Web-Engineering

Test von Web-Applikation / Web-Content

Hochschule Niederrhein
University of Applied Sciences
Elektrotechnik und Informatik
Faculty of Engineering and Computer Science

Literatur

- Franz, Klaus: Handbuch zum Testen von Web-Applikationen, Springer Xpert.press, 2007

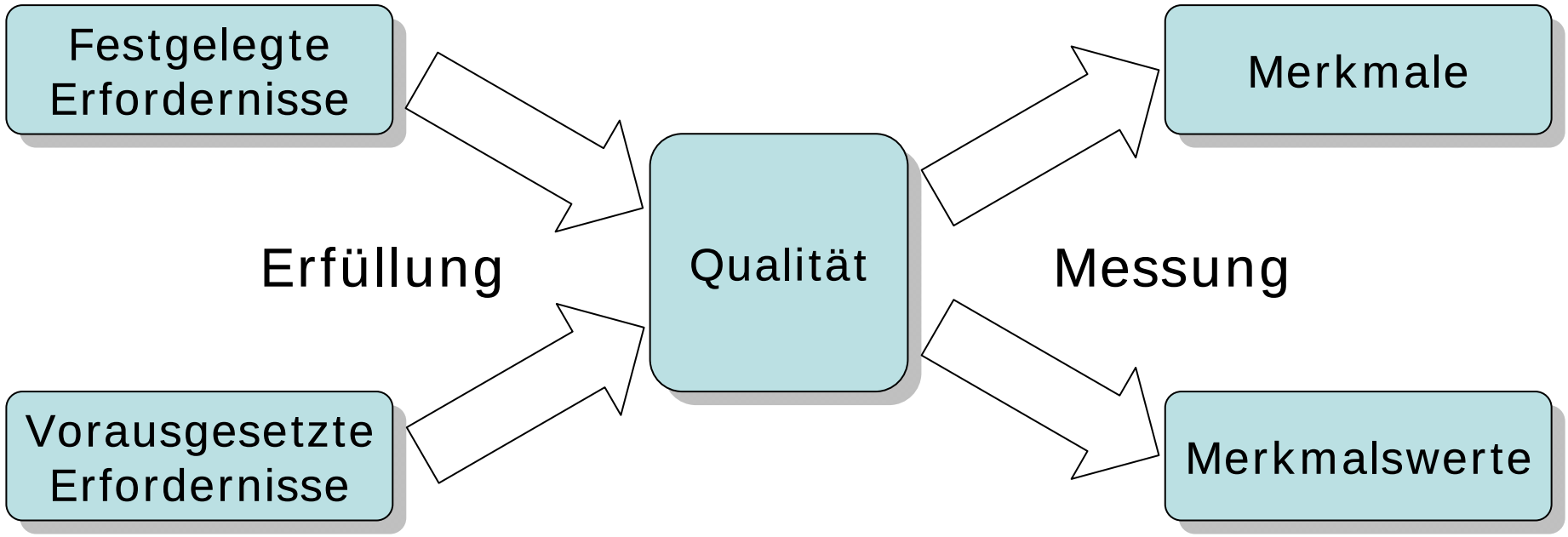
Der Unverbesserliche

Wer testet, ist feige !

Was ist ein Test (nicht) ?

- Nicht:
  - 'läuft schon ... irgendwie'
  - Einzelne Effekte beobachten
  - Mit einem Debugger exemplarisch einige Programmzeilen überprüfen
  - Herumprobieren
- Sondern:
  - **Messen**, in welchem Umfang vorgegebene Qualitätskriterien erfüllt werden
  - Somit: **Soll-Ist-Vergleich**

Was ist Qualität ?



Begriffe (1)

- Fehler: Nichterfüllung einer festgelegten Forderung
- Testobjekt: wird einem Test unterzogen
- Statischer Test: Prüfung Testobjekt ohne Rechnerunterstützung (insbesondere: ohne Ausführung)
- Dynamischer Test: Prüfung Testobjekt mit Rechnerunterstützung (insbesondere: mit Ausführung)

Begriffe (2)

- Testfall: Anweisung zur Durchführung eines Tests
  - Notwendige Vorbedingungen definieren / herstellen
  - Eingabedaten ("Testdaten")
  - Notwendige Eingabeaktionen (*)
  - Erwartete Ergebnisse – Werte wie Systemzustände
  - Aktionen, die zur Überprüfung der Ergebnisse notwendig sind (*)
  - (*) sind Bestandteil des Testrahmens / Testtreibers
- Testspezifikation: (vereinfacht) Beschreibung und Begründung der durchzuführenden Testfälle

Begriffe (3)

- Überdeckungsgrad : Maß für die Vollständigkeit eines Tests
- Regressionstest : Wiederholung von Tests nach Änderungen

