

TU Wien:Software-Qualitätssicherung VU (Biffl)/Prüfung 2013-12-19

< TU Wien:Software-Qualitätssicherung VU (Biffl)

Contents

Qualitätsmanagement VS Qualitätssicherung, Prinzipien des Qualitätsmanagements nennen.

Integration der Qualitätssicherung (Verifikation und Validierung) im V-Modell (Skizze war dabei) einzuschreiben.

Ablauf eines Review beschreiben (Skizze + Stufen kurz erklären) Vorteile eine Reviews gegen den manuellen Test nennen.

Black-Box Test vs. White-Box Test. Ist es möglich den White Box Test mit JUnit Framework durchzuführen. Wie? Begründung!

Kontrollflussgraph war gegeben. Testfälle bezüglich 100% Anweisungsüberdeckung und 100% Bedingungsüberdeckung waren gefragt.

Continous Integration Workflow erklären, Einfluss des CI-Systems an die Qualitätssicherung beschreiben.

Teststufen (Modul, Integration, Lasst, Regression, System, Akzeptance) mit der richtigen Fällen (Erklärungen) verbinden.

Testdoubles nennen und kurz beschreiben. Wie führt man die Isolation beim Testen durch?

Qualitätsmanagement VS Qualitätssicherung, Prinzipien des Qualitätsmanagements nennen.

Qualitätsmanagement beschreibt die Gesamtheit aller Maßnahmen und Aktivitäten, die zur Herstellung und Wahrung von Qualität notwendig sind. Dazu zählen unter anderem Qualitätsplanung, -lenkung, -sicherung und insbesondere die Qualitätsverbesserung. Wichtige Prinzipien:

- Kontinuierliche Verbesserung durch Bewertung von Maßnahmen und Prozessen.
- Frühzeitige Entdeckung und Behebung von Fehlern
- Unabhängigkeit bei Qualitätsprüfungen (vgl. QS-Abteilung getrennt von Entwicklung)
- Produkt- und projektabhängige Planung von QS-Maßnahmen

Qualitätssicherung hingegen beschreibt Verfahren, Aktivitäten und Prozesse, welche sicherstellen, dass ein Produkt gegebenen Anforderungen (Spezifikation und Kundenanforderungen) gerecht wird (kurz: Testen, Reviewen, ...).

Integration der Qualitätssicherung (Verifikation und Validierung) im V-Modell (Skizze war dabei) einzuschreiben.

- Aufgabe in Zeichnung zu lösen

Ablauf eines Review beschreiben (Skizze + Stufen kurz erklären) Vorteile eine Reviews gegen den manuellen Test nennen.

1. **Planung:** In der ersten Phase werden die Teilnehmer für das Review ausgewählt und das zu beurteilende Objekt, sowie die Ziele des Reviews festgelegt.
2. **Vorbereitung:** In der zweiten Phase bereiten sich die Teilnehmer des Reviews individuell vor. Dazu behandeln sie den Testling ihrer Rolle entsprechend und notieren sich Auffälligkeiten (Fehler, Fragen und Kommentare).
3. **Review-Sitzung:** In der Hauptsitzung werden die Erkenntnisse dann besprochen und protokolliert.
4. **Nachbearbeitung:** In dieser Phase werden die gefundenen Fehler behoben und zugehörige Dokumente aktualisiert.
5. **Bewertung:** Eine abschließende Bewertungsphase wird dazu genutzt, den Reviewprozess für künftige Reviews zu optimieren.

Black-Box Test vs. White-Box Test. Ist es möglich den White Box Test mit JUnit Framework durchzuführen. Wie? Begründung!

Black-Box Tests: Der Tester hat keine Kenntnis (oder ignoriert seine Kenntnis) über den Aufbau der zu testende Softwarekomponente. Der Test basiert daher rein auf der Spezifikation, sprich welche Ausgabe bei entsprechender Eingabe erwartet wird.

White-Box Tests: Bei dieser Testart hat der Tester Kenntnis über den Aufbau einer Komponente und nutzt dieses Wissen zum Schreiben der Tests aus. Es werden also Code-Teile, wie z.B. Bedienungsanweisungen, mit genau diesem Wissen im Hinterkopf überprüft ("Logik getriebene Tests"). Dadurch können verschiedene Codeabdeckungsarten erreicht werden (Co-C3). Es ist selbstverständlich möglich, mit JUnit White-Box Tests durchzuführen. Dazu schreibt man Tests, die beispielsweise alle Bedingungen oder Kanten in einer Komponente explizit durchlaufen.

Kontrollflussgraph war gegeben. Testfälle bezüglich 100% Anweisungsüberdeckung und 100% Bedingungsüberdeckung waren gefragt.

- Multiple-Choice Frage

Continous Integration Workflow erklären, Einfluss des CI-Systems an die Qualitätssicherung beschreiben.

Kontinuierliche Integration beschreibt ein System, das bei Änderungen (= neue Version im Versionsverwaltungssystem) automatisch Tests durchführt und ein Softwareprodukt zusammenführt. Dadurch wird gewährleistet, dass Fehler früh erkannt und behandelt werden können. Kurzum ist ein CI-System also dafür zuständig, bei Änderungen die Funktionalität einer neuen Version sicherzustellen. Im Idealfall übernimmt das CI-System auch entsprechende Dokumentationsarbeiten (Test-Berichte u.Ä.).

Teststufen (Modul, Integration, Lasst, Regression, System, Akzeptance) mit der richtigen Fällen (Erklärungen) verbinden.

- *Quasi* Multiple-Choice Frage

Testdoubles nennen und kurz beschreiben. Wie führt man die Isolation beim Testen durch?

- **Dummy-Object:** Verfügt über keinerlei Logik, dient lediglich als leerer Methodenparameter.
- **Fake Object:** Ist eine ausführbare Implementierung, die allerdings keine Echtdaten liefert. Wird oft zur zügigeren Testausführung herangezogen.
- **Stub:** Liefert vordefinierte Resultate bei Funktionsaufrufen.
- **Mock:** Ist im Prinzip ein konfigurierbares Stub, das auch entsprechende Logik enthalten kann. Prüft beispielsweise die Eingabeparameter und antwortet entsprechend.

Die Isolation wird durch die Injektion von Testdoubles durchgeführt. Dadurch wird sichergestellt, dass etwaige Fehler nicht durch abhängige (= darunterliegende) Objekte verursacht werden.

Retrieved from "[https://vowi.fsinf.at/wiki?title=TU_Wien:Software-Qualitätssicherung_VU_\(Biffi\)/Prüfung_2013-12-19&oldid=85748](https://vowi.fsinf.at/wiki?title=TU_Wien:Software-Qualitätssicherung_VU_(Biffi)/Prüfung_2013-12-19&oldid=85748)"

This page was last edited on 17 December 2015, at 11:42.

Content is available under GNU Free Documentation License 1.3 unless otherwise noted.