



the
POWER
of
JAVA™



Inside Eclipse Callisto: The New, the Useful, and the Lessons Learned

Mike Milinkovich and Bjorn Freeman-Benson

Executive Director and Director, Open Source Process
Eclipse Foundation

<http://www.eclipse.org/>

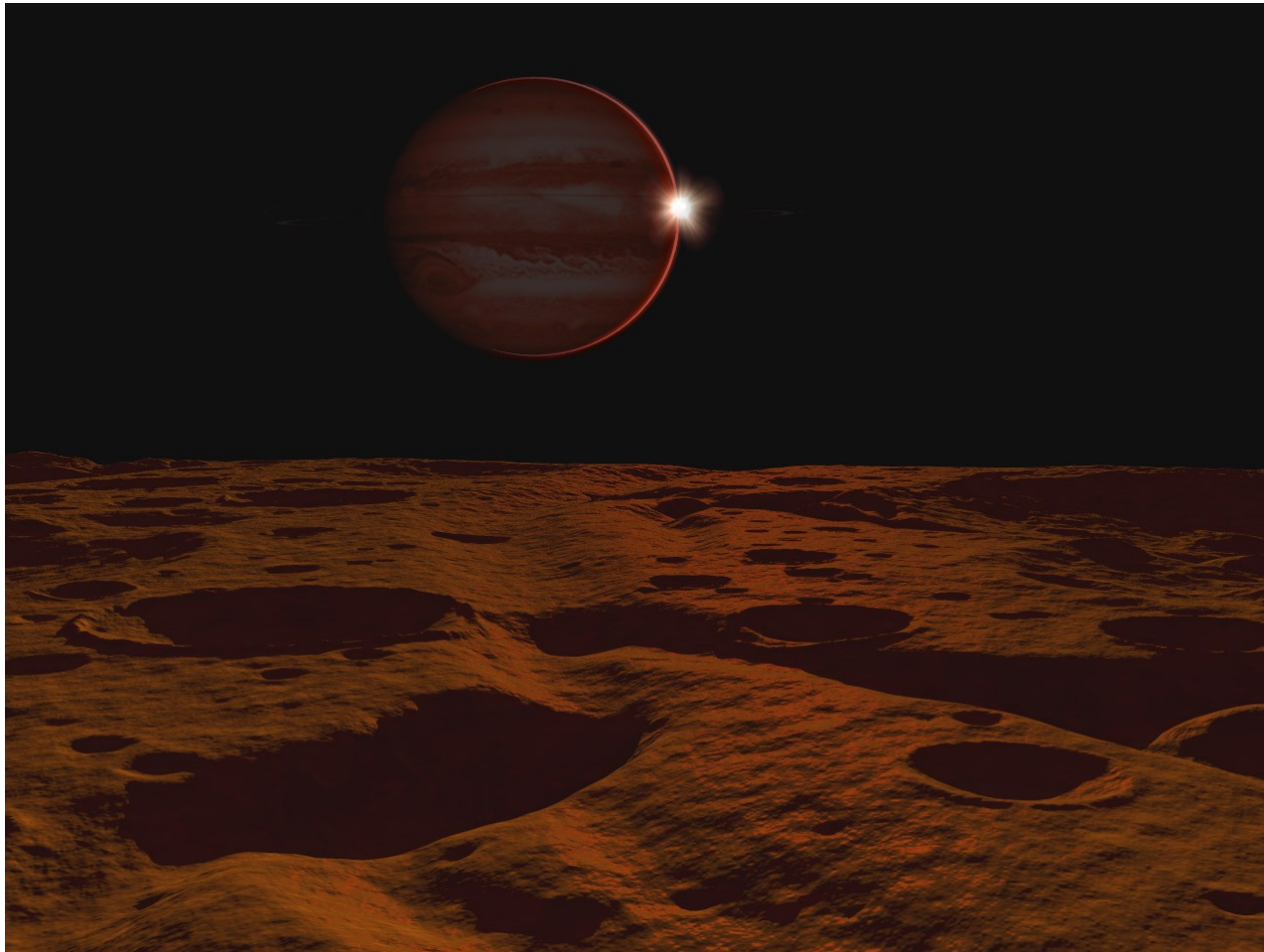
TS-4311

What Is Eclipse Callisto?

And what's in it for me?

Learn new features and frameworks in Eclipse Callisto; learn how we are putting Callisto together

Eclipse of Jupiter Seen from Callisto



<http://www.arcadiastreet.com/cgvistas/>

Eclipse Callisto Simultaneous Release

What Is Callisto?

Why Is the Eclipse Community Building Callisto?

New Feature Highlights

New Java™ Technology Development Features

Lessons Learned

Why Is It Named Callisto?

Eclipse Callisto Simultaneous Release

What Is Callisto?

Why Is the Eclipse Community Building Callisto?

New Feature Highlights

New Java™ Technology Development Features

Lessons Learned

Why Is It Named Callisto?

What Is Callisto?

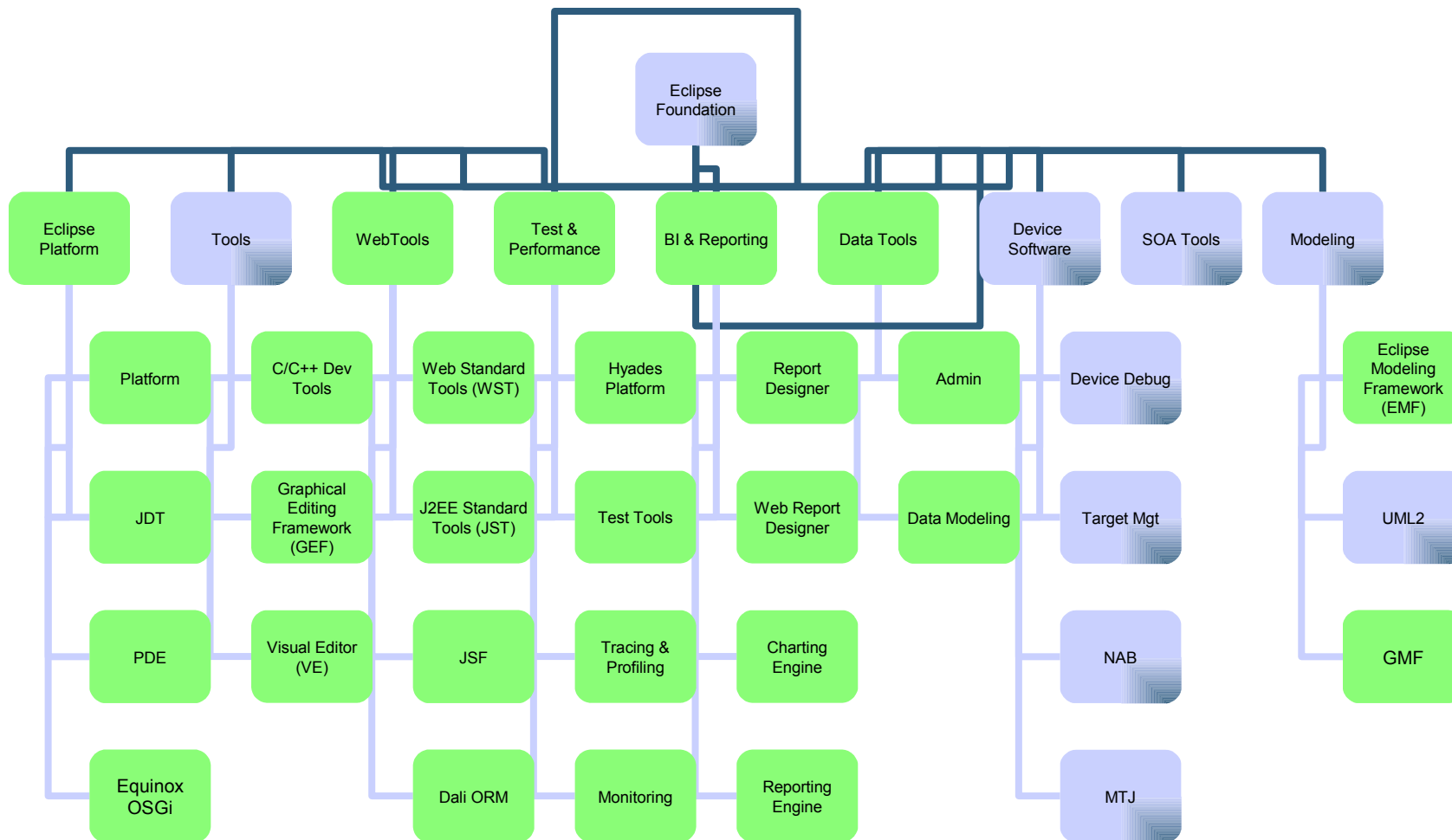
- **Officially:** The simultaneous release of ten major Eclipse projects
- **Simply:** To make it easier for you to adopt and use Eclipse technology
- **Larger Picture:** New features and new APIs from all the Eclipse projects
- **ISVs:** Eliminate the “which versions of which projects” question

What Is Callisto?

