

Softwaretechnikprojekt: „Agile Softwareentwicklung in einem Open-Source-Projekt“

Tag 2

Sommersemester 2012

Franz Zieris, M.Sc.
AG Software Engineering
Freie Universität Berlin
04.09.2012

Ankündigung

- Eure heutige Anwesenheit entspricht einer verpflichtenden Anmeldung im Campus Management.
- Nicht-Teilnahme ist demnach eine 5.0





Hausaufgabenkontrolle

1. Lesen der Entwicklerdoku
 - <http://saros-build.imp.fu-berlin.de/sarosGettingStarted>
2. Installieren der Software (Eclipse, Saros, JTourbus, EGit)
3. Anlegen eines Accounts auf saros-con (aus Saros heraus)
 - Bekanntgabe des Account-Namens auf se_p_agil@lists.spline.inf.fu-berlin.de
4. Anlegen eines Kontos auf Sourceforge.net
 - Kurze Mail an zieris@inf.fu-berlin.de
5. Bugtracker durchstöbern: Wie viele offene Bugeinträge gibt es zur Zeit?
6. Abonnieren von dpp-devel@lists.sourceforge.net
 - Archiv auf Gmane finden.
7. Anlegen eines Accounts für Gerrit
 - <http://saros-build.imp.fu-berlin.de/gerrit>
 - Kurze Mail an zieris@inf.fu-berlin.de
8. Git Erstkonfiguration, lokales Repository einrichten (siehe Entwicklerdoku)
9. Saros-Grobarchitektur:
 - In welche Komponenten wird Saros eingeteilt?
 - Was ist deren jeweilige Aufgabe?
10. JTourBus: Tour „Some Basics“ nachvollziehen
11. Saros macht was komisches! → Suche und evtl. Eintragen im Bugtracker.

Hausaufgabenkontrolle

1. Lesen der Entwicklerdoku
 - Gab's Probleme?
2. Installieren der Software (Eclipse, Saros, JTourbus, EGit)
 - Gab's Probleme?
8. Git Erstkonfiguration, lokales Repository einrichten (siehe Entwicklerdoku)
 - Gab's Probleme?
10. JTourBus: Tour „Some Basics“ nachvollziehen
 - Etwas gelernt?
11. Saros macht was komisches!
 - Suche und evtl. Eintragen im Bugtracker. Gemacht?

Hausaufgabenkontrolle

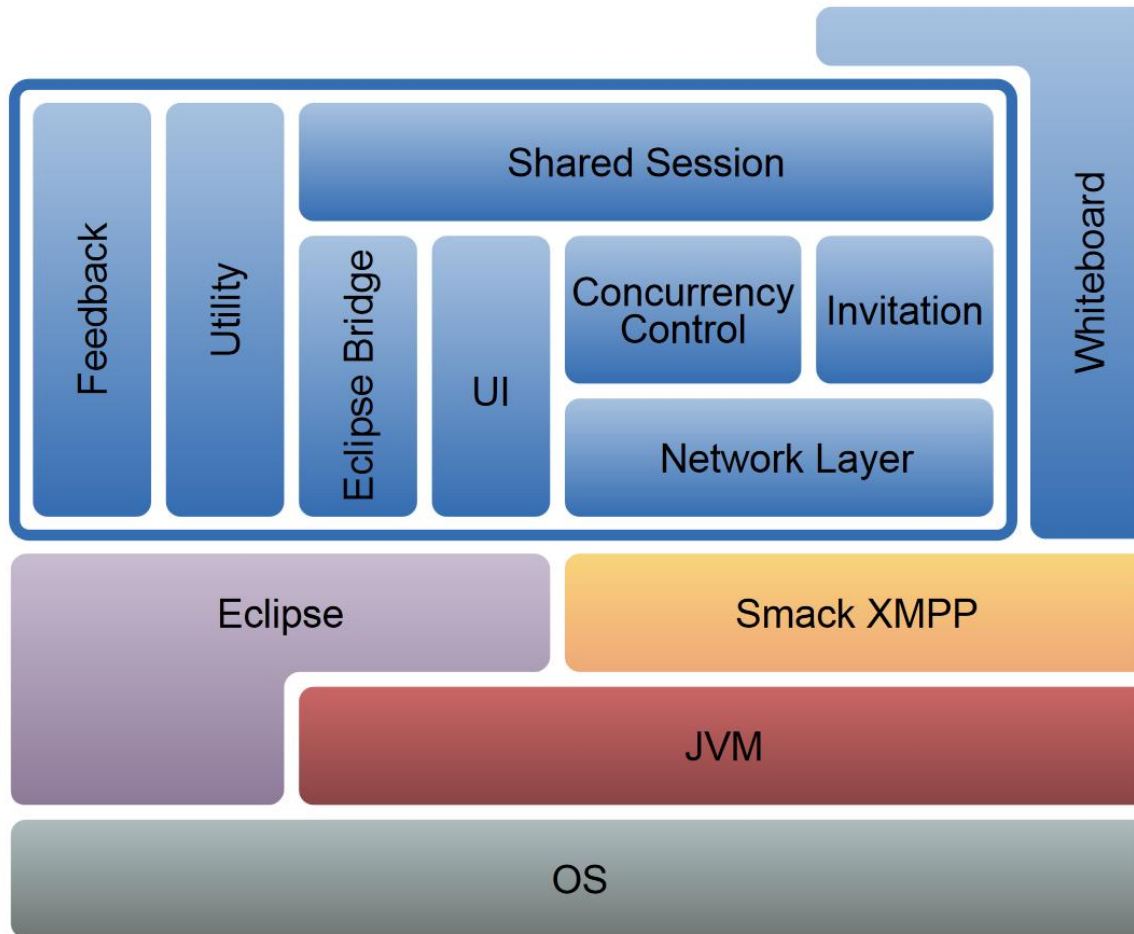
- | | | |
|---------------------------------|-------|---|
| 3. Accounts auf saros-con | (5/5) |  |
| 4. Accounts auf Sourceforge.net | (4/5) |  |
| 7. Accounts auf Gerrit | (4/5) |  |
| 6. Abonnieren dpp-devel@ | (3/5) |  |

