Nichtfunktionale User-Stories

Spezifikation und Test

Johannes Bergsmann
Berater, Trainer

© Software Quality Lab www.software-quality-lab.com

Über Software Quality Lab

LEISTUNGSPORTFOLIO

CONSULTING

- Management Consulting
- Prozesse und Vorgehensmodelle
- Requirements und User Experience
- Architektur und Modellierung
- Code Analyse und Metriken
- Teststrategie und -konzeption

OPERATIONAL SERVICES & TESTCENTER

- Testmanagement und -spezifikation
- Testautomatisierung
- Softwareverifikation und -validierung
- TestCenter
- Requirements Engineering
- Reviews und Code Analyse

ACADEMY

- Requirements und Usability
- Architektur und Modellierung
- Testen und Automatisieren
- Agile Softwareentwicklung
- Projektabwicklung und Vorgehensmodelle
- Lehrgänge und Zertifizierungen

TOOL EXPERTISE

- Tool Evaluation Center
- Beschaffung, Einführung und Schulung
- Tool-Integration und Schnittstellen
- Prozessautomatisierung
- Application Lifecycle Frameworks
- Tool-Studien und Tool-Expertisen

© Software Quality Lab www.software-quality-lab.com

software quality lab
Innovation Meets Quality

Über Software Quality Lab

Kunden (Auszug)

Energie & Versorgung

ANDRITZ Hydro, OMICRON, LINZ AG, Fronius, OMV, RheinEnergie, RE, SIEMENS, Tele Automation Components

Elektronik & Automation

bachmann., SALOMON AUTOMATION, TRIDONIC, sprecher automation, Infineon, B&R, SEZ, SKIDATA

Industrie & Engineering

MAGNA, BOSCH, ZF, LIEBHERR, PERI, ENGEL, TRUMPF, ZUMTOBEL, welser profile, Webasto, voestalpine, blum, Miba, BRP, SANDVIK, Gebrüder Weiss, AVL

Dienstleistungen & Handel

BMF, AMS, BG | Phoenics, vie, SPAR, BUNDESDRUCKEREI

Medizin & Pharma

ACH, Otto Bock, ÖÖGKK Forum Gesundheit, kav, VDW, Österreiches Rotes Kreuz, greiner bio-one, SVC, systema

Software & IT

BUSINESS SOFTWARE, Atos, kommunale informationsverarbeitung, inet, REVAL, NTS, mic, DATEV, PORSCHE INFORMATIK, T-Systems, utimaco

Finanz & Versicherung

RACON, kommunal KREDIT, BRZ, IT, Allianz, ÖNB, ÖBB, Österreichische Lotterien, Raiffeisen Wohn Bausparen, VERSICHERUNGSKAMMER BAYERN, GED

Mobilität & Kommunikation

FREQUENTIS, hr online.de, swisscom, kapsch, T-Mobile, TELEKOM AUSTRIA

© Software Quality Lab www.software-quality-lab.com

software quality lab
Innovation Meets Quality

Nichtfunktionale Anforderungen

Inhalt

- Agiles Requirements-Engineering
- Nicht-funktionale Anforderungen
- Spezifikation von NF User Stories
- Test(spezifikation) von NFA
- Resümee