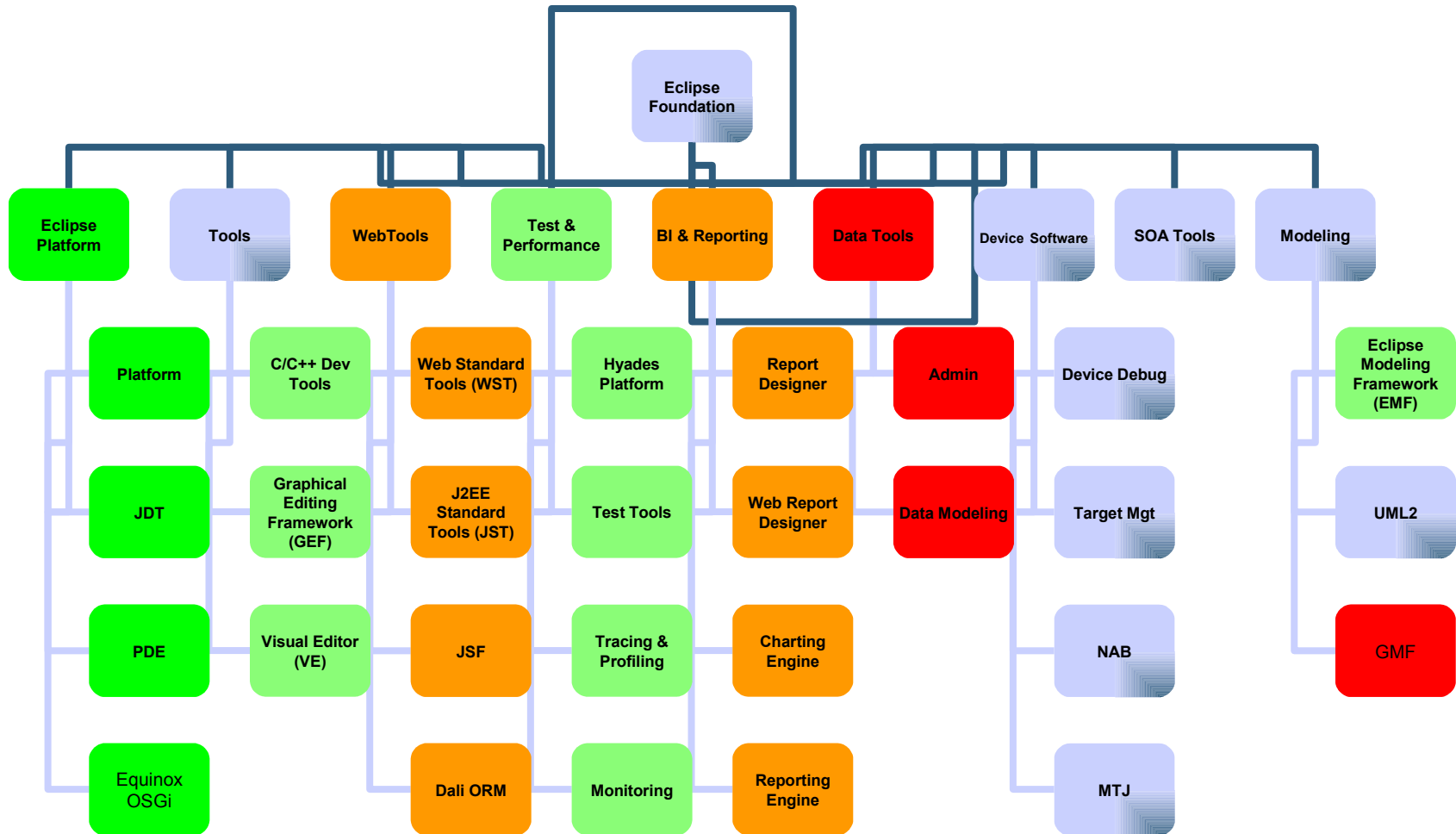
- (BIRT) Business Intelligence and Reporting
- (CDT) C/C++
- (DTP) Data Tools
- (EMF) Eclipse Modelling Framework
- (GEF) Graphical Editor Framework
- (GMF) Graphical Modelling Framework
- (Eclipse) Platform and Java Technology
- (TPTP) Test and Performance
- (WTP) Web Tools
- (VE) Visual Editor

<http://www.eclipse.org/projects/callisto.php>

Callisto Projects



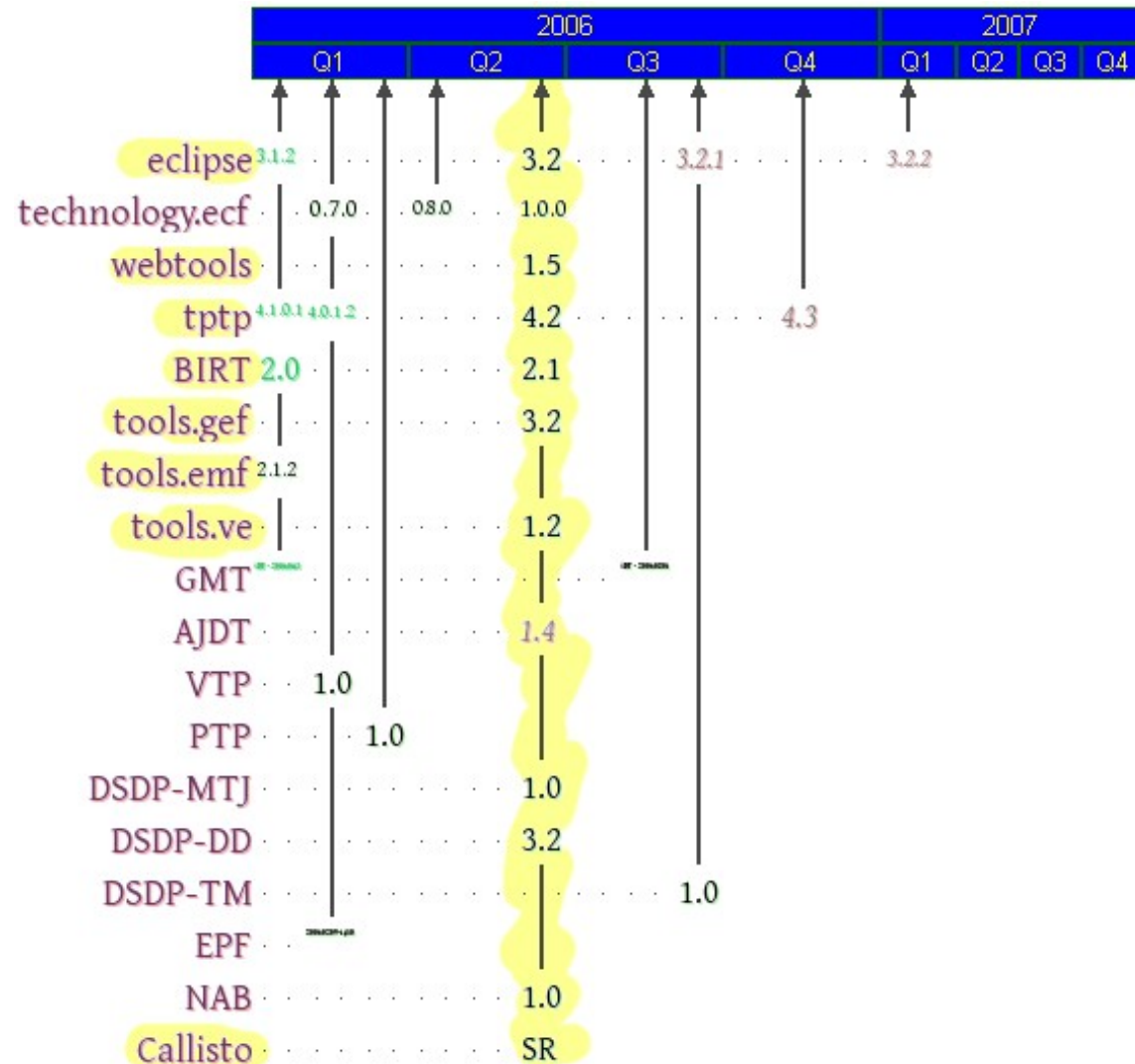
Callisto Projects: Maturity



Synchronized

All aboard!

- Synchronized versus Simultaneous
- Most projects want to join



What Is Callisto **Not**?

- It is not a single product
- It is not a seamless download experience
- But these are good ideas and there is always next year...

Eclipse Callisto Simultaneous Release

What Is Callisto?

Why Is the Eclipse Community Building Callisto?

