iteratec

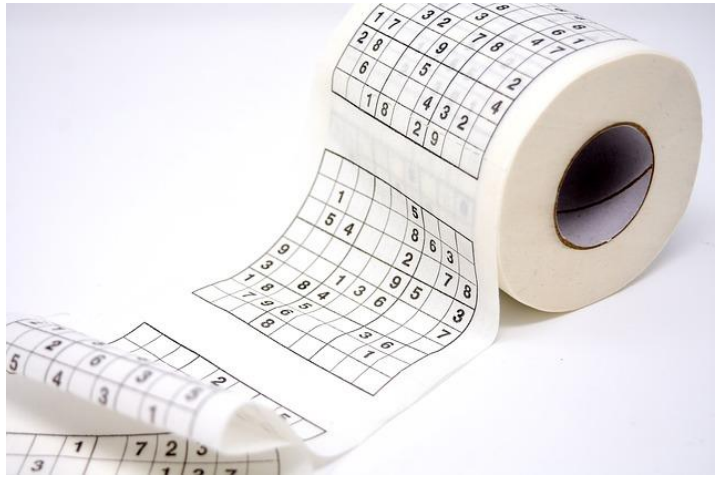
KOMPETENZ, DIE ENTLASTET

Test-Inventare - Ein leichtgewichtiger Ansatz für die Qualitätssicherung

Johannes Merkert

26.04.2019

Lasst uns mal über Toiletten reden



- Video Source: https://youtu.be/RdTNcTcoDGI

Akzeptanzkriterien:

- Alte Toilette raus
- Neue rein
- Toilette stabil und Sitzfläche eben
- Zufluss und Ablauf korrekt angeschlossen

Was tun?

- Begehung im Voraus
- Checklisten

„Versteckte Akzeptanzkriterien“?

- Tür geht noch auf und zu
- Die Bodenkacheln sind ok / noch da / schließen an der neuen Toilette
- Licht Schalter ist noch da / funktioniert noch?
- Lampe ist noch da / funktioniert noch?
- ???

Hmmm, was tun?

Warum?

- Risikoerkennung und -vermeidung
- Erkennung des Kontextes
- Erkennung von Problemen mit Akzeptanzkriterien
- Planung der Qualitätssicherung

Ansatz - Test Inventar

- „Einkaufsliste“ für Szenarien und Erwartungen
- Ziemlich formlos

Ziel?

- Nicht: „fehlerfrei“
- Sondern: „besser“

FAQs

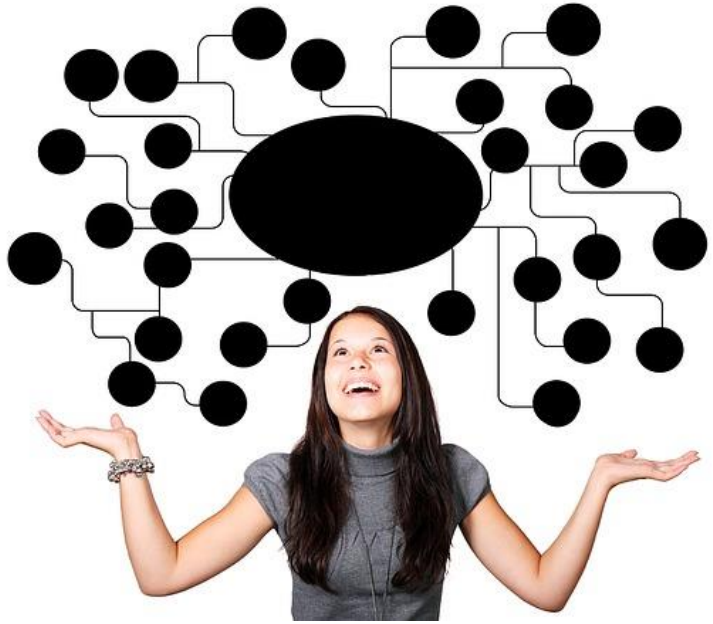
- Können wir nicht einfach die Akzeptanzkriterien übernehmen?
- Kann ich, bevor ich anfange, das überhaupt wissen?
- Das ist nicht Code-level, ist das nicht irrelevant?
- Ist das nicht Overhead?
- Kein Frontend, kein User, kein Testinventar?

