Eclipse Project 3.5 Release Review

Eclipse Project PMC
Highlights

• 3.5 new features:
  – New platforms: Mac Cocoa 32 and 64-bit, Solaris x86, os/390
  – Declarative services, block selection in editors, improved compare editors, PDE target management, extensible execution environments, improved test and build infrastructure

• API quality:
  – High. 7 changes in porting guide.
  – Binary compatible for compliant plug-ins
  – New API: 168 types, 351 methods
  – Deprecated API: 160 types, 51 methods, 23 fields
  – 7 breaking changes: Platform (7), JDT (0)

• End-of-life issues:
  – Bundle org.eclipse.pde.p2.ui removed, content merged into org.eclipse.pde.ui. No API involved.

• IP Clearance and Licenses:
  – All licenses and about files are in place as per the Eclipse Development Process, the Due Diligence Process was followed for all contributions

• Community and Committer Diversity:
  – 107 committers, 63 active in past 9 months
  – Organizations: IBM (51), Individuals (6), Code 9 (2), Adobe (1), Wind River (1), BestSolution (1), Red Hat (1)
  – Geographies: Canada, USA, France, Switzerland, Poland, Germany, Austria, Japan, India
  – Commits: IBM (91%), Individuals (4%), Wind River, Red Hat, Code 9, Adobe, BestSolution (1% each)
  – Consumed by many other Eclipse projects
Themes and Plan Items

• Scalability
  – Performance testing infrastructure improvements
  – Performance focus

• Consumability
  – User assistance improvements
  – Extensible execution environments
  – PDE build and export enhancements
  – Declarative services tooling
  – Debug user interface enhancements
  – PDE performance and target management
  – Team shareable working sets

• Platforms
  – Support Cocoa on Mac OS X
  – BIDI Improvements
  – Accessibility standards compliance

• Robustness
  – API Tools test suite and infrastructure
  – Compare editor improvements
  – Build process improvements
  – API Tools enhancements
  – Provide API for missing features

Themes and Plan Items

• Deferred plan items:
  – Team shareable working sets
  – Provide API for missing/internal features
  – Accessibility standards compliance

New and Noteworthy - Platform

- Eclipse on Cocoa
- Eclipse Solaris x86
- Eclipse on s390 and s390x
- Customize Perspective dialog
- Improved switch editors and multi-page editors
- Open Resource dialog allows choice of editor
- Improved Network Connections page
- Project Explorer improvements
- JFace dialog button order on GTK+
- Multi-instance Properties view pinning
- New workspaces preference page
- Compare Word documents
- New Compare With Other Resource dialog
- Compare editor enhancements
- Switchable compare viewers
- Custom icons in help system
- Buttons to change font size in Welcome
- Help quick search
- Install/Update Changes
- New software updates menu items
- New install wizard
- Install wizard only loads chosen repositories
- Install wizard usability and workflow improvements

- Installation details in About dialog
- New installation information dialog
- Auto-completion of repository locations in install wizard
- New available software sites preference page
- Block selection mode in textual editors
- Double-click collapses the folding region
- History in Tag Resources dialog
- Apply patch in the Synchronize view
- Synchronize schedule dialog improvements
- Debug view breadcrumb
- Declarative services added to the platform
- New compare core plug-in
- Improved look for filtered tree
- Multiple fonts in one cell
- JFace support for Mac Sheet dialogs
- New "check state" provider
- Show welcome checkbox
- New help content filter extension point
- Jetty version 6.1
- New release of ICU4J
- Extensible about dialog
- A new intro theme called "Slate" has been added
New and Noteworthy - Platform

- Add support for Cairo on AIX
- New tool for generating the SWT JNI code
- Block selection in StyledText
- JavaScript-to-Java communication in Browser widget
- Improved Java-to-Javascript support in Browser widget
- New caret listener API on StyledText
- New SWT text editor demo
- Improved search field
- Improved copy and paste support between Eclipse and Nautilus
- StyledText now supports hyperlinked text
- Configurable StyledText margin spacing and color
- Ability to disable Javascript in the Browser widget
- Create and access cookies in Browser
- Alternate button order on GTK
- New Shell modified state hint
- Support for drop-down style date/time widget

- SWT.SHEET style for Shell and Dialog on Mac OS X Cocoa
- Mozilla browser support on Solaris x86
- Improved DND feedback on Cocoa
- Cocoa drag and drop feedback
- Programmatically set paper orientation in PrintDialog
New and Noteworthy - JDT