Stand: 03.09.2012, 17:30

Hausaufgabenkontrolle

5. Wie viele offene Bugeinträge gibt es zur Zeit?
 - Open: 128 Pending: 16
6. Wie lautet die Adresse des Gmane-Archivs?
 - <http://dir.gmane.org/gmane.comp.ide.eclipse.saros.devel>
9. Saros-Grobarchitektur:
 - In welche Komponenten wird Saros eingeteilt?
 - Was ist deren jeweilige Aufgabe?

Grobarchitektur: Innensicht



siehe: <http://www.saros-project.org/Technicals>

Heute: Fahrplan

- Starten von Saros im Entwicklungsmodus
- PicoContainer
- Jenkins-Prinzipien
- JUnit, STF
- Starten von Saros im STF-Modus

- Etwas Neues 😊

Lehrstoff

Nachtrag Softwaretechnik



Dependency Injection

- Klasse `ImageTool` für Bildbearbeitung
 - benötigt einen Speicher, z.B. auf der Festplatte:
`HarddriveStore`
- **naiv: direkte Abhängigkeit:**

```
class ImageTool {  
    protected IStore store = new HarddriveStore();  
}
```
- **Besser: reinreichen**
 - via Konstruktor: `new ImageTool(new HarddriveStore());`
 - via Setter: `imageTool.setStore(new HarddriveStore());`
 - bei Singletons: via `PicoContainer @Inject IStore;`

PicoContainer

```
225 Component.create(XMPPReceiver.class),  
226 Component.create(XMPPTransmitter.class),  
227
```

„PicoContainer: Hier hast du einen XMPPTransmitter, wenn mal jemand einen braucht.“

```
94 public class SarosSessionManager implements ISarosSessionManager {  
95  
96     private static final Logger log = Logger  
97         .getLogger(SarosSessionManager.class.getName());  
98  
99     @Inject  
100     protected SarosSessionObservable sarosSessionObservable;  
101  
102     @Inject  
103     protected DiscoveryManager discoveryManager;  
104  
105     @Inject  
106     protected XMPPTransmitter transmitter;  
107  
108     @Inject  
109     protected SessionIDObservable sessionID;  
110
```

„PicoContainer: Ich brauche einen XMPPTransmitter!“

PicoContainer

- Mehr zum Hintergrund (Dependency Injection)
 - <http://martinfowler.com/articles/injection.html>
- Mehr zur Implementierung
 - <http://picocontainer.codehaus.org/>
- Wir haben aktuell zwei Container für Singletons:
 - SarosContext
 - langlebig, wie das Plugin in Eclipse
 - SarosSession.sessionContainer
 - für die Dauer einer Session

