

CASE STUDY

MOVIEPLEX7

EINLEITUNG

Die Fallstudie dient dazu das im Unterricht vermittelte theoretische Wissen und die vermittelten Testkonzepte und Konzepte zur Software Qualitätssicherung an einem konkreten Anwendungsfall umzusetzen und zu praktizieren um das erlangte Wissen zu festigen und zu vertiefen.

FALLSTUDIE

Als Fallstudie dient die JEE Movieplex Web Applikation aus dem JEE Tutorial von Glassfish [1], welches auf dem Buch „Java EE Essentials“ von Aaron Gupta [2] basiert.

This hands-on lab builds a Java EE 7 web application that allows customers to view the show timings for a movie in a 7-theater Cineplex and make reservations. Users can add new movies and delete existing movies. Customers can discuss the movie in a chat room. Total sales from each showing are calculated at the end of the day. Customers also accrue points for watching movies."

(aus <https://glassfish.java.net/hol/javaee7-hol.pdf>)

Im Rahmen der Fallstudie werden Sie an dem Projekt verschiedenste Testverfahren anwenden, und fortgeschrittene Konzepte der verteilten Entwicklung und der Build Automatisierung anwenden und auf ihren Nutzen, Vor- und Nachteile analysieren.

LERNZIELE

- Sie können fortgeschrittene Konzepte des verteilten Versionenmanagement effizient für die Softwareentwicklung nutzen
- Sie können fortgeschrittene Konzepte der Build Automatisierung und der kontinuierlichen Integration effizient für die Softwareentwicklung einsetzen
- Sie können auch komplexe verteilte Software mit geeigneten Testverfahren auf verschiedenen Ebenen und für verschiedene Zwecke automatisiert testen.
- Sie können fortgeschrittene Konzepte der Software-Qualitätskontrolle effizient anwenden.
- Sie sind in der Lage Qualitätssicherungsmaßnahmen zu planen.

INHALT UND ABLAUF

UMFANG UND THEMEN

Die Fallstudie besteht aus insgesamt 11 Teilaufgaben zu folgenden Schwerpunktthemen:

Hinweis der Autoren:

Dies ist die erste Ausführung dieses neuen Moduls über fortgeschrittene Konzepte der Softwarequalitätssicherung und des Software-Testens. Der vorgestellte Ablauf stellt eine Planung dar, die eventuell aufgrund von unterschiedlicher Bearbeitungsgeschwindigkeit oder gemachten Erfahrungen während des Semesters angepasst werden muss. Die Autoren behalten sich daher vor, organisatorische Anpassungen, Änderungen bzgl. Ablauf oder inhaltliche Anpassungen nach Bedarf während dem Semester vorzunehmen.

Aufgabe	Thema
1	Software Qualitätssicherung planen <i>Sie entwickeln ein Konzept und evaluieren geeignete Tools zur Sicherung der Softwarequalität in einem Softwareprojekt.</i>
2	Build Automatisierung <i>Automatisiertes Build, Packaging, Testing und Deployment von Software.</i>
3	Dependency Management <i>Automatisierte und sichere Verwaltung Abhängigkeiten von 3rd Party Softwarekomponenten.</i>
4	Distributed Version Control <i>Sie arbeiten im Team mit einem verteilten Versionskontrollsystem und wenden dabei geeignete Workflows an.</i>
5	Traceability <i>Effiziente Verwaltung der Nachverfolgbarkeit von Anforderungen und Änderungen in der Software mittels Issue Trackersystemen.</i>
6	Continuous Integration <i>Eigene Konfiguration eines Continuous Integration Server zur vollständigen Automatisierung und Kontrolle des Build Prozesses.</i>
7	Web UI Testing <i>Testen von Web UI mittels geeigneter Testing Applikationen.</i>
8	Web App Testing <i>Testen von Web Applikationen mittels geeigneter Testing Applikationen (UI unabhängig).</i>
9	Automated Acceptance Testing <i>Entwickeln und Schreiben von automatisierten Akzeptanztests.</i>
10	DB Testing <i>Entwickeln und Schreiben von Tests für datenbankbasierte Applikationen.</i>
11	Software Quality Control <i>Einsatz von Werkzeugen zur kontinuierlichen Softwarequalitätskontrolle</i>

AUSGABE DER AUFGABEN

Die Übungen werden in der Regel im wöchentlichen Rhythmus auf den Active Directory veröffentlicht. Wir empfehlen Ihnen **dringend**, die Aufgaben jeweils auch im Wochenrhythmus zu bearbeiten. Falls Fragen auftauchen, stellen Sie diese sobald als möglich; im Unterricht, per Email an die Dozenten oder schauen Sie einfach im Büro vorbei (5.2B18 oder 5.2B16).

DURCHFÜHRUNG DER ÜBUNGEN UND ABGABE DER LÖSUNGEN

Testat

Die erste Übung beinhaltet die Erstellung von einem Software Quality Assurance Plan (SQAP) mit einem integrierten Software Testing Plan. Die Rückmeldung erfolgt durch einen gegenseitigen Review durch die Studierenden selber. Es ist deshalb nötig, dass alle mitmachen, daher gilt für diese Übung folgende Testatbedingung:

1. **Abgabe des SQAP bis 27. September 2015** und
2. **Abgabe des Reviews zum SQAP bis 4. Oktober 2015**

Wird einer der beiden Termine nicht eingehalten, so wird das Testat nicht erteilt, das Modul gilt als nicht bestanden!

Stichzeit für die Abgabe ist jeweils **23:59** (Lokalzeit in Windisch) der oben genannten Tage.

Warnung: Es gibt immer wieder Wartungszeiten der IT-Services, an denen das System offline sein kann. Checken Sie rechtzeitig ein. **Offline-Zeiten sind kein Grund für spätere Abgaben!!!**

Alle anderen Übungen (ausser SQAP)

Hier gibt es keine Testatpflicht. Die Übungen sind mit einem Abgabetermin versehen, wer diesen einhält bekommt ein Feedback. Auf zu späte Abgaben kann kein Feedback garantiert werden.

Abgabe

Die Abgabe erfolgt jeweils als *annotated tag* im Git Repository der Übungsgruppe.

REFERENZEN

- [1] Glassfish. *Java EE 7 Hands-on Lab*. <https://glassfish.java.net/hol/javaee7-hol.pdf>, retrieved 11.09.2015.
- [2] Aaron Gupta. *Java EE 7 Essentials*. O'Reilly Media; 1 edition (September 6, 2013). ISBN-13: 978-1449370176.