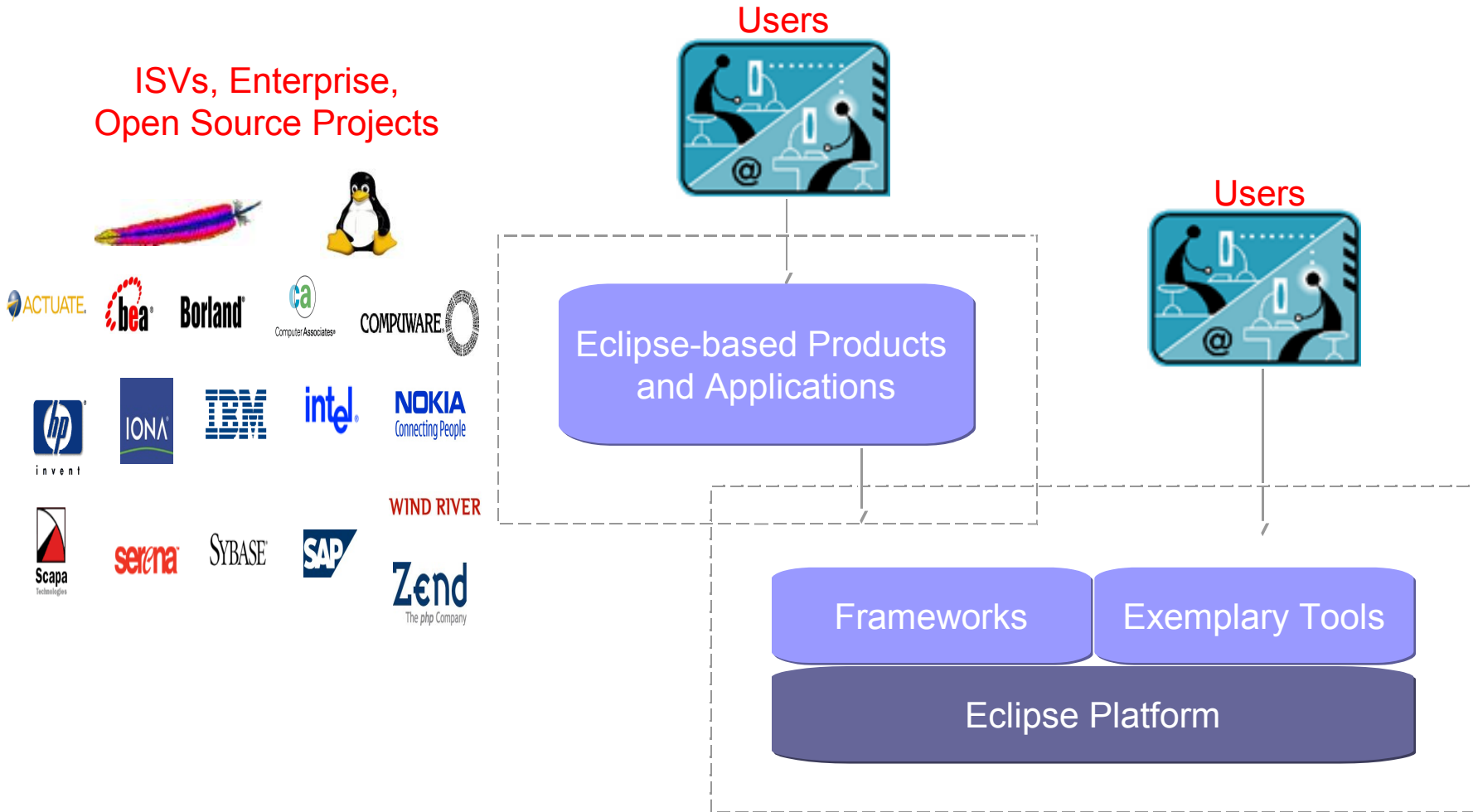
New Feature Highlights

New Java Technology Development Features

Lessons Learned

Why Is It Named Callisto?

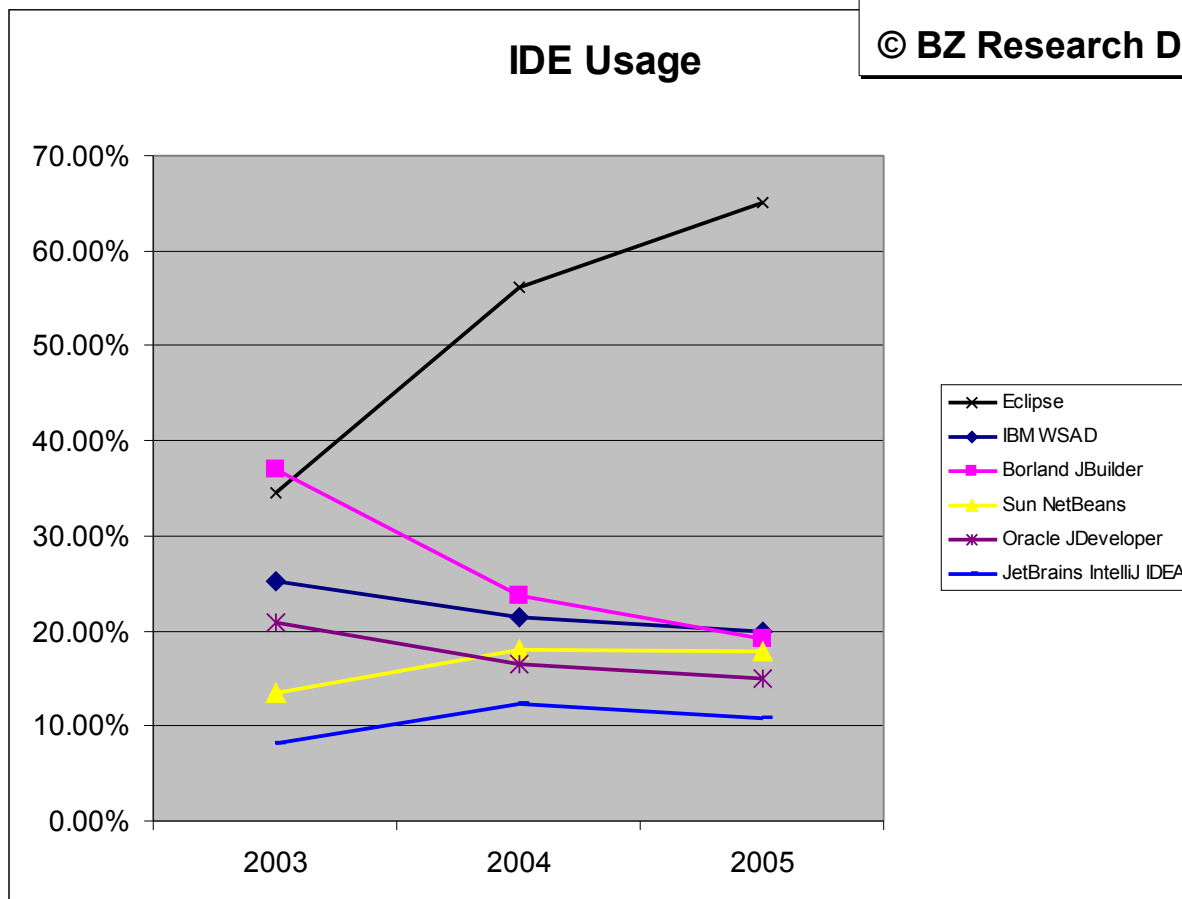
Measuring Success for Eclipse



How Is Eclipse Doing?

Java Use and Awareness Study

© BZ Research December 2005



So We're Doing OK, But...

