



# Orion Project 0.2 Release Review

# Features



- First incubating release of new web-based tool integration platform
- Tooling user interface written in HTML, JavaScript, and CSS
  - Navigator providing standard file/folder manipulations: create, delete, rename, copy, move, etc
  - High performance editor component supporting many of the same text manipulation commands as the Eclipse desktop editor
  - Basic Git client, sufficient for self-hosted development
  - Search, help system, bookmarks, compare editor
  - Site launching feature for testing and debugging apps
- JavaScript plugin architecture, allowing integration of tools and functionality across domains
- Initial Java-based server, using Jetty and Equinox. Search implementation provided by Apache Solr/Lucene

# Non-Code Aspects



- Orion User Guide and Orion Developer Guide included with release. Documentation developed using Mylyn WikiText
- Developer Guide includes server administration guide that describes how to setup and manage the server
- Public beta server hosted by Eclipse Foundation: [orionhub.org](http://orionhub.org)
- Blog for New and Noteworthy updates: [planetorion.org](http://planetorion.org)
- Community is very active
  - Mailing list has steady activity
  - Firefox and Firebug investigating Orion adoption and integration
  - Wiki content is growing
    - <http://wiki.eclipse.org/Orion>
  - Orion webinar: <http://live.eclipse.org/node/1006>

# Non-Code Aspects



- **Internationalization**
  - No internationalization or localization of messages has been done in this release
  - Editor is capable of displaying non-latin characters and supporting bidirectional text
- **Accessibility**
  - Some accessibility testing done
  - Screen readers handle Orion editor fairly well
  - Some known accessibility bugs: many commands and components are not keyboard accessible

# API



- Orion has initial JavaScript API for client side services and components
- HTTP API for accessing server functionality
- No Java API on the server to allow for alternate server implementation languages and technologies
- Client API documented using jsdoc
- No APIs are final while Orion is incubating

# Tool Usability



- Major focus on tool usability in this release to support self-hosting
- Task-focused UI design puts different developer tasks on different pages
- Page load and refresh times are critical for web applications: extensive work done on caching and build-time page optimization
- JavaScript tooling support is rudimentary at this point. Lacking strong content assist and refactoring support
- Capable search infrastructure, but missing search & replace functionality is a major drawback at this time

# Architectural Issues



- Strong architecture and modularity of Orion client
- CommonJS module format used to facilitate client side reuse
- JavaScript plugin architecture allows for integration via linking rather than a monolithic application: architecture designed for the web
- Editor and “text view” components factored out with minimal external dependencies to enable reuse by other applications
- Modular server using OSGi, Jetty, Servlets, Java SE 6
- HTTP Server API provides maximum flexibility in server implementations
- Need to find ways to leverage/work with JavaScript Development Tools from WTP, such as server side JS analysis
- Targeting modern HTML5-compliant browsers: Opera 11/12, Firefox 4/5, Internet Explorer 9

# End of Life Issues



- None.



# Bugzilla



- Between January 1, 2011 and June 24, 2011 (RC3)
  - 1,250 reports were created
  - 865 were resolved
- Current state (RC3) is:
  - 0 blockers, 1 critical
  - 0 P1, 6 P2

