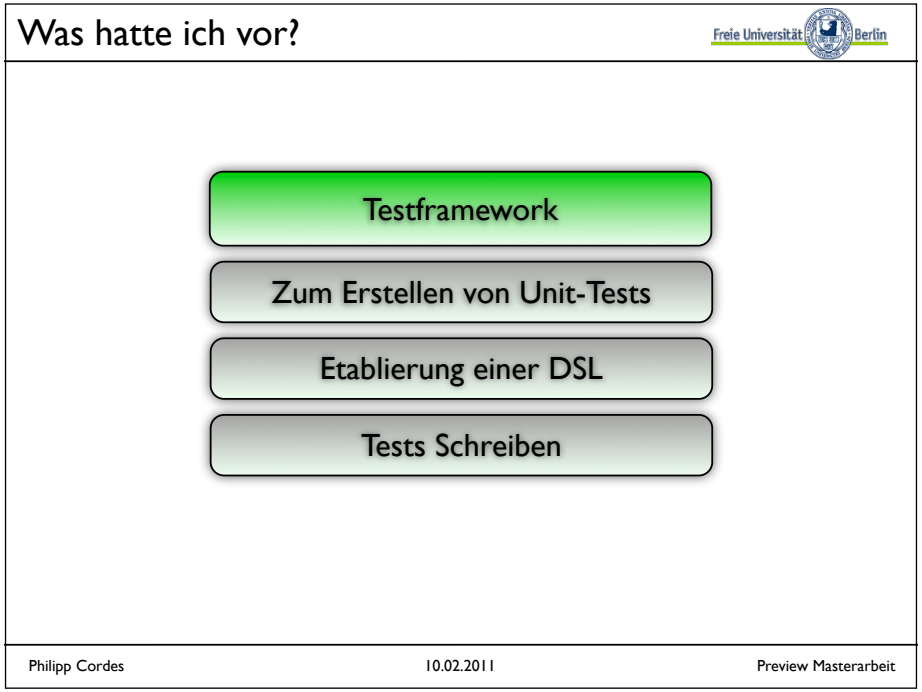
Erstellung eines Mockbasierten Testframeworks für Saros

Preview

Zu Beginn

Freie Universität Berlin

- Voraussetzungen
- Saros
- JUnit
- Dependency Injection

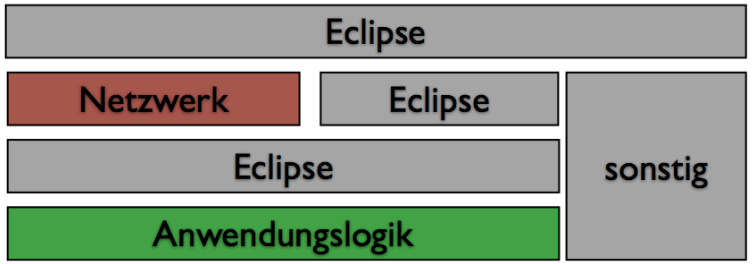


* Schnelles, Einfaches Erstellen von Tests
* Einfaches, Schnellles Ausführen
* Keine Setup notwendig