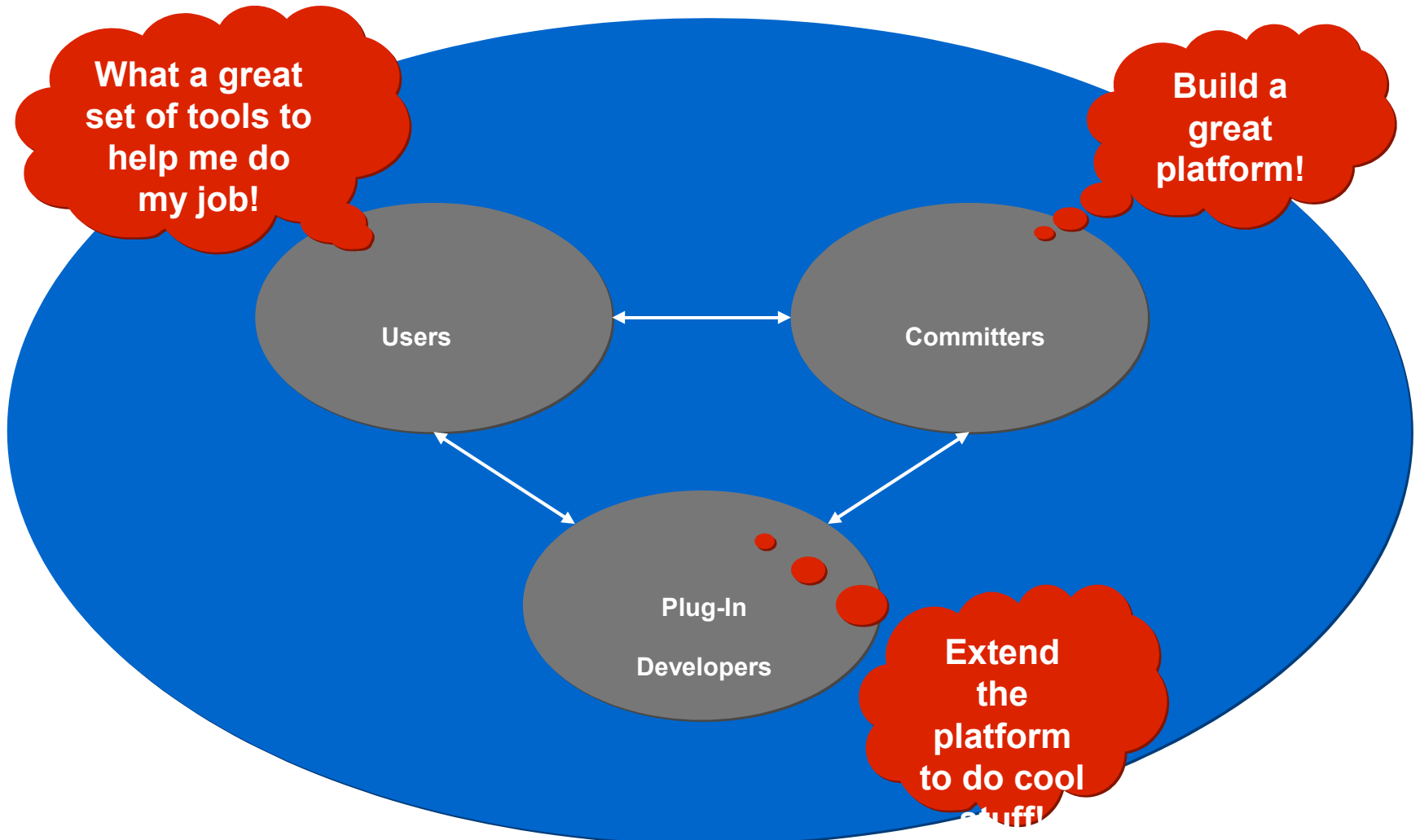
...What is our community saying?

The Anatomy of a Community

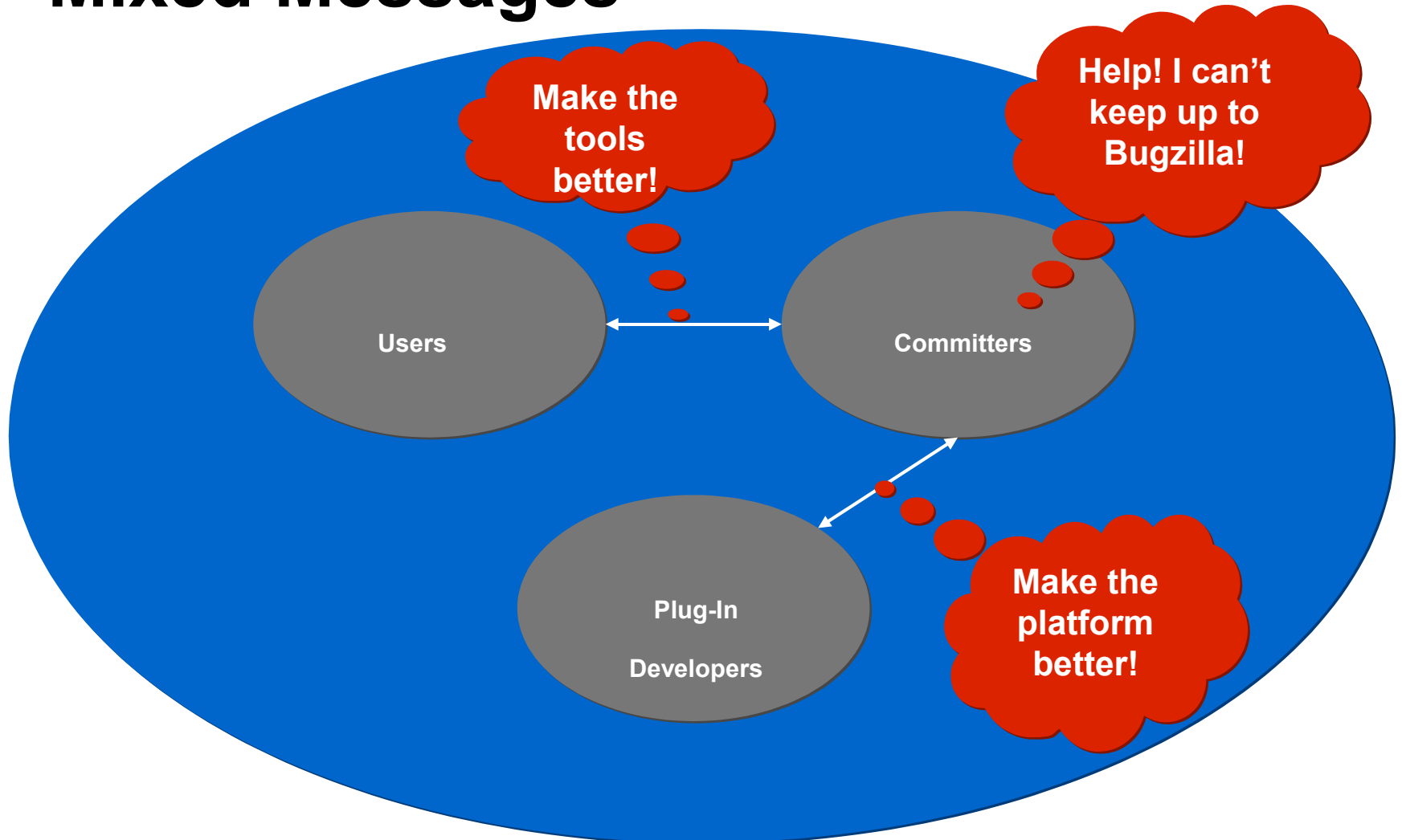
A large, solid blue oval that occupies most of the slide's content area. It is centered horizontally and vertically.

The Eclipse Community

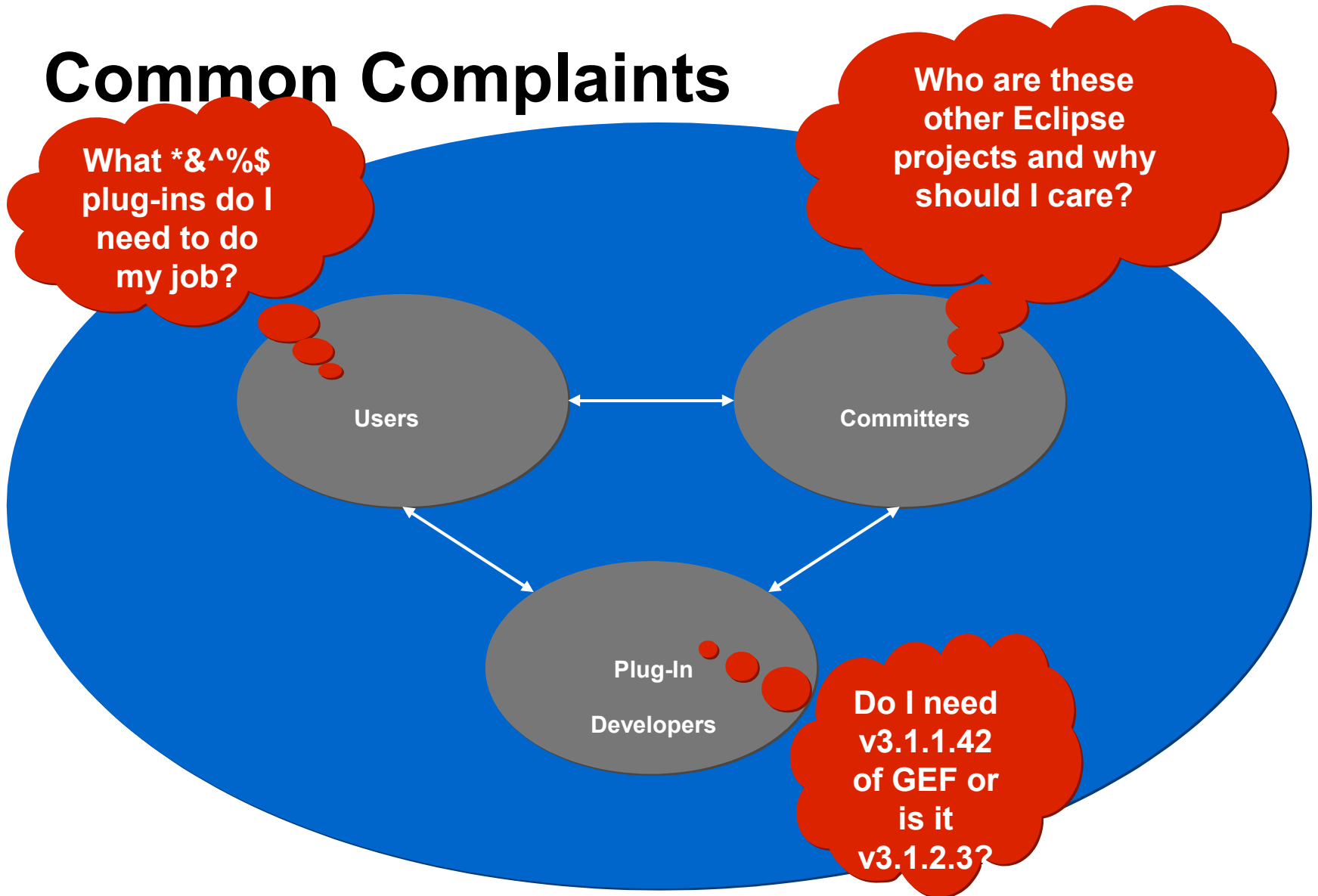
Role-based Communities



Mixed Messages



Common Complaints



What Problems Does Callisto Solve?

