verkaufen, kaufen) sind implementiert.
- Alle Aktionen sind mit einem passenden Sound unterlegt.

Autobahn-Story für Gegenstand

- Der Spieler kann eigene Gegenstände erschaffen.

Das Beispiel zeigt, dass die vier Ausbaustufen des Dimensional Planning nützlich sind, um Storys zu zerteilen. Kontextabhängig können mehr oder weniger als vier Stufen sinnvoll sein. So kann man sich zwischen den vier beschriebenen Storys leicht weitere Zwischenausbaustufen vorstellen. Diese würde man definieren, wenn sich herausstellt, dass die beschriebenen vier Storys immer noch zu groß sind.

Iterationen und Releases

Dimensional Planning funktioniert nicht nur für Storys, sondern auch für Iterationen und Releases. So lässt sich eine erste funktionierende Version des Systems in einer Feldweg-Iteration erstellen, indem man die Grundfunktionalitäten alle als Feldweg-Storys implementiert. So kann man den grundsätzlichen Nutzen des Systems sehr früh zeigen und Feedback dazu einholen, ob die Systemfunktionalität tatsächlich Anwenderbedürfnisse befriedigt oder konkrete Probleme löst.

Man spricht auch von einem Walking Skeleton (laufendes Skelett), an das man schrittweise Fleisch, Muskeln, Haut etc. dranhängt.

Dimensional Planning auf Iterationen anzuwenden, ist in der Spiele-Entwicklung durchaus üblich. Dort möchte man die Spiele nicht zu früh in der Öffentlichkeit bekannt machen. Stattdessen will man auf einem großen Spiele-Event das Spiel mit viel Tamtam dem Publikum präsentieren. Also versucht man, sich das Feedback innerhalb des eigenen Unternehmens zu organisieren, indem die Entwickler das Spiel selbst spielen. Da Entwicklern der Umgang mit der Konsole vertraut ist, implementiert man zunächst eine Feldweg-Implementation, bei dem das Spiel über die Tastatur bedient wird und alle Ausgaben nur als Text stattfinden. Schrittweise werden Grafik, Interaktion und Sound verbessert.

Wenn man in der ersten Iteration die Grund-Funktionalitäten in Feldweg-Qualität implementiert hat, würde man in ▶

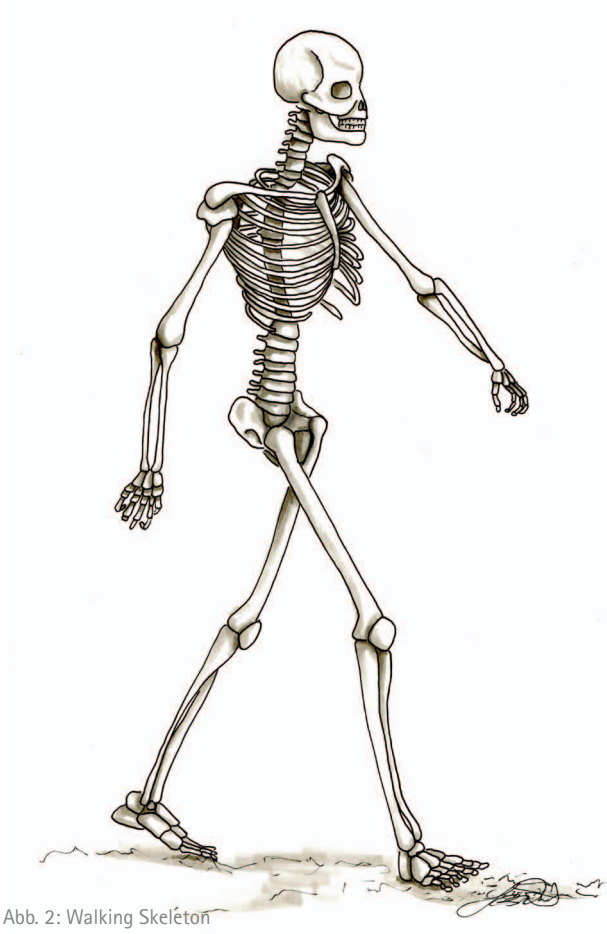

Abb. 2: Walking Skeleton

der folgenden Iteration die Funktionalitäten schrittweise auf höheres Niveau bringen. Dabei würde man typischerweise nicht mit der Gießkanne vorgehen, sondern sich genau überlegen, wo wann welche Ausbaustufe sinnvoll ist. So kann man z. B. die Konfiguration des Loggings sehr lange auf Feldweg-Niveau und die Stammdatenverwaltung lange auf Kopfsteinpflaster-Niveau halten. Stattdessen würde man die werthaltige Kernfunktionalität des Systems mit höherer Priorität auf Asphalt- oder gar Autobahn-Qualität heben.