* Schnellläufig
* Einfach
* Schnelles Erstellen von Tests

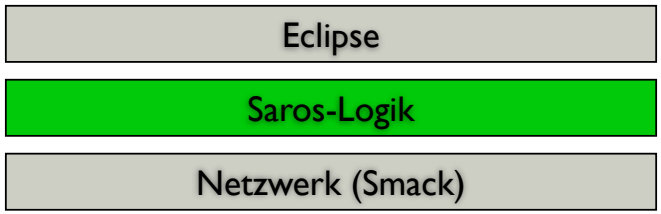
Abgrenzung damals

Freie Universität Berlin



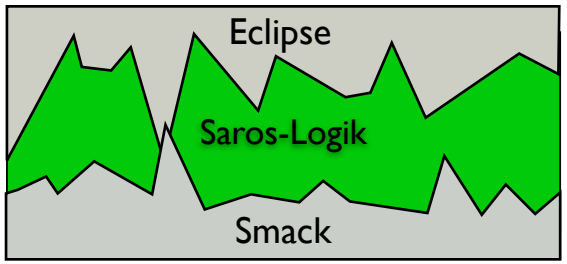
Einordnung heute

Freie Universität Berlin



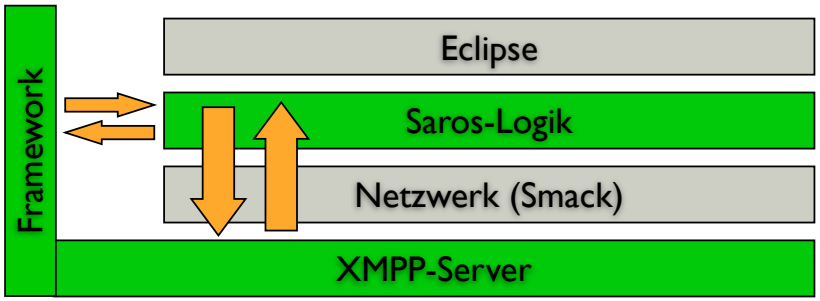
Die Wirklichkeit

Freie Universität Berlin



Einordnung heute

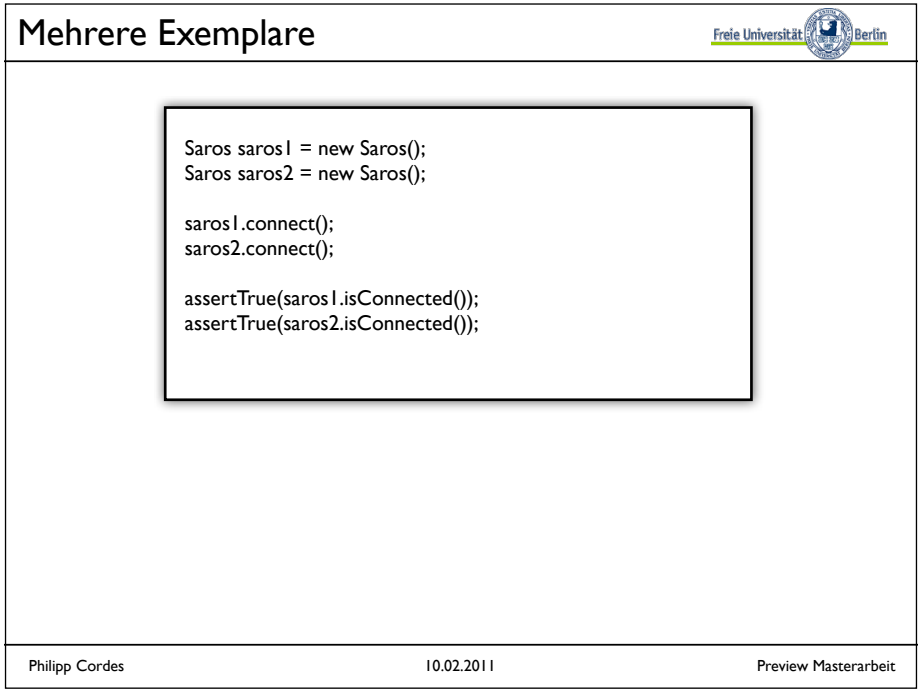
Freie Universität Berlin



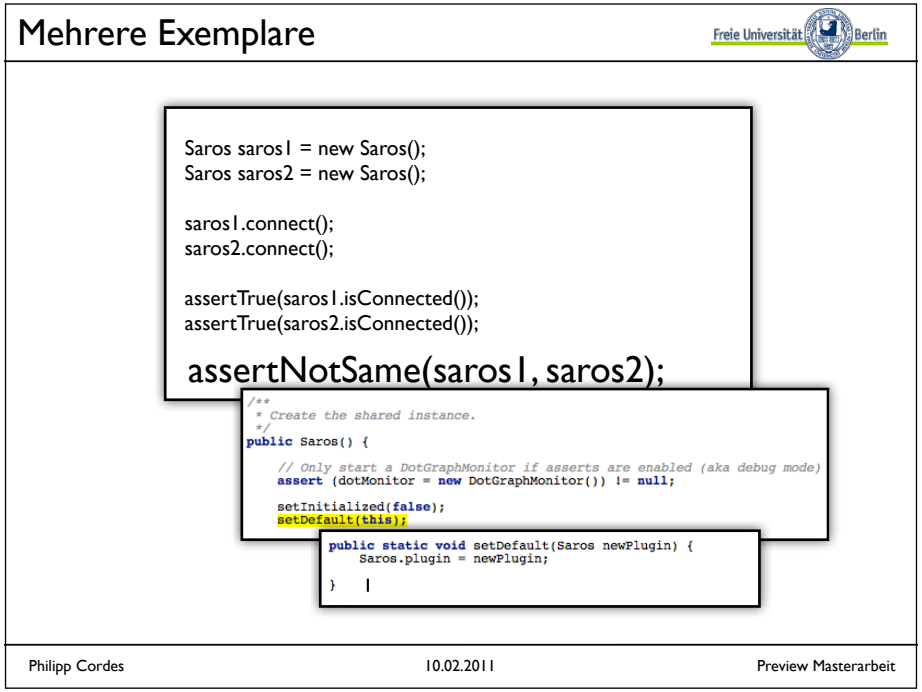
Der erste Test

Freie Universität Berlin

```
Saros saros = new Saros();

saros.connect();

assertTrue(saros.isConnected());
```



