Advanced Software Engineering

FOSS – Free and Open Source SW and the Eclipse Ecosystem

DI Dr. techn. Mario Bernhart

About myself

- Researcher at Vienna University of Technology
- In 2012 researcher at MIT Aero/Astro
- Eclipse project lead and committer (Mylyn Reviews)
- 6 years in software engineering for air traffic
  - 2000-2005 Electronic Flight Strips (EFS) at Frequentis AG
- 3 years in software engineering for airport operations
  - 2010-2012 AODB for Vienna International Airport

- Contact: mario.bernhart@inso.tuwien.ac.at

Preview: ASE series of 5 lectures

- Example Project: Vienna International Airport AODB Core System
- Release your stuff 3 times a day
  - Dependency Management
  - Build Management and -Automation
  - Continuous Integration, Continuous Delivery
- Five challenges you solve for every project
  - Error Management
  - Transaction Management
  - Logging
  - Auditing
  - Declarative Authentication and Authorization
- Build for ten years and more
  - Layered Software Design / API Design
  - Modularization / Service Design
  - Decoupling / Event Driven Design
  - Interfacing / Integration
- From prototype to product (make it work 24/7)
  - Clustering
  - Performance
  - Monitoring
  - Automating Operational Tasks

What is FOSS? (def. by Bruce Perens) I

- Free Redistribution
  - The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.
- Source Code
  - The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.
- Derived Works
  - The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

What is FOSS? (def. by Bruce Perens) II

- Integrity of the Authors Source Code
  - The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
- No Discrimination against Persons or Groups
  - The license must not discriminate against any person or group of persons
- No Discrimination against Fields of Endeavor
  - The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.
- Distribution of License
  - The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

What is FOSS? (def. by Bruce Perens) III

- License must not be specific to a Product
  - The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.
- License must not restrict other Software
  - The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.
- License must be Technology-Neutral
  - No provision of the license may be predicated on any individual technology or style of interface.

FOSS Organisations

- Free Software (Free Software Foundation – FSF)
  - Philosophical model: Free as in freedom (not free beer)
  - Usually copyleft: all rights to the user
  - Major licence: GPL Licence (strong copyleft)
- Open Source Software (Open Source Initiative - OSI)
  - Technical model: Access to sourcecode
  - Usually permissive: all rights to the contributor
  - Major licence: BSD Licence (commercial friendly)
- Philosophical dispute, but prominent coops e.g. GNU/Linux
  - Licenses approved by FSF alone: 51(by 2007)
  - Licenses approved by OSI alone: 43
  - Licenses approved by both: 16

Cathedral and the Bazaar

- Essay by Eric. S. Raymond with two different free software development models:
  - The **Cathedral model**, in which source code is available with each software release, but code developed between releases is restricted to an exclusive group of software developers. E.g. Emacs and GCC
  - The **Bazaar model**, in which the code is developed over the Internet in view of the public. Raymond credits Linus Torvalds, leader of the Linux kernel project, as the inventor of this process.
- Linus' Law: "given enough eyeballs, all bugs are shallow”

State of the art: Code reviews in Androi

Gerrit

- Code review system for GIT
- Based on Google Mondrian

FOSS cathedral model:

- Google develops latest version
- Released state is open source

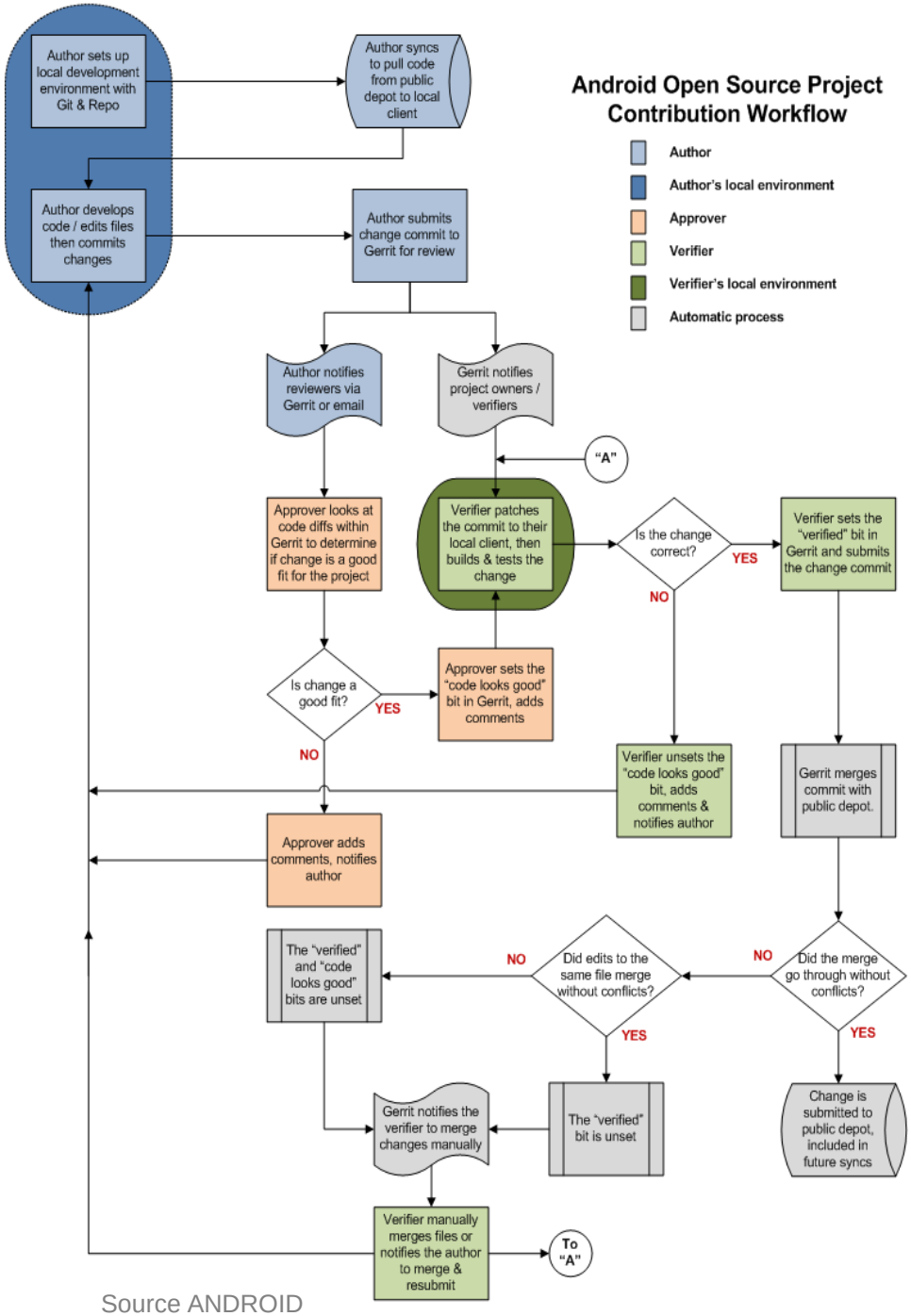
Manual approver

- Code review with voting

Automatic verifier

- Build Server

Eclipse Integration via Mylyn



Source ANDROID

FLOSS Business Models I

- Sell services: Customization, Integration, Training and Documentation
  - e.g. Books
- Sell Extensions/Components/Features
  - e.g. Tasktop connectors
- Forking (commercial) and porting-back with delay
  - e.g. Eclipse Mylyn and Tasktop
- Forking (commercial) and financing maintainance
- Dual License models
  - Copyleft and commercial friendly License combined (e.g. MySQL)

