



TESTAUTOMATISIERUNG MIT SELENIUM

- TEILAUTOMATISIERTE GENERIERUNG VON PAGE OBJECTS -

Fakultät für Informatik und Mathematik
der Hochschule München

Masterarbeit

vorgelegt von

Matthias Karl

Matrikel-Nr: 03280712

im <Datum>

Prüfer: Prof. Dr. Ullrich Hafner

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Studienarbeit selbstständig und nur unter Verwendung der von mir angegebenen Quellen und Hilfsmittel verfasst zu haben. Sowohl inhaltlich als auch wörtlich entnommene Inhalte wurden als solche kenntlich gemacht. Die Arbeit hat in dieser oder vergleichbarer Form noch keinem anderem Prüfungsgremium vorgelegen.

Datum: _____ Unterschrift: _____

Zusammenfassung / Abstract

Inhaltsverzeichnis

Eidesstattliche Erklärung	I
Zusammenfassung / Abstract	II
1 Einleitung	1
2 Grundlagen	2
2.1 Software-Qualität	2
2.2 Softwaretest	5
2.3 Testprozess	5
2.3.1 Testplanung und Steuerung	5
2.3.2 Testanalyse und Testdesign	5
2.3.3 Testrealisierung und Testdurchführung	5
2.3.4 Testauswertung und Bericht	5
2.3.5 Abschluss der Testaktivitäten	5
2.4 Softwarelebenszyklus	5
2.4.1 V-Modell	5
3 Testautomatisierung	6
3.1 Warum Testautomatisierung	6
3.2 Bereiche der Testautomatisierung	6
3.2.1 Testdesign	6
3.2.2 Testcodeerstellung	6
3.2.3 Testdurchführung	6
3.2.4 Testauswertung	6
3.3 Schnittstellen der Testautomatisierung zum System	6
3.3.1 API	6
3.3.2 GUI	6

4	Testautomatisierung mit Selenium	7
4.1	Selenium	7
4.2	Testdurchführung mit Selenium	7
4.3	Testcodeerstellung mit Selenium	7
4.3.1	Record-and-playback	7
4.3.1.1	Vorteile von Record-and-playback	7
4.3.1.2	Probleme von Record-and-playback	7
4.3.2	Manuell	7
4.3.3	Page Object Pattern	7
4.3.3.1	Vorteile des Page Object Pattern	7
4.3.3.2	Probleme des Page Object Pattern	7
5	Teilautomatisierte Generierung von Page Objects	8
5.1	übersicht über die Idee	8
5.2	einordnung des Testharness und gui in die Gesamtstruktur (Deploymentdiagramm)	8
5.3	übersicht über Aufbau des Systems	8
5.3.1	pro modul ein kapitel	8
5.4	Vorteile und Probleme	8
5.5	Anwendung	8

1 Einleitung

2 Grundlagen

2.1 Software-Qualität

Nahezu jeder Programmierer ist schon einmal mit dem Begriff der Software-Qualität in Berührung gekommen. Diesen Qualitätsbegriff jedoch genau zu fassen erweist sich als schwierig. Die DIN-ISO-Norm 9126 definiert den Begriff Software-Qualität wie folgt:

„Software-Qualität ist die Gesamtheit der Merkmale und Merkmalswerte eines Software-Produkts, die sich auf dessen Eignung beziehen, festgelegte Erforderniss zu erfüllen.“[ISO]

