

c-Akademie

**Software Requirements Engineering:
Vom Kundenwunsch zum Entwicklerticket**

**Zusammenfassung
des Vortrags**

**Stefan Matt
Geschäftsleiter**

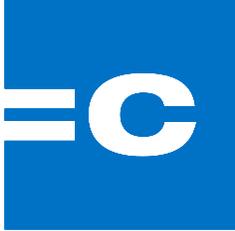


Ein Blick zurück

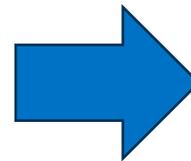
Historische Entwicklung im Requirements
Management

Foto: Tim Trad / Unsplash

Historische Entwicklung im Requirements Management



Zusammenfassend:



Von der **Beschreibung des Systems** zu den **Bedürfnissen der Benutzer**

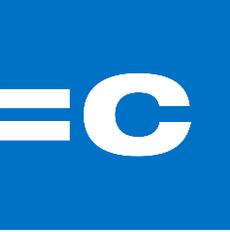
Agile Requirements

Eine Liste mit User Stories ist nicht genug



Grafik: <https://www.jamesbond.de/wp-content/uploads/2022/09/quad-theworldisnotenough.jpg>

Die wichtigsten zwei Kritikpunkte der Agilen am klassischen Requirements Engineering



Verschwendung

- Anforderungen die **erfasst** und dokumentiert werden, aber **nicht umgesetzt** und geliefert, sind Verschwendung.
- Vorgängig durchgeführte Arbeiten sind wie **Halbware** am Lager.

Änderungen

- Die Stakeholder und Benutzer wissen nicht was sie wollen. Es wird sich im Laufe des Projekts sowieso **alles ändern.**

Das richtige Mass finden

- Man muss einen **Überblick** haben, um wegzulassen und zu priorisieren
- Man braucht **nicht alle Details** ganz früh im Projekt
- Hier ist Abwägen gefragt und das Bewusstsein, dass es **Unsicherheiten** gibt
- Anforderungen können auch **rollierend** weiter detailliert werden (Agile Anforderungsentwicklung)

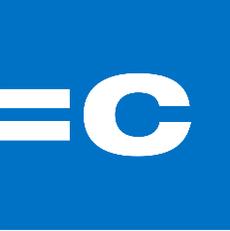


Foto: Christophe Hautier / Unsplash

Was User Stories ignorieren: Die Domäne

Was der Benutzer will, ist nicht unbedingt das, was er braucht

- **Domänenanforderungen** (Gesetze, Physik, Business-Regeln, etc.) werden von User Stories nicht abgedeckt
- Domänenmodelle (statisch, dynamisch) sind essentiell zum **Verständnis der Domäne** und für eine gemeinsame Sprache. Die Domäne lässt sich mit User Stories nicht beschreiben



Was User Stories gut können

- Beschreibung kleiner **testbarer** Funktionalitäten
- User Stories stellen immer den **Benutzer** in den Mittelpunkt
- Gut merkbare **Kurzbeschreibung** von Programmier-Aufgaben
- Geeignet für Task-Boards