FLOSS Business Models II

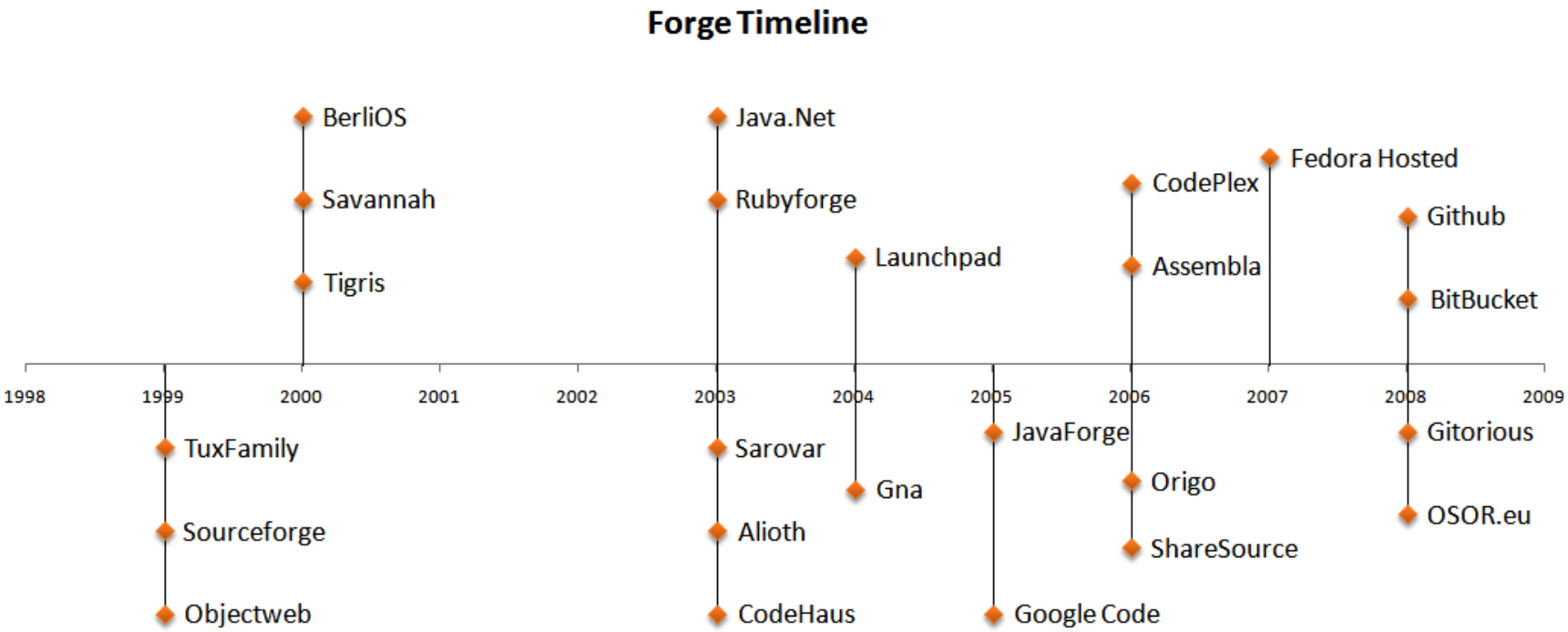
- Publicity for other software or services
- Personal publicity of the author and career opportunities
- Sponsorship and donations
- Academic research e.g. kerberos, findbugs, ...

FOSS Communities

- Large projects are usually managed by a separate legal entity e.g. Eclipse Foundation
- Communication (usually completely open)
  - Website, Mail, Newsgroup, Wiki
  - Blogs and Twitter
  - Conference calls
  - Conferences, BoFs
  - Local meetups and local usergroups
- Infrastructure (usually forge or sponsored)
  - Code repository
  - Bug repository
  - Common integration and build infrastructure

FOSS Forges

- Forges provide the most common infrastructure for the projects
- Forge metadata from http://flossmole.org/



Example: Eclipse

- 10+ years since the start of the Eclipse open source project
- 273 open source projects at Eclipse.org
- 1057 committers located around the world
- more than half in Europe
- 50+ million lines of code across all Eclipse projects
- 174 member companies of the Eclipse Foundation

Eclipse Cornerstones I

- Eclipse has an estimated **65% market share in the Java IDE** space and over 6 million users. It has been instrumental in the worldwide success and adoption of Java itself.
- The **Eclipse C/C++** IDE (CDT) has become the **de facto standard** developer IDE in the embedded and real-time operating system market.
- Eclipse Modeling: majority of **UML modeling tools**
- Eclipse Mylyn: over 70 Mylyn connectors
- An annual Eclipse Release Train for the last **eight years**

Eclipse Cornerstones II

- Indigo: **62 project teams**, 408 developers, 49 organizations, and **46 million LOC**.
- The Eclipse ecosystem has millions of individuals, thousands of companies and thousands of universities

