User Stories

Anforderungen

- Eine **Anforderung** ist eine Aussage über die notwendige Beschaffenheit oder Fähigkeit,
  - die von einer Person zur Erreichung eines Ziels benötigt wird.
  - die ein System oder Systemteile erfüllen oder besitzen muss, um einen Vertrag zu erfüllen oder einer Norm, einer Spezifikation oder anderen, formell vorgegebenen Dokumenten zu entsprechen.

- In der (Software-)Technik ist eine **Anforderung** (häufig englisch *requirement*) eine Aussage über eine zu erfüllende Eigenschaft oder zu erbringende Leistung eines Produktes, Systems oder Prozesses.
- Anforderungen werden in der Anforderungserhebung aufgenommen, analysiert, spezifiziert und verifiziert. Der Prozess ist in das Anforderungsmanagement, welches die Anforderungen verwaltet, eingebettet. Anforderungen werden üblicherweise in einem Dokument (z. B. Lastenheft) zusammengefasst.

- Quelle: Wikipedia

Anforderungsermittlung

- Ziel:
  - In Gesprächen mit Anwendern ihre Wünsche und Vorstellungen der gewünschten Software/des SW-Systems ermitteln
  - basierend hierauf Anforderungsmodell erstellen
- Elementarer Schritt, entscheidend für Projekterfolg
- Schwierig da unterschiedliche Denkmuster und Sprache zwischen Analytiker und Anwender
- Gesprächsdisziplin und konsequente **Klärung potenzieller Missverständnisse** sind unumgänglich!

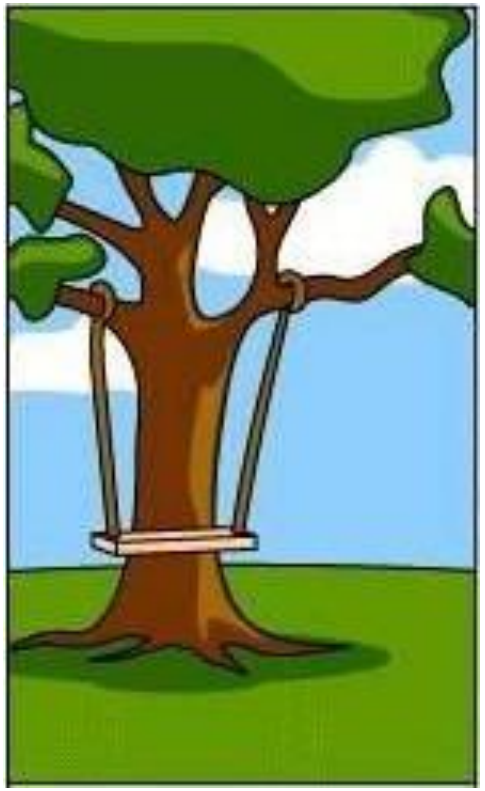
Eine User Story ist ein in der Sprache des Kunden beschriebene Anforderung an das [zu entwickelnde] System, die einen konkreten und für den Nutzer sichtbaren Mehrwert liefert.


Wie?
Was?
Hä?

