

## Praxis der Softwareentwicklung, Sommersemester 2019

### Qualitätssicherung

In dieser Phase soll das fertige Software-Produkt validiert (insbesondere getestet) werden. Die Qualitätssicherung soll in einem Dokument zusammengefasst werden. Die Qualitätssicherung umfasst **insbesondere** (weitere Anforderungen sind im Dokument **Tipps & Tricks**<sup>1</sup> des IPD Snelting zu finden):

#### Unit-Tests

- Führen Sie Unit-Test soweit möglich automatisch durch (mit **JUnit**<sup>2</sup>). Insbesondere sollen dabei die im Pflichtenheft genannten Testfälle abgearbeitet werden.
- Geben Sie dazu die erreichte Abdeckung an (z.B. mit **JaCoCo**<sup>3</sup>). Um welche Art von Abdeckung handelt es sich?

#### Integrations- und Belastungstests

- Testen Sie die GUI nach einem selbstverfassten Testplan.
- Führen Sie die im Pflichtenheft genannten Testszenarien durch.
- Testen Sie, ob Ihr Programm auf verschiedenen Systemen (Hardware/Software) lauffähig ist.
- Testen Sie die Qualität der Funktionalitäten. Überlegen Sie sich zunächst, welche Qualitätsanforderungen wie getestet werden können. Erstellen Sie Statistiken und aussagekräftige Grafiken dazu.
- Untersuchen Sie Laufzeit und Speicherverbrauch Ihres Produkts durch Debugging und Profiling. Geben Sie Statistiken an. Beschreiben Sie eventuelle Optimierungen.

#### Fehlerbehebung

- Dokumentieren Sie jeden gefundenen Fehler (sowohl in den Anforderungen als auch der Implementierung). Benutzen Sie dazu eine Bugtracking-Software (z.B. **Mantis**<sup>4</sup> oder **github**<sup>5</sup>).
- Korrigieren Sie Ihren Entwurf bzw. Ihre Implementierung entsprechend (ohne den Funktionsumfang zu erweitern).
- Schreiben Sie Regressionstests zu den gefundenen Fehlern.
- Führen Sie weitere Refactorings durch, die die Wartbarkeit des Produkts erhöhen.

---

<sup>1</sup><https://git.scc.kit.edu/IPDSnelting/pse-tips>

<sup>2</sup><http://junit.org/>

<sup>3</sup><https://www.jacoco.org/jacoco/>

<sup>4</sup><http://www.mantisbt.org/>

<sup>5</sup><http://github.com>