Or “Why bother?”

- Eclipse project teams are independent
 - Independent project plans
 - Independent schedules
 - Yet the projects’ code is inter-dependent
 - Adopters often utilize multiple frameworks
 - Thus each adopter/user had to solve “which version of which project works with the others?”
 - Why not share the burden and do this once?
- ...And adopters get the whole stack 6–8 weeks faster

Callisto Benefits

- For ISVs and Plug-in Developers
 - Access to a complete stack of Eclipse projects once a year to build products on
- For Developers and Users
 - Better integration of multiple projects
 - More complete development solution (WTP, BIRT, etc.)
- For Eclipse Committers and Projects
 - Forcing greater co-operation and interaction ---- community building
 - Learning and sharing experience and best practices
 - Forcing new projects to mature faster
- For Everyone
 - Continued focus on Quality and Predictability

Eclipse Callisto Simultaneous Release

What Is Callisto?

Why Is the Eclipse Community Building Callisto?

New Feature Highlights

New Java Technology Development Features

Lessons Learned

Why Is It Named Callisto?

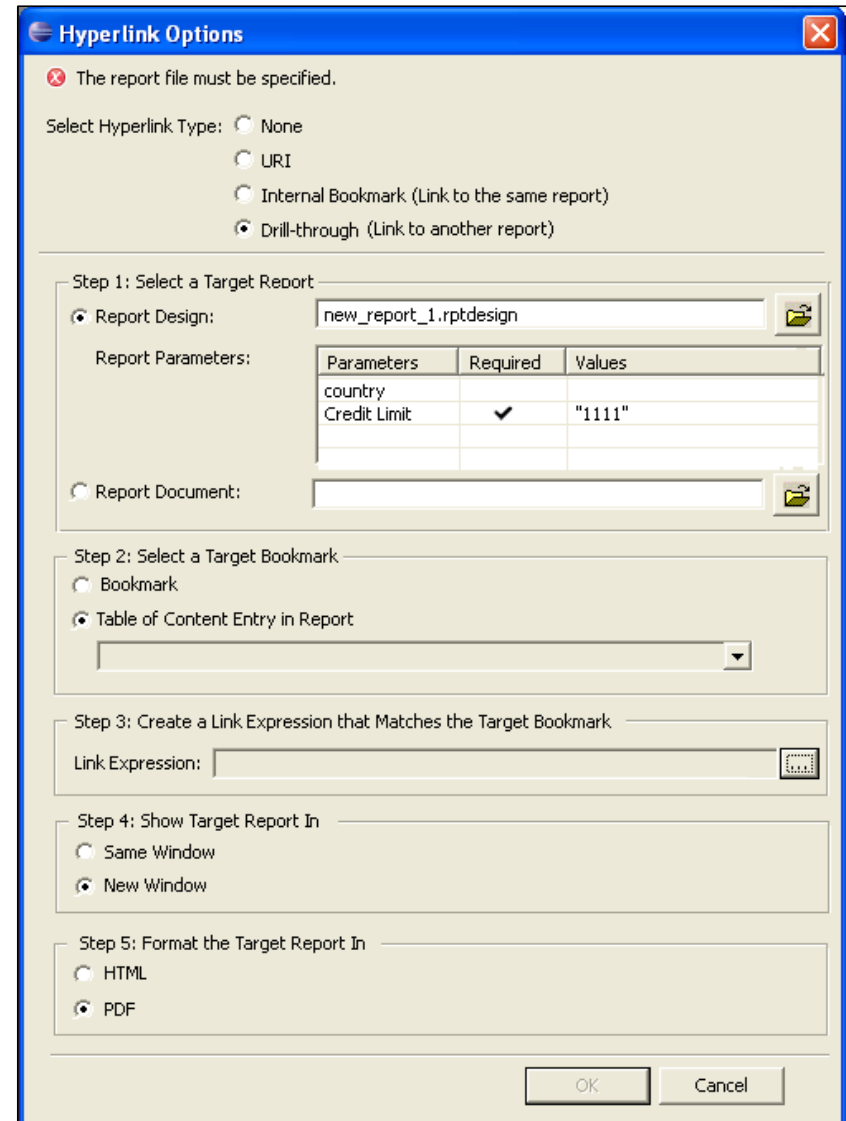
Platform and Java Development Tools

- Support for JSE 6 (“Mustang”) features
- Preview support for Windows Vista
- JUnit 4.1 integration
- Switch to ICU4J
- More separation from relying on the filesystem
 - Logical models, flexible resource model
- Enhanced debug platform for non-Java debug
 - E.g., embedded multi-core, parallel tools
- ...And more

BIRT

New features

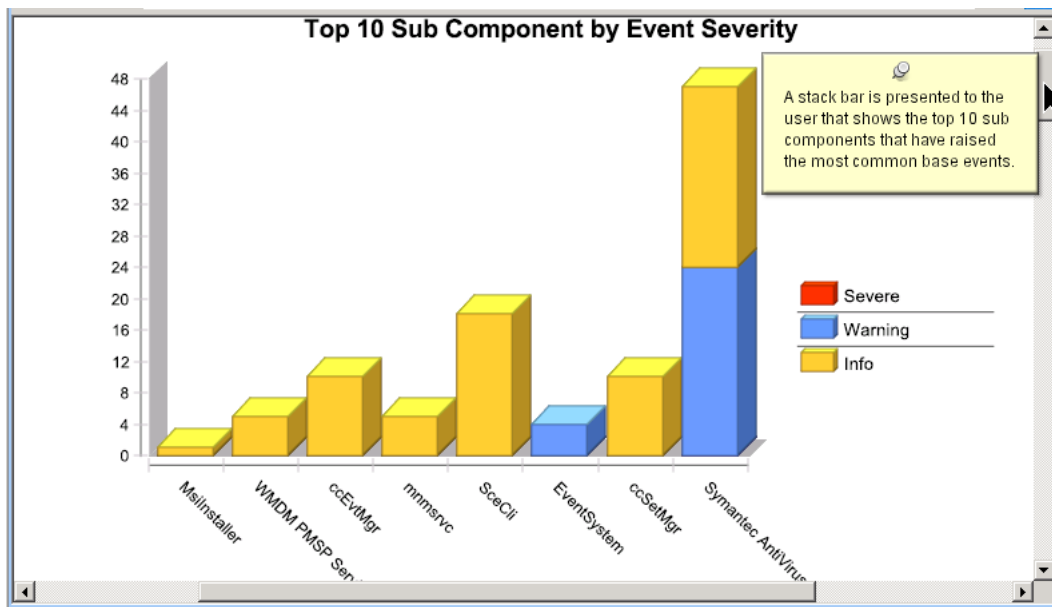
- Improved report viewing
- Enhanced XML data source
- More charting features
- And more...



TPTP

Filling out the feature set

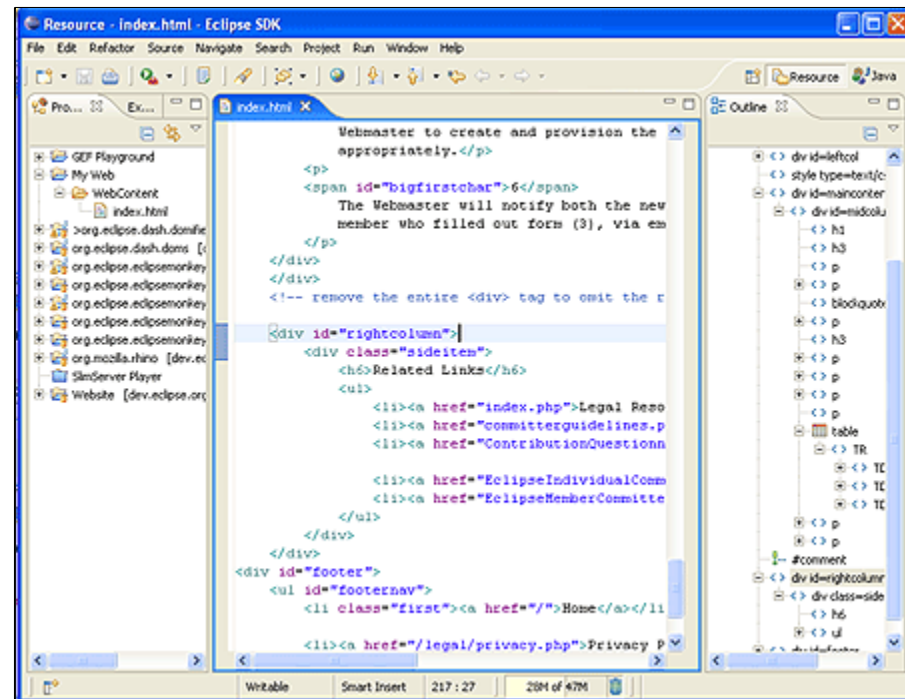
- Profiler support for Java Virtual Machine Tool Interface
- Dynamic BCI
- BIRT integration
- Static analysis tools
- Import Java-based Logging files
- And more...