Jenkins

- Kernprinzip: Jenkins verwaltet Jobs
- die Merkmale eines Jobs sind
 - Trigger: Was löst den Job aus?
 - Zeitgesteuert (wie ein Cron-Job)
 - Ereignisgesteuert (nach einem anderen Job)
 - SCM (bei Änderungen in der Versionsverwaltung)
 - Inhalt: Was macht der Job?
 - Bauen mit Ant
 - Testen mit JUnit
 - Testen mit STF
 - Post-Build:
 - Metriken berechnen
 - veröffentlichen von Artefakten
 - nächsten Job anstoßen


JUnit

- im Order „test/junit“
- Unittests, Steuerung über Annotations
 - @Test
 - Ein Testfall
 - @Before, @After
 - Vor bzw. nach jedem Testfall
 - @BeforeClass, @AfterClass
 - Vor dem ersten bzw. nach dem letzten Testfall
- An jede Methode?
 - Nein, muss „public void“ sein

Saros Test Framework - STF

- im Order „test/stf“
- sind auch JUnit-Tests
- GUI-Ende-zu-Ende-Tests:
 - Starten mehrerer Eclipse-Instanzen
 - Ansprechen der GUI-Elemente der „Teilnehmer“ via SWTBot
 - <http://www.eclipse.org/swtbot/>
 - Simulieren ganzer Sitzungen

```
54 @Test
55 public void testActivateAccountButton() throws Exception {
56     ALICE.superBot().menuBar().saros().preferences()
57         .addAccount(new JID("a@bar.com"), "a");
58
59     ALICE.superBot().menuBar().saros().preferences()
60         .addAccount(new JID("b@bar.com"), "b");
61 }
```



Bei Nutzer ALICE im Menü „Saros“ auswählen, dann die „Preferences“ öffnen und zwei Accounts anlegen.

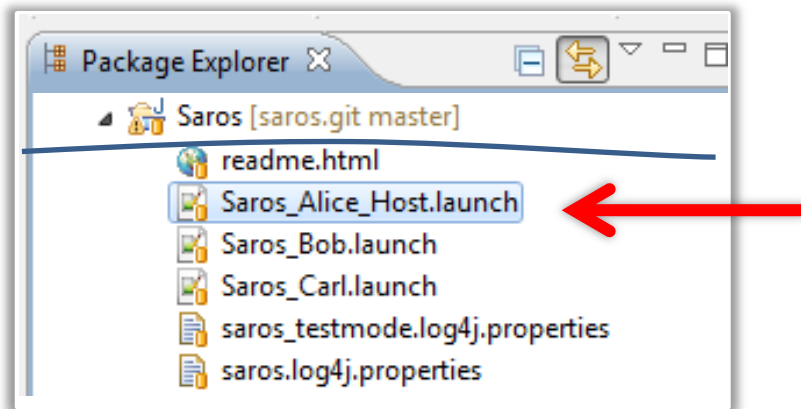
Entwickeln

Saros aus Entwicklersicht

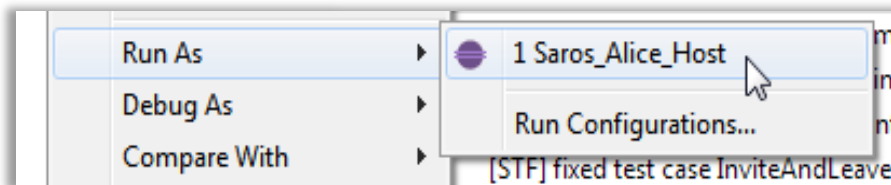
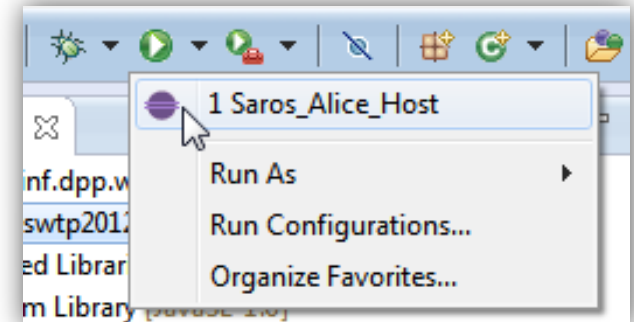


