Software-Qualität: Übung 1
Evolutionäres Entwickeln

Versionskontrolle
Automatisiertes Testen
Bugtracking

University of Zurich
Department of Informatics

ifi

© 2007, University of Zurich, Department of Informatics, Samuel Fricker

Inhalt

Ziele

- Übung 1 erklären
- Überblick über Werkzeuge geben
  - JUnit
  - CVS
  - Bugzilla

Agenda

- Zielsetzung, Struktur und Ablauf der Übung 1
- Aufgabe 1: Regressionstests mit JUnit
- Aufgabe 2: Kollaborative Entwicklung mit JUnit, CVS, und Bugzilla

Zielsetzung

- Die Rolle von Regressionstests für die Minimisierung von Fehlern bei ändernder Software verstehen
- Die Rolle von Regressionstests, Konfigurationsmanagement, und Bugtracking für die gemeinschaftliche Softwareentwicklung verstehen
- Verbreitete Werkzeuge verwenden können
  - Eclipse mit JUnit, CVS und Bugzilla

Struktur

- Teil A: Regressionstests
  - A.1: Anwendung im Test-First Ansatz entwickeln
  - A.2: Probleme verstehen, welche bei Änderungen entstehen
  - A.3: Probleme eliminieren
- Teil B: Kollaborative Entwicklung
  - Aufträge über Bugtracking-System entgegennehmen und nachverfolgen
  - Entwicklung mit kontinuierlicher Integration
  - Test-First Entwicklungsansatz weiter vertiefen

Administratives

- Ablauf
  - 2. April: Ausgabe der Übung und Einführung
  - 10. April: Fragemöglichkeiten
  - 11. April: Aufträge and Partner in Bugzilla eingetragen (Übung 2)
  - 17. April: Fragemöglichkeiten
  - 20. April, 18:00 CET: Deadline für Abgabe
- Form
  - Übung A: Individuell
  - Übung B: in Gruppen von 2 Personen
    - 1 Person für Konto.java, 1 Person für Verwaltung.java
- Infrastruktur
  - Fragen: Uniboard Diskussionsforum und cramer@ifi.unizh.ch
  - Abgabe: cramer@ifi.unizh.ch und fricker@ifi.unizh.ch

Inhalt

Ziele

- Übung 1 erklären
- Überblick über Werkzeuge geben
  - JUnit
  - CVS
  - Bugzilla

Agenda

- Zielsetzung, Struktur und Ablauf der Übung 1
- Aufgabe 1: Regressionstests mit JUnit
- Aufgabe 2: Kollaborative Entwicklung mit JUnit, CVS, und Bugzilla

Übung A: Regressionstests

A.1

- MathematikBibliothek
  - public boolean isNumber(Object entry)
  - public int multiplyByHundred(int number)
  - public Vector getMultiplyByHundred(Vector entries)
- CalculatorTestCase
  - public void testIsNumber()
  - public void testMultiplyByHundred()
  - public void testGetMultiplyByHundred()
  - public static Test suite()

Übung A: Regressionstests

A.2

- MathematikBibliothek2
  - public boolean isNumber(Object entry)
  - public boolean isPrime(int number)
  - public Vector getPrimeNumbers(Vector entries)
- CalculatorTestCase2
  - public void testIsNumber()
  - public void testIsPrime()
  - public void testGetPrimeNumbers()
  - public static Test suite()

Übung A: Regressionstests

A.3

- MathematikBibliothek3
  - integration von MathematikBibliothek und MathematikBibliothek2
- CalculatorTestCase
  - integration von CalculatorTestCase und CalculatorTestCase2

Übung A: Vorgehen

Vorbereitung

- Eclipse installieren
- Skeleton von CVS Repository laden
- Projekt von CVS enkoppeln
  Rechte Maustaste > Team > Disconnect