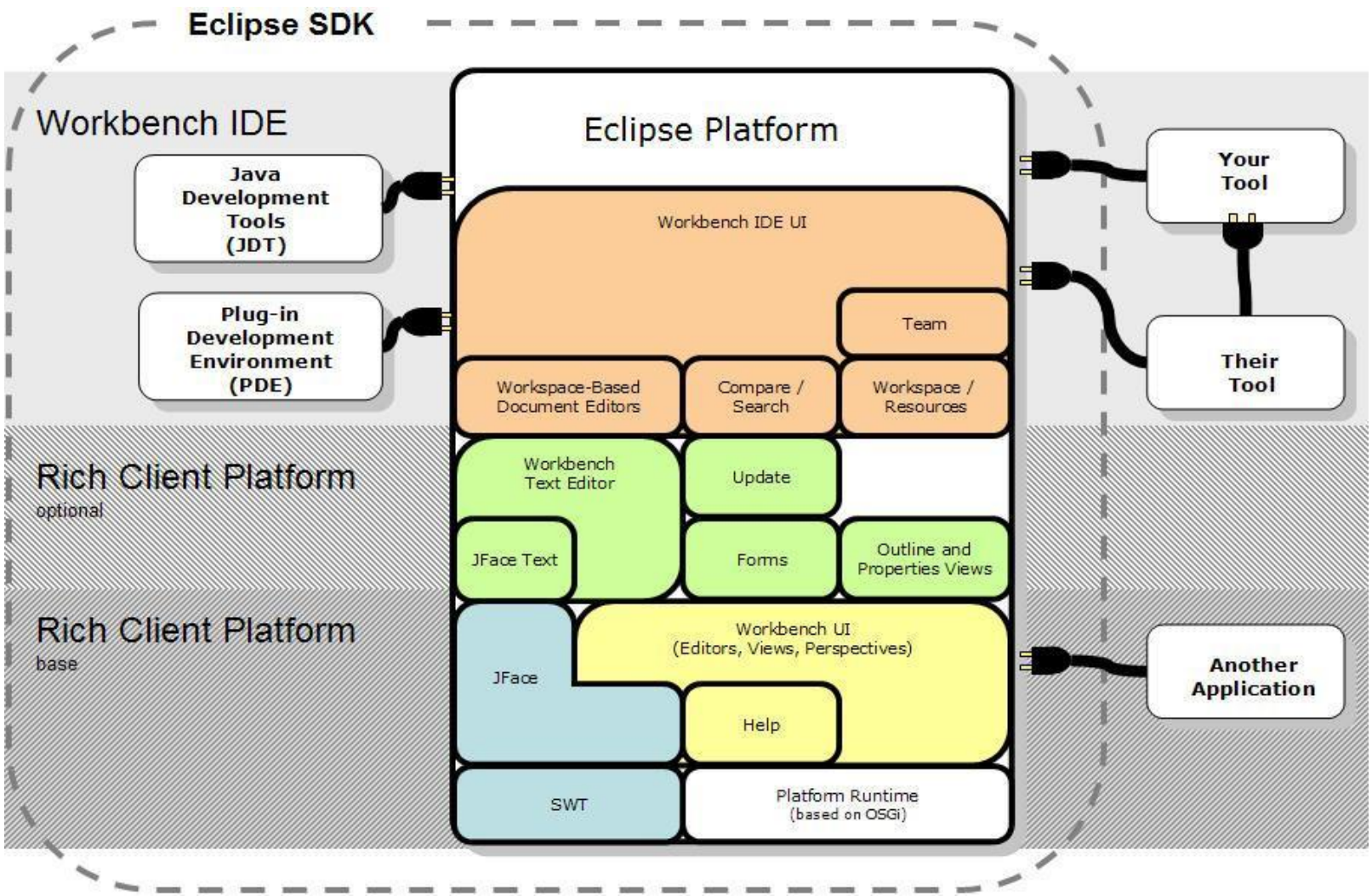
Eclipse Organization

- Eclipse Foundation is funded by its members and responsible for:
  - IT Infrastructure
  - IP Management
    - EPL
  - Development Process
    - Eclipse Development Process (EDP)
  - Ecosystem Development
    - EclipseCon, EclipseCon Europe, local Democamps
- The 3 Eclipse Communities
  - Contributors and Committers
  - Users
  - Adopters

Eclipse Councils

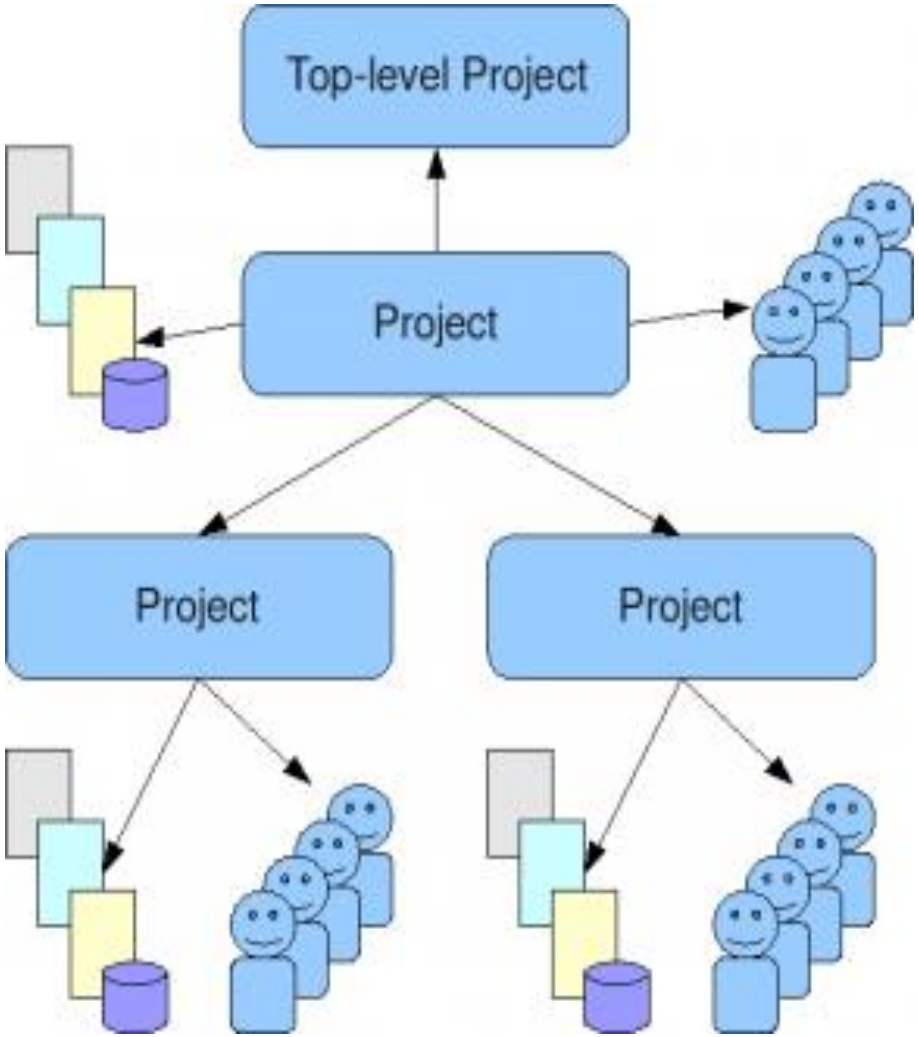
- Planning
  - The Planning Council is responsible for establishing a coordinated Platform Release Plan that supports the Roadmap, and balances the many competing requirements. The Platform Release Plan describes the themes and priorities that focus these Releases, and orchestrates the dependencies among Project Plans.
- Architecture
  - The Architecture Council is responsible for (i) monitoring, guiding, and influencing the software architectures used by Projects, (ii) new project mentoring, and (iii) maintaining and revising the Eclipse Development Process subject to the approval of the Board. More explanation of the Architecture Council can be found in the Eclipse Development Process and in the guidelines and checklists for the Architecture Council.