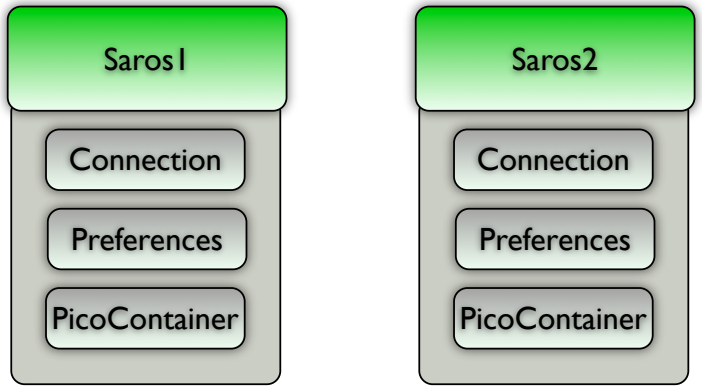
* wir wollen ja auch die Kommunikation zwischen zwei Exemplaren testen können



* wir wollen ja auch die Kommunikation zwischen zwei Exemplaren testen können

Kontext

Freie Universität Berlin



PicoContainer

Freie Universität Berlin

```
PicoBuilder picoBuilder = new PicoBuilder(new CompositeInjection(
    new ConstructorInjection(), new AnnotatedFieldInjection()))
    .withCaching().withLifecycle();

/*
 * If given, the dotMonitor is used to capture an architecture diagram
 * of the application
```

```
/* Core Managers
this.container.addComponent(ChangeColorManager.class);
this.container.addComponent(ConsistencyWatchdogClient.class);
this.container.addComponent(ConsistencyWatchdogServer.class);
this.container.addComponent(DataTransferManager.class);
this.container.addComponent(DiscoveryManager.class);
this.container.addComponent(EditorAPI.class);
this.container.addComponent(EditorManager.class);
this.container.addComponent(ErrorLogManager.class);
this.container.addComponent(FeedbackManager.class);
this.container.addComponent(JDTFacade.class);
this.container.addComponent(LocalPresenceTracker.class);
this.container.addComponent(MUCManager.class);
this.container.addComponent(MUCManagerSingletonWrapperChatView.class);
this.container.addComponent(PingPongCentral.class);
this.container.addComponent(PreferenceManager.class);
this.container.addComponent(PreferenceUtils.class);
this.container.addComponent(PermissionManager.class);
this.container.addComponent(RosterTracker.class);
this.container.addComponent(SarosRosterListener.class);
this.container.addComponent(SarosUI.class);
this.container.addComponent(SarosSessionManager.class);
this.container.addComponent(SessionViewOpener.class);
this.container.addComponent(SharedResourcesManager.class);
this.container.addComponent(SkypeManager.class);
this.container.addComponent(StatisticManager.class);
this.container.addComponent(StopManager.class);
this.container.addComponent(StreamServiceManager.class);
this.container.addComponent(AudioServiceManager.class);
this.container.addComponent(MixerManager.class);
this.container.addComponent(SubscriptionListener.class);
this.container.addComponent(UndoManager.class);
```

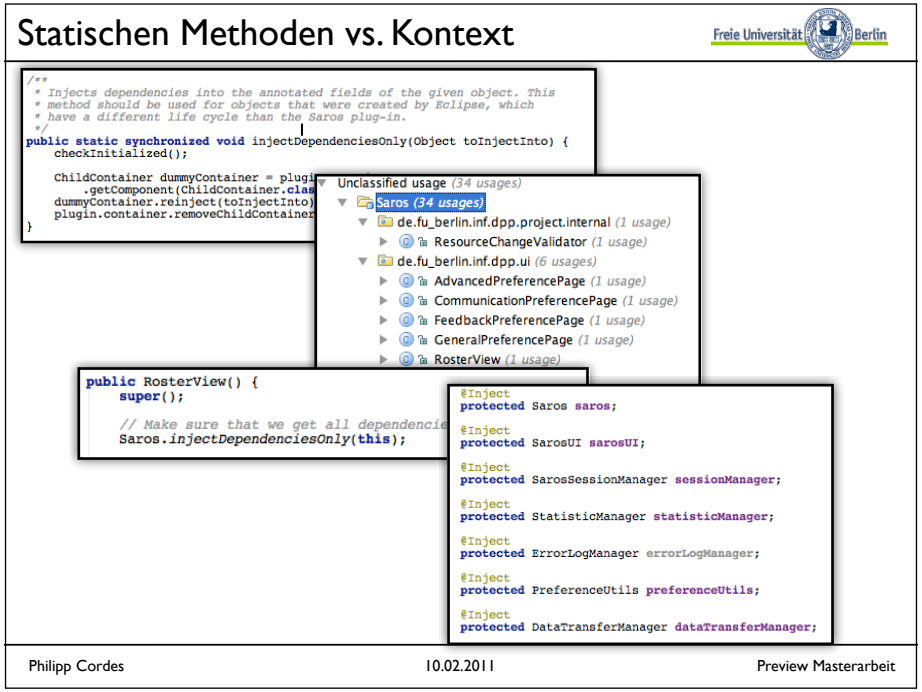
```
@Inject
protected Saros saros;

@Inject
protected RosterTracker rosterTracker;
```

