



©iStockphoto.com/evirgen

Geschichtenerzählen in der Softwareentwicklung

Besser erzählen

Seit einigen Jahren spielen Geschichten in der Softwareentwicklung eine wichtige Rolle. Aber nur gut erzählte Geschichten führen ein Projekt zum Happy End. Literatur, Film und Fernsehen zeigen, wie's geht.

von Adrian Krummenacher

Ein Softwareprodukt gleicht einer Unzahl von Kurzgeschichten mit verwobenen Handlungssträngen, die zusammen ein gewaltiges Epos bilden können. Während der Entwicklung werden diese Geschichten weitergegeben, verändert, gekürzt, präzisiert und manchmal auch komplett entfernt. Sich dabei nicht zu verlieren, ist keine leichte Aufgabe.

Auftritt User Stories

Mit dem Aufkommen der agilen Methoden bekamen die Geschichten hinter dem Produkt auch im Alltag des Entwicklers einen höheren Stellenwert. Viele agile Teams arbeiten heutzutage mit einer besonderen Form der Geschichte, der User Story (Kasten: „User Story“).

Leider ist es selten der Benutzer selbst, der den Entwicklern seine Geschichten erzählt. Stattdessen wird eine User Story erst viele Male weitergegeben. Und die Personen, die sie weitergeben, sind oft keine geübten Erzähler. Darum gibt es in der Softwareentwicklung immer wieder viele missverstandene Geschichten.

Lehrmeister gesucht

Die Softwareentwicklung braucht also bessere Geschichtenerzähler. Nun ist die Kunst des Geschichtenerzählens in der Softwareentwicklung aber noch sehr jung und passende Vorbilder sind dementsprechend rar. Glücklicherweise gibt es andere Branchen, die das Geschichtenerzählen schon seit Längerem perfektionieren. Literatur, Film, Fernsehen und Werbung, alle diese Branchen leben von Geschichten. Aber lässt sich von diesem Wissen

profitieren? Die Unterschiede sind groß. So wird eine User Story wohl nie eine überraschende Wendung enthalten oder ein dramatisches Ende nehmen. Aber eine Geschichte bleibt eine Geschichte. Und jede Geschichte beginnt irgendwo ...

Drei Akte und ein totes Pferd

Eine Geschichte hat einen Anfang, eine Mitte und ein Ende. Eine einfache Form ist die Geschichte in drei Akten (Abb. 1):

- Akt I: Der Protagonist tritt auf und stößt auf erste Schwierigkeiten.
- Akt II: Es kommt zu Komplikationen und Rückschlägen.

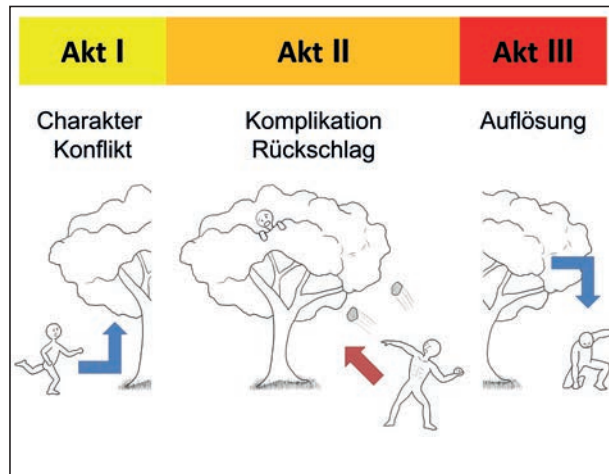


Abb. 1: Geschichte in drei Akten

User Story

Eine User Story ist eine kurze Beschreibung einer Aufgabe, die ein Benutzer mit einem Produkt bewältigen will. Sie wird vom Benutzer oder zusammen mit dem Benutzer geschrieben und enthält in wenigen Sätzen das „Wer“, „Was“ und „Warum“. Da User Stories die Zusammenarbeit mit dem Kunden fördern und den Fokus auf die Problemlösung statt auf die Details der Realisierung setzen, sind sie ein beliebtes Artefakt in der agilen Softwareentwicklung.