Eclipse Plug-In Architecture



Eclipse Project Structures

- Eclipse top level projects
  - Eclipse
    - JDT
    - PDE
  - BIRT
  - Datatools
  - Modelling
  - Mylyn
  - RT
  - SOA
  - Technology
  - Tools
  - TPTP
  - WTP

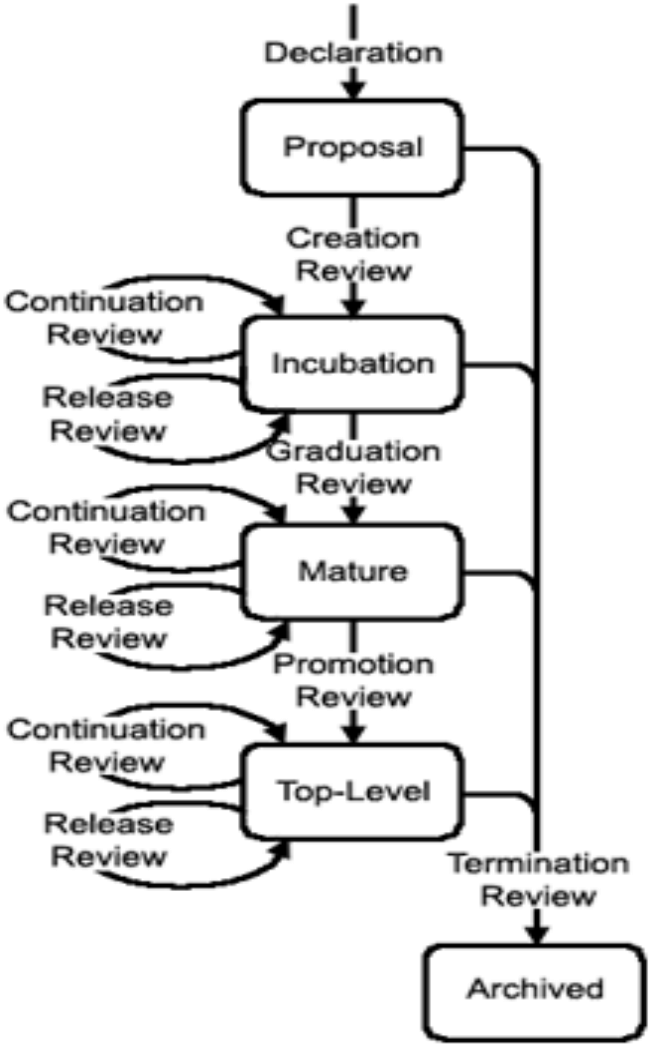


Eclipse Top-Level PMC

- Top-level projects are managed by a Project Management Committee (PMC).
- A PMC has one or more PMC Leads and zero or more PMC Members.
- Together the PMC provides oversight and overall leadership for the projects that fall under their top-level project.

Eclipse Project Lifecycle

- Preconditions for a Creation Review:
  - Proposal document is complete
  - Top-level project along with any containing (i.e. second-level project) is indicated
  - Project has a well-defined scope
  - Project lead is identified
  - Biographical information (as it relates to the project) is provided for each initial committer
  - Two Architecture Council Mentors are identified
  - PMC approval has been obtained
  - Trademark assignment is complete
  - All questions have been satisfactorily answered in the communication channel
  - Qualitative requirements are satisfactorily addressed



Eclipse Public License (EPL)

- Weak-copyleft business friendly free licence:
  - According to article 1(b) of the EPL, additions to the original work may be licensed independently, including under a commercial license, provided such additions are "separate modules of software" and do not constitute a derivative work. Changes and additions which do constitute a derivative work must be licensed under the same terms and conditions of the EPL, which includes the requirement to make source code available.
  - Not compatible with GPL (!)
  - Rigorous IP process: Full IP Log required for a Release Review

Example Eclipse Mylyn

- Mylyn is a Task-Focused Interface for Eclipse
- Integrating rich and offline editing for repositories such as Bugzilla, Trac, and JIRA
- ALM tools integration platform
- Presented at EclipseCon 2005
- Top level Eclipse project promotion 2010
- Mylyn Subprojects:
  - Mylyn Build, Mylyn Commons, Mylyn Context
  - Mylyn Docs, Mylyn Incubator, Mylyn Reviews
  - Mylyn Tasks, Mylyn Versions
- Commercial version: Tasktop

Eclipse Mylyn

Java - Attach error log entries into new bug reports - Eclipse SDK

File Edit Navigate Search Project Run Window Help

Package Exp | Hierarchy

> org.eclipse.mylyn.bugzilla.core [dev
src
org.eclipse.mylyn.internal.bugz
BugzillaAttributeMapper.jav
BugzillaAttributeMappe
BugzillaClient.java
BugzillaClient
postAttachment(St
BugzillaClientFactory.java
BugzillaClientFactory
taskRepositoryLocat
org.eclipse.mylyn.bugzilla.ui [dev.ecl
src
org.eclipse.mylyn.internal.bugz
BugzillaUpdateAttachment
BugzillaUpdateAttachm
currentSelection
init(IViewPart)

BugzillaAttributeMapper.java | *216640: Attach error log entries into new bug r

Bug 216640 Eclipse

Attach error log entries into new bug reports

P5 Status: NEW Bug: 216640 Opened: Jan 25, 2008 Modified: Jan 29, 2008 2:32 AM

Attributes

Attachments (0)

Description

Comments (1)

1: Mik Kersten, Jan 29, 2008 1:54 AM

New Comment

Looking into this now, see also related bug 124224.

Actions

Leave as NEW
Accept (change status to ASSIGNED)
Resolve as FIXED
Duplicate of
Reassign to m
Reassign to defau
Submit

mik.kersten@tasktop.com
mylyn-inbox@eclipse.org

People

Assigned to: Mylyn Inbox <mylyn-inbox@e
Reporter: Willian Mitsuda <wmitsuda@g
QA Contact:
Add CC:

jacek.pospychala@pl.ibm.com
mik.kersten@tasktop.com

(Select to remove)

Planning | Context | Bugzilla

Task List

Find: All Allow to ed...

Mylyn - Mik [Bugzilla]
162335: [context] Allow "Make Landi
267524: create Mylyn 3.2 release plar
Mylyn - Rob [Bugzilla]
124224: Allow to edit priorities in
Project - Steffen [Trac]
83: filter sort results
84: upgrade ws api
5: address web site nits
Target - Rob [Atlassian]
PLE-290: logged in as wrong user wh
PLE-327: Documentation for Eclipse
PLE-172: Create Crucible review with
PLE-238: add an action to a cha

Synchronize

Change Sets for CVS (Workspace)

Allow to edit priorities in the task list
> org.eclipse.mylyn.bugzilla.core [d
streamline task attachments

State of the art: Code reviews in Androi

Gerrit

- Code review system for GIT
- Based on Google Mondrian

AOSP cathedral model:

- Google develops latest version
- Released state is open source

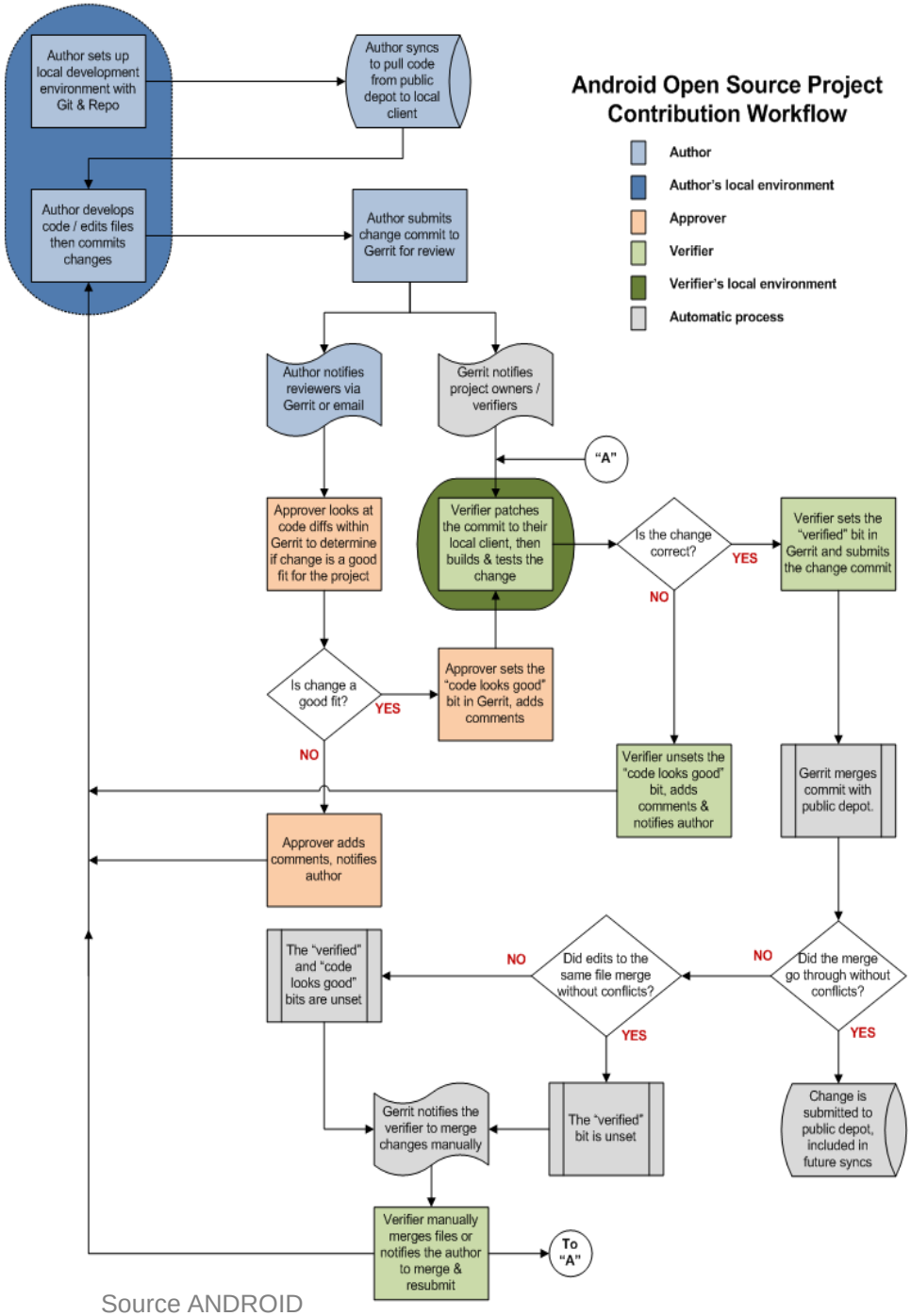
Manual approver

- Code review with voting

Automatic verifier

- Build Server

Eclipse Integration via Mylyn



Source ANDROID

CR important code review tools projects

Many projects at Google and others (i.e. open source) doing continuous reviews manually e.g. via email or bugzilla

2001 CodeStriker

2006 Google Mondrian

2008 ReviewClipse (project lead)

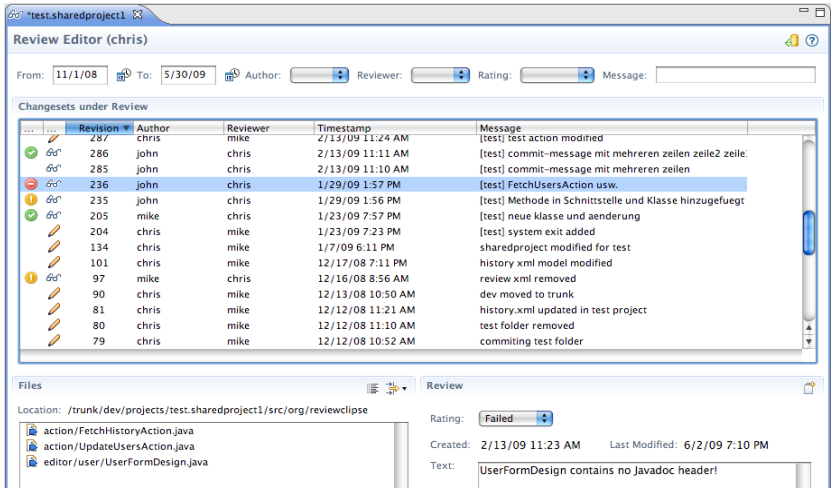
2009 Google Gerrit (e.g. for Andriod)

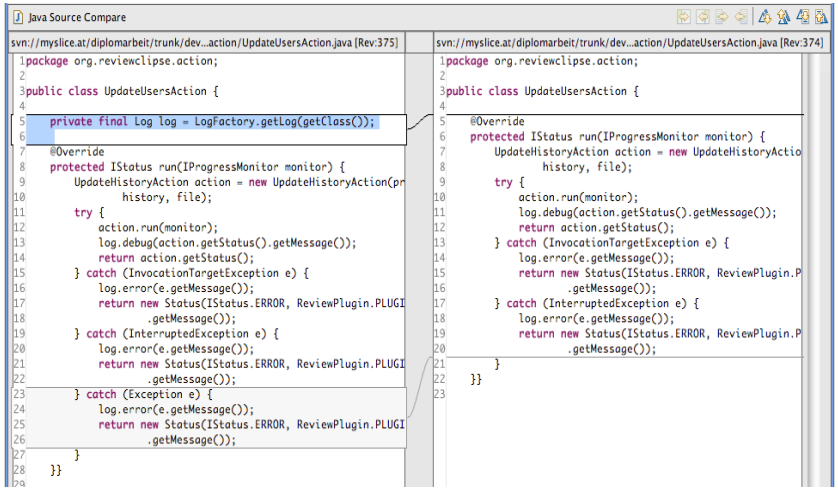
2010 Eclipse Mylyn for Frequentis (project lead)