Zwei Kontexte

Freie Universität Berlin

```
Saros saros1 = new Saros();
Saros saros2 = new Saros();

saros1.connect();
saros2.connect();

assertTrue(saros1.isConnected());
assertTrue(saros2.isConnected());
assertNotSame(saros1, saros2);
```


* der weg war geebnet



* die Komponenten müssten sich mit entsprechenden PicoContainer initialisieren

Was wir brauchen?

Freie Universität Berlin

SarosContext

Kapselt PicoContainer

bietet reinject

bietet initComponent

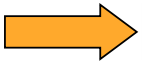
Refactoring

Freie Universität Berlin

Saros.class

```
PicoBuilder picoBuilder = new PicoBuilder(new CompositeInjection(
    new ConstructorInjection(), new AnnotatedFieldInjection()))
    .withCaching().withLifecycle();

/*
 * If given, the dotMonitor is used to capture an architecture diagram
 * of the application
 */
if (dotMonitor != null) {
    picoBuilder = picoBuilder.withMonitor(dotMonitor);
}

// Initialize our dependency injection container
this.container = picoBuilder.build();
```

```
// Core Managers
this.container.addComponent(ChangeColorManager.class);
this.container.addComponent(ConsistencyWatchDogClient.class);
this.container.addComponent(ConsistencyWatchDogServer.class);
this.container.addComponent(DataTransferManager.class);
this.container.addComponent(DiscoveryManager.class);
this.container.addComponent(EditorAPI.class);
this.container.addComponent(EditorManager.class);
this.container.addComponent(ErrorLogManager.class);
this.container.addComponent(FeedbackManager.class);
this.container.addComponent(JDTFacade.class);
this.container.addComponent(LocalPresenceTracker.class);
this.container.addComponent(MUCManager.class);
this.container.addComponent(MUCManagerSingletonWrapperChatView.class);
this.container.addComponent(PingPongCentral.class);
this.container.addComponent(PreferenceManager.class);
this.container.addComponent(PreferenceUtils.class);
this.container.addComponent(PermissionManager.class);
this.container.addComponent(RosterTracker.class);
this.container.addComponent(SarosProtocol.class);
this.container.addComponent(SarosSC.class);
this.container.addComponent(SarosSessionManager.class);
this.container.addComponent(SessionViewOpener.class);
this.container.addComponent(SharedResourcesManager.class);
this.container.addComponent(SkypeManager.class);
this.container.addComponent(StatisticManager.class);
this.container.addComponent(StopManager.class);
this.container.addComponent(StreamServiceManager.class);
this.container.addComponent(AudioServiceManager.class);
this.container.addComponent(MixerManager.class);
this.container.addComponent(SubscriptionListener.class);
this.container.addComponent(UndoManager.class);
```

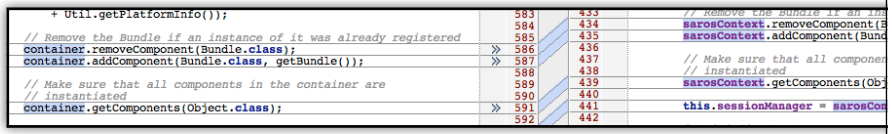


SarosContext.class

Refactoring

Freie Universität Berlin

Benutze SarosContext statt PicoContainer



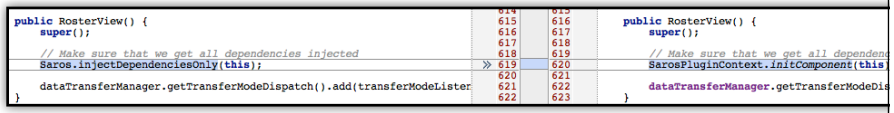
```
    + Util.getPlatformInfo());

// Remove the Bundle if an instance of it was already registered
container.removeComponent(Bundle.class);
container.addComponent(Bundle.class, getBundle());

// Make sure that all components in the container are
// instantiated
container.getComponents(Object.class);
```

```
// Remove the Bundle if an ins
sarosContext.removeComponent(B
sarosContext.addComponent(Bund

// Make sure that all componen
// instantiated
sarosContext.getComponents(Obj

this.sessionManager = sarosCon
```