Es gibt viele unterschiedliche Formen von User Stories. Eine der bekanntesten behandelt Mike Cohn [1]. Ins Deutsche übersetzt sieht sie etwa so aus: Als (Rolle) will ich, (Etwas) damit (Nutzen). Oder in einem Beispiel: Als Student will ich Unterrichtsmaterial über mein Smartphone abrufen, damit ich unterwegs lernen kann, ohne schwere Bücher mitzuschleppen.

Anzeige



Whitepapers360 –
Ihre zentrale Anlaufstelle, wenn es um technische Informationen geht!

Aktuelle Whitepapers:

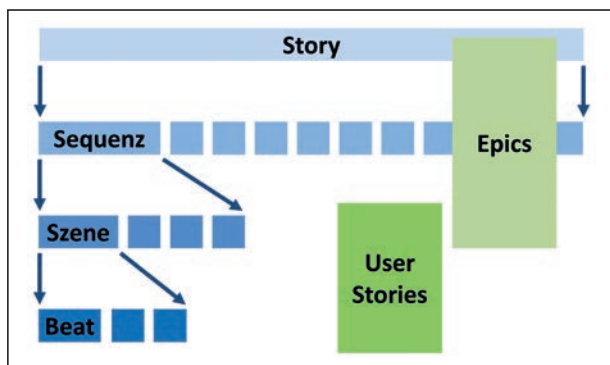
Optimize SharePoint Storage with BLOB Externalization

BCS Connector für SharePoint 2010

Getting Started with Project Templates

Best Practices in Software Monetization – Secure License Management

Abb. 2:
Story,
Sequenz,
Szene und
Beat



- Akt III: Die Geschichte erreicht den Höhepunkt und es kommt zur Auflösung.

Oder mit den Worten von Nancy Beiman [2]:

- Akt I: Bring deinen Helden auf einen Baum.
- Akt II: Lass jemanden Steine nach ihm werfen.
- Akt III: Hole ihn wieder vom Baum runter.

Spätestens im zweiten Akt kommt es zum Konflikt. Linda George [3] verwendet dafür ein einprägsames Bild: Eine gute Geschichte hat ein „totes Pferd im Wohnzimmer“. Ein totes Pferd im Wohnzimmer stellt ein großes Problem dar. Der Held kann versuchen, es zu ignorieren. Aber nach einiger Zeit wird das Pferd beginnen zu verwesen und fürchterlich zu stinken. Ein totes Pferd im Wohnzimmer ist also ein Konflikt, der den Helden zum Handeln zwingt.

Erfüllen User Stories diese Kriterien? Haben sie einen Anfang, eine Mitte und ein Ende? Haben sie ein totes Pferd im Wohnzimmer? Hier ein Beispiel einer User Story für ein Diagnosegerät in einem Labor eines Krankenhauses: „Als Laborant will ich die Detailinformationen zu Reagenzien auf dem Diagnosegerät XY anschauen, damit ich entscheiden kann, ob noch genügend Reagenzien vorhanden sind.“ Das scheint bestenfalls ein Anfang zu sein. Mitte und Ende fehlen. Und was ist mit dem Konflikt? Sind es die fehlenden Reagenzien? Was wären die Folgen, falls der Laborant diesen Konflikt ignorieren würde?

Hier eine etwas längere Version derselben Geschichte: „Am Morgen hat der Laborant das Diagnosegerät XY mit Reagenzien beladen. Das Gerät führt nun schon seit einigen Stunden Tests aus und bald könnten die Reagenzien ausgehen. Da einige Reagenzien eine Vorbereitung von einer Stunde benötigen, muss der Laborant rechtzeitig nachladen. Anderenfalls würde das Gerät stillstehen. Bei einer dringenden Anfrage aus einem Operationsaal könnte ein solcher Stillstand sogar den Tod eines Patienten bedeuten. Der Laborant überprüft die verbleibenden Reagenzien auf dem Gerät und sieht, dass sie tatsächlich ausgehen. Er holt sofort neue Reagenzien aus dem Laborkühlschrank und belädt das Gerät damit.“ Diese Geschichte hat einen Anfang, eine Mitte und ein Ende. Außerdem gibt es auch das tote Pferd im Wohnzimmer. Falls der Laborant das Problem ignoriert, stehen im schlimmsten Fall sogar Menschenleben auf dem Spiel.