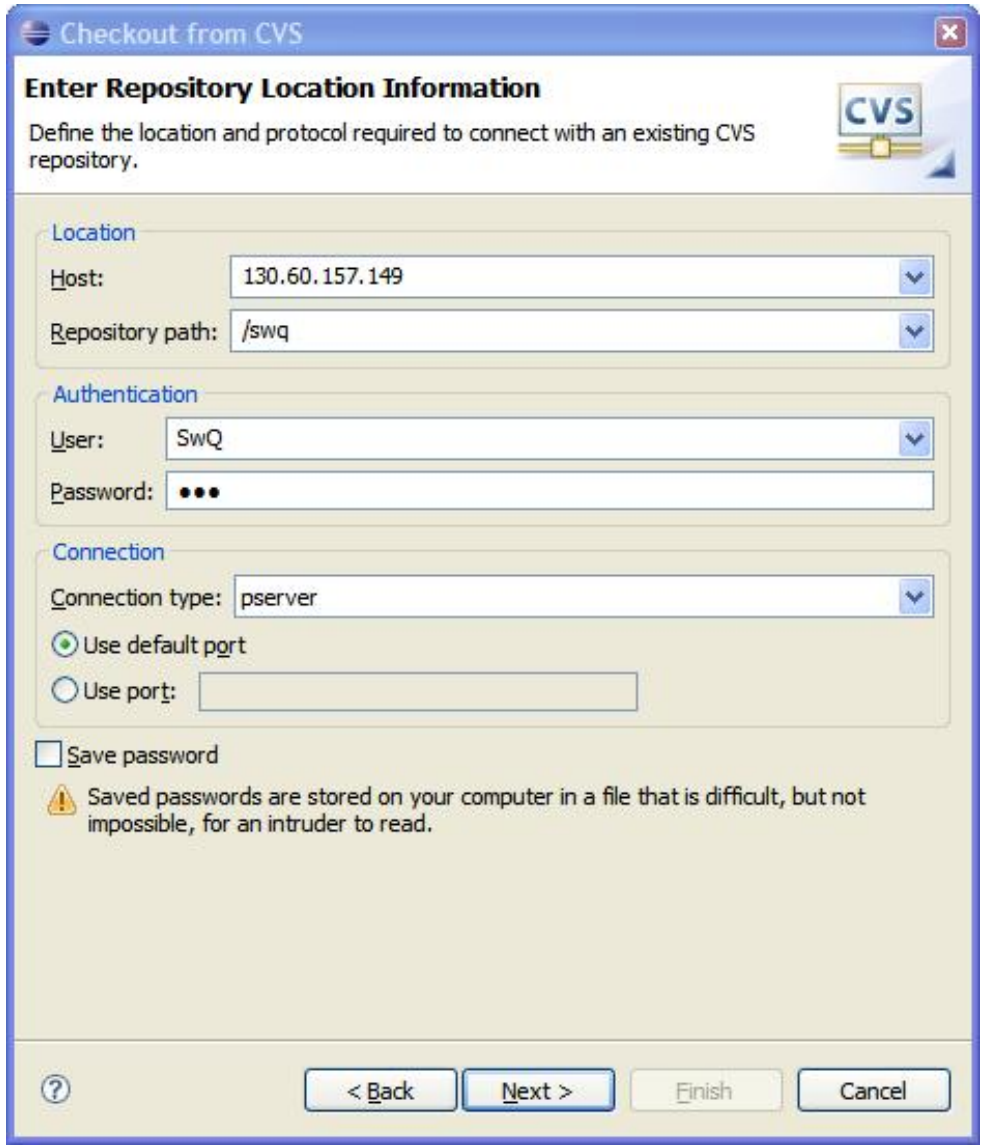
Implementierung

- Methodensignaturen schreiben
- Für jede Methode
  - Tests schreiben
  - Methode implementieren
  - Testen
- Fragen beantworten

Rückgabe

- per e-Mail (ZIP-Datei)

