

Studienprojekte als Experimentierraum für QS

Karin Vosseberg

Hochschule Bremerhaven
Studiengänge Informatik und Wirtschaftsinformatik
An der Karlstadt 8
27568 Bremerhaven
karin.vosseberg@hs-bremerhaven.de

Abstract: Der „Test-First-Ansatz“ und damit eng verbunden eine Testautomatisierung wird mit agilen Vorgehensmodellen in der Softwareentwicklung vorangetrieben. Testautomatisierung ist jedoch mehr als nur Testfälle implementieren. Was müssen Studierende lernen, um Testautomatisierung gewinnbringend zur Verbesserung der Softwarequalität einzusetzen.

1 Motivation

Mit der Diskussion über „Test Driven Development“ in Verbindung mit agilen Vorgehensmodellen rücken die Aufgabenbereiche der Rollen „Tester“ und „Entwickler“ immer stärker zusammen. Testen wird nicht länger als separate Aufgabe am Ende des Projekts gesehen sondern als gemeinsame Projektaufgabe zur Verbesserung der Qualität des Softwareprodukts. Mit Aufgaben der Testautomatisierung brauchen Tester verstärkt auch Softwareentwicklungskompetenzen sowie Entwickler Wissen über systematisches Testen [VS13].

2 Studentische Projekte

In einjährigen Bachelorprojekten habe ich an der HS Bremerhaven eine Umgebung geschaffen, in der Studierende der Studiengänge Informatik und Wirtschaftsinformatik mit einem „Test Driven Development“-Ansatz in der Softwareentwicklung experimentieren konnten. Ziel der Projekte war im Rahmen der Entwicklung einer kleinen Applikation die Testautomatisierung der Komponenten-, Integrations- und Systemtestfälle bereits von Beginn des Projekts an zu berücksichtigen. Im Fokus der Projekte stand das Experimentieren mit Werkzeugen für die Einrichtung einer adäquaten Entwicklungs- und Testumgebung. Um eine realistische Projektumgebung zu simulieren, waren externe Auftraggeber involviert, z.B. Bredex GmbH, Braunschweig oder die Forschungstransferstelle der HS Bremerhaven.

Neben Eclipse als Entwicklungsumgebung haben die Studierenden Jubula, Junit und DBUnit zur Implementierung von Testfällen auf den verschiedenen Teststufen eingesetzt. Die Studierenden haben schnell erkannt, dass Testautomatisierung mehr ist als Testfälle zu implementieren. Testautomatisierung ist eng verbunden mit Konzepten der Versionskontrolle, des Daily-Builds und des Continuous Integration. Daraufhin haben sie zusätzlich mit Werkzeugen wie Jenkins, Subversion, Maven, EclEmma eine Werkzeugkette für ihre nächtlichen Testläufe aufgebaut.

Flankiert wurden die Aktivitäten mit einer Vorlesung zum Thema „Systematisches Testen“ auf Basis des internationalen Lehrplans zum ISTQB® Certified Tester Foundation Level [CT11]. Die Studierenden hatten die Möglichkeit hier zusätzlich das Zertifikat zu erwerben.

Fazit

Projekte bieten genügend Spielraum, um die praktischen Erfahrungen in der Softwarequalitätssicherung sammeln zu können. Die Studierenden konnten ihre Projektergebnisse, z.B. auf der ignite 2010 [VF10] oder dem Eclipse Testing Day 2011 [SWV11] vorstellen. Ein externer Auftraggeber ist sehr hilfreich, um den Realitätsbezug herzustellen, aber führt auch dazu, dass durch Termindruck die Entwicklung - wie in realen Projekten - wieder stärker in den Vordergrund rückt. Nach wie vor schwierig gestaltet sich die Frage, wie der theoretische Input zum systematischen Testen in die Bewältigung der Aufgabe zu integrieren ist, ohne das Projekt zu überfrachten. Aktuell lege ich in der Veranstaltung „Grundlagen Qualitätsmanagement“ den Fokus auf die Vermittlung der methodischen Grundlagen und versuche im Übungsbetrieb gemeinsam mit Kollegen der HEC GmbH, Bremen die Testautomatisierungskette zu thematisieren.

Literaturverzeichnis

- [CT11] Certified Tester Foundation Level Syllabus, Version 2011 1.0.1, www.german-testing-board.info/service/information/lehrplaene.html#c72 25.4.2014.
- [SWV11] Markus Sagurna, Jonas Wolniczak, Karin Vosseberg: Testautomatisierung – mehr als nur schreiben von Testfällen, Eclipse Testing Day 2011, Neuss.
- VF10] Karin Vosseberg, Marco Fischer: Evaluation of Test Tools – a Student Education Project. ignite 2010, Düsseldorf.
- [VS13] Karin Vosseberg, Andreas Spillner: Traditionell cum Agil. Am Beispiel Test-Driven Development. ASQF SQ-Magazin, Juni 2013.