2011 Eclipse Mylyn Gerrit Connector (project lead)

ReviewClipse 2008; IDE-integrated review tool

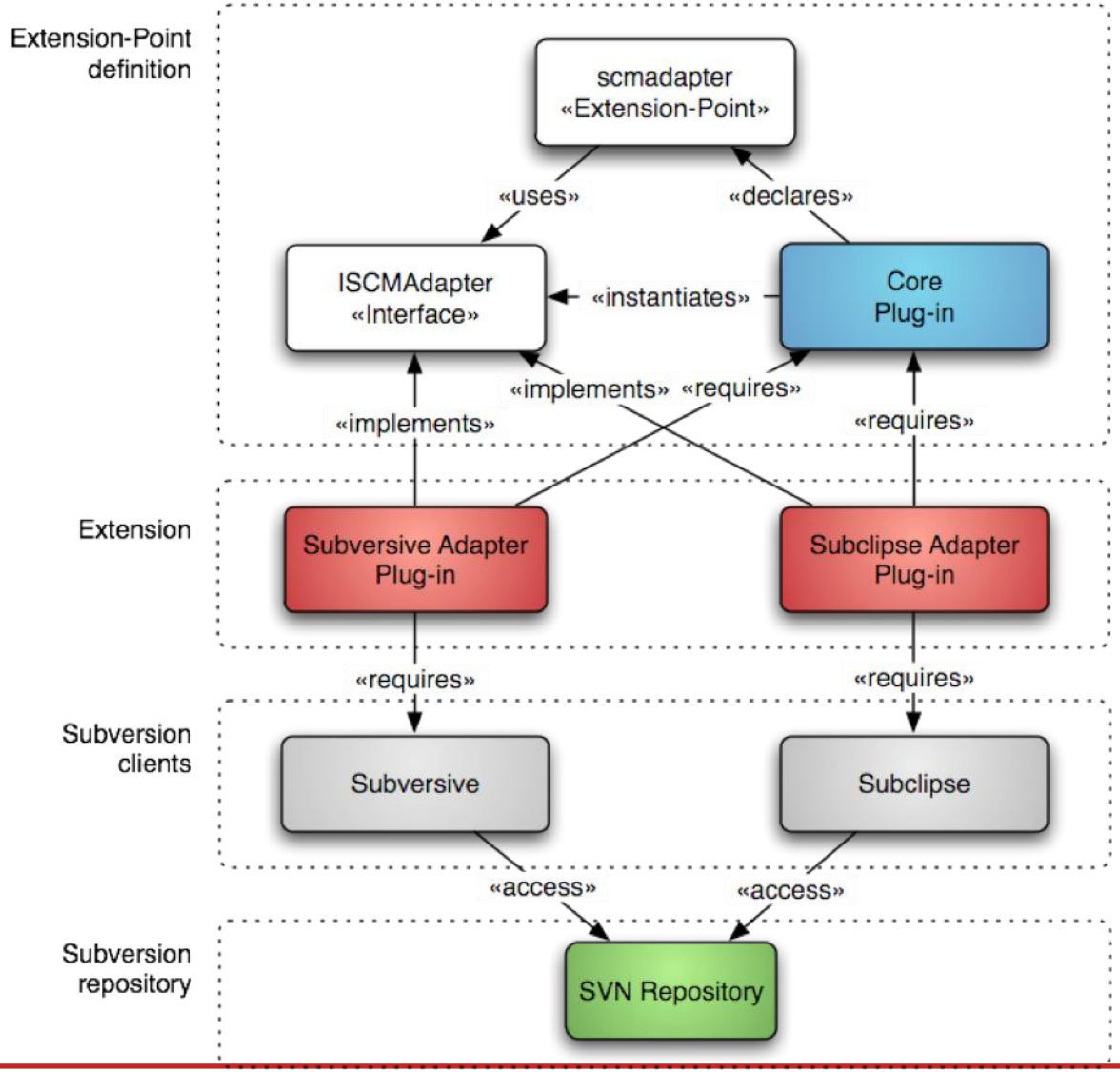
- Eclipse integrated easy to use UI
- Review scope = one changeset in the SVN
- No additional server (e.g. classroom use)
- Works offline (XML/SVN based)
- Works with Subversive and Subclipse
- Flexible filtered review assignments e.g. *Test*.java
- Create Mylyn tasks out of (bad) reviews
- Freeware , 10.000+ downloads (by 2009)
- www.inso.tuwien.ac.at/projects/reviewclipse/





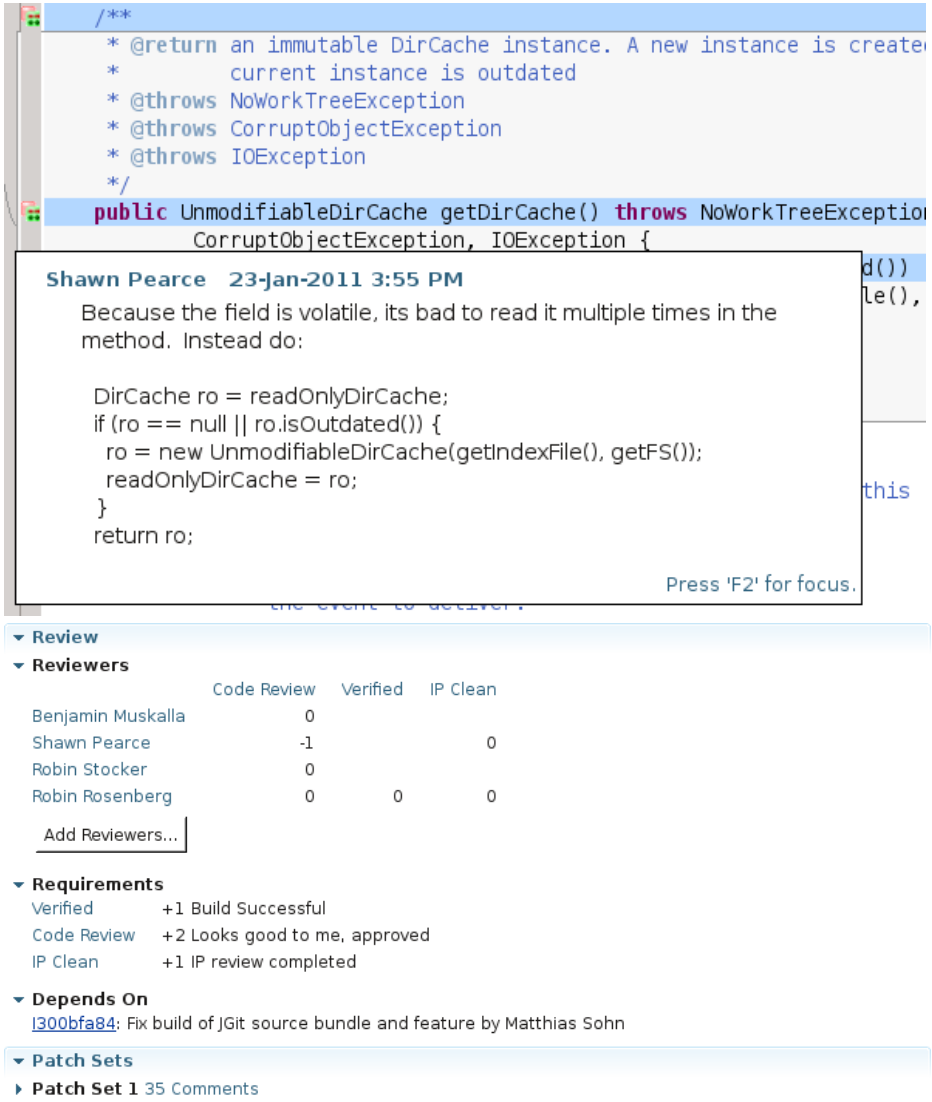
Source INSO [9]

