

Woche 08

ISW-Tutorium

Xel Pratscher

Orga

Weihnachtsmarkt

- Montag, 18.12.
- 18 Uhr Treffpunkt Sparkasse

Klassendiagramme

- **Klausurrelevant**
- Nur Assoziationsbeziehungen werden abgefragt
- Bei Unklarheiten an Frau Prof Paech wenden
 - paech@informatik.uni-heidelberg.de

Vorlesung

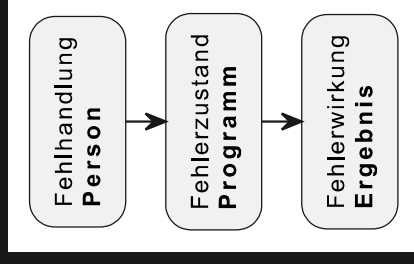
- Wichtige Begriffe Qualität
- Organisatorisches Testen
- Testen (Dynamische Codeprüfung)

Wichtige Begriffe

Fehler vs. Mangel

- **Fehler:**
Ist \neq Soll
- **Mangel:**
gestellte Anforderung nicht genug erfüllt

Weitere Fehlerbegriffe



Fehlhandlung

- menschliche Handlung führt zu falschem Ergebnis
- **Developers:** Fehlerzustände in Software
- **Users:** Fehlbedienung mit unerwünschtem Ergebnis
- Führt ggf. zu Beeinträchtigung der Softwarefunktion

Fehlerzustand

- **Mangel** in Arbeitsergebnis, sodass Anforderungen / Spezifikationen nicht erfüllt werden
- Inkorrektes Teilprogramm

Fehlerwirkung

- sichtbares Auftreten von Fehler
- Ereignis, in welchem Komponente / System geforderte Funktion nicht wie gewünscht ausführen
- Abweichung zw. erwartetem Ergebnis und tatsächlichem Ergebnis

Qualitätsmanagement vs -sicherung

- **Qualitätsmanagement**
etablierter Prozess zum Umgang mit Qualität
- **Qualitätssicherung**
konkretes Vorgehen zur Sicherstellung der Qualität

Organisatorische vs Konstruktive vs Analytische QS

- **Organisatorische QS**
Qualität durch geeignete Planung / Orga sicherstellen
- **Konstruktive QS**
Qualität durch systematische Entwicklung sicherstellen
- **Analytische QS**
Qualität durch Anforderungsspezifikationen sicherstellen

Organisatorisches Testen

- Testen ist **Ausführung eines Programmes** mit dem Ziel **Fehlerwirkungen** zu finden

Testbegriffe I

- **Testbasis**
 - Dokumente mit Anforderungen an Komp/Sys
- **Testfall**
 - Menge von Eingabewerten, Vor-/Randbed., erwartete Ergebnisse + Nachbed.
- **Vorbedingung**
 - Zustand Testobjekt um Testfall / Testsuite ablaufen zu lassen

Testbegriffe II

- **Nachbedingung**
 - Zustand Testobjekt nach Ausführung von Testfall / Testsuite
- **Testlauf**
 - Ausführung eines / mehrerer Testfälle / Testsuiten mit best. Version des Testobjekts
- **IST-Verhalten**
 - Folge erzeugter Ist-Ausgaben des Testobjekts
 - Mit Soll-Reaktion abgleichbar

Testbegriffe III

- **Testspezifikation**
 - Festlegung Testobjekte und Testfälle
 - Testendekriterien (wie lange testen)
- **Testskripterstellung**
 - Definition konkreter Testfälle und -daten
 - Implementierung des Testskripts (Vorbed., Testschritte, Set-up, Tear-down Routinen, Testdatengenerierung, Testorakelermittlung)

Umgang mit Fehlern

- **Fehlermanagement**
 - Nachvollziehbarkeit / Finden von Fehlern
 - Beseitigen von Fehlern
 - Übermittlung und Verwaltung von Fehlermeldungen
 - zentrale *Fehlerdatenbank*
 - einheitliches Schema
- **Debugging**
 - Wie systematisch von Fehlerwirkung zum Fehlerzustand