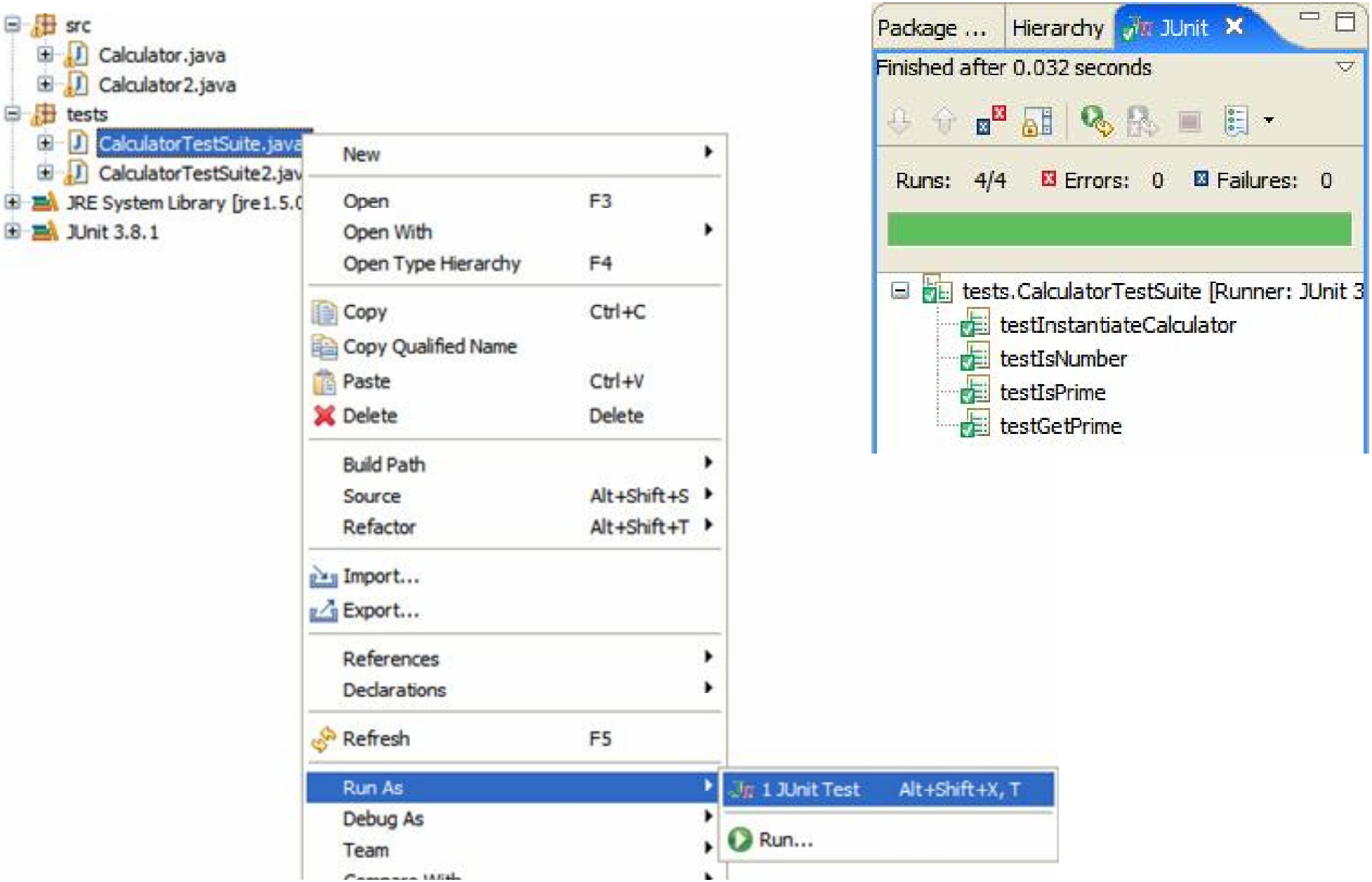
File > Import > CVS > Project from CVS



Übung A: JUnit Tests

```
package tests;

import src.Calculator;
import junit.framework.Test;
import junit.framework.TestCase;
import junit.framework.TestSuite;

public class CalculatorTestSuite extends TestCase {

    //hier werden die Testfälle geschrieben.
    private Calculator calculator;

    public void testInstantiateCalculator() {

    public void testIsNumber() {

        // initialize
        calculator = new Calculator();

        // test sensible inputs and outputs
        assertTrue(...);
        assertFale(...);
        assert...(...);
        ...;
    }

    public void testMultiplyByHundred() {

    public void testGetMultiplyByHundred() {

    public static Test suite() {
        TestSuite suite = new TestSuite(CalculatorTestSuite.class);
        return suite;
    }
}
```

Übung A: JUnit Tests



Inhalt

Ziele

- Übung 1 erklären
- Überblick über Werkzeuge geben
  - JUnit
  - CVS
  - Bugzilla