Ausgehend von der längeren Geschichte wird es sich der Entwickler gut überlegen, ob er die Information über die Reagenzien wirklich in einer Detailansicht versteckt. Ihm kommt vielleicht sogar die Idee, dass das Gerät den Laboranten aktiv über ausgehende Reagenzien warnen sollte.

Szenen und Beats

Muss eine User Story immer eine ausführliche Geschichte erzählen? Nein. Die notwendigen Details hängen von der Nähe zum Benutzer und vom Domänenwissen der Entwickler ab. Eine User Story „Pizza bestellen“ kann tatsächlich nur aus zwei Worten bestehen, da die meisten Entwickler sich mit Pizza bestellen sehr gut auskennen.

Aber auch wenn es um Herzchirurgie geht, ist eine User Story üblicherweise kürzer. Wieso das so ist, lässt sich am Beispiel eines Spielfilms erklären. Ein Spielfilm erzählt eine Geschichte. Die Geschichte besteht aus einzelnen Sequenzen, die Sequenzen aus einzelnen Szenen und die Szenen aus einzelnen Beats (Abb. 2).

Im Spielfilm Indiana Jones und der letzte Kreuzzug rettet Indiana Jones den Heiligen Gral. Eine Sequenz des Spielfilms ist die Suche nach einem Hinweis auf den Gral in den Katakomben von Venedig. In einer Szene findet Indiana das Grab eines Kreuzritters mit dem gesuchten Hinweis. In einem Beat, dem kleinsten Element einer Geschichte, öffnet Indiana das Grab.

Überträgt man dies in die Welt der agilen Softwareentwicklung, sind User Stories nur Szenen oder Beats einer größeren Geschichte. So wie eine Szene oder ein Beat muss auch die User Story nicht die gesamte Geschichte erzählen. Trotzdem ist es wichtig, die übergeordnete Geschichte zu kennen. Komplexe User Stories, die aus kürzeren User Stories bestehen, heißen Epics.

Epics benötigen einen Anfang, eine Mitte und ein Ende sowie ein totes Pferd im Wohnzimmer. Aus solchen Epics entstehen bessere User Stories und weniger Missverständnisse.

Story Beats entwickeln

Eine Geschichte schreibt sich nicht von selbst. Und mit Epics und User Stories verhält es sich nicht anders. Kann die Unterhaltungsindustrie auch hier etwas bieten, was das Leben in der Softwareentwicklung vereinfacht? Tatsächlich gibt es genügend Gemeinsamkeiten zwischen dem Ausarbeiten von User Stories und dem Erstellen eines Skripts für einen Spielfilm oder eine Fernsehsendung, so dass es sich lohnt, einen Blick darauf zu werfen (Abb. 3).

Ein Skript beginnt mit einer Idee, die der Autor in einer kurzen Log Line festhält. Für den Film Indiana Jones und der letzte Kreuzzug wäre dies: „Abenteuerlustiger Archäologieprofessor rettet Vater und Heiligen Gral.“

Als Nächstes erstellt der Autor daraus eine Prämisse. Das ist ein kurzes Dokument, welches die Geschichte in groben Zügen skizziert: „Henry Jones, Vater des berühmten Archäologieprofessors und Schatzjägers Indiana Jones, sucht schon sein ganzes Leben nach dem Heiligen Gral. Als Henry Jones plötzlich verschwindet, verfolgt Indiana dessen letzten Schritte auf der Grals-

suche, um ihn wiederzufinden. Während Indiana den finalen Hinweis auf den Gral entdeckt, stellt er fest, dass sein Vater von bösen Mächten entführt wurde, die ebenfalls nach dem Gral suchen. Indiana rettet seinen Vater und beide versuchen den Gral rechtzeitig zu finden und so die Welt zu retten.“