ReviewClipse design

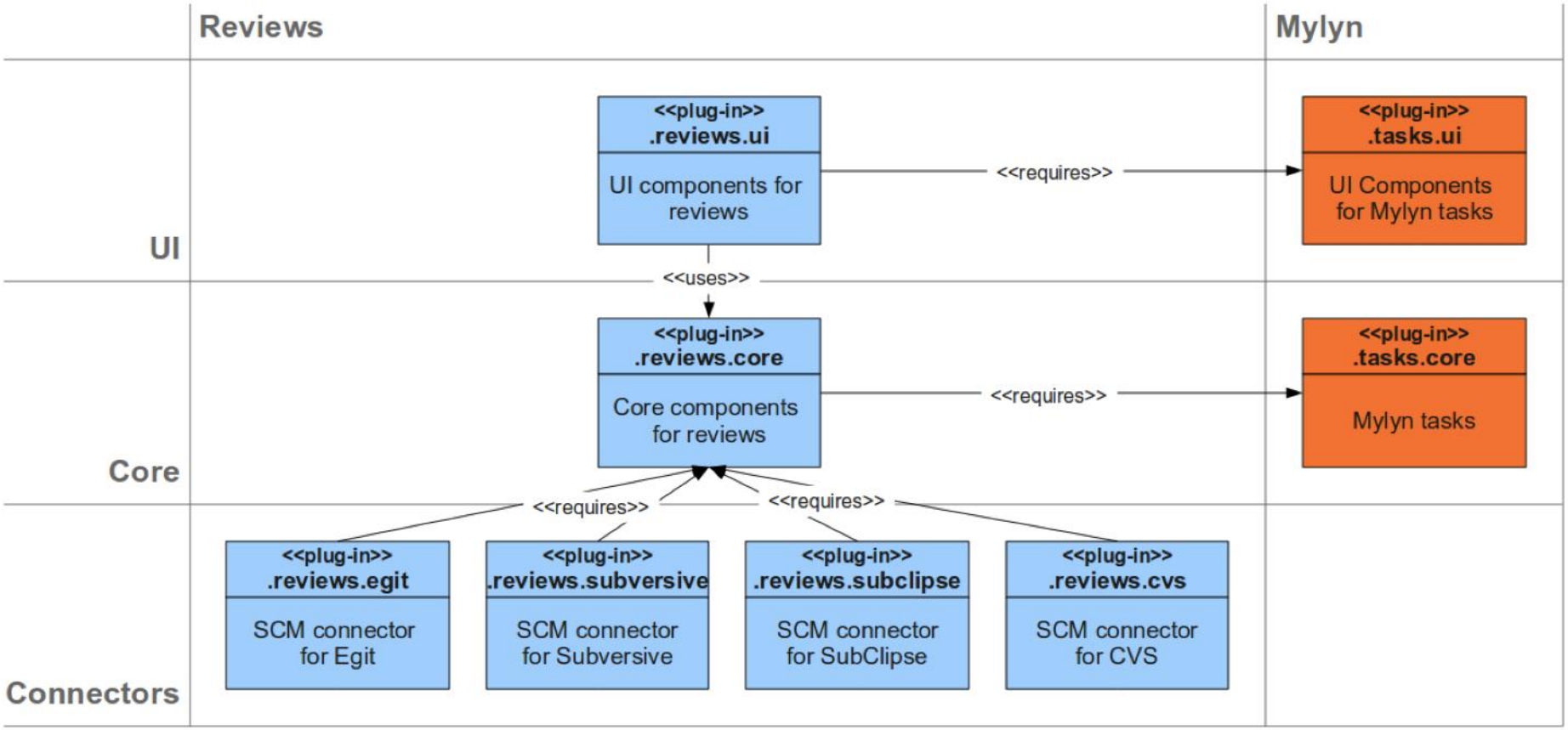


Eclipse Mylyn Reviews

- Eclipse is the defacto-standard IDE for Java
- ReviewClipse has been contributed to Eclipse
- Mylyn Reviews is a code review plugin that integrates with Eclipse Mylyn and GIT
- Provides client integration (e.g. inline editing) for the development environment
- Uses Gerrit as review server
- 14 committers from INSO, Twitter, SAP, Google, Tasktop, Ericsson, …
- www.eclipse.org/reviews



Mylyn Reviews plugin design



Common prefix for all plug-ins org.eclipse.mylyn!