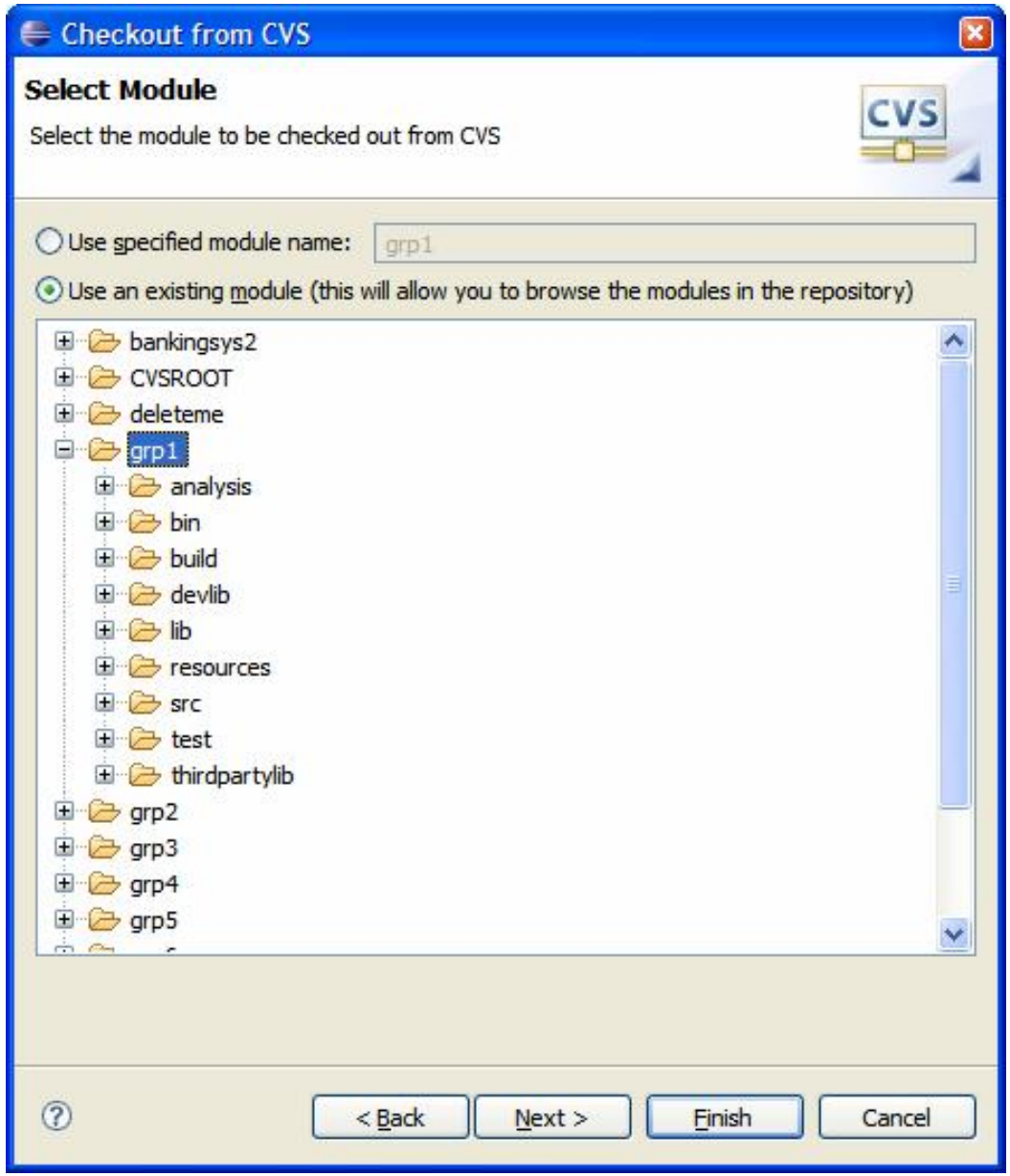
Agenda

- Zielsetzung, Struktur und Ablauf der Übung 1
- Aufgabe 1: Regressionstests mit JUnit
- Aufgabe 2: Kollaborative Entwicklung mit JUnit, CVS, und Bugzilla