- Code completion in constructors
- Open Implementation hyperlink
- Select comment with double click
- Quick Fix to start Rename refactoring
- Links in Javadoc headers
- Open in Properties File action in NLS Hover
- Formatter option to preserve user line breaks
- Coded formatted with never join lines
- The Java Compare editor now supports:
  - Move/copy/delete line
  - Formatting
  - Hyperlinking
  - Content assist
  - Javadoc hovers
  - Quick outline
  - Reconcile Java structure while typing
- Generate toString()
- Optionally use blocks when generating hashCode() and equals()
- Comparing identical values compiler warning
- Missing synchronized modifier compiler warning
- Problem hover with quick fix for missing synchronized modifier
- Missing hashCode() method detection
- New compiler dead code detection
- Classpath resolution honors the 'Class-Path' header of JAR manifest file
- Build path supports "..
- Build path resolution tolerates duplicate entries
- Compiler compliance follows execution environment
- Emphasized matching characters in Open Type dialog
- Sort working sets in Package Explorer
- Delete working sets from the Package Explorer
- Paste patch into Package Explorer
- Call Hierarchy can expand with constructors
- Expand With Constructors
- Improved Javadoc view and hover
- Open *.jardesc files with JAR Export Wizard
- Open test result files in JUnit view
- Importing test run history
- Moved up JUnit version to 4.5
New and Noteworthy - PDE

- New declarative services editor
- DS tooling supports latest DS specification
- Jump to Console from stack trace
- New compiler options to validate version numbers on packages and bundle
- New properties editor for .options
- Bundle start levels and license info added to product editor
- You can now initialize product definitions from OSGi launch configurations.
- New NL fragment generation wizard
- Menu introspection added to Plug-in Spy
- Plug-in registry view now supports browsing OSGi services
- Eclipse launch configurations now support start levels and auto-start settings
- JUnit plug-in tests in non-UI thread
- The Plug-in Registry view now includes fragments.
- Plug-in export wizard now supports installing into the running platform
- Patched plug-ins will appear in the installed software list and can also be uninstalled.
- Export options have been enhanced:
  - Support added for generating source bundles
  - Export features, plug-ins, and products with binary cycles.
  - Export existing class files from the workspace.

- New category editor to support publishing feature categories
- JAR signing support expanded to include support for keypasses.
- Improved API tooling compatibility options
- @noextend restriction now supported on interfaces
- System library validation in API tools
- Warnings and quick fix for stale problem filters
- Ant tasks added to perform API analysis and generate simple HTML reports
- New Compare With > API Baseline in API tools
- PDE Build support added to sort plug-ins across feature boundaries
- Parallel compilation support added in PDE build
- Update integration between PDE/Build and p2
- Qualifier replacement support on .product files
- Added support to fetch artifacts for build using p2
- p2 repositories supported as a PDE/Build target
- New target platform and target definition page
- Support for multiple targets and target switching
- Target editor enhanced to support features in targets
- New Target Platform State view displays details about current target platform
3.5 Plug-in Changes from 3.4

Added Plug-ins (7)
• org.eclipse.compare.core
• org.eclipse.core.databinding.observable
• org.eclipse.core.databinding.property
• org.eclipse.pde.ds.core
• org.eclipse.pde.ds.ui
• org.eclipse.pde.ua.core
• org.eclipse.pde.ua.ui

Removed Plug-ins (1)
• org.eclipse.pde.p2.ui

Added 3\textsuperscript{rd} Party Plug-ins
• org.apache.commons.codec
• org.apache.commons.httpclient
• org.eclipse.ecf.provider.filetransfer.httpclient
• org.eclipse.ecf.provider.filetransfer.httpclient.ssl
• org.eclipse.equinox.concurrent
• org.eclipse.equinox.ds
• org.eclipse.equinox.p2.publisher
• org.eclipse.equinox.p2.repository
• org.eclipse.equinox.p2.repository.tools
• org.eclipse.equinox.p2.ui.sdk.scheduler
• org.eclipse.equinox.util
• org.mortbay.jetty.server
• org.mortbay.jetty.util