User Stories sind ein passendes Mittel, um die Umsetzung zu strukturieren

Vom Erfassen zum Umsetzen

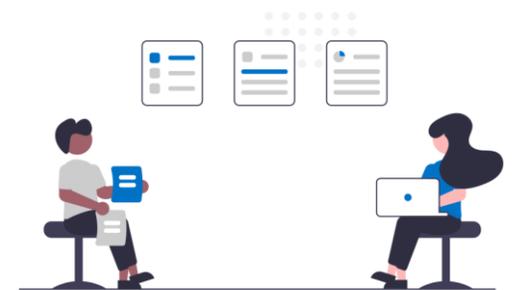
Ein Blick auf das Gesamte



Foto: Vienna Reyes / Unsplash

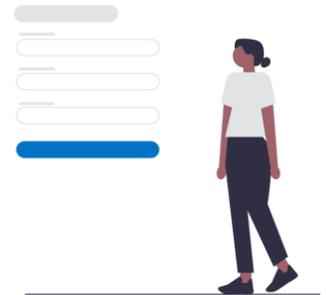
Fachbereichsanalyse

- Projektüberblick, **Kontextdiagramme** zum System, Grenzen
- **Nutzer**(-gruppen), andere **Anspruchsgruppen** (Stakeholder)
- Dokumentation des **bestehenden Systems** (auch wenn es ohne IT funktioniert)
- statische und dynamische Modellierung der Domäne um eine **Taxonomie** zu erhalten

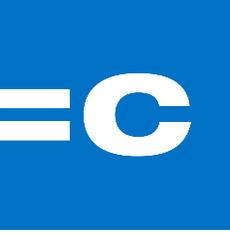


Schnittstelle zum Benutzer

- Die **Struktur der Anforderungen** entspricht nicht unbedingt der **Struktur der Realisierung**
 - Releases, Features, Module, Administrative Aufgaben, Fehler, etc. kommen in den Anforderungen nicht vor.
- Wir setzen hier auf **Anwendungsfälle** für Anforderungen → Benutzerziele
- Wir setzen auf **Tasks** für die Realisierung



Schnittstellen zu anderen Systemen / Interoperabilität

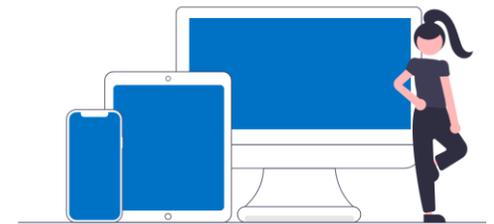


- **Datenübernahme** vom bestehenden System
- **Kommunikation** mit anderen Systemen (auch manuell)
- **Dateiformate**



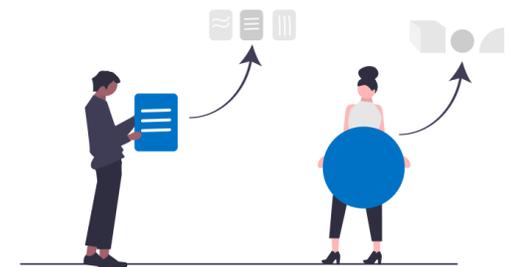
Schnittstellen zur Laufzeitumgebung

- Technologie **Benutzeroberfläche**
- **Systemleistung**
- **Sicherheit:** Datensicherheit / Betriebssicherheit / Systemsicherheit
- **Auslieferung:** Installation / Deployment
- **Verwendete Funktionen** der Laufzeitumgebung (Portierbarkeit)



Weitere Vorgaben zur Durchführung

- Einschränkungen für **Design** und **Implementation** (Normen, Reaktionszeiten, Speicherplatz, Technologien, Werkzeuge, ...)
- Gewichtung der **Softwarequalitätseigenschaften** (ISO 25010)
- Realisierungsprioritäten, **Termine**
- Wie wird **getestet**?
- Abschluss (vom Testbetrieb zum Echtbetrieb und Abnahme)



Es kommt drauf an...

One Size does not fit all



Was bestimmt den Anforderungsprozess?

- **Projektgrösse** (wieviele involvierte Partner)
- Projekt **Kompliziertheit / Komplexität**
- Formale/regulatorische Anforderungen
- Inhouse vs. Auftragsverhältnis
- Neuentwicklung vs. Weiterentwicklung

Die simpelste Form ist es, direkt mit der Umsetzungsstruktur zu arbeiten. Das ist das, was die Agilisten vorschlagen.



Vielen Dank!



creasoft ag

Software: Planen. Entwickeln. Testen.

Let's go!

creasoft ag

Software: Planen. Entwickeln. Testen.

Weiherring 115 . FL 9493 Mauren . T +423 375 8200 . www.creasoft.li