Schema zur Fehlermeldung

- Nachverfolgbarkeit / Finden
 - EntdeckerIn, Datum, Betroffenes Produkt, ausgeführte QS-Aktivität, Beschreibung
- Einordnung zur weiteren Behandlung / Beseitigung
 - Fehlerklasse, Fehlerpriorität
 - Status
 - Änderungshistorie
- Verbesserung des Entwicklungsprozesses
 - Schlussfolgerungen
 - Globale Probleme

Issue Tracking

- Verwaltung von Fehlermeldungen und neuen Features
- Verteilte Erfassung von Issues
- Verteilung von Arbeit und Verfolgung
- JIRA heute weit verbreitet

Debugging

- Suche nach Fehlerzustand
- **Systematisches Vorgehen**
 - Reproduziere Fehlerwirkung
 - Suche Fehlerzustand
 - Verbessere Fehlerzustand

Testen (Dynamische Codeprüfung)

Teststufen

Vor- und Nachteile von Testen

Vorteile

sichtbares
Systemverhalten

reproduzierbar

wiederverwendbar

Prüfung auch der
Testumgebung

Nachteile

Testen nicht
ausreichend

aufwändig

Testfallspezifikation

- Legt das Soll-Verhalten fest
- 2 Abstraktionsebenen
 - Logische Testfälle
 - Konkrete Testfälle
- Testfallbeschreibung
 - Name, Getestete Anforderung, Typ, Vor- / Nachbed., Testinfrastruktur, Testschritte

Positiv-Testfall vs Negativ Testfall

- **Positiv-Testfall:**
 - korrekte Eingaben
 - erwarte korrektes Ergebnis
- **Negativ-Testfall**
 - unzulässige Eingabe
 - erwarte sinnvolle Ausnahmebehandlung

Wir brauchen beides

Black-Box vs White-Box

Black-Box

Außwirkung des
Testobjekts testen

Keine Steuerung des
Ablaufs

Nutzen Kenntnisse über
Schnittstelle,
Spezifikation

White-Box

gezieht versch.
Bestandteile / Abläufe
testen

Nutzen Kenntnisse
über inneren Aufbau

Äquivalenzklassentest

- Testfälle decken alle Äquivalenzklassen ab
- Grenzwerte spezifisch testen
- **Typische Äquivalenzklassen**
 - Gültige Eingabe-Äquivalenzklasse
(Zulässige Datenbereiche)
 - Ungültige Eingabe-Äquivalenzklasse
(unzulässige Datenbereiche)
 - Gültige Ausgabe-Äquivalenzklasse
(Unterteilung zulässiger Eingaben)
 - Ungültige Ausgabe-Äquivalenzklasse
(Ausnahmen)

Übungsblatt 09

9.1 (Testat, Team)

Calculate Overall Rating betrachten

1. Äquivalenzklassen angeben (12)
2. Testfälle skizzieren (6)
3. Minimale Testfallmenge bilden
4. 4 Testfälle spezifizieren (min. 1 (R2))

9.2 (Testat, Einzeln)

- Testfälle implementieren
- Testfälle konkretisieren

9.3 (Kein Testat, Einzeln)

- Weiter Thema Inspektion vorbereiten

9.4 (Testat, Team)

- Funktionale Anforderung dokumentieren
- User Task
- Domänendatenmodell
- Systemfunktion
- Veränderte Systemfunktionen
- UI-Strukturdiagramm

9.5 (Testat, Einzeln)

- Wiederholungsaufgaben überprüfen und verbessern

9.6 (Probeklausur, Kein Testat, Einzeln)

- **30.01.** Technologievorlesung Nachbesprechung
- Wenn Feedback gewünscht -> Auf Moodle hochladen
 - Deadline **23.01.**

Danke fürs Zuhören
und Mitmachen :)