- *Blackbox-Test*: kein Einblick in die Interna des Testobjekts
- *Whitebox-Test*: innere Struktur Testobjekt prüfen, d.h. alle Interna sind bekannt

Funktionalität testen (1)

- Qualitätsmerkmal Funktionalität überprüfen:
  - Web-Anwendung realisiert geforderte Funktionen
    - Angemessen
    - Vollständig
    - Korrekt
  - Komponenten fehlerfrei integriert zu Gesamtsystem
  - Sicherheit der Daten gewährleistet

Funktionalität testen (2)

- "Klassische" funktionale Tests:
  - Klassentest:
    - Einzeltest
    - Integrationstest
    - Z.B. mit Überdeckungstests (Zweig-, Pfad-Überdeckung)
  - Komponententest (Modultest):
    - Nachweisen, dass die Komponente der Spezifikation entspricht
    - Überprüfen der Schnittstellen
    - Überprüfen der Zustände der Komponente

Funktionalität testen (3)

- "Klassische" funktionale Tests (Forts.):
  - Integrationstest:
    - Funktionales und technisches Zusammenwirken von HW- und SW-Komponenten prüfen
    - Fehlerwirkungen in Schnittstellen und Kommunikation aufdecken
  - Integrationsstrategien:
    - Top-Down, Bottom-Up: bei Web-Anwendungen ungeeignet
    - "Big Bang": Alles direkt zusammentesten: NIE !
    - Anwendungsfallorientiert vorgehen !

Funktionalität testen (4)

- "Klassische" funktionale Tests (Forts.):
  - Funktionaler Systemtest:
    - Gesamtsystem in Bezug auf die funktionalen Anforderungen testen
- Web-spezifische Tests:
  - Link-Test:
    - Fehlerfreiheit, Rechtmäßigkeit von internen und externen Links
    - Checklisten verwenden
    - Link-Checker, z.B. vom W3C, einsetzen

Funktionalität testen (5)

- Web-spezifische Tests (Forts.):
  - Cookie-Test
    - Fachliche, technische Funktionalität eingesetzter Cookies prüfen
    - (hier sind Ergänzungen notwendig: WebStorage etc.)
  - Plugin-Test
    - Fehlerfreie und korrekte Benutzerführung im Hinblick auf benötigte Plugins prüfen
  - Sicherheitstest:
    - Prüft die Eignung, Korrektheit, Unumgänglichkeit und Wirksamkeit der eingesetzten Sicherheitsmaßnahmen

Benutzbarkeit testen (1)

- Content-Test:
  - Erfüllung von Benutzererwartungen
  - Rechtskonformität
  - Erfüllung von Aufklärungspflichten, z.B.
    - Anbieterkennzeichnung
    - Nennung von Autoren, Vertretungsberechtigten
    - Gesetzliche Anforderungen
- Oberflächentest:
  - Einhaltung Dialogrichtlinien, Korrektheit von Standardfunktionen überprüfen

Benutzbarkeit testen (2)

- Browser-Test
- Usability-Test:
  - Gebrauchstauglichkeit überprüfen
  - Unter Beachtung der jeweiligen Zielgruppe !
- Zugänglichkeitstest: (Barrierefreiheit testen)
- Auffindbarkeitstest:
  - Direkter Aufruf per URI möglich?
  - Problemloses Auffinden in Suchmaschine

Effizienz und Zuverlässigkeit testen (1)

- 'Performance' testen:
  - Werden geforderten Funktionen im Rahmen geforderter Bedingungen erbracht?
    - Zeitverhalten
    - Mengenverarbeitung
    - Ressourcenverbrauch

    bei normalen Systembedingungen.
- Lasttest:
  - Verhalten bei steigender Systemlast

Effizienz und Zuverlässigkeit testen (2)

- Skalierbarkeit testen
- Auf Speicherlecks hin prüfen
- Ausfallsicherheit testen:
  - "Failover" prüfen: wird problemlos auf eine vorhandene Sekundärkomponente umgeschaltet?
- Verfügbarkeitstest:
  - Verfügbarkeit / Ausfallrate überprüfen

Werkzeuge (Beispiele)

- Treiber und Platzhalter
- Capture Replay Tools:
  - Aufzeichnung von Tests
  - Wiederholung und damit Automatisierung von Tests
- Code Coverage Tools:
  - Überdeckungen bestimmen
- Monitore:
  - Schnittstellen beobachten
- Logging-Tools:
  - Protokolle zu Abläufen erstellen und auswerten
- Automatisierung der Benutzerinteraktionen:
  - z.B. Selenium IDE / Selenium WebDriver

Der Einsichtige

Wer testet, ist mutig !