Removed 3\textsuperscript{rd} Party Plug-ins (1)
• org.mortbay.jetty

Note: 3\textsuperscript{rd} party plug-ins are plug-ins consumed in the Eclipse SDK but not produced by the Eclipse Project
Non-Code Aspects

• The 3.5 release will contain updated User and ISV documentation

• Community is very active
  – Mailing lists and newsgroups have steady activity
    • new E4 mailing list: e4-dev@eclipse.org
  – Blogs dedicated to Eclipse are active e.g.
    • http://www.planeteclipse.org
  – Wiki content is growing
    • http://wiki.eclipse.org/index.php/Eclipse_Project
    • E4 wiki: http://wiki.eclipse.org/E4
Non-Code Aspects

• Internationalization
  – Latin1 and Latin2 locales are supported in all operating environments
  – DBCS locales are supported on all platforms
  – BIDI locales (with mirroring) supported on Windows and Linux GTK, BIDI text supported on Mac.
  – GB18030-1 Chinese codepage standard is supported on Windows, Linux GTK and Mac.

• Localization
  – Tested for Localization and participating in Babel Project

• Accessibility
  – Tested for accessibility
  – Open accessibility bugs: 8 major, 0 critical, 0 blocker
Non-Code Aspects

• Articles, examples, and tutorials
  – New and updated articles and tutorials on eclipse.org (3)
  – Numerous Webinars and Podcasts
  – Some of the new/updated articles and tutorials were provided by the Eclipse community
  – Older articles need to be reviewed and updated for 3.5, if applicable
Platform Quality API

- API quality is a collaborative effort that involves the experience of the developers working on the Eclipse project, and feedback from consumers.
- API changes and proposed API additions are often broadcast to mailing lists to raise awareness of the changes and encourage discussion and feedback.
- API changes between 3.4 and 3.5 are checked automatically by API tooling integrated into integration build process.
- The 3.5 migration guide identifies 7 changes:
  - For each, a description of the change, what code is affected, and the action that needs to be taken is described.
  - We are not aware of any API compliant plug-ins breaking as a result of these changes.
  - The 3.5 migration guide also describes changes required to adopt mechanisms and APIs that are new in 3.5.
- The PMC is comfortable supporting the API that is in the Eclipse project 3.5
3.5 API – Platform

New

- API to defer updates when changing command context
- Added API for Ant launch configuration attributes
- Added extension point for breakpoint import participants
- Added API to retrieve the debug context associated with a command’s execution event
- Added API for debug preference settings
- API allows detail panes to specify a selection provider
- Added API for re-useable working directory selection UI component in launch tabs
- Added API to export/import breakpoints to/from a string buffer
- Added extension point to contribute editor/debug model specific breakpoint factories
- New API to allow a web server other than Jetty to be used as the help server.
- New API to allow preprocessing to be performed on html and xhtml pages returned from the help server.
- New API to support split button role for accessibility
- New API to support authentication in the Browser
- New API to support cookies in a Browser
- New API to support calling between Java and JavaScript
- New API to track the locations of the caret in a StyledText control
- New API to support block selection in a StyledText control
- New API to offset the source image when a drag is started
- Support for embedded links was added to StyledText
- Support for shared OpenGL contexts was added to GLCanvas
- Support for landscape printing was added to Printers
- Support for setting the Mac modified state was added to Shells

New

- Support for Mac sheet windows was added
- IFileBufferManager now allows executing a runnable in the manager’s synchronization context
- New method on FindReplaceDocumentAdapter to allow escaping a string for usage in a regular expression
- New APIs to support block selection in text views and editors
- New API in DefaultTextDoubleClickStrategy to allow subclasses to provide the double-click selection
- New editor action constructors for creation with default label
- New API in TemplateStore for restoring default values
- WhitespaceRule now allows setting a custom whitespace token
- New API in DefaultHyperlinkPresenter to support native hyperlink color
- New API to remove a search query
- New API for Properties, making it easier to set up bindings, and to implement custom observables
- Better support for binding to JFace viewers
- Support listening for disposal of data binding observables
- More general-purpose data binding observables
- Support opening multiple editors at once
- API to hide and show editors based on current perspective
- API for multi-instance properties views
- API to support an extensible About dialog
- Filling in missing API in org.eclipse.ui
- API to support check state and radio state for commands
- New API added for applying patches
- Better support for configuring and customizing compare editors
3.5 API – Platform

