

Übung 8: Softwarequalität und Testen

Aufgabe A: Softwarequalität

Beantworten Sie stichpunktartig folgende Fragen:

- Was ist Softwarequalität, welche Facetten gibt es?
- Welche Möglichkeiten gibt es Softwarequalität **qualitativ** oder **quantitativ** zu bestimmen?

Aufgabe B: Metriken

Bestimmen Sie die Codegröße (Lines of Code) und die Komplexität (Cyclomatic Complexity) von (a) der Originalimplementierung des Lok-Beispiels auf der Webseite und (b) ihrer aktuellsten Implementierung des Lok-Beispiels. *Recherchieren Sie ggf. die Funktionsweise der Metriken; Sie können die Metrik manuell bestimmen oder bestehende Werkzeuge benutzen.*

Aufgabe C: Anti-Pattern

Recherchieren Sie zwei Anti-Pattern (etwa aus dem Katalog <http://c2.com/cgi/wiki?AntiPatternsCatalog>: wählen Sie Pattern aus unterschiedlichen Kategorien) und erläutern Sie diese anhand eines Beispiels. Welchen Wert haben Anti-Pattern?

Aufgabe D: Code Smells und Refactorings

In der Vorlesung wurden eine Reihe von Code Smells und zugehörigen Refactorings vorgestellt.

Geben Sie für jeden der folgenden Code Smells ein Beispiel aus dem Lok-Szenario oder Bank-Szenario:

- Replizierte Methode in unterschiedlichen Klassen
- Große Klasse mit vielen Methoden
- Lange Parameterliste
- Feature Neid (Feature Envy)
- Switch-Statements
- Middle Man

Wenn der Code Smell dort nicht auftaucht, erweitern Sie das Beispiel entsprechend (Erweiterung des UML Diagrams reicht in der Regel aus).

Wie kann man den Code Smell jeweils beseitigen? Diskutieren Sie mögliche Refactorings und Ergebnisse (erneut mit UML falls hilfreich).

Ihre Lösung dieses Übungszettels geben Sie bitte wie gewohnt bis zum 09.12.2012 23:59 Uhr per Email an selecture@mathematik.uni-marburg.de ab.