Scrum außerhalb der Software-Entwicklung

Immer mehr Bereiche außerhalb der Software-Entwicklung liebäugeln inzwischen mit Scrum (vgl. [3]). Häufig ist es in diesen Bereichen schwierig, in einem Sprint von einem Monat oder weniger ein potenziell auslieferbares Produktinkrement zu erschaffen – insbesondere, wenn Hardware im Spiel ist. Häufig behindern dabei antrainierte Denkmuster bei der Planung. Hier kann Dimensional Planning helfen, neue Perspektiven zu schaffen und außerhalb des Gewohnten zu denken.

Ein schönes Beispiel für Dimensional Planning außerhalb der IT stellt die Entwicklung eines Autos mit einem Verbrauch von 1,5 Liter / 100 km (in den USA: 100 Meilen per Gallone) durch die Wikispeed-Gruppe dar (vgl. [4]).

Bei der Entwicklung hat man Wert darauf gelegt, sehr früh ein fahrbares Auto zu haben, und hat dieses dann schrittweise verbessert. So hatte die Gruppe nach nur drei Monaten einen Pkw gebaut, der den angestrebten Verbrauch erreichte. Wie man auf dem Foto auf Abbildung 3 leicht erkennen kann, war es zwar noch nicht besonders hübsch – aber es erfüllte bereits den wichtigsten Zweck.

Eine weitere Ausbaustufe, die 2011 auf der Autoshow in Detroit gezeigt wurde (vgl. Abb. 4), sah schon deutlich ansprechender aus und verfügte bereits über etwas mehr Komfort. Allerdings war auch diese Version noch ein gutes Stück entfernt von dem Ziel eines mehrsitzigen Autos, das in puncto Bequemlichkeit mit handelsüblichen Pkws mithalten kann. Die nächste Ausbaustufe sieht nicht mehr wie ein Rennwagen aus, sondern wie ein sportliches mehrsitziges Alltagsfahrzeug (vgl. Abb. 5).

Tatsächlich hat die Wikispeed-Gruppe nicht nur Dimensional Planning und Walking Skeleton für die Planung eingesetzt, sondern insgesamt massiv auf agile Entwicklungstechniken gesetzt, wie Steve Denning beschreibt (vgl. [5]).

Organisationsentwicklung

Einen weiteren interessanten Einsatzbereich für Dimensional Planning bildet die Organisationsentwicklung. Führt man beispielsweise Scrum in größeren Unternehmen ein, hat es sich bewährt, ein Scrum-Team für die Unternehmenstransition zu bilden, das sogenannte Enterprise Transition Team (vgl. [6]). Dieses Team organisiert sich natürlich nach Scrum und arbeitet mit Sprints. In Scrum muss am Ende jedes Sprints ein potenziell auslieferbares Produktinkrement stehen. Diese Forderung auf die Scrum-Einführung zu übertragen, ist ebenso anspruchsvoll wie wertvoll.

Nehmen wir an, das Transitionsteam stellt nach einer Weile fest, dass die Arbeitsweise des Einkaufs einer unternehmensweiten Scrum-Einführung im Wege steht. Denn der Einkauf beauftragt Dienstleister immer nur nach Festpreis, was für Scrum nicht optimal ist. Das Transitionsteam möchte erreichen, dass Scrum-Projekte als Aufwandsprojekte (Time&Material) vergeben werden können. Dazu muss es eine Reihe von Bedingungen herstellen: Der Einkaufsleiter muss verstehen, warum die Vertragsform wichtig für den Projekterfolg ist. Es müssen Strategien entwickelt werden, wie man bei Aufwandsprojekten die Anbieterauswahl vornimmt, wie man die Gewährleistung handhabt und wie die Schnittstelle zum Controlling bzgl. der Aktivierung in der Bilanz aussieht. Es ist sehr einfach, sich an diesen Bedingungen beim Schreiben der Storys zu orientieren. Leider landet man dann bei einer ganzen Reihe von Powerpoint-Präsentationen. Zuerst erstellt man eine Präsentation für den Einkaufsleiter, die erläutert, warum die Vertragsform so wichtig ist. Und dann eine mit Vorschlägen für das Verfahren zur Anbieterauswahl usw. Keine dieser Präsentationen stellt einen beobachtbaren Fortschritt in Richtung des Ziels der Unternehmensagilisierung dar.

Hilfreicher ist es, sich wieder an dem ursprünglichen Ziel zu orientieren (Aufwandsprojekte vergeben) und sich mit Dimensional Planning die Frage zu stellen, was der kleinste messbare Fortschritt in Richtung des Ziels sein könnte. Und dann kommt man vielleicht zu folgenden Storys (wohlgemerkt: Wir sprechen hier nicht von Produktentwicklung, sondern von Organisationsentwicklung):

- Ein Pilot-Projekt wurde so nach Festpreis beauftragt, dass die Abnahmeentscheidung vollständig beim Product Owner des Auftraggebers lag (und nicht mehr bei der QS-Abteilung oder gar dem Einkauf).
- Es wurde ein Vertragsmuster im Einkauf etabliert, das es generell erlaubt, die Abnahme in die Hände des