Unterschiedliches Verständnis von Anforderungen

Was der Kunde erklärte

Unterschiedliches Verständnis von Anforderungen

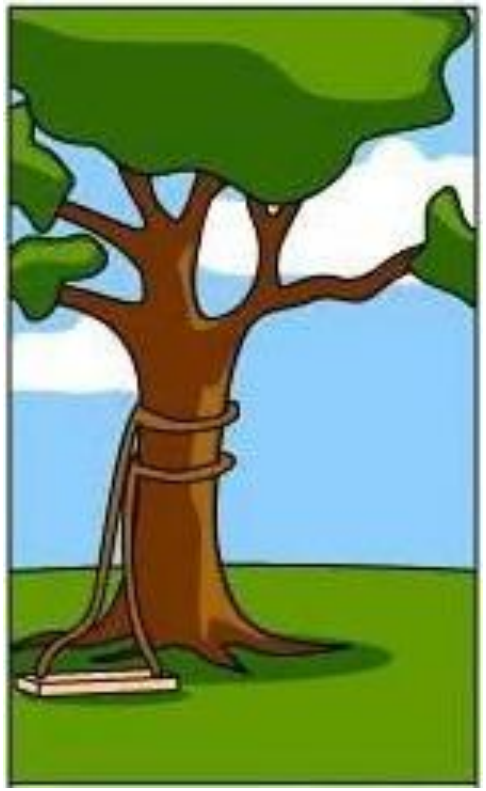


Was der Projektleiter verstand

Unterschiedliches Verständnis von Anforderungen

Was der Analytiker entwarf

Unterschiedliches Verständnis von Anforderungen

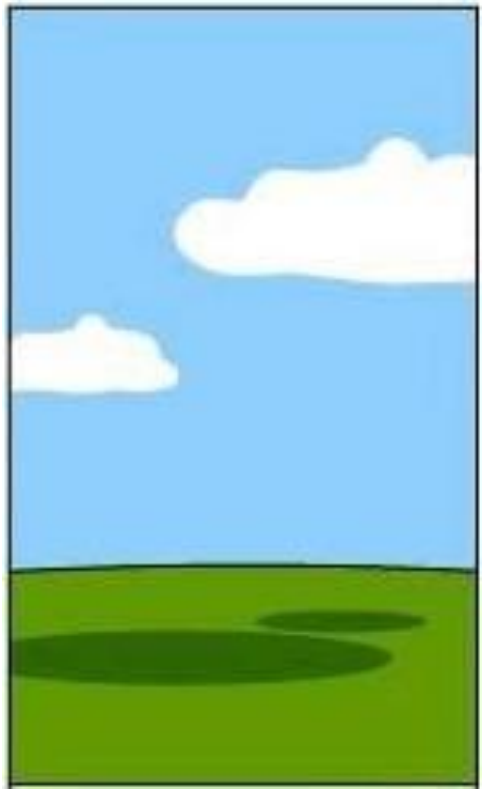


Was der Programmierer programmierte

Unterschiedliches Verständnis von Anforderungen

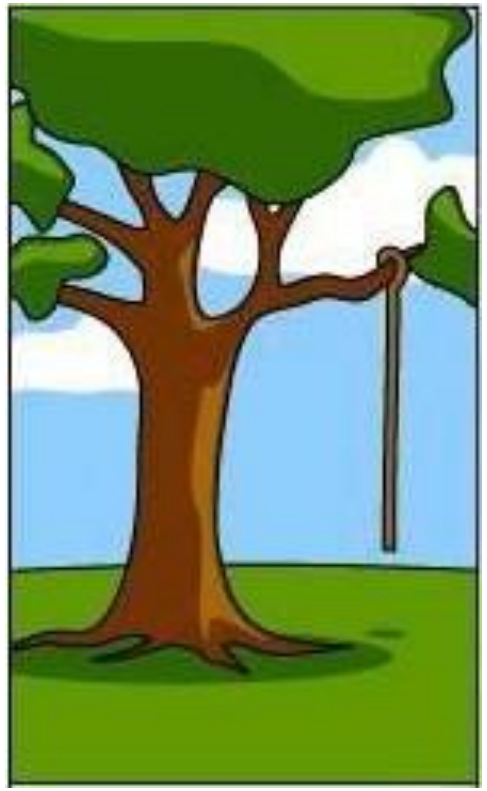
Was der Berater definierte

Unterschiedliches Verständnis von Anforderungen



Wie das Projekt dokumentiert wurde

Unterschiedliches Verständnis von Anforderungen

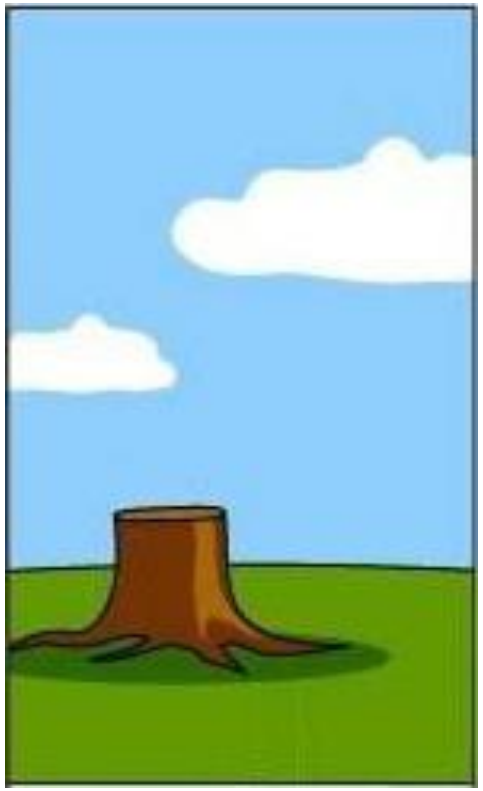


Was installiert wurde

Unterschiedliches Verständnis von Anforderungen

Was dem Kunden in Rechnung gestellt wurde

Unterschiedliches Verständnis von Anforderungen



Wie es gewartet wurde

Unterschiedliches Verständnis von Anforderungen

Was der Kunde wirklich gebraucht hätte

User Stories

- Eine User Story ist ein in der Sprache des Kunden beschriebene Anforderung an das [zu entwickelnde] System, die einen konkreten und für den Nutzer sichtbaren Mehrwert liefert.
- Ziel: Missverständnisse aus dem Weg räumen

https://www.noz.de/deutschland-welt/kultur/artikel/66057/jede-pointe-ein-echter-treffer-karikaturisten-zeigen-ihre-bilanz-des-jahres-2012#gallery&0&0&66057

User Stories **vermitteln** zwischen denen, die die

Software haben wollen

und

Software entwickeln

Unterscheidung User Stories und Tasks

Wunsch des Kunden: Ich möchte, dass man das Spiel nicht anonym spielen kann.

Reaktion der Entwickler: Ok, dann brauchen wir einen Login und eine Registrierungsmöglichkeit

Wir brauchen dann:

- Eine Datenbank
- Eine Login-Maske
- Ein CSS für die Maske
- Eine Netzwerkkommunikation

Damit kann der Kunde nicht viel anfangen

User Stories

- beschreiben Anforderungen an ein Softwaresystem
- Aktiver Stil
- Konkreter und sichtbarer Mehrwert für den Kunden:
  - „Als Nutzer will ich meine persönlichen Daten ändern können."
    → Nach Implementierung hat System einen Mehrwert
  - Keine Userstory: „Die Software soll in Java implementiert werden."
- Unschärfer als Anwendungsfälle (vgl. VL Software Technik … später)
- In einem knappen Satz sollte stehen, was getan werden soll
  - Als Gast möchte ich mich als neuer Nutzer registrieren um mitspielen zu können.
  - Als Nutzer möchte ich meine persönlichen Daten ändern können.
- User-Story-Muster:
  - Als <Benutzerrolle> will ich <das Ziel> [,so dass <Grund für das Ziel>].
- WICHTIG!
  - Stories sind nur ein PLATZHALTER!!
  - Erstmal (alles) aufschreiben, damit es nicht vergessen wird :-)

User Stories sind Platzhalter

- Platzhalter für Konversationen (mit dem Kunden/Product Owner)
- Zu Beginn eines Projektes typischerweise: Anforderungsworkshop
- Wenn es an die Umsetzung geht, muss konkret geklärt werden, worum es geht!
  - Das kann dann z.B. auch durch einen Anwendungsfall geschehen
  - Vorher nicht zu viele Details, evtl. wird die Story nie umgesetzt
- Im der ursprünglichen Verwendungsweise:
  - User Stories werden auf eine Story-Karte geschrieben (DIN A5)
  - An die Wand (Scrum-Board) gehängt (ggf. einfach wieder verworfen)
  - werden um weitere Aspekte erweitert
- Fragen des Entwicklers an den Kunden können vermerkt werden
- Unvollständigkeit von Stories sind Teil ihres Konzeptes
- Finale User-Story wird iterativ entwickelt!