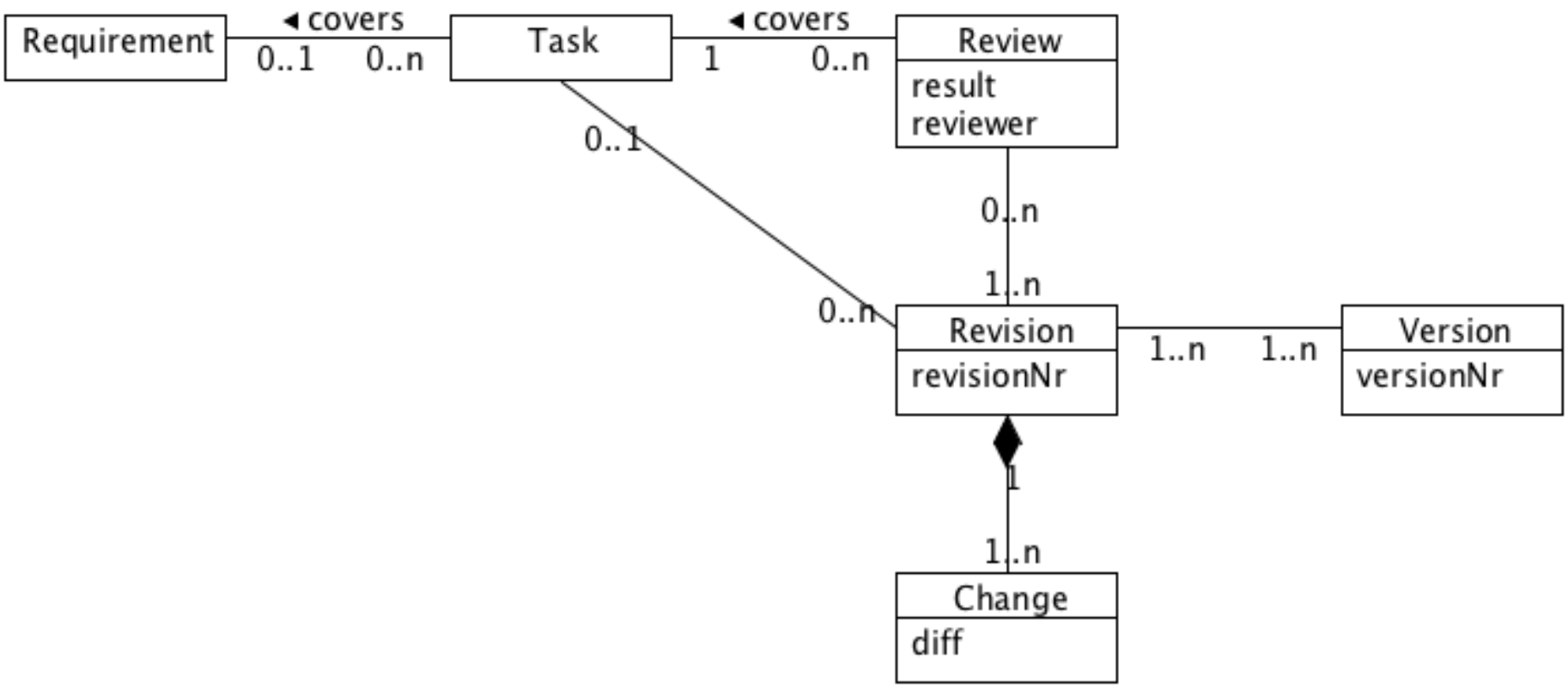
Mylyn Reviews for Frequentis AG

- Development of safety-critical communication- & information-systems
- Located in Vienna/Austria, 1000+ employees
- Adoption of Mylyn Reviews for SW-Dev-unit to support the RTCA DO278B

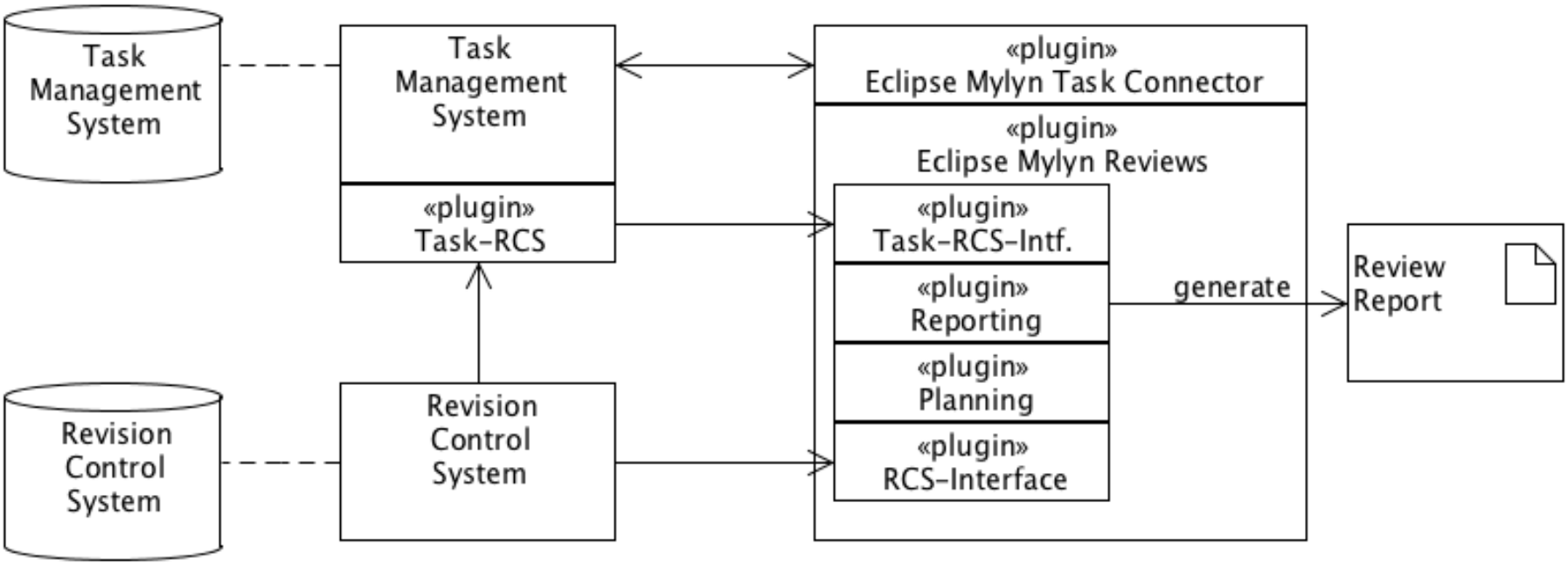
RTCA DO-278ED109 requirements

- For Communication, Navigation, Surveillance and Air Traffic Management Systems (CNS/ATM)
- DO-278/ED109 provides guidance for determining that an acceptable level of confidence has been achieved with regards to the software of airborne systems.
  - Each procedure of a review, analysis and test can pass or fail. It has to be possible to determine the (1) result of each procedure. Furthermore, it has to be possible to determine the final pass/fail result for each activity.
  - The software (2) version of a particular reviewed, analyzed or tested software item needs to be identifiable.
  - Results of tests, reviews and analyses, including (3a) coverage analyses and (3b) traceability analyses need to be available.

Frequentis review model



Frequentis Mylyn Reviews Plugin Architecture



Frequentis Mylyn Reviews Screenshot

Java - Refactor data model - Eclipse Platform

File Edit Navigate Search Project Reviews Run Window Help

373: Refactor data model

Task 373 Synchronize to update editor contents demo_repository Submit

Review-Summary

Review-Id	Revisions	Author	Reviewer	Result	Comment
374	179, 178	sreiterer	sue	? NONE	
375	178, 177	sreiterer	bill	? NONE	
376	179, 177	sreiterer	john	PASSED	ok, looks fine
378	178	sreiterer	phil	WARNING	missing source code comments

Remove Changeset(s)

Changesets / Patches

Revis...	Description
179	#373 - phone number added
178	#373 - removed continent
177	#373 - removed country from city

Create review

Context Mantis Reviews

Create Review for selected Changeset(s)

Create new review task

Assigned to: stefan find Reviewer...

Assign changeset to an existing review task

Task Id	Changesets	Assigned To
374	179, 178	sue
375	178, 177	bill
376	179, 177	john
378	178	phil

OK Cancel

Next lecture preview

Mark Struberg

Firma: INSO, TU Vienna

Mark Struberg ist Softwarearchitekt mit über zwanzig Jahren Programmiererfahrung. Er arbeitet seit 1996 mit Java und ist aktiv in Open-Source-Projekte im Bereich Java und Linux involviert. Mark ist Apache Software Foundation Member und PMC bei Apache OpenWebBeans, MyFaces, DeltaSpike und vielen anderen Apache-Projekten. Als Java Expert Group Member arbeitet er aktiv an der CDI und anderen EE-Spezifikationen mit. Er arbeitet unter anderem für die Research Group for Industrial Software (INSO) der TU Wien.