Web Tools

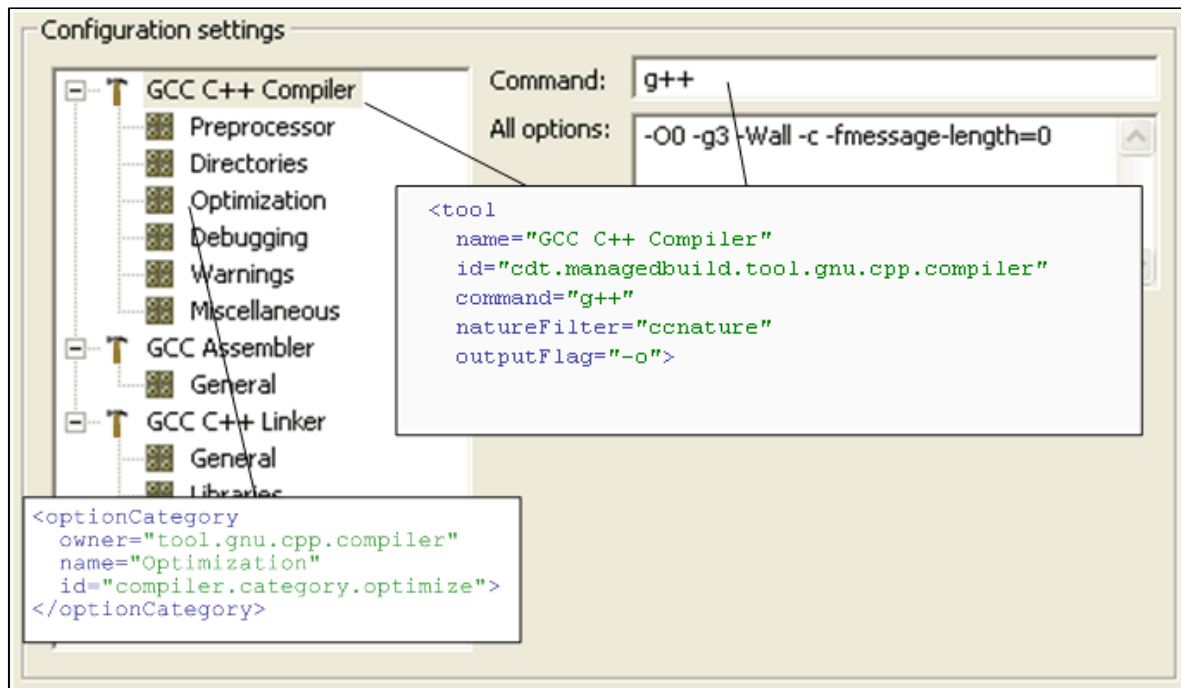
Adopter ready

- Common navigator
- Common undo stack
- Axis 1.3
- Generic server support
- More public API
- And more...



C/C++ Development Tools

- New indexer
- Managed builds



Eclipse Callisto Simultaneous Release

What Is Callisto?

Why Is the Eclipse Community Building Callisto?

New Feature Highlights

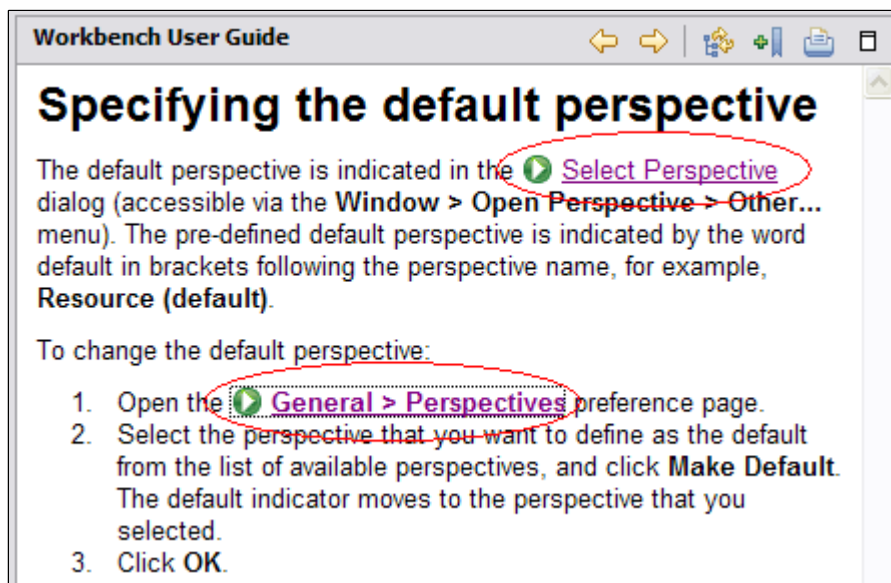
New Java Technology Development Features

Lessons Learned

Why Is It Named Callisto?