New
- Common Navigator enhancements:
  - capabilities support,
  - Go Into,
  - better drag&drop support for clients,
  - enable subclassing
- New API for better Open With support
- Support view folders that remain visible after their last child is closed
- API to configure new look for FilteredTree
- Allow programatic additions to the menu system via the extension point
- Improvements to error handling API
- New API to support toolbar configurability
- Enable setting style bits for JFace dialogs (for SWT.SHEET)
- Various new convenience API added to JFace:
  - Helper dialog to make viewers with columns keyboard-accessible
  - Common API for testing which platform is running
  - Improve support for trees and tables with checkboxes
  - Add font support for StyledCellLabelProvider
  - Enable better performance for TableViewers with SWT.VIRTUAL
  - New API to create content describers for XML files
  - New API in IProxyService to handle URI

Deprecated
- Deprecated old Navigator view
- Deprecated org.eclipse.update component
- Deprecated org.eclipse.core.runtime.Preferences API
- Deprecated data binding BidirectionalMap and MappedSet in favor of BidiObservableMap
- Deprecated text editor SaveAction, replaced by ActionFactory.SAVE
- 51 assorted method deprecations
- 5 assorted constructor deprecations
- 23 assorted field deprecations
3.5 API – JDT

**New**
- Added API to create a wildcard type binding key from a generic type, a bound kind, a bound type and a rank.
- Added APIs for constructor invocation completion proposals.
- Added APIs that allow the cancellation of code completion.
- Added code assist options to define the prefixes and suffixes when completing a static final field name.
- New compiler option to report comparison of identical expressions.
- Added compiler option to report dead code.
- Added compiler option to report missing hashCode() method.
- Added compiler option to report a missing synchronized modifier on inherited method.
- Added naming convention API to get the base name from a variable name.
- Added APIs to allow a working copy owner to define a type or package that would be otherwise unknown to the type system.
- Added API to convert a char array to upper case.
- Added DOM/AST binding API to get the generic type of a wildcard type.
- Added DOM/AST binding API to obtain rank of a wildcard type.
- Added DOM/AST API to find the covered node and the covering node for a given range.
- Added code formatter options to join wrapped lines in code and to join lines in comments.
- Added search pattern API to get the regions in a name matching a pattern using a match rule.

**New**
- Previously internal CompilationUnitChange promoted to API
- Previously internal RefactoringSaveHelper promoted to API
- Many new APIs to support new clean up extension point
- New APIs to provide type parameter labels
- GenerateToStringAction introduced to invoke toString() wizard
- New API to allow inclusion of empty directories when creating a runnable JAR
- New API to configure a new Java project with a given compiler compliance
- New API in JUnitCore for finding tests for a given Java element
- New API to retrieve/specify compiler compliance options and system packages for execution environments

**Deprecated**
- Deprecated JDT “show in” actions, replaced by generic Navigate > Show In
3.5 API – PDE

New
• Added API for Eclipse Application launch configuration attributes
• Added API to support JUnit plug-in tests run in non-UI thread
• Added API to create a plug-in project creation wizard based on a specific template

Deprecated
• None
Tool Usability

• Eclipse is a superior IDE for Java tooling and plug-in development

• Many usability enhancements made in 3.5 to continue this tradition
  – Declarative services tooling
  – Execution environment validation in PDE and JDT
  – Powerful new PDE target definition support
  – Block selection in editors
  – Full editing capabilities in compare editors
  – Help quick search
  – More powerful perspective customization
  – Breadcrumb debug view
  – Content assist for constructors
  – Many new Java compiler diagnostics, including dead code analysis
  – And many more tooling improvements!
Architectural Issues

• Primary runtime is still a 1.4 JRE. Complementary functionalities on Java SE 5 (junit4, APT 5) and Java SE 6 (APT 6, compiler API)

• Minimum execution environment for RCP moved up from Foundation 1.0 to Foundation 1.1

• 20 new plug-ins, 2 removed plug-ins (org.mortbay.jetty and org.eclipse.pde.p2.ui)
  – 3 new plug-ins in platform
  – 4 new plug-ins in PDE
  – 13 new plug-ins due to external dependencies (Equinox, Orbit, ECF)
End of Life Issues