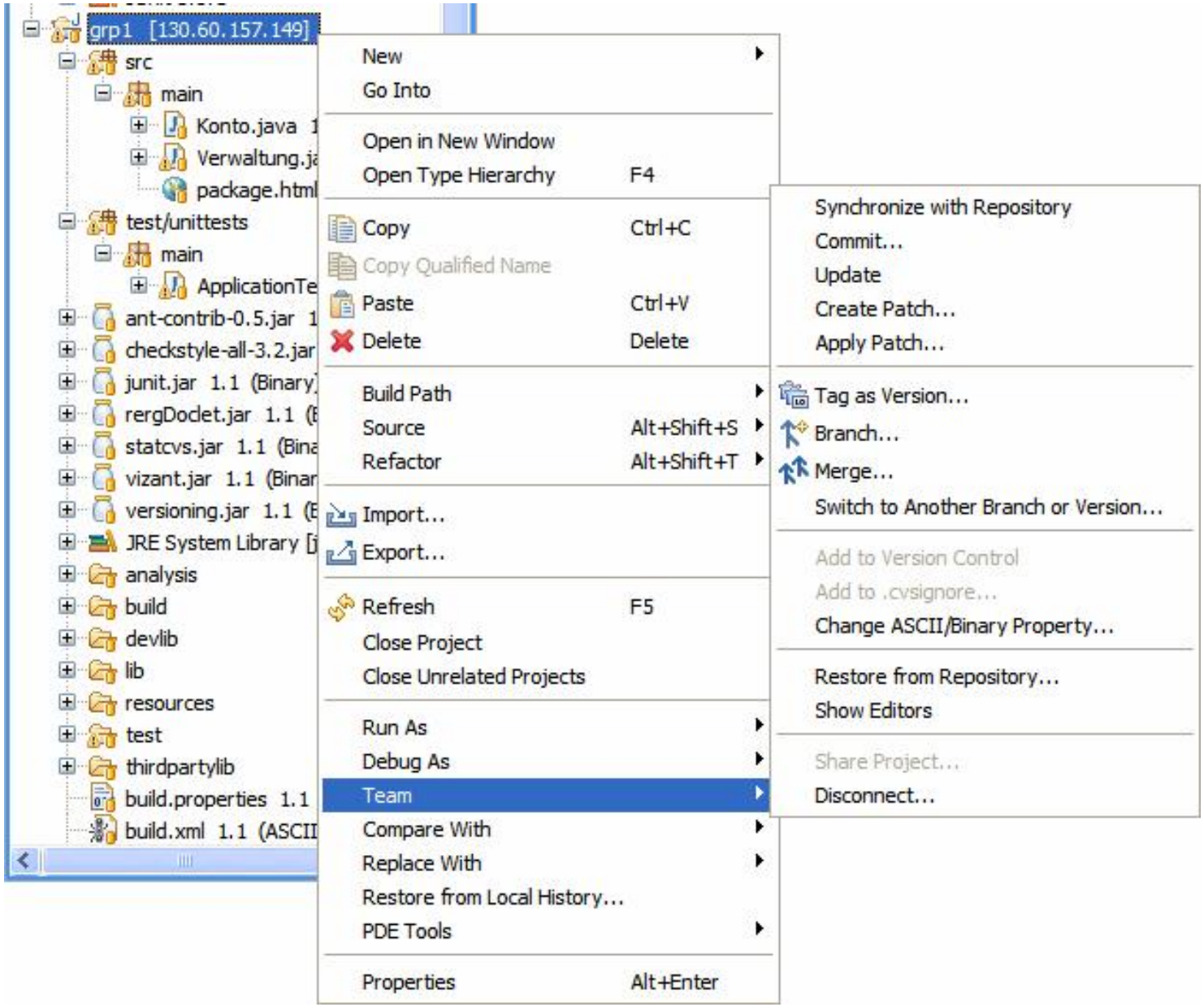
Übung B: Kollaborative Entwicklung

- Projekt „grpX“ von CVS herunterladen
  - (X=Gruppennummer)
- Bugzilla Account eröffnen
  - http://130.60.157.149
  - Bugzilla Produkt „grpX“ öffnen, um zu den Änderungswünschen zu gelangen
- Für jeden Änderungswunsch
  - JUnit Test
  - Implementierung
  - Integration in CVS
  - in Bugzilla Quittieren
- 4 weitere Änderungswünsche für Ihren Partner in Bugzilla eintragen
- Änderungswünsche von Ihrem Partner implementieren