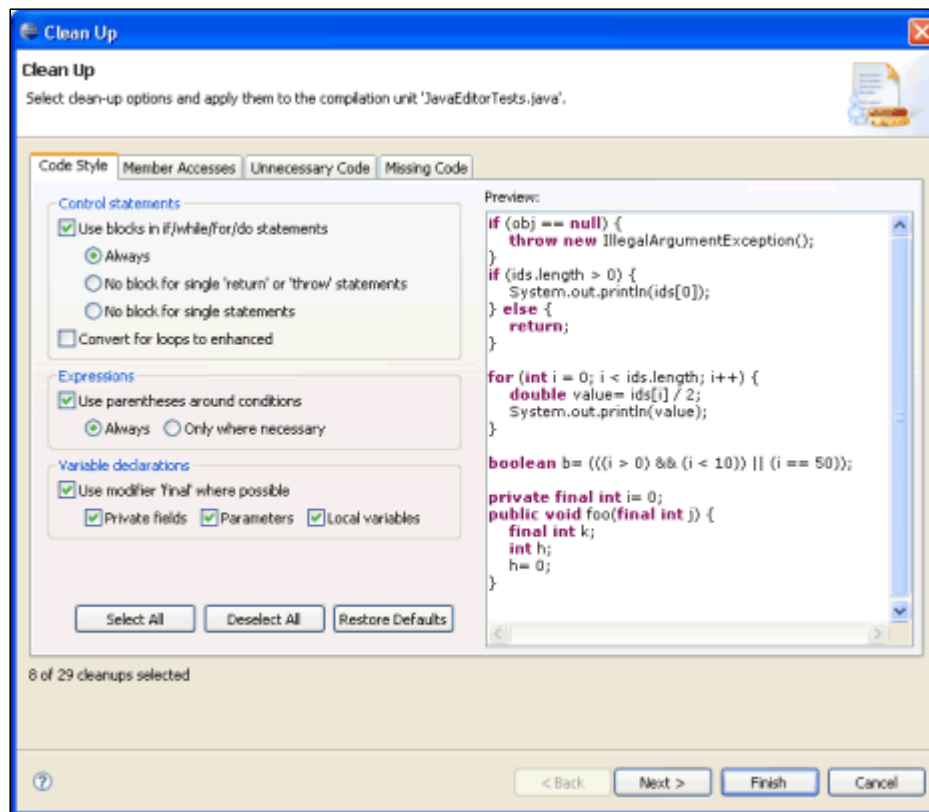
Cheat Sheet Embedded Commands

Simple to use



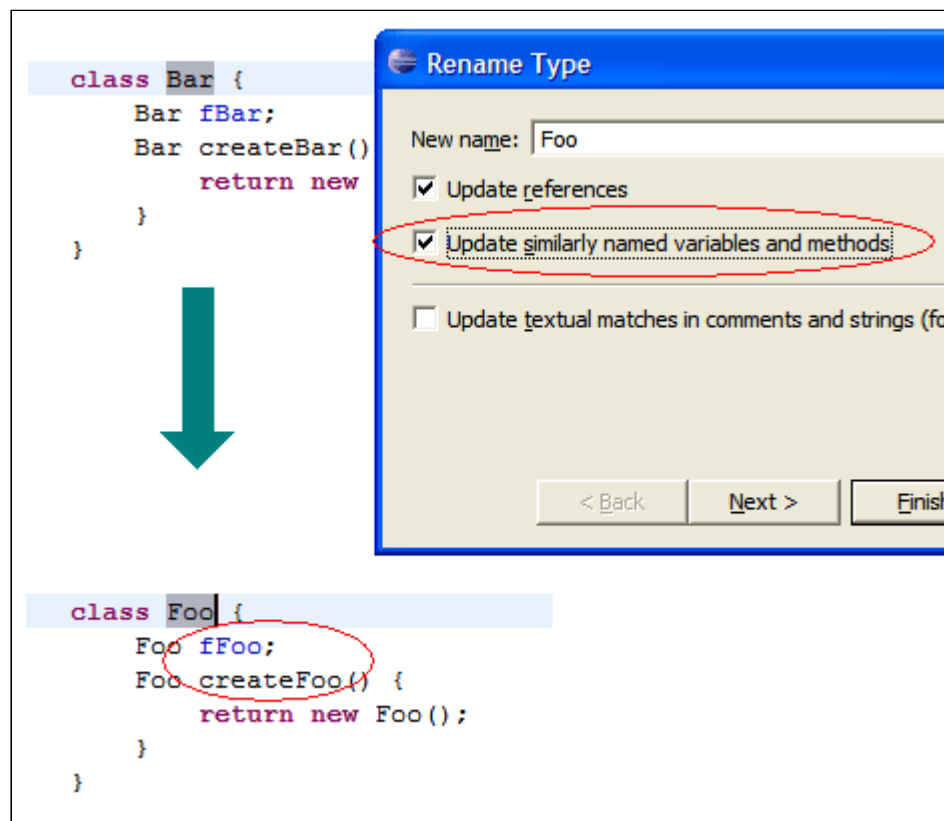
Clean Up Wizard

Simple to use



Update Similar Names

Simple to use



```

class Bar {
    Bar fBar;
    Bar createBar ()
        return new
    }
}

```

↓

```

class Foo {
    Foo fFoo;
    Foo createFoo() {
        return new Foo ();
    }
}

```

Rename Type

New name:

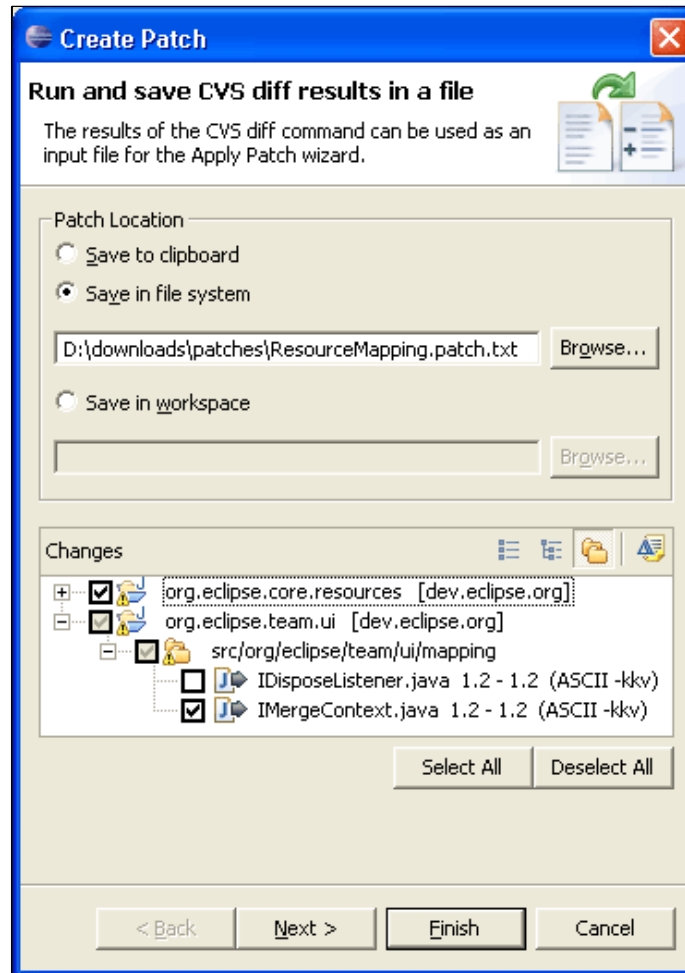
Update references

Update similarly named variables and methods

Update textual matches in comments and strings (fo

< Back Next > Finish

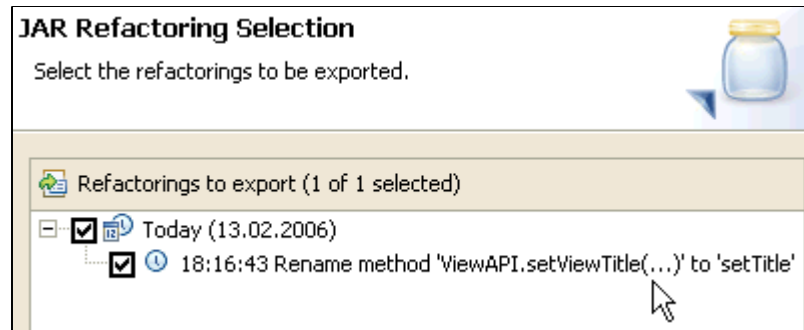
Better Create a Patch



<http://download.eclipse.org/eclipse/downloads/> » 3.2M6 » New & Noteworthy

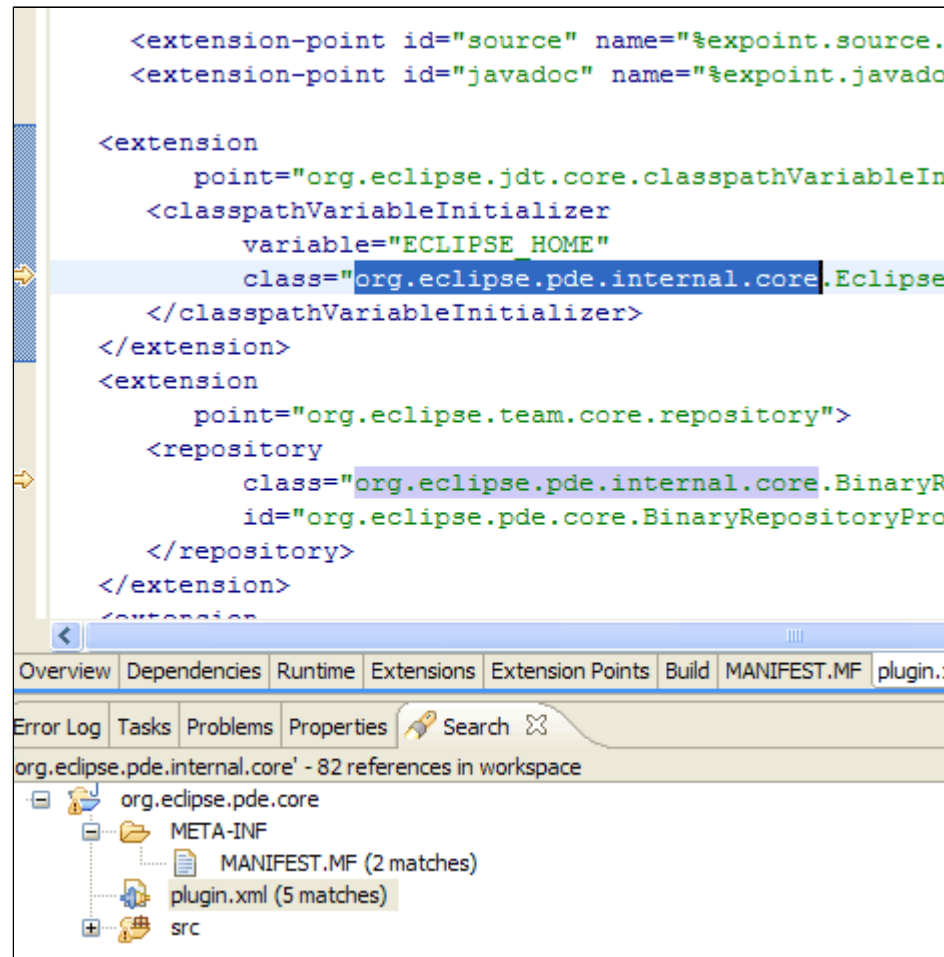
Refactoring Scripts

Scaling up



Java-based Search in Manifest Files

Enterprise ready



The screenshot shows the Eclipse IDE interface. The top pane displays XML code from a manifest file. The code includes extension points and extensions. One extension is highlighted with a blue background:

```

<extension
  point="org.eclipse.jdt.core.classpathVariableIn
  <classpathVariableInitializer
    variable="ECLIPSE_HOME"
    class="org.eclipse.pde.internal.core.Eclipse
  </classpathVariableInitializer>
</extension>
<extension
  point="org.eclipse.team.core.repository">
  <repository
    class="org.eclipse.pde.internal.core.BinaryR
    id="org.eclipse.pde.core.BinaryRepositoryPro
  </repository>
</extension>
<extension