• When evolving API the Eclipse Platform will, whenever possible, deprecate the affected API methods and continue to keep them operational.
• Exceptions to this rule are in the 3.5 migration guide.
• All API and functionality provided by the org.eclipse.update component is deprecated in 3.5. This component is superseded by the Equinox p2 component.
• The platform moved to a new major version of the Jetty web server (from 5.1 to 6.1). There are breaking changes between these two major releases.
Bugzilla

• Between June 25, 2008 and May 30, 2009 (RC3)
  – More than 10,500 reports were created
  – Over 9,100 were resolved
  – Over 3,800 were resolved without changing code
    • invalid, duplicate, worksforme, etc...
  – 395 were backported to 3.4.x maintenance

• Current state (RC3) is
  – 18 blockers, 41 critical
  – 0 P1, 49 P2 (3 planned for 3.5.1)

• 3.4 final state was
  – 18 blockers, 57 critical
  – 0 P1, 34 P2
# Bug resolution during 3.5

## Bug Resolution Distribution

### RESOLVED

<table>
<thead>
<tr>
<th>Type</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>M7</th>
<th>RC1</th>
<th>RC2</th>
<th>RC3</th>
<th>RC4</th>
<th>3.5</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>blocker</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>35</td>
</tr>
<tr>
<td>critical</td>
<td>2</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>19</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>77</td>
</tr>
<tr>
<td>major</td>
<td>18</td>
<td>19</td>
<td>34</td>
<td>38</td>
<td>52</td>
<td>61</td>
<td>84</td>
<td>30</td>
<td>11</td>
<td>14</td>
<td>0</td>
<td>11</td>
<td>372</td>
</tr>
<tr>
<td>normal</td>
<td>242</td>
<td>295</td>
<td>416</td>
<td>395</td>
<td>386</td>
<td>681</td>
<td>697</td>
<td>217</td>
<td>68</td>
<td>43</td>
<td>4</td>
<td>130</td>
<td>3574</td>
</tr>
<tr>
<td>minor</td>
<td>26</td>
<td>24</td>
<td>14</td>
<td>21</td>
<td>15</td>
<td>27</td>
<td>31</td>
<td>15</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>185</td>
</tr>
<tr>
<td>trivial</td>
<td>20</td>
<td>21</td>
<td>15</td>
<td>15</td>
<td>16</td>
<td>29</td>
<td>38</td>
<td>14</td>
<td>3</td>
<td>13</td>
<td>1</td>
<td>2</td>
<td>187</td>
</tr>
<tr>
<td>enhancement</td>
<td>34</td>
<td>40</td>
<td>65</td>
<td>74</td>
<td>58</td>
<td>105</td>
<td>57</td>
<td>8</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>18</td>
<td>462</td>
</tr>
<tr>
<td>Total</td>
<td>343</td>
<td>408</td>
<td>553</td>
<td>558</td>
<td>540</td>
<td>925</td>
<td>932</td>
<td>290</td>
<td>88</td>
<td>80</td>
<td>6</td>
<td>169</td>
<td>4892</td>
</tr>
</tbody>
</table>

### FIXED

<table>
<thead>
<tr>
<th>Type</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>M7</th>
<th>RC1</th>
<th>RC2</th>
<th>RC3</th>
<th>RC4</th>
<th>3.5</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>blocker</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>8</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>critical</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td>17</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>63</td>
</tr>
<tr>
<td>major</td>
<td>14</td>
<td>12</td>
<td>32</td>
<td>35</td>
<td>48</td>
<td>56</td>
<td>77</td>
<td>28</td>
<td>11</td>
<td>7</td>
<td>0</td>
<td>9</td>
<td>335</td>
</tr>
<tr>
<td>normal</td>
<td>226</td>
<td>240</td>
<td>385</td>
<td>372</td>
<td>364</td>
<td>618</td>
<td>647</td>
<td>197</td>
<td>63</td>
<td>19</td>
<td>4</td>
<td>79</td>
<td>3236</td>
</tr>
<tr>
<td>minor</td>
<td>26</td>
<td>23</td>
<td>13</td>
<td>19</td>
<td>13</td>
<td>26</td>
<td>28</td>
<td>14</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>178</td>
</tr>
<tr>
<td>trivial</td>
<td>20</td>
<td>20</td>
<td>14</td>
<td>14</td>
<td>16</td>
<td>29</td>
<td>37</td>
<td>14</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>enhancement</td>
<td>32</td>
<td>34</td>
<td>61</td>
<td>64</td>
<td>56</td>
<td>97</td>
<td>55</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>13</td>
<td>423</td>
</tr>
<tr>
<td>Total</td>
<td>319</td>
<td>335</td>
<td>511</td>
<td>517</td>
<td>508</td>
<td>843</td>
<td>866</td>
<td>267</td>
<td>84</td>
<td>72</td>
<td>6</td>
<td>107</td>
<td>4435</td>
</tr>
</tbody>
</table>
Resolved bugs
including fixed, invalid, …