Mache SarosContext an den kontextsensitiven Stellen zugänglich (z.B. SarosSessionManager)

```
@Inject
protected SarosContext sarosContext;
```

Ersetze „Saros.injectDependenciesOnly“



```
public RosterView() {
    super();

    // Make sure that we get all dependencies injected
    Saros.injectDependenciesOnly(this);

    dataTransferManager.getTransferModeDispatch().add(transferModeLister
}
```

```
public RosterView() {
    super();

    // Make sure that we get all dependenc
    SarosPluginContext.initComponent(this

    dataTransferManager.getTransferModeDis
}
```

Der weitere Weg

Freie Universität Berlin

- Schreiben von Tests
- Erstellen von XMPP-Account
- Hinzufügen von Buddies
- Session starten
- In Session einladen

Mit Gewalt

Freie Universität Berlin

```
if (!Saros.isTestSaros()) {
    // SWT-, Eclipse-Code etc.
}
```

Angebotene Funktionalitäten

Freie Universität Berlin

```
TestSaros saros =
    buildSaros()
    .withUserName("alice")
    .andReturn();
```

```
TestSaros saros =
    buildSaros()
    .withTestWorkspace()
    .withTestProject()
    .andReturn();
```

```
subscribeUsers(asList(saros1, saros2));
```

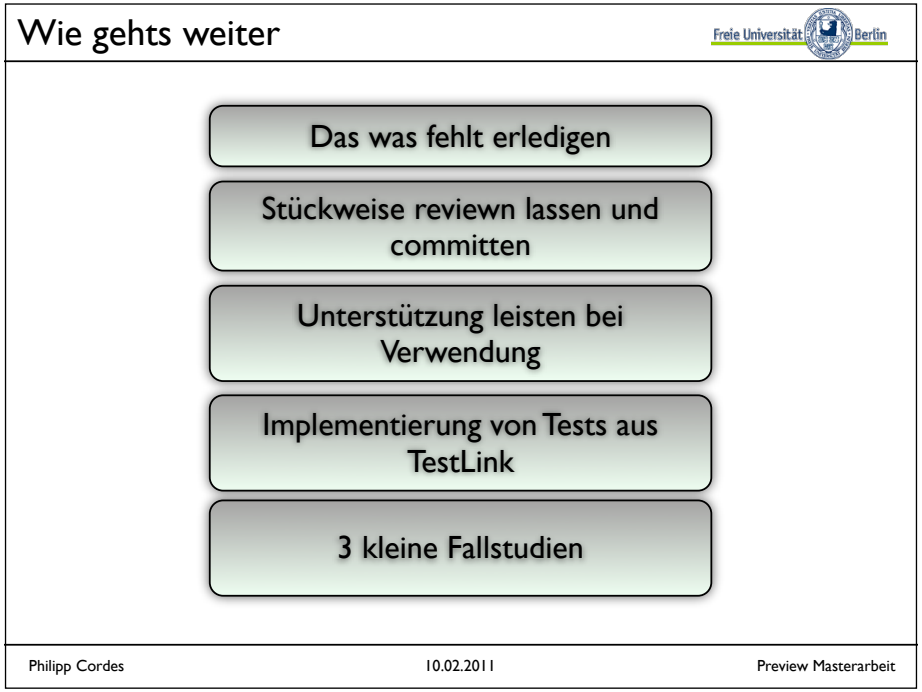
```
final ExpectedPacket expectedPacket =
    getExpectedPacketOnSaros(sarosInstance)
      .containsData("workspace_alice")
      .withTransferdescription(description)
      .andReturn();
```

```
waitFor(new TCondition("received packet") {
    boolean isFullfilled() {
        return expectedPacket.wasReceived();
    }
});
```

Was fehlt?

Freie Universität Berlin

- Refactoring der Stellen Saros.isTestSaros
- Starten/Stoppen XMPP-Server
- Funktionalitäten für Senden/ Empfangen von Aktivitäten
- Weitere Standardfunktionalitäten



* bei reviews werde ich mögliche Tests skizzieren

Demo