

mehr zum thema:

agileproductdesign.com/presentations/user_story_mapping/index.html
github.com/techtalk/SpecLog-Resources/wiki/Story-Map

STORY-MAPS: DIE AGILE PROJEKTENT- WICKLUNG ERFOLGREICH MACHEN

Die agile Projektentwicklung erfüllt oft nicht die hoch gesteckten Erwartungen aller Beteiligten. Story-Maps unterstützen einen wichtigen Mechanismus, der agile Projekte erfolgreich macht und der häufig außer Acht gelassen wird.

Agile Methoden dringen unaufhaltsam in alle Bereiche der Softwareentwicklung vor. Wo IT-Abteilungen den Schritt (noch) nicht wagen wollen, drängt die Business-Seite zu mehr Flexibilität und kürzeren Lieferzyklen, um im Wettbewerb bestehen zu können. Häufig erbringt die Umstellung auf agile Softwareentwicklung jedoch nicht jenen Produktivitätszuwachs, der erhofft und möglich wäre. Die Ursache hierfür ist meistens, dass der agile Kernmechanismus blockiert wird: *Build – Measure – Learn*. Wenn dieser Motor im Projekt nicht anspringt, hilft weder die eifrige Befolgung agiler Rituale, noch die Lieferung in kurzen Iterationen.

Story-Maps sind eine Methode für die Arbeit mit dem *Product Backlog*, mit der dieser Motor in Gang gebracht werden kann: Sie unterstützen beim iterativen Produktdesign, das bei der bloßen Priorisierung einer Liste von User-Stories im *Product Backlog* häufig Acht gelassen wird.

Story-Maps fokussieren immer auf ein gewünschtes Ergebnis, das im Endeffekt erzielt werden soll. Bei einem Web-Shop ist das der Erhalt der bestellten Ware, bei einer Applikation im Finanzbereich kann das beispielsweise die konsolidierte Bewertung eingegangener Risiken sein. Oft lassen sich die Ziele von Projekten auf eine Handvoll solcher Ergebnisse reduzieren. Alle weiteren Anforderungen des Projekts dienen lediglich der Unterstützung dieser eigentlichen Projektziele.

Die Funktion einer Story-Map ist es nun, das System iterativ so zu planen, dass es möglichst früh genügend Funktionalität zur Verfügung stellt, um das gewünschte Ziel zu erreichen. Ab diesem Zeitpunkt kann das Business wesentlich sinnvoller Feedback zu Iterationslieferungen geben und der agile Motor springt an.

Beim Aufbau einer Story-Map werden dazu Benutzeraktivitäten entlang einer Zeitlinie aufgetragen, deren Abfolge das eigentlich gewünschte Ergebnis herbeiführt (siehe Abbildung 1). Dabei wird nicht versucht, einen strikten Workflow zu modellieren, der in vielen Fällen auch gar nicht existiert.

Stattdessen werden die Aktivitäten in einer linearen Abfolge aneinandergereiht, wie sie in einem realen Anwendungsfall auftreten könnten. Dabei kann es durchaus vorkommen, dass zwischen einzelnen Aktivitäten in der Map hin- und hergesprungen wird.

Benutzeraktivitäten beschreiben noch keine konkreten Systemfunktionen, sondern lediglich, welche Aufgabe der Benutzer im jeweiligen Schritt erfüllen muss. Erst die nächste Ebene der Story-Map beschreibt, wie das zu entwickelnde System die jeweilige Benutzeraktivität unterstützen kann. Dies erfolgt in Form von User-Stories, die mögliche Optionen zur Ausprägung des Systems repräsentieren. Üblicherweise gibt es mehr als eine Funktion pro Benutzeraktivität, um unterschiedliche Varianten und Ausprägungen oder einzelne Tätigkeiten innerhalb der Aktivität zu unterstützen.

Die Priorisierung der User-Stories erfolgt nun nicht mehr in einer Liste, sondern pro Benutzeraktivität, sodass möglichst schnell ein minimaler Funktionsumfang zur Verfügung steht, der den gewünschten Endeffekt des Systems liefert. Je kleiner der Funktionsumfang, umso früher beginnt der Mechanismus *Build – Measure – Learn* zu funktionieren. Dabei schneidet man User-Stories pro Aktivität zu Beginn auf ein absolutes Minimum zurück. Das „absolute Minimum“, eine bestimmte Aktivität zu unterstützen, kann bedeuten, dass diese gar nicht unterstützt wird und ganz ausgelassen werden kann.



Christian Hassa

(ch@techtalk.at)

ist Managing Partner bei TechTalk. Bei der XP-Konferenz im Juni 2013 in Wien gibt er eine halbtägige Einführung in Story-Maps.

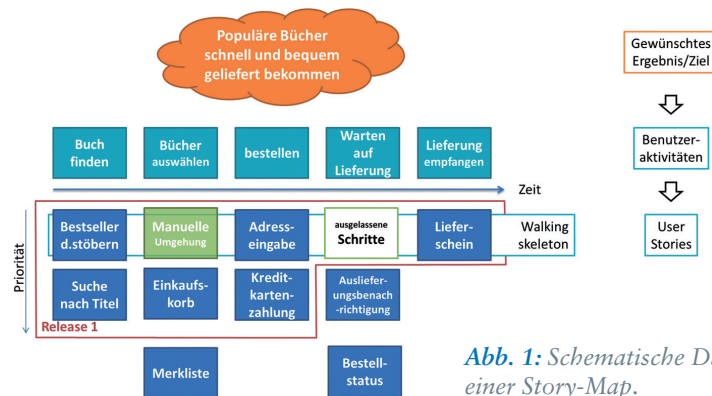


Abb. 1: Schematische Darstellung einer Story-Map.

Die erste Ausprägung des Systems nennt man *Walking Skeleton*, das oft nach wenigen Iterationen erreicht werden kann. Es ist zwar häufig noch nicht produktiv einsetzbar, liefert aber wertvolles Feedback für die weitere Planung des Projekts. Weitere Releases des Projekts können ebenfalls horizontal entlang der Story-Map geschnitten werden. So lassen sich auch produktiv einsetzbare Releases identifizieren, bevor das eigentliche Ende des Projekts erreicht wird. Produktive Zwischen-Releases liefern noch aussagekräftigeres Feedback und sollten in keinem agilen Projekt fehlen.

Unsere Erfahrung zeigt, dass selbst in kleineren Projekten (<=500 Personentage) in bekannten Domänen ca. 25 % des ursprünglich geforderten Funktionsumfangs gar nicht benötigt werden, dafür aber weitere 25 % erst im Laufe des Projekts (bzw. nach der ersten Produktivsetzung) identifiziert werden können. Richtig eingesetzt erschließen Story-Maps dieses Potenzial von agiler Methodik, das allzu oft leider nicht genutzt wird: Zu liefern, was tatsächlich benötigt wird, anstatt sich auf die effizientere Lieferung nicht benötigter Funktionalität zu konzentrieren. ■

Die „14th International Conference on Agile Software Development“ (XP 2013) findet vom 3. bis zum 7. Juni 2013 in Wien statt. Siehe: xp2013.org