![Bar chart showing the distribution of resolved bugs across different releases and categories.](image-url)
Fixed bugs (only)

![Graph showing fixed bugs over different releases and severity levels]
Standards

- Annotation Processing APIs
  - com.sun.mirror 1.5
  - javax.annotation.processing 1.6
- Java compiler API
  - javax.tools 1.6
- User Assistance consumes (parses) a small subset of RSS 1.0 to get news from eclipse.org
- JUnit 3.8.2 and JUnit 4.3.1
- J2SE
  - Tools are built against J2SE 1.4
  - Compiler can generate 1.3, 1.4, 1.5, and 1.6 code
  - Clients can run 1.4, 1.5 or 1.6.
- SWT
  - Win32, GDI, GDI+, OLE, IE, Carbon, Cocoa, Core Graphics, Quick Draw, Safari, ATSUI, X Windows, X/t, Motif, GTK, GDK, Pango, cairo, ATK, Mozilla, Uniscribe, WPF, OpenGL
UI Usability

• Strings are externalized to support translation into other languages.
• Extensive use of mnemonics and shortcut keys in the user interface enhances usability.
• Full Bidirectional support (mirroring) on Windows and Linux GTK, bidirectional text on Mac OS X
• Accessibility support for Windows, Linux GTK and Mac OS X
• Open accessibility bugs:
  – 8 major, 0 critical, 0 blocker
Schedule

- **Milestones every 6 weeks, 6 cycle duration**
  - API frozen on March 13 (M6), Feature freeze May 1 (M7)
  - Adjusted M5/M6 duration (resp. 7 weeks and 5 weeks) for EclipseCon

- **Tracked schedule**
  - All milestones except M5 delivered as promised
    - M5 three days late due to last minute Eclipse Foundation certificate change

- **End game (release candidate) milestones for 4 cycles**
  - Duration reduced from 2-week to 1-week cycles at RC2 milestone
  - No new features or API allowed without proper approvals
  - Development to end on June 12, 2009
  - Increasingly stringent approval, checking, and change notification requirements in this stage
Process

• The Eclipse project is developed using an open, transparent, and inclusive process
• Teams rely on Bugzilla, mailing lists and newsgroups for input
• Weekly planning calls conducted with the PMC and component leads
  – Meeting minutes posted to the eclipse-dev mailing list
• Component teams have publicly available milestone plans
  – Use project’s web space on eclipse.org to broadcast component milestone plan items and provide status on each item, per milestone
Community

• Eclipse team members are active in Bugzilla, newsgroups, and mailing lists

• Blogs started by Eclipse committers are evolving
  – Use blogging infrastructure at Eclipse.org
  – http://www.planeteclipse.org

• Some teams are using the eclipse-dev IRC channel
  – irc.freenode.net#eclipse-dev
  – irc://irc.freenode.net/#eclipse-e4
  – also see: http://wiki.eclipse.org/index.php/IRC

• The Eclipse team participates in code camps, conference presentations, and tutorials, including
  – EclipseCon, JavaOne, JavaWorld, JAOO, Eclipse Summit Europe, Eclipse Forum Europe, JAX, JAX Asia

• The Eclipse team interacts with other open source projects, standards bodies, and other projects on eclipse.org, including
  – OSGi, Apache Ant, JLS, WTP, Apache Harmony, GCJ, GTK
IP Issues

• All significant and third party contributions have been reviewed and approved by Eclipse legal.
• About files and license files are complete and correct.
• Draft project logs:
Project Plan for Eclipse 3.6/4.0

• Still in planning stage
• Areas in exploration in e4 currently include:
  – Declarative definition of user interfaces
  – Model-based workbench
  – Skinnable UIs
  – Support for multiple languages, in particular Javascript and Clojure
  – A cleaner, simpler, and more powerful programming model
  – Compatibility layer for running full 3.x Eclipse platform on e4