© Software Quality Lab www.software-quality-lab.com

software quality lab
Innovation Meets Quality

Agiles Requirements-Engineering

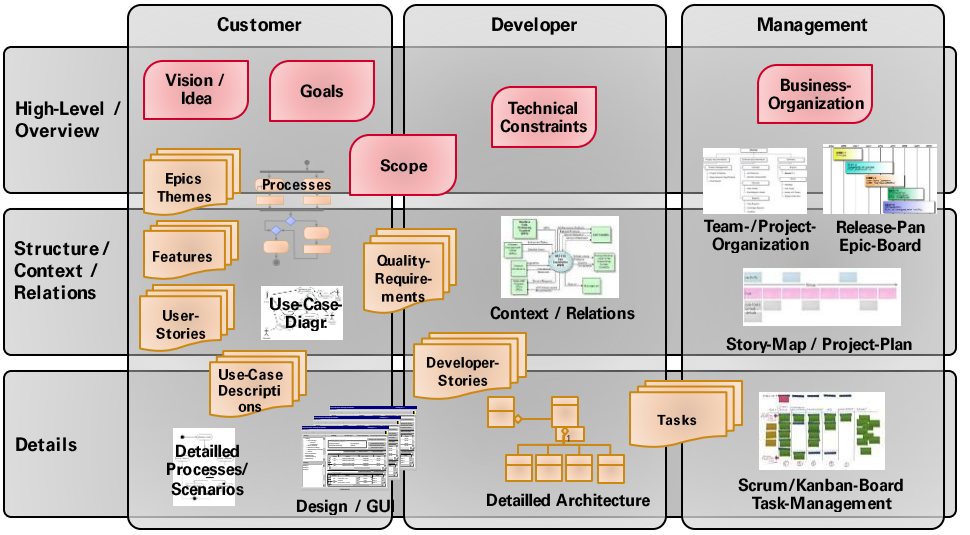
Kurzüberblick über die Artefakte

© Software Quality Lab www.software-quality-lab.com

software quality lab
Innovation Meets Quality

RE im agilen Umfeld

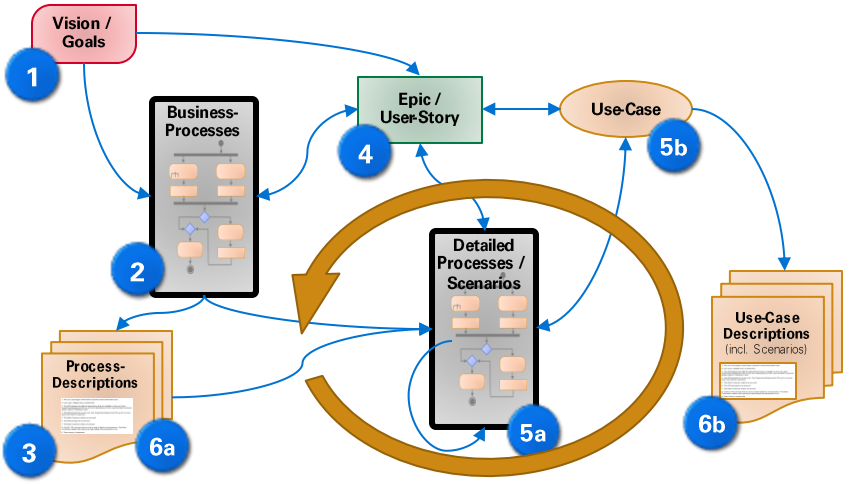
Artefakte und Tools im Überblick



© Software Quality Lab www.software-quality-lab.com

RE im agilen Umfeld

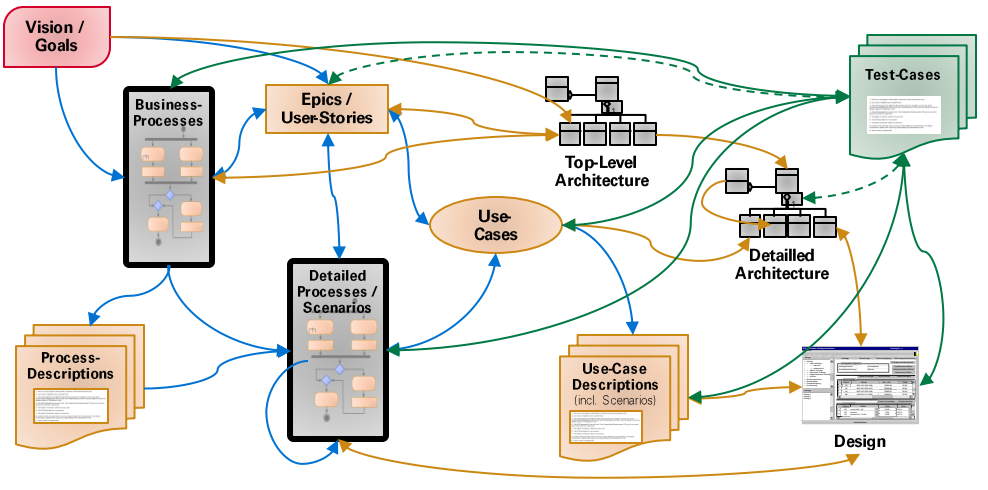
RE-Artefakte und Zusammenhänge



© Software Quality Lab www.software-quality-lab.com

RE im agilen Umfeld

Requirements – Architektur – Design - Test



© Software Quality Lab www.software-quality-lab.com

software quality lab
Innovation Meets Quality

Nicht-funktionale Anforderungen (NFA)