Aus dieser Definition wird deutlich, dass es sich bei dem Begriff der Software-Qualität eine multikausale Größe handelt. Das bedeutet, dass zur Bestimmung der Qualität einer Software nicht ein einzelnes Kriterium existiert. Vielmehr verbergen sich hinter dem Begriff eine ganze Reihe verschiedener Kriterien die je nach den gestellten Anforderungen in ihrer Relevanz variieren.[Hof, vgl. Seite 6 ff.] Sammlungen solcher Kriterien werden in sogenannten Qualitätsmodellen zusammengefasst. Die DIN-ISO-Norm 9126 bietet selbst ein solches Qualitätsmodell und definiert damit eine Reihe von wesentlichen Merkmalen, die für die Beurteilung der Software-Qualität eine Rolle spielen. Diese Merkmale sind in der Abbildung 2.1 zusammengefasst. Eine nähere Definition der einzelnen Begriffe des Qualitätsmodells kann beispielsweise dem Buch Software-Qualität von Dirk W. Hoffmann entnommen werden. [Hof, Seite 7 ff.] Um die Qualität einer Software zu Steigern bietet die moderne Software-Qualitätssicherung eine Vielzahl von Methoden und Techniken. Ein Teil der Methoden geht dabei davon aus, dass ein qualitativ hochwertiger Prozess der Produkterstellung die Entstehung von qualitativ hochwertigen Produkten begünstigt. Das Augenmerk wird hierbei also auf die Prozessqualität gelegt. Diese Methoden fallen in den Bereich der Prozessqualität. Die klassischen Vorgehensmodelle der Softwareentwicklung werden z.B. hier eingeordnet. Einen weiteren Bereich bilden die Methoden die zur Verbesserung der Produktqualität dienen. Bei diesen Methoden wird das Softwareprodukt direkt bezüglich der Qualitätsmerkmale überprüft. Dieser Bereich unterteilt sich in die konstruktiven und analytischen Qualitätssicherung. Unter konstruktiver Qualitätssicherung versteht man den Einsatz von z.B. Methoden, Werkzeugen oder Standards die dafür sorgen, dass ein Produkt bestimmte Forderungen



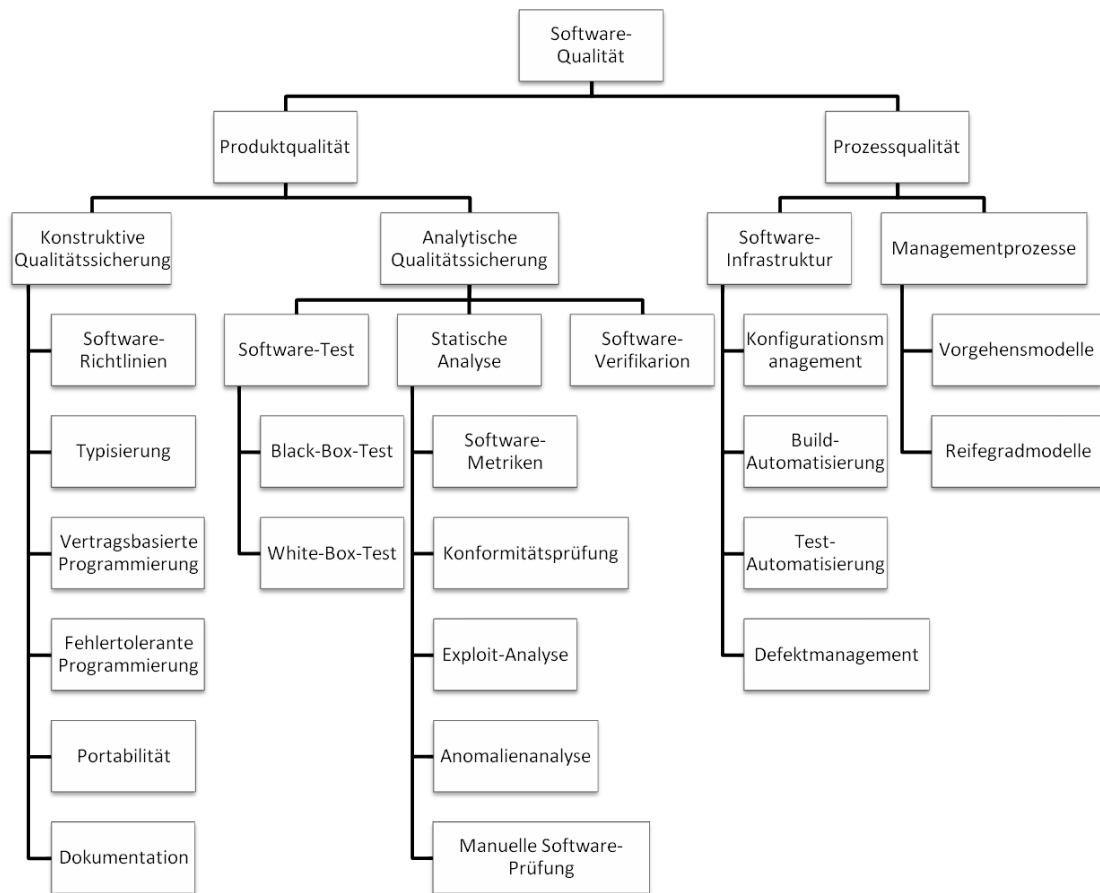
Quelle: [ISO]