Woraus besteht eine User Story: 3C

- Card
- Conversation
- Confirmation

3C

Die Karte (card)

- Physisches Token
- Wird für die Planung verwendet
- Erinnerung für ein Gespräch
- Wird häufig kommentiert

Inhalt der Karte: Beschreibt in einem Satz den Kern der umzusetzenden Anforderung

- <Rolle> - wer führt die Aktion durch oder ist der Nutznießer (z.B. ein anderes System)
- **<Aktivität>** - Aktion die von der Rolle durchgeführt wird
- <Business value> - Geschäftswert, der durch die Aktion geliefert werden soll

- Als Nutzer möchte ich mich **beim Spiel anmelden**, um mit anderen Spielern spielen und chatten zu können.

3C

Die Konversation (u.a. mit dem Kunden)

- Der Anforderung
- Mündliche Konversation / Workshops
- Kann mit Dokumenten / Mockups / … ergänzt werden

- Weitere Details der Story:
  - Die Nutzerdaten sollen in einer relationalen Datenbank gehalten werden.
  - Der Nutzer soll sich über ein Formular anmelden können.
  - Wenn der Nutzer bereits eingeloggt ist, soll die andere Sitzung beendet werden.
  - Nach fünf Fehlversuchen vom selben Client soll die Eingabemöglichkeit zunächst für zwei Minuten gesperrt werden. Nach weiteren fünf Fehlversuchen für weitere 30 Minuten.
  - …

3C

Bestätigung (Confirmation)

- Akzeptanzkriterien: Werden dazu verwendet, um festzustellen, ob die Story abgeschlossen ist
- Tipps für die Akzeptanzkriterien
  - Schreibe die Akzeptanzkriterien zusammen mit dem Kunden
  - Definiere “vollständig” gemeinsam mit dem Kunden
  - Alle Kriterien müssen erfüllt sein, bevor die User Story abgeschlossen werden kann
  - Alle Risiken, Voraussetzungen, Probleme und Abhängigkeiten aufnehmen
  - Verwende möglichst eine Vorlage
- Beispiele für Akzeptanzkriterien:
  - Die Loginmaske ist übersichtlich.
  - Der Nutzer bekommt Feedback, ob das Login fehlgeschlagen ist. Es wird dabei aus Sicherheitsgründen nicht gesagt, ob der Nutzername nicht existiert oder das Passwort falsch war.
  - Die Klassen, die das Login durchführen sind vollständig getestet worden. (JUnit 100% Abdeckung bei den Verzweigungen und im Code)
  - Das Passwort wird verschlüsselt in der Datenbank gespeichert.

3C

User Stories

- enthalten **keine Technik** und keine Details über die grafische Umsetzung (z.B. Login-Button)
  - Constraints (Technical Stories sollten separat verwaltet werden)
- Eigenschaften guter User Stories: INVEST
  - I: Independent: User Stories sollten unabhängig voneinander sein
  - N: Negotiable: User Stories sollten verhandelbar sein
  - V: Valueable: User Stories sollten einen Wert für den Kunden haben
  - E: Estimatable: User Stories sollten schätzbar sein
  - S: Small: User Stories sollten klein sein
  - T: Testable: User Stories sollten testbar sein.
- Große Stories: Epic
  - Noch nicht ganz klar, was Details sind
  - Aus Epic werden i.d.R. Reihe von User-Stories.