Kurzüberblick

© Software Quality Lab www.software-quality-lab.com

software quality lab
Innovation Meets Quality

Arten von Anforderungen

Definition

Funktionale Anforderung

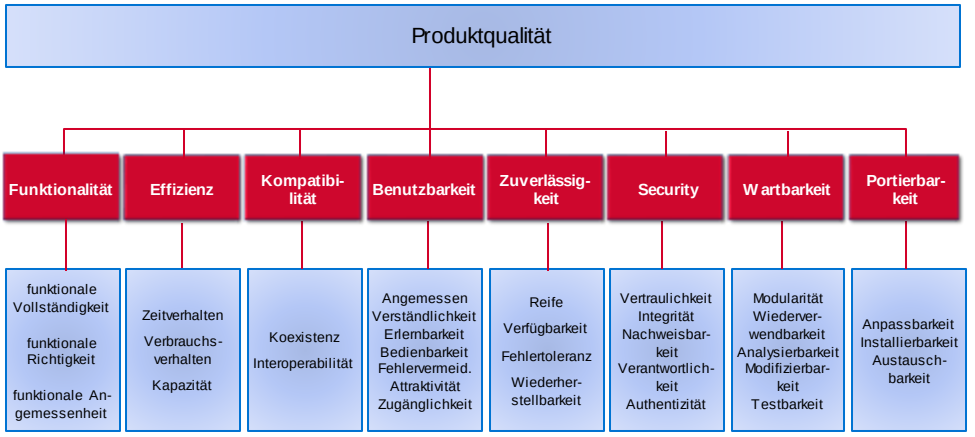
- definiert eine vom System oder von einer Systemkomponente **bereitzustellende Funktion** des betrachteten Systems.
- Typischerweise kann der Benutzer das Ergebnis direkt beobachten, wenn das System in Betrieb ist.

NICHT-funktionale Anforderung (NFA, Qualitätsanforderung)

- definiert eine **qualitative Eigenschaft**, die das System oder einzelne Funktionen des betrachteten Systems aufweisen sollen.
- ist eine Beschreibung einer Beschaffenheit oder einer Charakteristik, die ein System aufweisen muss, die meist den Freiheitsgrad bei der Konstruktion des Systems (also die Umsetzung der funktionalen Anforderungen) einschränkt.

© Software Quality Lab www.software-quality-lab.com

Software-Qualität nach ISO 25010



© Software Quality Lab www.software-quality-lab.com

Beispiele für Qualitätsanforderungen nach ISO 25010

Usability / Benutzbarkeit (1/2)

- **Usability** beschreibt Kriterien, die für die Benutzung und die individuelle Beurteilung der Benutzung durch eine bestimmte Benutzergruppe erforderlich sind.

- **Understandability / Verständlichkeit**
  - Möglichkeiten des Benutzers, das System-Konzept zu verstehen
  - Möglichkeiten zu erkennen, ob das System für seine Anforderungen passt und wie er mit diesem System umgehen soll.
- **Ease of Learning /Erlernbarkeit**
  - Aspekte für den Benutzer, um das System zu erlernen
  - z.B. Eingabe/Ausgabe, Bedienung, ...
  - Diese Anforderung sollen den Designern zeigen, wie die Anwender das System erlernen.

© Software Quality Lab www.software-quality-lab.com

Beispiele für Qualitätsanforderungen nach ISO 25010

Usability / Benutzbarkeit (2/2)

- **Operability / Bedienbarkeit**
  - Anforderungen an das Handling / die Steuerung der Anwendung.
  - Berücksichtigung ergonomischer Aspekte.
- **User Error Protection (Fehlervermeidung)**
  - Wie verhindert das System, dass der Benutzer Fehler macht?
- **Attractiveness (Look&Feel)**
  - Aspekte, welche die Attraktivität für den Benutzer steigern.
  - spezifizieren Vorstellung des Kunden vom Erscheinungsbild des Systems
- **Accessibility / Zugänglichkeit**
  - Aspekte, die die Möglichkeiten für den Benutzer beschreiben, um einfachen Zugang zum System zu erhalten (z.B. Benutzer mit Einschränkungen).

© Software Quality Lab www.software-quality-lab.com

Beispiele für Qualitätsanforderungen nach ISO 25010

Effizienz des Systems

- **Effizienz** ist die Fähigkeit eines Produkts, unter festgelegten Bedingungen eine angemessene Leistung zu erbringen, bezogen auf den Umfang eingesetzter Betriebsmittel.

- **Zeitverhalten**
  - Antwort-/Verarbeitungszeiten sowie Durchsatz bei Funktionsausführung.
- **Verbrauchsverhalten**
  - Anzahl und Dauer der benötigten Betriebsmittel bei der Erfüllung der Funktionen. Ressourcenverbrauch, wie CPU-Zeit, Festplattenzugriffe usw.
- **Kapazitätsgrenzen**
  - Gibt es Kapazitätsgrenzen und passen diese zu den Anforderungen?

© Software Quality Lab www.software-quality-lab.com

Innovation Meets Quality

Nichtfunktionale Anforderungen

Auswirkungen

- Meist große Auswirkungen auf innere und äußere Struktur eines Systems.