Test Inventar Erstellung

Vorgehen

Fragen:

- Was sind Inputs, was sind Outputs?
  - (Haben wir besondere Fehlerfälle?)
- Haben wir Zustände (States)?
- Haben wir Rollen?
- Was sind potentiell betroffene Features?
- Gibt es Upstream Themen?
- Gibt es Downstream Themen?
- Gibt es umgebungsspezifische Besonderheiten?



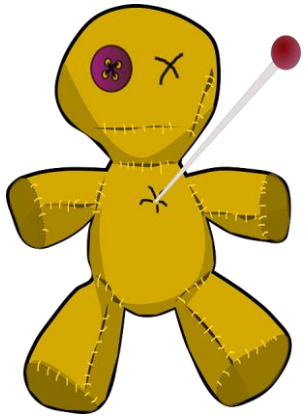
Vorgehen

- Fragen beantworten
- Vorstellen wie man den Soll-Zustand durchspielt
- Probleme aufnehmen
- Szenarien sammeln

Test Inventar-Erstellung

Wie kommt man von den Antworten zum Inventar?

Fragen



Antworten und Voodoo?

Test Inventar

Test Inventar Erstellung – Beispiel 1

Webshop um Bestellhistorie erweitern

Story

„Als Besteller möchte ich über das Profil oder direkt über die URL die Bestellhistorie aufrufen können, um dort meine vergangenen Bestellungen anzusehen.“

Akzeptanzkriterien

1. Über einen Klick auf „Bestellhistorie“ im Profil öffnet sich die Bestellhistorie
2. Ein Aufruf von /orders/history öffnet die Bestellhistorie für den eingeloggten User
3. Gibt es keine Bestellungen in der Historie, wird eine leere Tabelle angezeigt
4. Stornierte Bestellungen werden als storniert angezeigt

Test Inventar Erstellung – Beispiel 1

Webshop um Bestellhistorie erweitern

Antworten

- Was sind Inputs, was sind Outputs?
  - Input: Inhalt der Historie (leer, Bestellungen, stornierte Bestellungen)
  - Output:
    - Anzeige der Seite
    - http Status
- Haben wir Zustände (States)?
  - Zugang über Profil: User ist da bereits eingeloggt und Anwendung initialisiert
  - Zugang direkt über Historien-URI

„Szenarien“ (Test-Inventar)

- Historie mit Inhalten aus Profil aufrufen → Historie mit Inhalten für den User wird aufgerufen (http 200)
- Historie mit Inhalten per URI aufrufen (mit valider aktiver Session) → Historie mit Inhalten für den User wird aufgerufen (http 200)
- Historie ohne Inhalte aus Profil aufrufen → Historie mit leerer Tabelle wird aufgerufen (http 200)
- Historie mit Inhalten per URI aufrufen (mit valider aktiver Session) → Historie mit leerer Tabelle wird aufgerufen (http 200)
- Historie ohne Session aufrufen → Login Abfrage kommt, bei Abbruch http 401
- Historie mit stornierten Bestellungen anzeigen → werden korrekt gekennzeichnet

Test Inventar Erstellung – Beispiel 2

Backend Umstellung: Statt Fehlertext Fehlerobjekt mit „code“ und „text“

Story

„Als Service-Verantwortlicher möchte ich, dass das Backend bei allen Fehlerantworten auf Anfragen einen eindeutigen Fehlercode zusammen mit einem Fehlertext übergibt, damit eine Fehlerantwort besser nachvollzogen werden kann. “

Akzeptanzkriterien

1. Für alle APIs vom Typ GET/POST/PUT/DELETE sollen die Antworttexte im Fehlerfall gegen ein Fehlerobjekt mit Fehlertext „text“ und eindeutigem Fehlercode „code“ ersetzt werden