Nun entwickelt der Autor aus der Prämisse Story Beats. Der Story Beat ist der Herzschlag der Geschichte, ein kleines Stück Handlung, welches die Geschichte vorantreibt. Als Erstes sucht der Autor nach solchen Beats in der Prämisse:

1. Henry Jones sucht nach dem Gral
2. Henry Jones verschwindet
3. Indiana folgt den Spuren seines Vaters
4. Indiana findet den finalen Hinweis auf den Gral
5. Indiana erfährt, dass sein Vater gefangen gehalten wird
6. Indiana rettet seinen Vater
7. Indiana und Henry Jones retten den Gral

Im nächsten Schritt sucht der Autor nach Beats, welche vorhanden sein müssen, damit der Ablauf der Geschichte einen Sinn ergibt, und entfernt Beats, die für die Geschichte nicht relevant sind. Wie ist es Indiana überhaupt möglich, die Spuren seines Vaters zu verfolgen? Wie erfährt er, dass sein Vater verschwunden ist? Und wie entdecken die bösen Gegenspieler das Versteck des Grals?

1. Henry Jones sucht nach dem Gral
2. Henry Jones verschwindet
3. **Indiana erhält das Tagebuch seines Vaters**
4. **Bösewicht erzählt Indiana über Gral und verschwundenen Vater**
5. Indiana folgt den Spuren seines Vaters
6. Indiana findet den finalen Hinweis auf den Gral
7. Indiana erfährt, dass sein Vater gefangen gehalten wird
8. **Indiana reist nach Deutschland, um seinen Vater zu retten**
9. **Die Bösen nehmen Indiana das Tagebuch ab**
10. Indiana rettet seinen Vater
11. ...

Als Nächstes entwirft der Autor eine Outline. Dieses Dokument beschreibt die gesamte Geschichte in Prosa und jede Szene, welche auch im finalen Skript enthalten sein wird. Zum Schluss schreibt der Autor das Skript. Ein Skript besteht aus der Beschreibung der Umgebung, den Aktionen und dem Dialog.

Log Line und Prämisse haben eine direkte Entsprechung in der Softwareentwicklung: Den Titel einer Epic und deren Beschreibung: „Prüfe die Qualität des Diagnosegeräts XY. Das Diagnosegerät XY führt Tests mit Blutproben durch. Falsche Resultate bedeuten falsche Diagnosen. Darum überprüft das Labor regelmäßig die Qualität der Resultate. Dazu verwendet der Laborant

statt einer Blutprobe ein vorbereitetes Material, von dem das Resultat bereits bekannt ist. Der Laborant überprüft das Resultat, und wenn es innerhalb der erlaubten Toleranz ist, ist die Qualität in Ordnung.“

Auch aus einer Epic lassen sich Story Beats entwickeln:

1. Fabrik stellt Material für Qualitätskontrollen her
2. Laborant lädt Kontrollmaterial in Gerät
3. Laborant führt Test aus
4. Laborant prüft, ob Resultate in der Toleranz liegen

Nun werden die notwendigen Beats für einen logischen Ablauf ergänzt:

1. Fabrik stellt Material für Qualitätskontrollen her
2. **Labor erhält Kontrollmaterial**
3. **Laborant legt Kontrollmaterial in Kühlschrank**
4. **Laborant wird informiert, dass eine Qualitätskontrolle notwendig ist**
5. **Laborant holt Kontrollmaterial von Kühlschrank**
6. Laborant lädt Kontrollmaterial in Gerät
7. Laborant führt Test aus
8. ~~Laborant prüft, ob Resultate in der Toleranz liegen~~
9. **Gerät prüft, ob Resultate in der Toleranz liegen**
10. **Gerät informiert Laborant, wenn eine Kontrolle fehlschlägt**

Story Beats mögen bereits reichen, um aus einer Epic einzelne User Stories zu entwickeln. Trotzdem ist es zumindest für eine komplexe Domäne ratsam, auch eine Outline für die Epic zu erstellen. Sie dient dann als Orientierung für die Entwickler, wenn sie einzelne User Stories umsetzen.