→ Erfahrenen Architekten einbinden!

→ Ev. auch UX/Usability-Experten für Benutzbarkeit und GUI-Definition.

- Abnahmerelevanz beachten (wie testet man das?)

© Software Quality Lab www.software-quality-lab.com

Innovation Meets Quality

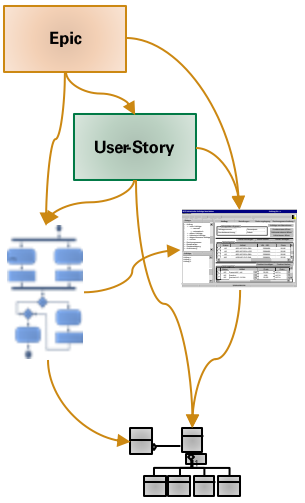
Spezifikation von NFAs

als User-Stories

© Software Quality Lab www.software-quality-lab.com

Nichtfunktionale Anforderungen

Spezifikations-Ebenen / Auswirkungen



- Direkte NFA Spezifikation als
  - Epic
  - User-Story
- Primäre Auswirkungen auf
  - User-Interface
  - Prozesse
  - Use-Cases / Use-Case Szenarien
- Sekundäre Auswirkungen auf
  - Features
  - Architektur
  - Technisches Design

© Software Quality Lab www.software-quality-lab.com

Nichtfunktionale Anforderungen

NFA als Epic

„Das Zeiterfassungssystem soll eine **tolle Usability** haben."

„Das Zeiterfassungssy… …ne **schnelle Reaktionszeit** hab…

„Das Zeiterfass… …gssystem soll **gut wartbar** sein."

Darf in der Spezifikation NICHT so stehen bleiben!

- Sehr grobe Spezifikationsebene → primär „**Merkliste**"
- Für **Umsetzung** und **Tests** noch **NICHT geeignet**!

© Software Quality Lab www.software-quality-lab.com

Beispiel NFA als Story/User Constraint

NFA EFF.1

User-Constraint:

Effizienz - Zeitverhalten:

„Als Benutzer möchte ich, dass spätestens 5 Sekunden nach Abschließen einer Maske eine Reaktion des Systems erfolgt."

Akzeptanzkriterien:

- Test aller Masken mit einer typischen Verwendungshäufigkeit von mehr als 1x pro Tag auf dieses Kriterium.

Test-Umgebung: Testsystem X

→ Hinweis: Achtung auf Universalquantoren bei NFAs: Hier die Frage stellen: „Wirklich für *alle* Masken?"

© Software Quality Lab www.software-quality-lab.com

Beispiel NFA als Story/User Constraint

NFA EFF.2

User-Constraint:

Effizienz - Zeitverhalten:

„Als Controller möchte ich, dass die monatliche Verarbeitung von 1000 eingegebenen Projektzeitdatensätzen innerhalb von max. 10 Minuten erfolgen muss."

Akzeptanzkriterien:

- Monatliche Datenverarbeitung am Controller-Client für 10 Mitarbeiter mit je 20 Tageseinträgen und je Tag 5 Projektbuchungen.
- Dauer je Durchführung < 10 Minuten

Test-Umgebung: Testsystem X

→ Hinweis: Zeitangaben nicht ohne Umgebungsdefinition

© Software Quality Lab www.software-quality-lab.com

software quality lab
Innovation Meets Quality

Test(Spezifikation) von NFA

Ein Beispiel

© Software Quality Lab www.software-quality-lab.com

software quality lab
Innovation Meets Quality

Nichtfunktionale Anforderungen

Detail-Angaben

- **User Constraint Ebene** ist oft noch zu **ungenau**. Bei NFAs sind oft genauere Angaben notwendig (wegen Vertrag, Abnahme):
  - Unter welchen Voraussetzungen wird hier gemessen?
  - Welche Hardware, welche Software, etc. liegen zugrunde?
- Spätestens **bei den Abnahmekriterien immer Messumgebung** angeben:
  - Genaue verwendete Hardware
  - Genaue verwendete Software (Version, evtl. auch Patches, etc.) sowohl für die getestete Software als auch für Betriebssystem und sonstige auf dem System laufende Software
  - Angaben über die zugrundeliegenden Testdaten (z.B. Datensätze in der DB)
  - Betriebsdaten wie Auslastung der Prozessoren, Netzwerk, Speicher, etc.
  - usw.