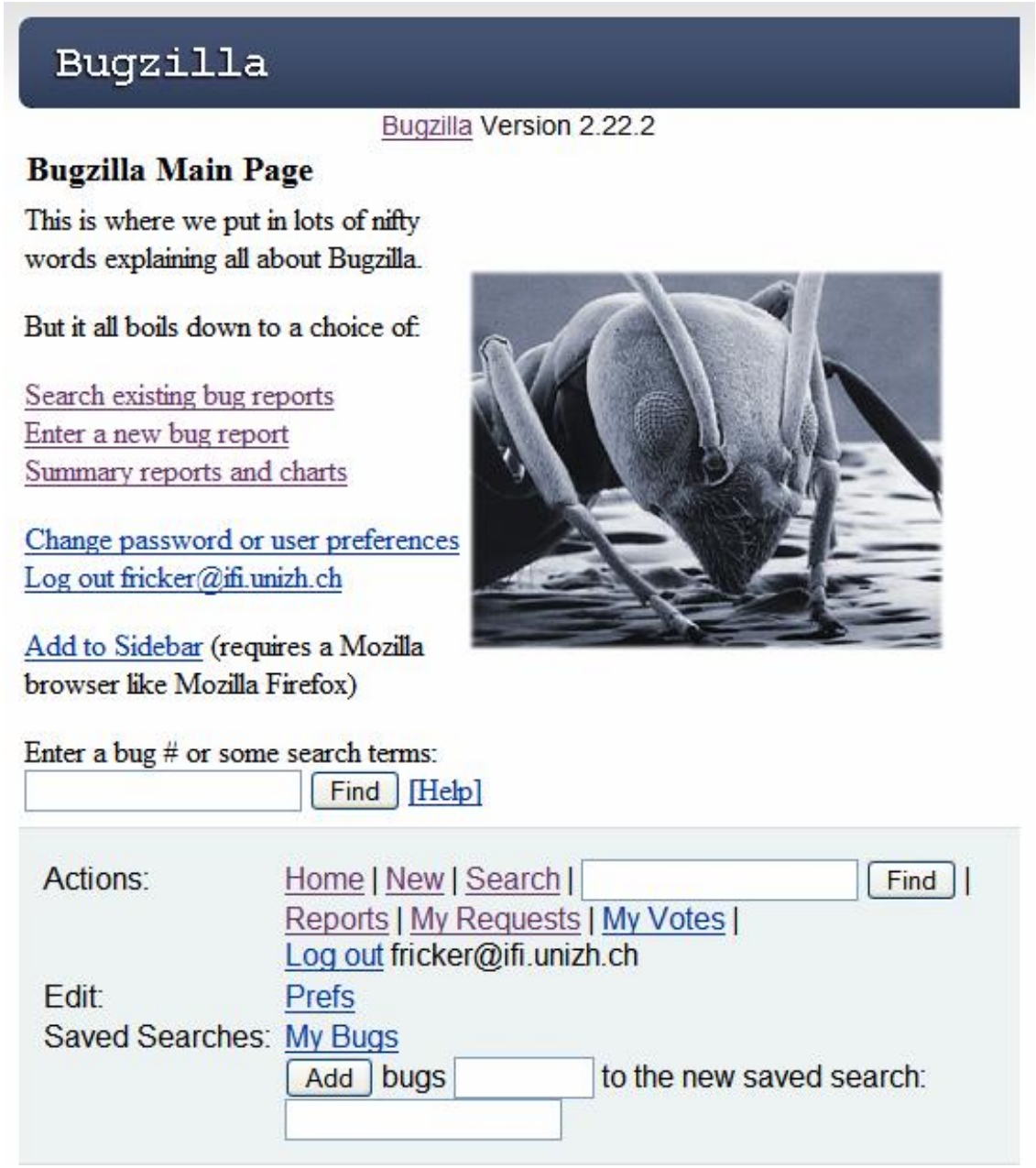
Übung B: CVS Operationen



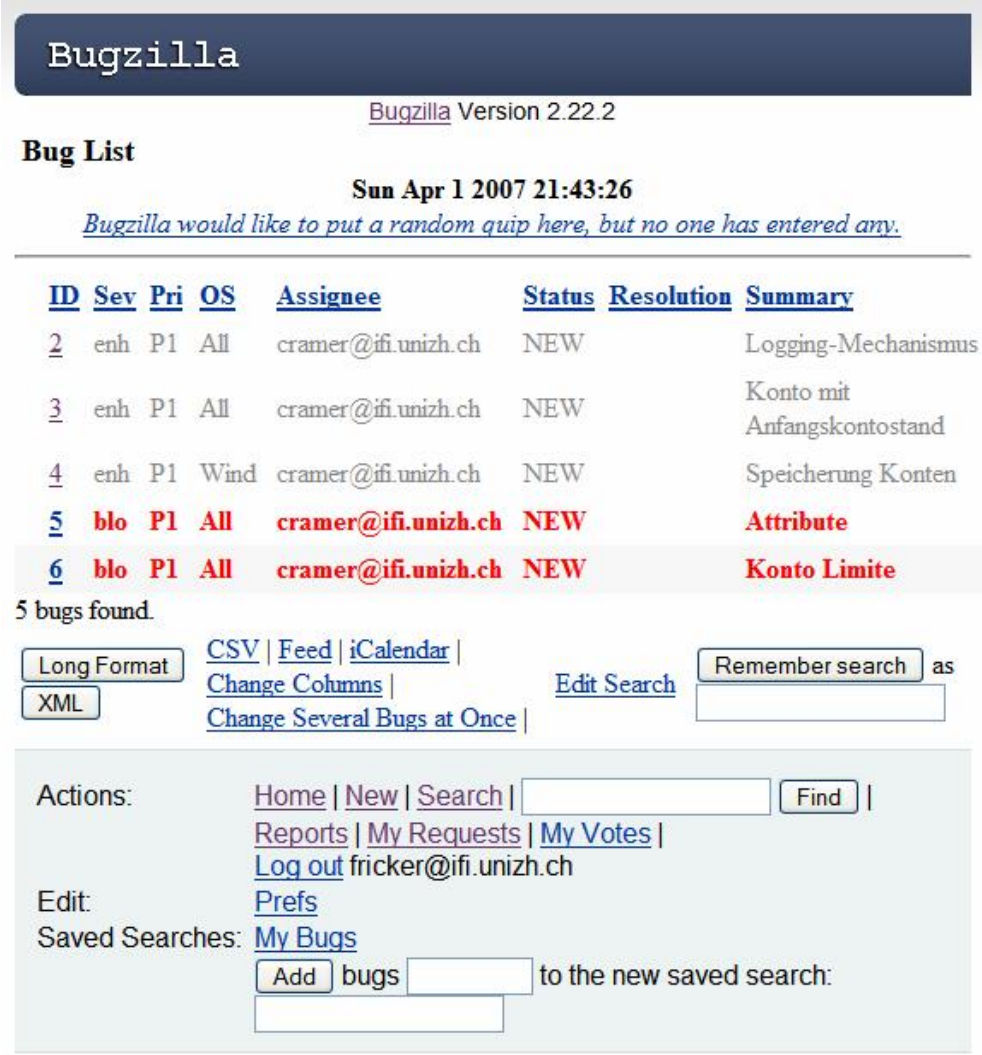
Übung B: CVS Operationen



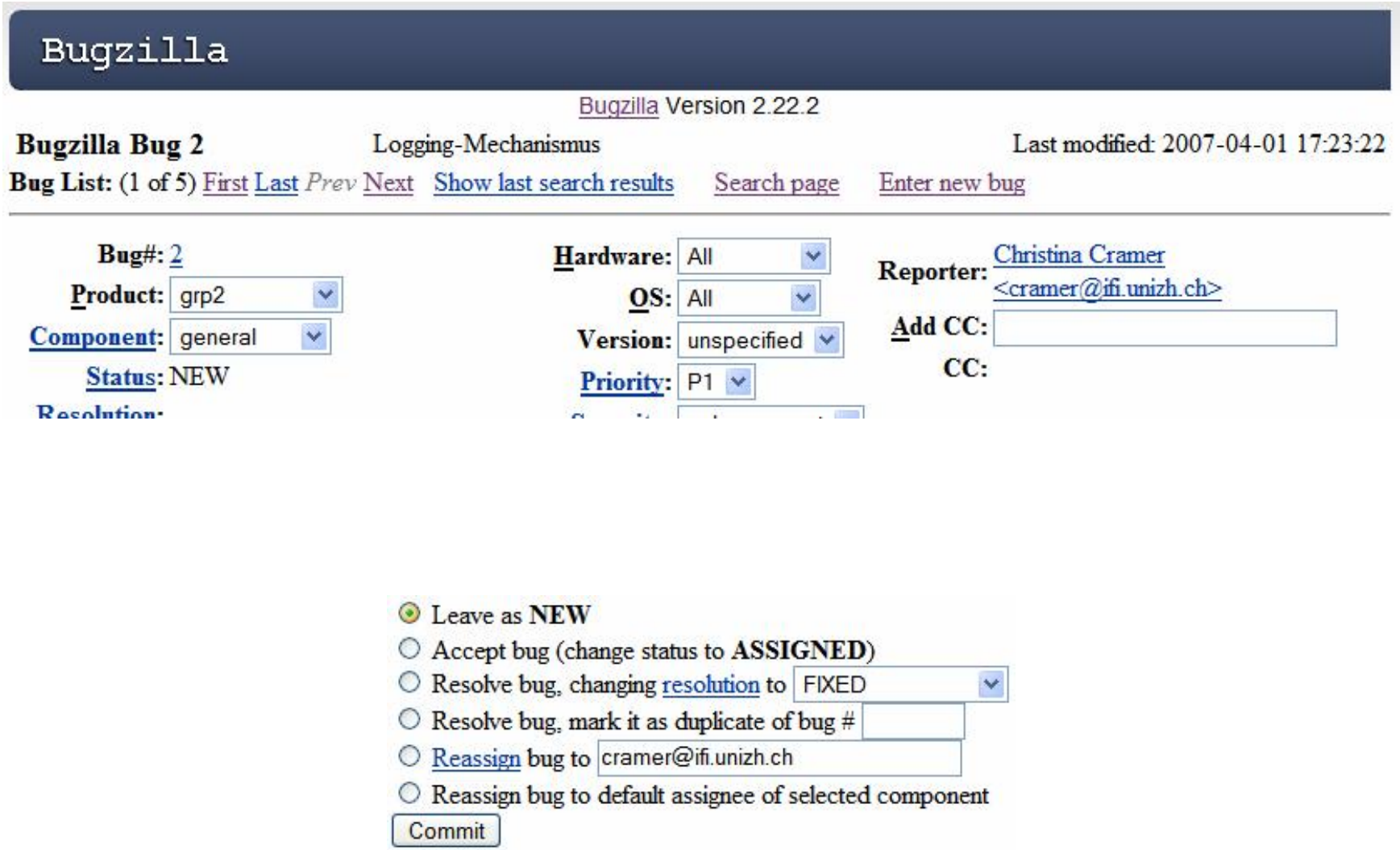
Übung B: Bugzilla Operationen



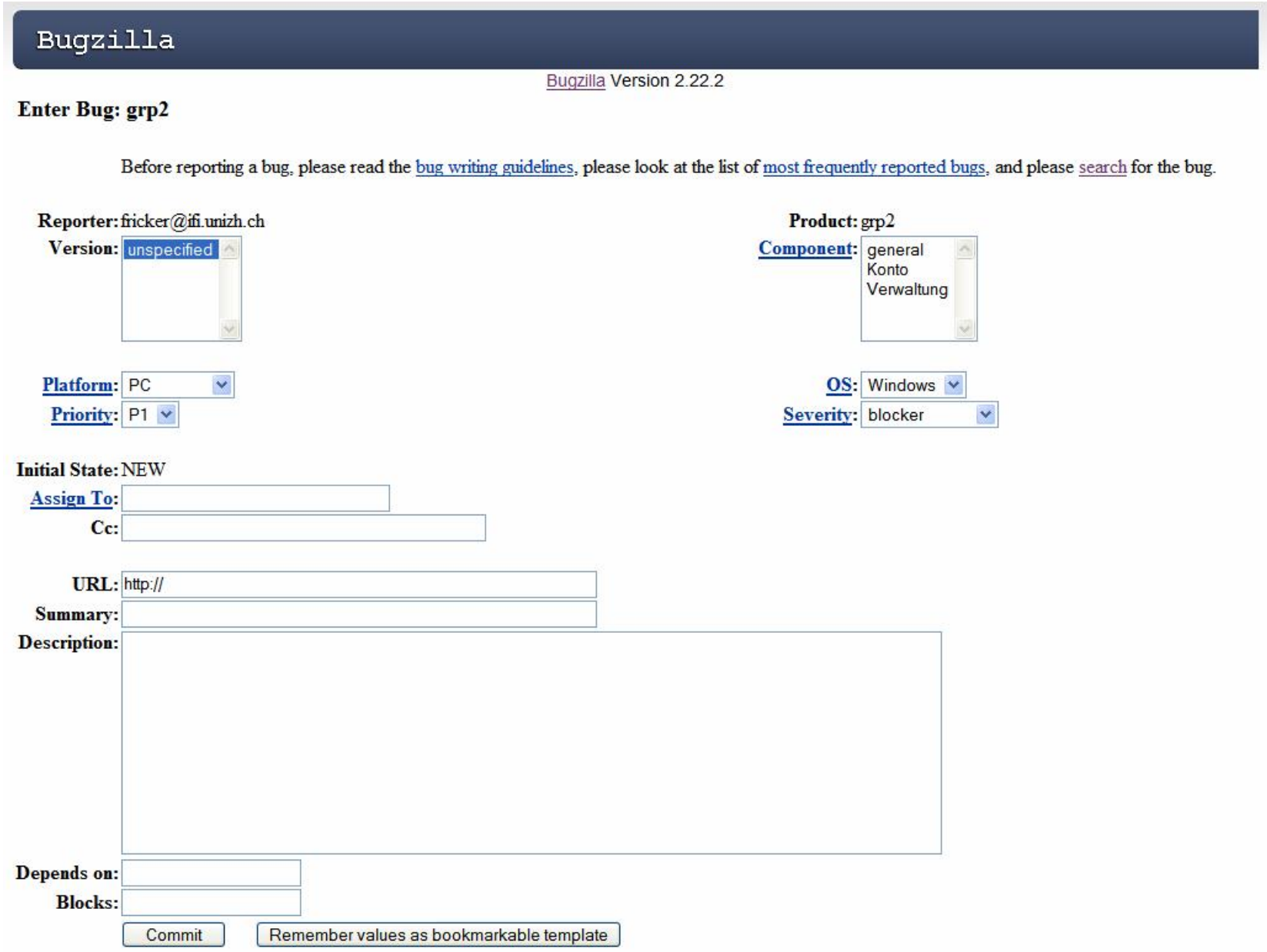
Übung B: Bugzilla Operationen



Übung B: Bugzilla Operationen



Übung B: Bugzilla Operationen



Viel Glück!

- Ablauf
  - 2. April: Ausgabe der Übung und Einführung
  - 10. April: Fragemöglichkeiten
  - 11. April: Aufträge and Partner in Bugzilla eingetragen (Übung 2)
  - 17. April: Fragemöglichkeiten
  - 20. April, 18:00 CET: Deadline für Abgabe
- Form
  - Übung A: Individuell
  - Übung B: in Gruppen von 2 Personen
    - 1 Person für Konto.java, 1 Person für Verwaltung.java
- Infrastruktur
  - Fragen: Uniboard Diskussionsforum und cramer@ifi.unizh.ch
  - Abgabe: cramer@ifi.unizh.ch und fricker@ifi.unizh.ch