Das Gegenstück zum Skript ist das Detaildesign des Entwicklers. Das Skript ist die Basis, um den Film zu drehen, das Detaildesign die Basis, um Code zu schreiben.

Storyboard

Ein Storyboard besteht aus einer Reihe von Illustrationen, die den kompletten Inhalt eines Spielfilms in der Form eines Comic Strips visualisieren. Ein Pionier des Storyboards war Walt Disney, der es bereits 1928 für das Tonfilmdebut von Micky Maus in Steamboat Willie einsetzte. Die Disney Studios verwendeten Storyboards sehr erfolgreich für ihre Animationsfilme. Bald bedienten sich auch Regisseure im Realfilm des Storyboards. Heutzutage gibt es kaum mehr eine Filmproduktion, die keine Storyboards einsetzt. Das Storyboard lässt den Filmemacher überprüfen, was funktioniert und was nicht, bevor die teure Produktion beginnt. Außerdem gibt es während dem Prozess des Filmemachens, an dem viele Personen beteiligt

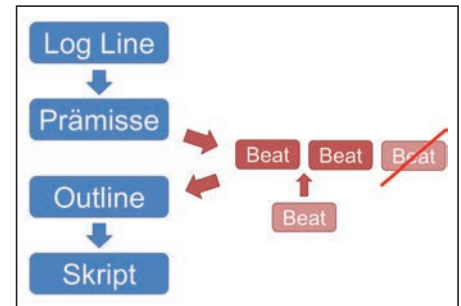


Abb. 3: Der Weg zum Filmskript

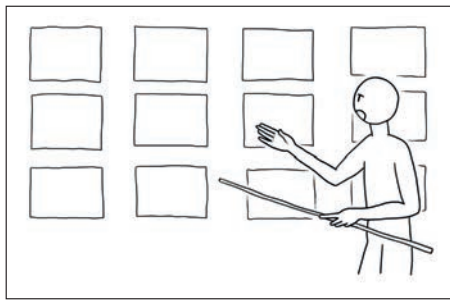


Abb. 4: Story Pitch

sind, allen den nötigen Überblick. Manche Regisseure stellen die Storyboards sogar beim Dreh auf, damit die Szenen für alle sichtbar sind.

Es gibt Storyboards aber nicht nur im Filmgeschäft. Fernsehen, Computerspiele, Theater

und Werbung, alle bedienen sich dieser Technik. In der allgemeinen Softwareentwicklung trifft man sie jedoch nur selten an. Am häufigsten sind sie im User-Interface-Design zu finden. Diese Storyboards sind aber schon sehr nahe an der Implementation. Was der Softwareentwicklung fehlt, sind Storyboards für Epics und User Stories.

Die Stellenbeschreibung „Storyboard Artist“ fehlt üblicherweise auf den Ausschreibungen für ein Entwicklerteam. Aber selbst wenn im Team keinerlei gestalterisches Talent vorhanden ist, ist das noch kein Grund, auf Storyboards zu verzichten. Manche Buchautoren bedienen sich Plotting Boards. Statt mit Bildern füllen sie eine Wand mit Post-its, die die Charaktere, die Beats, die Konflikte und Wendepunkte repräsentieren, und planen so den Verlauf der Geschichte.

Aber welche Vorteile bringt ein Storyboard der Softwareentwicklung? Das Team kann eine Geschichte prüfen, bevor die teure Implementation beginnt. Bei einer Produktentwicklung, wo neben dem Softwareteam noch viele weitere Personen mitwirken, bietet es allen die notwendige Übersicht. Und während der Implementation dient es dem Team als Orientierung und verhindert, dass die Entwickler beim Umsetzen von User Stories die übergeordneten Epics aus den Augen verlieren.

Story Pitch

Walt Disney war nicht nur ein Pionier in der Anwendung von Storyboards sondern auch in der Durchführung von Story Pitches. Während einem Story Pitch präsentiert der Zeichner das Storyboard einem Publikum. Dabei deutet er mit einem Zeigestab auf das aktuelle Panel, beschreibt die Handlung und spricht den Dialog (Abb. 4).