- Epic und User Story in Jira verwaltbar (Scrum: Product Backlog)

3C

Eigenschaften guter User Stories: INVEST

Independent

Negotiable

Valuable

Estimatable

Small

Testable

- Abhängige Stories erzeugen ein Reihenfolgenproblem in Bezug auf ihre Umsetzung
- Stories sollten sich möglichst nicht überlappen
- Fassen Sie ggf. Stories zusammen
- ABER: u.U. Probleme mit zu großen Stories → Über Prioritäten arbeiten, d.h. natürliche Ordnung finden

- Keine guten Stories:
  - Zahlung mit Visa
  - Zahlung mit Mastercard
  - Zahlung mit American Express
- Besser
  - Zahlung mit Kreditkarte
- Abhängigkeiten: Vor dem Einloggen muss man sich registrieren können

3C

Eigenschaften guter User Stories: INVEST

Independent

Negotiable

Valuable

Estimatable

Small

Testable

- Stories sind die Basis für die Gespräche mit dem Kunden
- Sie sind kein Vertrag, sondern eine Gesprächsgrundlage
- Können jederzeit verändert oder gar verworfen werfen

3C

Eigenschaften guter User Stories: INVEST

Independent

Negotiable

Valuable

Estimatable

Small

Testable

- Die Stories sollen aus der Sicht des Kunden geschrieben werden
- Sie müssen verständlich (in der Sprache des Kunden) geschrieben werden
- Sie sollen einen Mehrwert für den Kunden haben

3C

Eigenschaften guter User Stories: INVEST

Independent

Negotiable

Valuable

Estimatable

Small

Testable

- Man soll in der Lage sein, den Aufwand einer Story anhand ihrer Beschreibung abschätzen zu können
- Dafür muss das Team, welches die Story gemeinsam schätzt, die Story vollständig verstanden haben

3C

Eigenschaften guter User Stories: INVEST

Independent

Negotiable

Valuable

Estimatable

Small

Testable

- Eine Story sollte möglichst klein sein
- Sie sollte auf jeden Fall innerhalb eines Sprints abgeschlossen werden
- Wenn Story zu groß, wird es u.U. schwierig alle Leute des SWP-Teams zu beteiligen

3C

Eigenschaften guter User Stories: INVEST

Independent

Negotiable

Valuable

Estimatable

Small

Testable

- Eine Story muss eine definierte Definition on Done haben:
  - Wann ist eine Story vollständig abgeschlossen
  - Was soll eine Story alles enthalten

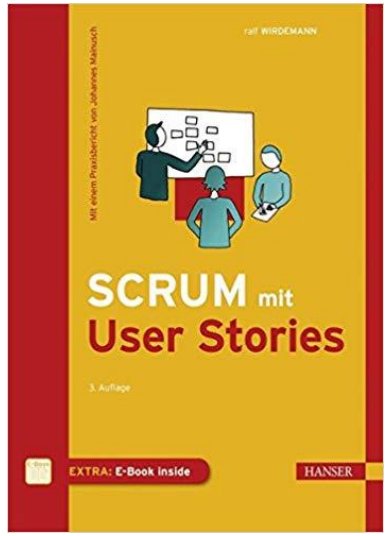
3C

Schätzen von User Stories

- User Stories sind die Dinge, die in einem sog. Sprint in Scrum behandelt werden
- Jeder Sprint besteht aus einer Menge von User Stories
- Problem: Welche und wieviel Stories können wir in dem Sprint eigentlich schaffen?
- Wichtig: Das weiß man zu Anfang nicht! → Team Velocity ist ein Wert, der erst im Laufe der Zeit entsteht
- Warum soll man denn dann schätzen?
  - Damit man irgendwann einmal die Team Velocity bestimmen kann
  - UND NOCH VIEL WICHTIGER: Eine Story kann man u.U. auch dann nicht schätzen, wenn man sie noch nicht genau **verstanden** hat! ! → Schätzen als Mittel, ob die Story verstanden wurde

Quellen und weitere Informationen

- Ralf Wirdemann: Scrum mit User Stories, Hanser-Verlag



- Boris Gloger: Scrum: Produkte zuverlässig und schnell entwickeln, Hanser-Verlag