Abb. 3: Wikispeed-Auto in Release 1, der angestrebte Verbrauch von 1,5 Liter auf 100 Kilometer ist schon erreicht, aber das Design ist noch verbesserungsfähig (Abdruck mit freundlicher Genehmigung durch Wikispeed)

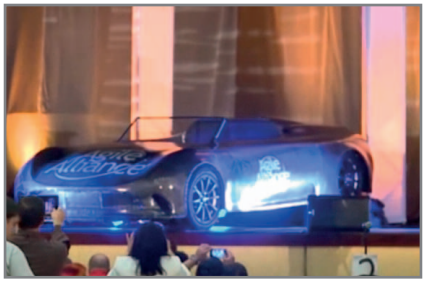

Abb. 4: Wikispeed-Auto SGT01, auf der Autoshow 2011 in Detroit. Dank seiner modularen Konstruktion konnte das Auto innerhalb von zwei Tagen zusammengebaut werden. (Abdruck mit freundlicher Genehmigung durch Wikispeed)

Abb. 5: Mehrsitzige Roadster-Version des Wikispeed-Autos. (Abdruck mit freundlicher Genehmigung durch Wikispeed)

Product-Owners zu geben.

- Ein Pilotprojekt wurde als Festpreisprojekt mit der Möglichkeit des Anforderungstausches beauftragt.
- Diese Vertragsform wurde in ein generelles Vertragsmuster überführt.
- Ein Pilotprojekt wurde so beauftragt, dass der Preis feststand, aber der Auftragnehmer keine Verpflichtung über den zu liefernden Funktionsumfang einging (Design to Cost).
- Diese Vertragsform wurde in ein generelles Vertragsmuster überführt.
- Ein Pilotprojekt wurde als reines Aufwandsprojekt beauftragt.
- Diese Vertragsform wurde in ein generelles Vertragsmuster überführt.

Jeder Schritt ist konkret, stiftet Nutzen und stellt einen Fortschritt in Richtung des eigentlichen Ziels dar.

If I had a hammer ...

Wenn ein kleines Kind einen Hammer in die Hände bekommt, sieht alles wie ein Nagel aus. In diese Falle kann man auch mit Dimensional Planning tappen. Es ist ein sehr mächtiges Werkzeug, und nicht immer das beste. Ausschlaggebend sollte immer sein, was mit der Zerlegung erreicht werden soll. Häufig ist es sinnvoll, zunächst den generellen Nutzen und Einsatzbereich des Produktes mit den Kundenbedürfnissen abzugleichen, und dafür liefert Dimensional Planning i.d.R. sehr gute Ergebnisse. Glauben wir allerdings, dass eine uns unbekannte, innovative Technologie entscheidende Vorteile bringt, sollten wir diese Technologie zunächst verproben. Dann würde Dimensional Planning tendenziell dazu führen, dass die neue Technologie sehr spät integriert wird (typischerwese erst bei der Autobahn-Story). Wenn sich dann herausstellt, dass wir die Technologie nicht beherrschen können, haben wir viel Zeit und Geld verschwendet. Wollen wir beispielsweise unser Fantasy-Adventure mit Holodeck-Technologie anbieten, sollte man mit der Modellierung von Gegenständen oder Akteuren in Holodeck-Technologie beginnen und den Spielfluss zunächst zurückstellen. Die Entscheidung für oder wider Dimensional Planning ist also eine Frage des Risikomanagements. Wo liegen die größten Risiken und wie kann ich sie am besten adressieren?

Dimensional Planning ist besonders gut geeignet, wenn

- eine erste Komplettversion in geringer Qualität hilft, wichtige Fragen zu beantworten und wichtigen Risiken zu begegnen.
- eine Komplettversion in geringer Qualität bereits von einer Nischenanwendergruppe verwendet werden kann.
- man nicht weiß, wie man mit klassischer Planung auslieferbare Produktinkremente in jedem Sprint entwickeln kann (z. B. Hardware, Scrum-Transition). ■

Literatur & Links

[1] Mike Cohn (2004): User Stories Applied: For Agile Software Development. Addison-Wesley.

[2] Koen Von Exem, Walter Hesius: „Dimensional Planning", XP-Days Benelux 2007

[3] Stefan Roock: „Scrum außerhalb der IT", Folien: http://de.slideshare.net/roock/scrum-ausserhalb-der-it

[4] http://www.wikispeed.com/

[5] Steve Denning: „Wikispeed: How A 100 mpg Car Was Developed In 3 Months", http://www.forbes.com/sites/stevedenning/2012/05/10/wikispeed-how-a-100-mpg-car-was-developed-in-3-months/

[6] Simon Roberts, Stefan Roock: http://enterprisescrum.de/

it-agile

Die agile review bequem im Abo!

ab 20 € pro Jahr (3 Ausgaben)

Hier Abo bestellen: www.agilereview.de