Auf den Story Pitch folgt die Turnover Session. Dabei diskutiert das Publikum das Storyboard und schlägt Anpassungen vor. Alle Ideen werden auf ein zusätzliches Storyboard geheftet. Am Ende entscheidet der Regisseur, welche Ideen er übernehmen will. Dazu dreht er die entsprechenden Panels im originalen Storyboard um und heftet die ausgewählte Idee auf die Rückseite. Panels, die komplett entfallen, werden nur auf die leere Seite gedreht. Story Pitches und Turnover Sessions werden solange wiederholt, bis der Regisseur das Storyboard akzeptiert.

Ein ähnliches Vorgehen gibt es auch in der agilen Softwareentwicklung. In Grooming Meetings werden User Stories vorgestellt, ausdiskutiert und überarbeitet. Ein Story Pitch unterscheidet sich jedoch in einigen wichtigen Punkten von einem Grooming Meeting.

Der offensichtlichste Unterschied ist der visuelle Aspekt. In einem traditionellen Grooming Meeting ist die Ausgangslage meist nur der Storytext und vielleicht ein, zwei Bildschirmskizzen. Der Story Pitch hingegen legt das Schwergewicht auf die Präsentation. Außerdem wird die Story nicht nur erklärt, sondern in erster Linie durchgespielt. Fragen, Diskussionen und Änderungsvorschläge gibt es im Gegensatz zum Grooming Meeting erst während der anschließenden Turnover Session.

Ein Grooming Meeting in der Form eines Story Pitches und einer Turnover Session bietet verschiedene Vorteile. Die Geschichte wird geprobt und nicht nur erklärt. Der Verantwortliche steht unter größerem Druck, sich gut vorzubereiten. Auf dem Storyboard sieht das Team die gesamte Geschichte auf einen Blick. Für Diskussion und Änderungen hat man etwas zum Anfassen. Durch das Drehen der Panels sind Korrekturen sofort ersichtliche. Und nicht zu vergessen: Nur eine Story, die den Pitch übersteht, wird auch umgesetzt.

Zum Schluss

Das Geschichtenerzählen ist für die Softwareentwicklung fremd und ungewohnt. Vieles was in Literatur, Film und Fernsehen gilt, ist für eine Software bedeutungslos oder sogar abträglich. So ist es spannend mitzufiebern, ob ein Held ein Problem bewältigt oder nicht. Aber niemand will gespannt sein, ob das Speichern einer Datei gelingt.

Trotzdem gibt es einiges von Literatur, Film und Fernsehen zu lernen. Die Bedeutung von Anfang, Mitte und Ende. Dass eine gute Geschichte ein totes Pferd im Wohnzimmer haben sollte. Dass User Stories nur Szenen oder Beats einer größeren Geschichte sind. Wie man Story Beats entwickelt, um so bessere User Stories zu schreiben. Dass Storyboards mit Missverständnissen aufräumen. Und schließlich, dass nur Geschichten, die einen Story Pitch überleben, stark genug sind, um da draußen zu bestehen.

Ist das alles, was uns die Welt des Geschichtenerzählens für die Softwareentwicklung zu bieten hat? Noch lange nicht. Aber um es mit den Worten von Michael Ende [4] zu sagen: Das ist eine andere Geschichte.



Adrian Krummenacher ist Softwareingenieur und Scrum Master in Softwareprojekten in der Medizintechnik. Er arbeitet seit zwölf Jahren bei bbv Software Services in der Schweiz. Geschichten liebt er, schon seit er denken kann.

✉ adrian.krummenacher@bbv.ch

Links & Literatur

- [1] Cohn, Mike: „User Stories Applied“, Addison-Wesley 2004
- [2] Beiman, Nancy: „Prepare to Board!“, Focal Press, 2007
- [3] George, Linda: „Fill-in-the-Blank Plotting“, Crickhollow Books, 2009
- [4] Ende, Michael: „Die unendliche Geschichte“, K. Thienemanns Verlag Stuttgart, 1979