

Software Engineering in der Praxis

SS 2008

26. März 2008

1 Motivation

Die Eclipse Plattform wurde geschaffen, um als Basis für Werkzeuge des gesamten Software-Entwicklungszyklus zu dienen. Im Bereich Design und Implementierung ist dies bereits gelungen, jedoch für die Verwaltung von Anforderungen ist noch keine praxistaugliche und quelloffene Erweiterung verfügbar.

Im Rahmen des Praktikums wird ein Eclipse Plugin entwickelt, das die Erfassung von Anforderungen innerhalb von Eclipse ermöglicht: *Reclipse*. Dabei wird besonderer Fokus auf eine flexible Architektur des Plugins gelegt.

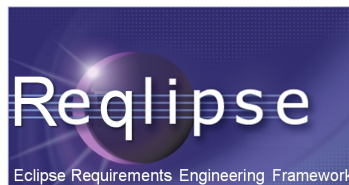


Abbildung 1: Reclipse Logo

Die Lehrveranstaltung führt in die Systementwicklung mit bewährten Software Engineering Methoden durch praktische Übungen ein. Dabei wird mit Eclipse eine für die industrielle Praxis hoch relevante Anwendungsplattform genutzt. Man lernt somit auch, wie große bestehende Systeme am Beispiel Eclipse verstanden und gezielt erweitert werden können.

2 Organisation der Lehrveranstaltung

An den ersten Veranstaltungsterminen erfolgt zunächst eine fundierte Einführung in die Eclipse Plattform und ihrer Erweiterungsmöglichkeiten. Darauf aufbauend werden die Anforderungen und die Architektur von Reclipse definiert. Im Anschluss daran erfolgt die Implementierung des Plugins, aufgeteilt in verschiedenen Teams.

3 Zeitplan

Folgender Zeitplan ist für die Vorlesung vorgesehen:

3.1 Termin 1 (KW 17): Die Eclipse Plattform

Inhalte:

- Die Eclipse Plattform: Übersicht, Architektur, Erweiterbarkeit
- Entwicklung von Plugins mit dem Eclipse Platform Development Toolkit (PDT)

Quellen:

- [Cla06]

3.2 Termin 2 (KW 18): Oberflächenprogrammierung auf Basis Eclipse

Inhalte:

- SWT: Das Standard Widget Toolkit zur Oberflächenprogrammierung
- JFace: Model-View-Controller für SWT
- Forms API: Programmierung von Formulareingaben

Quellen:

- SWT Homepage: <http://www.eclipse.org/swt>
- JFace Wiki: <http://wiki.eclipse.org/JFace>
- Einführung in die Forms API: <http://www.eclipse.org/articles/Article-Forms/article.html>, <http://www.eclipse.org/articles/article.php?file=Article-Forms33/index.html>
-

3.3 Termin 3 (KW 19): Die Eclipse Rich Client Platform

Inhalte:

- Architektur und Konzepte der Eclipse Rich Client Platform (RCP)
- Die Eclipse Workbench und ihre Extension Points

Quellen:

- Rich Client Platform Wiki: http://wiki.eclipse.org/index.php/Rich_Client_Platform
- Eclipse Platform Developers Guide: <http://help.eclipse.org/help33/nav/2>

3.4 Termin 4 (KW 20): Das Eclipse Modeling Framework (EMF)

Inhalte:

- Definition, Generierung und Instanziierung einer Modellhierarchie über EMF/ecore
- Zugriff auf EMF Modelle aus Java Quellcode
- Prüfung und Abfrage von EMF Modellen mit EMF Validation und EMF Query

Quellen:

- EMF Homepage: <http://www.eclipse.org/modeling/emf>
- EMF Wiki: <http://wiki.eclipse.org/EMF>
- EMF Dokumentation: <http://www.eclipse.org/modeling/emf/docs/>
- [Ste03]
- [Ste08]

3.5 Termin 5 (KW 21): Konstruktion

Auf Basis eines vorliegenden Anforderungskatalogs wird die Softwarearchitektur von Reclipse erarbeitet. Ziel dabei ist es, das System in sinnvolle Komponenten aufzuteilen und deren Schnittstellen untereinander und zur Eclipse Plattform zu definieren. Ferner sollen Erweiterungsschnittstellen für das Plugin konstruiert werden.

3.6 Termin 6 bis 13 (KW 22-29): Realisierung

Entwicklung des Plugins in verschiedenen Teams. Die Aufteilung der Teams erfolgt entsprechend den indentifizierten Komponenten der Softwarearchitektur. Die Realisierungsphase ist in mehrere Iterationen aufgeteilt, an deren Ende eine ausführliche Abschlussbesprechung stattfindet. Jede Iteration besteht aus einer eigenständigen Analyse-, Design-, Planungs-, Entwicklungs- und Test/Integrationsphase. Im Laufe der Realisierungsphase werden gängige Werkzeuge der Softwareentwicklung verwendet:

- IDE und Plattform: Eclipse 3.3
(<http://www.eclipse.org>)
- Modellierung: MID Innovator oder Borland Together
- Versionskontrolle: Subversion
(<http://subversion.tigris.org>)
- Projektmanagement: Thoughtworks Mingle
(<http://http://studios.thoughtworks.com/mingle-project-intelligence>)
- Qualitätssicherung: FindBugs, JUnitFactory
(<http://findbugs.sourceforge.net/>, <http://www.junitfactory.com/>)

Literatur

- [Cla06] Dan Clayberg, Eric Rubel. *Eclipse: Building Commercial-Quality Plugins*. Addison Wesley Professional, second edition edition, 3/2006 2006.
- [Ste03] Frank Merks Ed Steinberg, Dave Budinsky. *EMF - Eclipse Modeling Framework*. Prentice Hall International, 1st edition, 10/2003 2003. ISBN-10: 0131425420 ISBN-13: 978-0131425422.
- [Ste08] Dave Steinberg. Fundamentals of the eclipse modeling framework. In Eclipse Foundation, editor, *eclipsecon 2008*, 2008.