Abbildung 2.1: Qualitätsmerkmale von Softwaresystemen (ISO 9126)

erfüllt. Unter analytische Qualitätssicherung versteht man den Einsatz von analysierenden bzw. prüfenden Verfahren, die Aussagen über die Qualität eines Produkts machen. In diesem Bereich der Qualitätssicherung befindet sich beispielsweise der klassische Software-Test.[Hof, vgl. Seite 19 ff.] Eine Übersicht über das gesamte Gebiet der SoftwareQualitätssicherung, wie es sich uns gegenwärtig darstellt, ist in Abbildung 2.2 dargestellt.

2.2 Softwaretest

In kapitel.. hben wir gesehn, dass der softwertest in .. angesiedelt ist.



Quelle: [Hof, vgl. Seite 20]

Abbildung 2.2: Übersicht über das Gebiet der Software-Qualitätssicherung

2.3 Testprozess

2.3.1 Testplanung und Steuerung

2.3.2 Testanalyse und Testdesign

2.3.3 Testrealisierung und Testdurchführung

2.3.4 Testauswertung und Bericht

2.3.5 Abschluss der Testaktivitäten

2.4 Softwarelebenszyklus

2.4.1 V-Modell

3 Testautomatisierung

3.1 Warum Testautomatisierung

3.2 Bereiche der Testautomatisierung

3.2.1 Testdesign

3.2.2 Testcodeerstellung

3.2.3 Testdurchführung

3.2.4 Testauswertung

3.3 Schnittstellen der Testautomatisierung zum System

3.3.1 API

3.3.2 GUI

4 Testautomatisierung mit Selenium

4.1 Selenium

4.2 Testdurchführung mit Selenium

4.3 Testcodeerstellung mit Selenium

4.3.1 Recorde-and-playback

4.3.1.1 Vorteile von Recorde-and-playback

4.3.1.2 Probleme von Recorde-and-playback

4.3.2 Manuell

4.3.3 Page Object Pattern

4.3.3.1 Vorteile des Page Object Pattern

4.3.3.2 Probleme des Page Object Pattern

5 Teilautomatisierte Generierung von Page Objects

5.1 übersicht über die Idee

5.2 einordnung des Testharness und gui in die Geamtstruktur (Deploymentdiagramm)

5.3 übersicht über Aufbau des Systems

5.3.1 pro modul ein kapitel

5.4 Vorteile und Probleme

5.5 Anwendung

Abbildungsverzeichnis

2.1	Qualitätsmerkmale von Softwaresystemen (ISO 9126)	3
2.2	Übersicht über das Gebiet der Software-Qualitätssicherung	4

Tabellenverzeichnis

Literaturverzeichnis

[Hof] HOFFMANN, Dirk W.: *Software-Qualität*. 2013. Springer. – ISBN 9783540763222

[ISO] ISO/IEC: *ISO/IEC 9126. Software engineering – Product quality*. 2001. ISO/IEC