Test Inventar Erstellung – Beispiel 2

Backend Umstellung: Statt Fehlertext Fehlerobjekt mit „code“ und „text“

Antworten

- Was sind Inputs, was sind Outputs?
  - Input: Fehlercode, Fehlertext
  - Output: Objekt {code: number, text: string}
- Betroffene Features?
  - Alle eingehenden Schnittstellen
- Upstream?
  - Alle Antworten vom Typ Fehler in Services/Resourcen brauchen eindeutige Fehlercodes
- Downstream?
  - Fehleranzeige im Frontend muss angepasst werden => Nachricht muss nun aus Objekt ausgelesen werden

„Szenarien“ (Test-Inventar)

- Für jeden Fehlercode: Fehler auslösen → Korrektes Fehlerobjekt wird als Response ausgegeben
- Für jede Schnittstelle: Regressionstest über alle nicht-Fehlerobjekt antworten
- Im Frontend: für jede Schnittstelle x jeden Fehlercode x jede Sprache prüfen, dass die Fehlermeldung korrekt dargestellt wird
- In der Review: prüfen, dass alle Fehlercodes eindeutig sind

Test Inventar Erstellung – Beispiel 3

Geringfügige Erweiterung einer UI bei unterschiedlichen Clients

Story

„Als Benutzer möchte ich auf Knopfdruck alle Datumsfelder im Formula auf den ersten des aktuellen Monats setzen können, da ich ansonsten, wenn ich das Formular wegen eines Feiertags (z.b. 01.01. oder 01.05.), erst später öffne, alle Datumsfelder händisch ändern muss. “

Akzeptanzkriterien

1. Im Header ist ganz rechts ein zusätzlicher Button „Datum auf Anfang des Montags setzen“
2. Klick auf den Button überschreibt alle Datumsfelder mit dem Datum des ersten des aktuellen Monats
3. Das Formular enthält weiterhin beim Öffnen alle Datumsfelder vorbefüllt mit dem aktuellen Datum

Hinweis

Bei der Anwendung handelt es sich um eine Webanwendung für PC, Tablet und Mobile (hochkant)

Test Inventar Erstellung – Beispiel 3

Geringfügige Erweiterung einer UI bei unterschiedlichen Clients

Antworten

- Was sind Inputs, was sind Outputs?
  - Input: aktuelles Datum, Buttonclick
  - Output: Inhalt aller Datumsfelder
- Umgebungsspezifische Besonderheiten?
  - 3 Client Typen, dabei Mobile hochkant mit besonders schmaler Breite

„Szenarien“ (Test-Inventar)

- Initiales Laden das Formulars --> alle Datumsfelder sind mit aktuellem Datum vorbefüllt, Button wird im Header angezeigt
- Button anklicken (aktuelles Datum NICHT Anfang des Monats) --> im Formular werden alle Datumsfelder auf Anfang des Monats gesetzt
- Button anklicken (aktuelles Datum entspricht Anfang des Monats) --> im Formular sind alle Datumsfelder Anfang des Monats
- Nach klick auf Button Datumsfelder ändern, dann wieder Button klicken --> im Formular sind alle Datumsfelder Anfang des Monats
- Formular im Standard-Browser --> Header wird mit neuem Button anständig formatiert angezeigt
- Formular im Tablet --> Header anständig formatiert angezeigt
- Formular in Mobile --> Header anständig formatiert angezeigt

Overkill - Formelle Mechanismen zur Findung von Testfällen

- Formlos bleiben
- Schlicht halten
- Übersichtliche Darstellung
  - Liste
  - Mindmap
- Auffindbarkeit der Testinventare
  - Als Kommentar dem Ticket anhängen
  - Der Dokumentation beifügen
- Sammlung von Antworten vollständig verarbeiten
- Test Inventare mit Bugs vergleichen und ggf. Bugs aufnehmen

➢ Pragmatische Herangehensweise

Was kann man mit einem Test Inventar machen?

Anwendungsbeispiele

- Präzisierung unklarer Anforderungen / AKs
  - Aufdecken von Spezifikationslücken
  - Aufdecken und Widersprüchen
- Als Checkliste für technisches Review
- Als Vorlage für automatisierte Tests
- Als Checkliste für manuelle Tests
- Für die Validierung der Ergebnisse mit dem Kunden

iteratec

KOMPETENZ, DIE ENTLASTET

Danke für eure Aufmerksamkeit

www.iteratec.de