Starten von Saros im Entwicklungsmodus

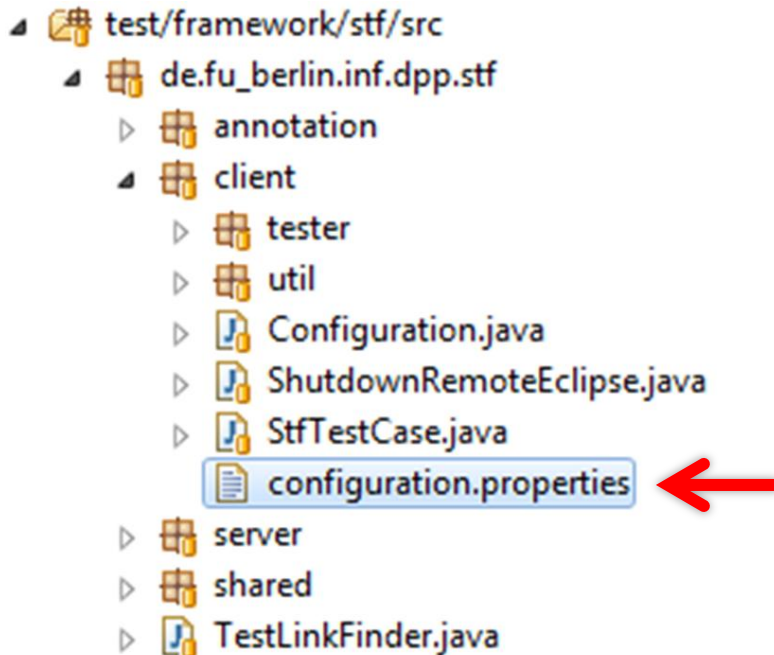
- Direktes Ausführen der aktuellen Quellen des Working Directory
- Starten von Saros: Launch Configurations



oder



STF-Tests vorbereiten



- Anlegen von 4 Test-Accounts
- Muster:
<klarname>_test<n>
- Achtung: Zeitlimit,
Kaffeepause einplanen

```
ALICE_JID = <insert user here>@saros-con.imp.fu-berlin.de/Saros
ALICE_PASSWORD = <insert password here>
ALICE_HOST = localhost
ALICE_PORT = 12345
```

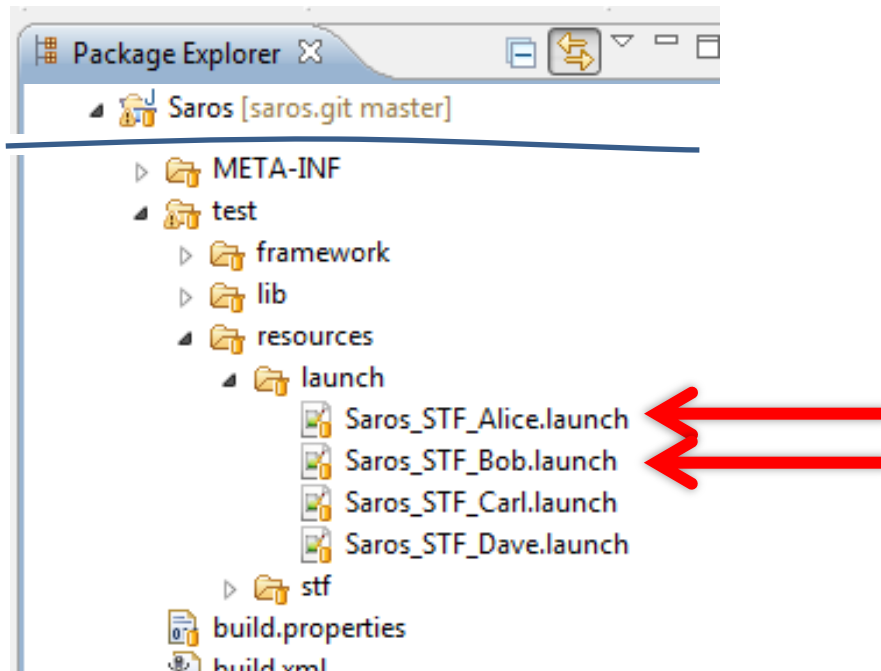
```
BOB_JID = <insert user here>@saros-con.imp.fu-berlin.de/Saros
BOB_PASSWORD = <insert password here>
BOB_HOST = localhost
BOB_PORT = 12346
```

```
CARL_JID = <insert user here>@saros-con.imp.fu-berlin.de/Saros
CARL_PASSWORD = <insert password here>
CARL_HOST = localhost
CARL_PORT = 12347
```