```

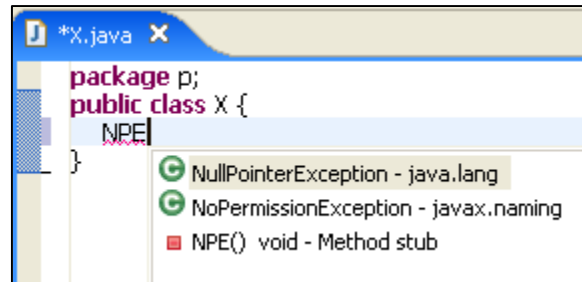
Below the code editor, the 'MANIFEST.MF' tab is active. The bottom pane shows search results for the query 'org.eclipse.pde.internal.core'. The results indicate 82 references in the workspace. The search results are displayed in a tree view:

- org.eclipse.pde.core
 - META-INF
 - MANIFEST.MF (2 matches)
 - plugin.xml (5 matches)
 - src

<http://download.eclipse.org/eclipse/downloads/> » 3.2M6 » New & Noteworthy

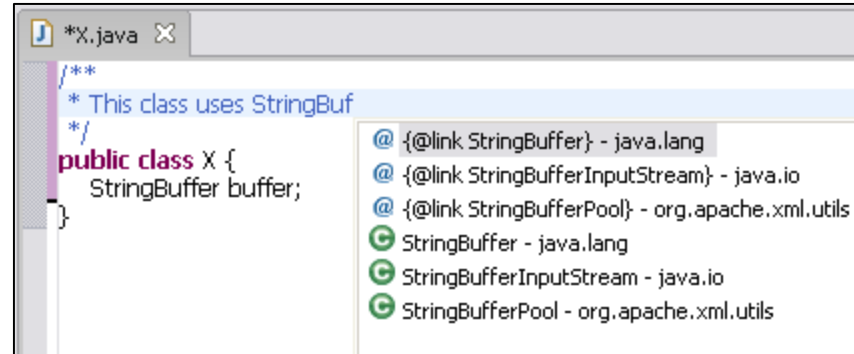
CamelCase in Code Completion

Appealing to the broader community



Code Completion in Javadoc™

Appealing to the broader community

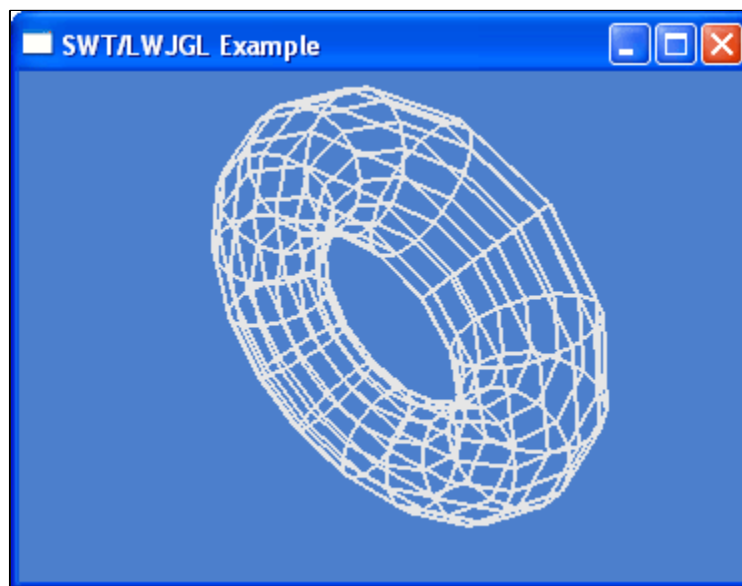


```
/**
 * This class uses StringBuffer
 */
public class X {
    StringBuffer buffer;
}
```

- {@link StringBuffer} - java.lang
- {@link StringBufferInputStream} - java.io
- {@link StringBufferPool} - org.apache.xml.utils
- StringBuffer - java.lang
- StringBufferInputStream - java.io
- StringBufferPool - org.apache.xml.utils

OpenGL in SWT

Be a better platform



Eclipse Callisto Simultaneous Release

What Is Callisto?

Why Is the Eclipse Community Building Callisto?

New Feature Highlights

New Java Technology Development Features

Lessons Learned

Why Is It Named Callisto?

Simultaneous Release

How hard could it be?

- If projects were independent, not too hard
- Due to dependencies, need coordination
- Project X releases after project P;
and project P releases after project A
 - $A \Rightarrow P \Rightarrow X$
 - Communication of changes
 - Each communication creates delay

First Lesson: Unintended Effects

Using un-documented features

- Eclipse projects have well-defined APIs
 - Ambiguities
 - Deliberate violations
- Upstream projects do not know who is using which of the APIs/non-APIs
- Upstream projects make a valid change that ends up breaking downstream projects
 - E.g., extension ids not allowed to have dots
 - Caused 3.2M5a
- Lesson: allow time for whole chain testing

Second Lesson: Distributed Control

Ownership

- The Eclipse projects are separate entities with separate leadership and separate schedules
 - But a simultaneous release requires coordinated schedules
- Most projects had to change their schedules
 - Time for new features
 - Length of iterations/milestones
 - Date of API freeze
 - Dates of mid-term releases
- Lesson: people do not want to give up control

Third Lesson: Update Manager

More than just code

- Planned to use the Eclipse Update Manager to distribute the ten Callisto projects
 - Users would download a tiny core
 - Then use Update Manager to get desired features
- Update manager was never tested with this much code
- Update manager was designed for a single project
 - Requires centralized “site.xml” file
 - We needed an “<include>” directive
- Not a simple one-click download experience
- Lesson: code is only part of the story –or– delivering a working system is more than just code

Fourth Lesson: Mirror Sites

Volunteers

- Not all mirror sites have all the Eclipse files
 - Callisto is going to make the disk space and bandwidth issues even larger
- Worse, a corrupt eclipse.org server seeded some mirrors with corrupt copies
- The Update Manager interacted badly with missing and corrupt files on mirrors
- Lesson: distributing bytes is more than just putting them on an http/ftp server

Fifth Lesson: Common Interface

Community demands more

- Callisto goal “simultaneous release”
 - Same code; simultaneous date
 - Worst UI problems solved
 - E.g., three project types with the same name
 - E.g., menu items for one project showing up in another project’s view
- Community understood “integrated product”
 - Better code and simultaneous date
- Have to use capabilities and perspectives
- Lesson: be careful what you say because people repeat what they want to hear

Sixth Lesson: Testing

Who owns the in-betweens?

- Everyone wants to be part of the release
- Nobody wants to test other projects' code
- Eclipse projects are fairly well tested individually
 - One can expect the Callisto bugs to be in-between
- Lesson: (don't have one yet)

Hard Work

It's easy to say but hard to accomplish

- **Simultaneous Release** is easy to say
 - And easy to agree to join
- The reality is more work than everyone expects
- N projects have $\sum_n i$ interactions
- Requires giving up some sovereignty and resources for a greater good
 - Always a trade-off
- Projects can opt-out and release “simultaneously” on their own

Eclipse Callisto Simultaneous Release

What Is Callisto?

Why Is the Eclipse Community Building Callisto?

New Feature Highlights

Problems We've Encountered

New Java Technology Development Features

Why Is It Named Callisto?

From Bill Arnet's Nine Planets

Humor

- Callisto seems to have **little internal structure**; however, there are signs from recent Galileo data that the interior materials have settled partially, with the **percentage of rock increasing toward the center**

Voted by the Community

Community input

- A variety of naming schemes were proposed
- Space-based names were the most popular
- The 63 moons of Jupiter provide a rich name space for future coordinated releases
 - But we'll have to think of a new naming scheme in 2069 ☹️

Why We Did Not Use Numbers

The serious reason

- Eclipse version numbers (X.Y.Z) semantics
 - X increments for breaking API changes
 - Y increments for non-breaking API changes
 - Z increments for bug fixes
- Projects have different numbers
 - Platform 3.2; TPTP 4.2; BIRT 2.1; CDT 3.1; EMF 2.2
- Needed to accommodate different release cycles for different projects
 - Once a year; twice a year; ...
- Didn't want to force all the projects to the same version numbers

You Can Help!

- Download Callisto Release Candidates and let us know what you think
- The Callisto Bug Hunt is on!
- Eclipse Callisto t-shirts for all nominated “great bugs”
- Draws for iPods
- Grand prize of a custom-built Eclipse mountain bike



Summary

- Eclipse Callisto is ten major projects...
- ...Releasing simultaneously
- Each project has nifty new APIs for adopters and new features for users
- Coordinating the releases is harder than it looks

Q&A

Mike Milinkovich

Bjorn Freeman-Benson



the
POWER
of
JAVA™



Inside Eclipse Callisto: The New, the Useful, and the Lessons Learned

Mike Milinkovich and Bjorn Freeman-Benson

Executive Director and Director, Open Source Process
Eclipse Foundation

<http://www.eclipse.org/>

TS-4311