© Software Quality Lab www.software-quality-lab.com

software quality lab
Innovation Meets Quality

Nichtfunktionale Anforderungen

Testbarkeit von NFA's

Oft schwer zu testen:

- Usability → subjektiv!
  - Wie testet man subjektive Eindrücke?
  - → Statistisch mitteln. Tester „nutzen ich ab"!
- Performance → von Details der Umgebung abhängig
  - Läuft grad ein Hintergrundprozess?
  - Wie ist der Netzwerk-Verkehr?
  - → Abgeschlossene klar definierte Umgebung
- Skalierbarkeit → Großes Umfeld
  - 10000 User sind nicht so einfach zu simulieren
  - → Passende Szenarien / Umgebung auswählen

© Software Quality Lab www.software-quality-lab.com

software quality lab
Innovation Meets Quality

Bespiel: Usability Test

Erlernbarkeit

- Aufwand für den Benutzer, die Anwendung zu erlernen (z.B. Bedienung, Ein-, Ausgabe).

1. Genaue (eigene) Definition des Begriffs „Erlernbarkeit"
2. „Neue User" lernen lassen (einmal erlernt → unbrauchbar)
3. Prüfung mit den Usern durchführen (Wissen abfragen)

→ User kennen das System und sind daher „verbraucht"
→ Neue User für neuen Test nötig – gleiches Verhalten?
→ Breite statist. Basis notwendig, ev. zeit-/kostenintensiv!

© Software Quality Lab www.software-quality-lab.com

Usability Test – Vielfältige Aspekte

Die Sache schaut auf den ersten Blick einfach aus:

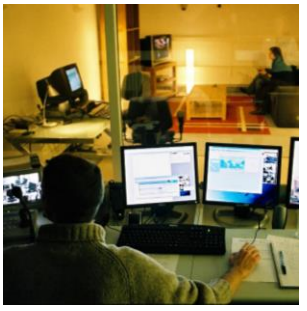
→ Wir setzen ein paar User hin und schauen, was sie uns zurück melden ….

Doch es ist komplexer als man denkt (wenn man es systematisch machen möchte):

- Usability Inspektion (**Experten**zentrierte Evaluation)
  - Heuristische Evaluation
  - Cognitive Walkthrough
- Usability Test (**Benutzer**zentrierte Evaluation)
  - Thinking Aloud
  - Question Asking (Protocol)
  - Interview
  - Fragebögen
  - Online-Fokusgruppe
  - Eye-Tracking

© Software Quality Lab www.software-quality-lab.com

Usability Test – die breite Praxis



Praxis zeigt:

- 1 qualifizierte Testperson findet ca. 30% aller Usability-Probleme
- 3 - 5 Tester finden 60 - 80% der Usability-Schwachstellen

→ Für die „normalen" Usability-Probleme ist der Aufwand überschaubar

→ Für High-End-Ausführungen steigt der Aufwand sehr stark an!

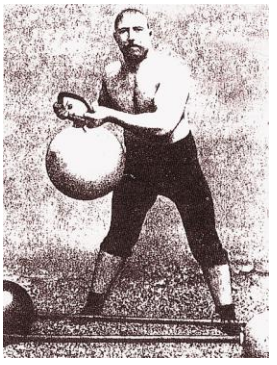
© Software Quality Lab www.software-quality-lab.com

Innovation Meets Quality

Resümee

Innovation Meets Quality

Resümee

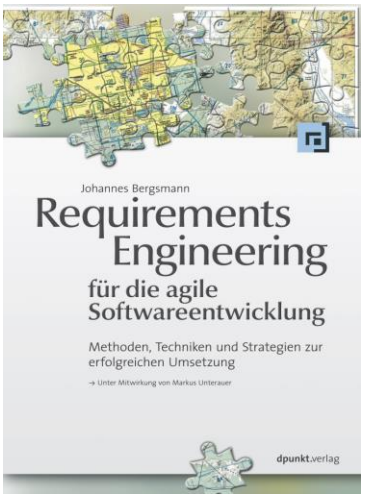


- NFA nicht vergessen oder vernachlässigen
- Nicht nur „oberflächlich“ spezifizieren (Abnahmerelevanz)
- Tests gut überlegen (Rahmenbedingungen, Detailaspekte)
- NFAs sind teilweise sehr aufwändig zu testen (siehe Beispiel Usability)

RE in agilen Vorgehensweisen

Das Buch zum Thema

- Requirements-Engineering für die agile Softwareentwicklung
- dpunkt.verlag
- Verfügbar im Buchhandel und online



© Software Quality Lab www.software-quality-lab.com

Software Quality Lab

INNOVATION MEETS QUALITY

Academy | Consulting | Operational Services | Tool Expertise

© Software Quality Lab www.software-quality-lab.com