University of Zurich
Department of Informatics

RE
RG

Software Qualität Übung 1

JUnit Tests, Versionskontrolle, Bugtracking

1 Informationen

1.1 Daten

- Ausgabe: 03.03.2008
- Abgabe: 17.03.2008 24:00
- Besprechung: 07.04.2008

1.2 Formales

Die Dateien, welche zu Ihrer Abgabe gehören, müssen in eine .zip-Datei gepackt werden (Diagramme und andere Dokumente als PDF, Quellcode als Textdateien in einem separaten Unterverzeichnis). Die Abgabe erfolgt per Email an reinhard@ifi.uzh.ch.

Die Aufgaben sollen mit der Java-IDE Eclipse (http://www.eclipse.org) gelöst werden. Auf den Rechnern in den Computerräume am IFI ist Eclipse installiert. Der abgegebene Code muss auf Java 5 (oder tiefer) lauffähig sein.

1.3 Gruppen

Die Übung ist in 2er Gruppen zu lösen. Falls die Aufgaben aufgeteilt werden, muss klar ersichtlich sein, wer welchen Teil bearbeitet hat. Alle Gruppenmitglieder müssen über alle Teile Auskunft geben können.

2 Vorbereitung

Laden Sie vom Repository die verschiedenen Komponenten herunter. Sie müssen sich dazu im Netz der Uni befinden (z.B. über VPN). Die Verbindungsinformationen (Host, Username, Password) erhalten Sie in der Vorbesprechung.

Loggen Sie sich mit den in der Vorbesprechung erhaltenen Angaben auf den Bugzilla-Server ein (nur innerhalb des Uni Netzes oder per VPN möglich). Im Bugzilla befindet sich ein Produkt welches die Ihrer Gruppe zugewiesenen Bugs enthält. Diese Bugs enthalten Ihre Arbeitsanweisungen.

3 Ausgangslage

Die Airline PinkAir möchte zusammen mit den Reisebüros Earthtrotter und Travelagent ihre Buchungssoftware weiterentwickeln. Mit Travelagent besteht schon eine langjährige Zusammenarbeit und eine funktionierende Software. Diese muss jedoch erweitert und verbessert werden. Gleichzeitig möchte das Reisebüro Earthtrotter mit PinkAir zusammenarbeiten. Earthtrotter hat schon eine eigene Buchungssoftware, die nun an den PinkAir-Server angebunden werden soll.

Nach vielen Entwicklungsstunden wurde ein Prototyp des Gesamtsystems von Fachleuten geprüft. Die Ergebnisse dieser Tests wurden mit Bugs, eingetragen in Bugzilla, festgehalten. Sie haben nun als Team die Aufgabe die Bugs zu lösen. Damit in Zukunft nicht mehr so viele Probleme auftreten, sollen gleichzeitig auch JUnit Tests für wichtige Funktionalitäten der Software geschrieben werden.

4 Aufgabenstellung

A: Beziehung zwischen den Bugs

Begutachten Sie die Bugs und suchen Sie nach allfälligen Beziehungen/Abhängigkeiten. Vermerken Sie die Beziehungen mit den von Bugzilla zur Verfügung gestellten Mitteln (*depends on, blocks, duplicate of*). Notieren Sie die entdeckten Beziehungen ebenfalls auf Ihrer Abgabe.

B: Testen

Im Package com.pinkair.tests des Moduls tests gibt es zwei Klassen mit denen Sie nun Tests erstellen sollen. Mit der Klasse GraphicalTest soll das Erstellen und Ausfüllen der GUIs automatisiert werden. Implementieren Sie dazu die vorgegebenen Methodenrümpfe. Nutzen Sie die Klasse ServerTest, um funktionale und Unittests für den PinkAir Server zu erstellen. Vervollständigen Sie dazu die bestehenden Methodenrümpfe. Die Angaben, was die einzelnen Methoden tun sollen, ist im Code selbst angegeben. Sie können die Arbeit in der Gruppe aufteilen, abgegeben müssen Sie jedoch eine Version die alle Tests enthält.

Schreiben Sie zu den Ihnen zugewiesenen Bugs, die den Kommentar “TEST” enthalten, entsprechende JUnit Tests. Für Tests, welche Eingaben über das graphische Interface benötigen, sollen wenn möglich die in der Klasse GraphicalTest implementierten Methoden benutzt werden. Die einzelnen Tests sollen als Methoden in der Klasse ServerTest erstellt werden. Verwenden Sie dazu die in der Klasse vorbereiteten Testmethoden.

C: Verteilte Entwicklung mit CVS

Implementieren Sie die in den Bugs geforderten Änderungen. Achten Sie dabei darauf, dass sie immer nur einen Bug beheben und danach ein "Commit" machen, damit sich auf dem Repository immer eine aktuelle Version befindet. Falls beim "Einchecken" Konflikte auftreten, beheben Sie diese sofort. Die unter B erstellten Tests müssen auf der Schlussversion korrekt funktionieren. Es ist möglich, dass Sie während der Implementierung die Tests anpassen müssen, da sich die Voraussetzungen durch die Bugs geändert haben.

D: Versionskontrolle

Beantworten Sie folgende Fragen:

- Warum ist es besser möglichst oft möglichst kleine Änderungen aufs Repository zu laden?
- Viele Tools zur Versionskontrolle benutzen, im Gegensatz zu CVS, einen Lock-Mechanismus (d.h. ein File kann immer nur von einer Person auf einmal bearbeitet werden). Was sind die Auswirkungen davon? Geben Sie Vor- und Nachteile an.
- Sie haben im Software-Praktikum an einem grösseren Projekt mit mehreren Teammitgliedern gearbeitet. Haben Sie damals ein Tool zur Versionskontrolle verwendet? Falls ja, beschreiben Sie die Erfahrungen, die Sie damit gemacht haben. Falls nein, überlegen Sie sich, wie sich die Arbeit verändert hätte, wenn Sie ein ein solches Tool eingesetzt hätten. Gab es spezifische Probleme, welche hätten verhindert werden können?

5 Abgaben

Es werden folgende Abgaben erwartet:

- Teil A: Die Abhängigkeiten der Bugs, sowohl in Bugzilla als auch auf dem Abgabeblatt.
- Teil B: Die JUnit Tests und Hilfsmethoden implementiert
- Teil C: Alle Bugs "gefixt". Die unter B erstellten Tests müssen auf der Implementierung geprüft werden und korrekt funktionieren.
- Teil B und C: Die Schlussversion muss sich auf dem Server befinden. Zusätzlich muss die Schlussversion als Zip-File abgegeben werden.
- Teil D: Beantwortete Fragen