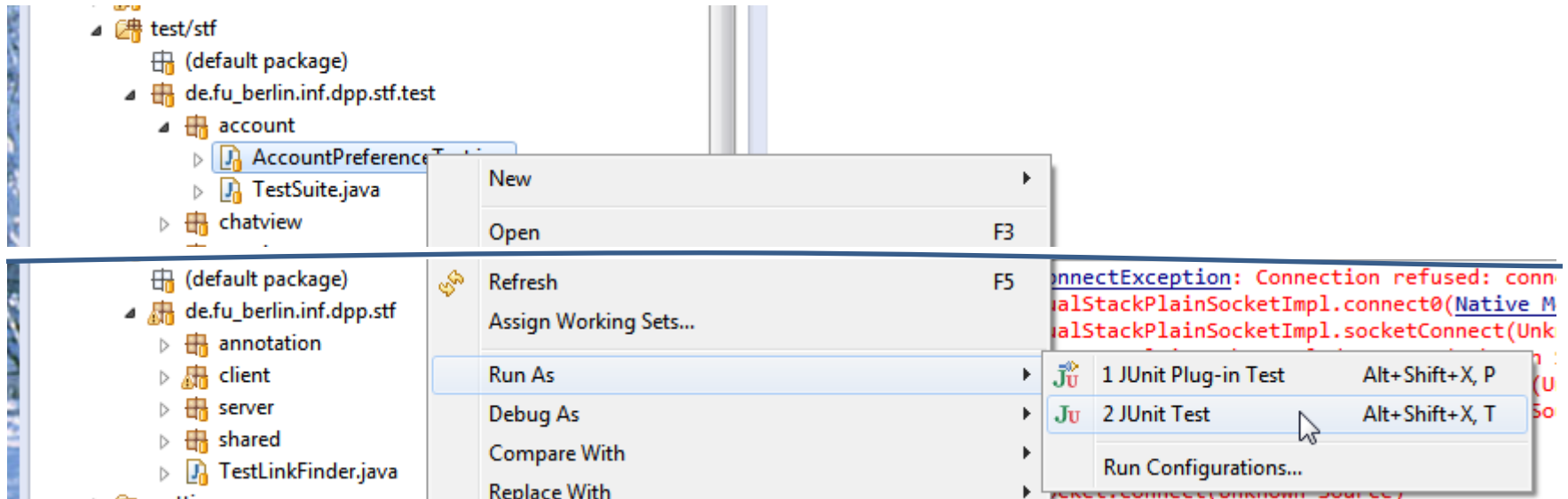
```
DAVE_JID = <insert user here>@saros-con.imp.fu-berlin.de/Saros
DAVE_PASSWORD = <insert password here>
DAVE_HOST = localhost
DAVE_PORT = 12348
```

STF-Tests vorbereiten

- Starten von n Saros: STF-Launch Configurations



STF-Tests starten



- Nicht verzweifeln, wenn es Fehler gibt, sondern systematisch nach den Ursachen suchen
 - Und: erinnert euch an den „Hilfeentscheidungsbaum“ von gestern

Hausaufgaben zu Morgen

0. Aufgaben von Heute nachholen!
1. Alle JTourBus-Touren machen
2. Bugtracker: Wie soll er benutzt werden?
 - Welche Prioritäten gibt es, und wie sollen sie benutzt werden?
 - Welche Status gibt es, und wie sollen sie benutzt werden?
3. STF-Dummyaccounts einrichten
 - Zeit einplanen.
4. Und: ...

Etwas Neues

Ein Quiz



Quiz

- Jeder bekommt 2-3 Fragen zugewiesen (durchnummeriert)
- Fragen erfordern
 - projektspezifisches Fachwissen
 - Zugriff auf verschiedene Informationsquellen
- Morgen: Zufälliger Aufruf
 - Teilnehmer X stellt bitte die Antwort und den Lösungsweg für Frage 13 vor
 - Jeder muss jede Frage beantworten können. Tauscht euch aus!
- Ich bringe die Fragen nachher in K 46

Bis morgen!

10 Uhr, Raum 055