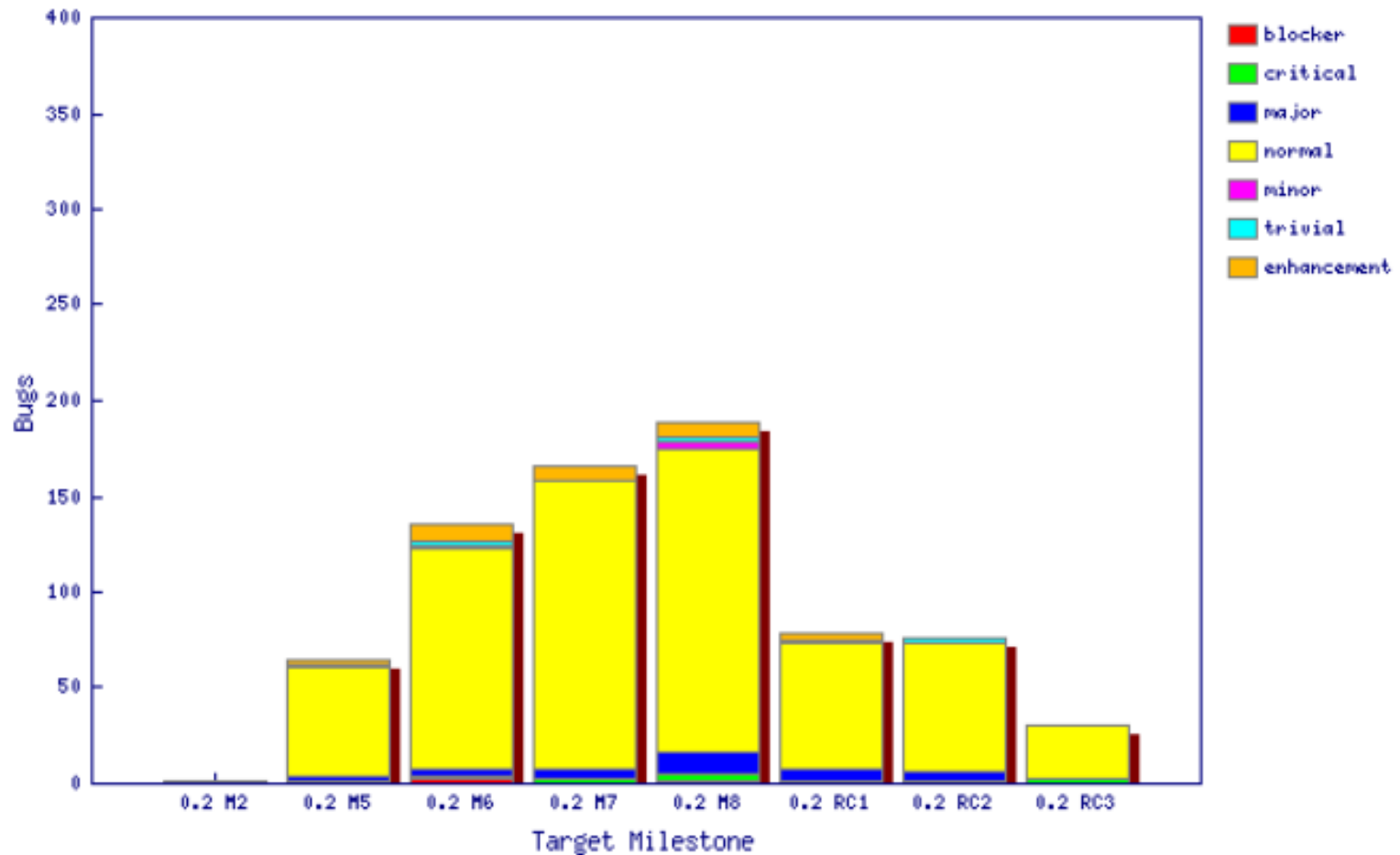
# Bugs fixed during 0.2



## Target Milestone

| Severity |             | 0.2 M2   | 0.2 M5    | 0.2 M6     | 0.2 M7     | 0.2 M8     | 0.2 RC1   | 0.2 RC2   | 0.2 RC3   | Total      |
|----------|-------------|----------|-----------|------------|------------|------------|-----------|-----------|-----------|------------|
|          | blocker     | .        | .         | <u>2</u>   | .          | <u>1</u>   | .         | .         | .         | <u>3</u>   |
|          | critical    | .        | <u>1</u>  | <u>2</u>   | <u>2</u>   | <u>4</u>   | <u>1</u>  | <u>1</u>  | <u>2</u>  | <u>13</u>  |
|          | major       | .        | <u>3</u>  | <u>4</u>   | <u>5</u>   | <u>12</u>  | <u>6</u>  | <u>5</u>  | <u>1</u>  | <u>36</u>  |
|          | normal      | <u>1</u> | <u>57</u> | <u>115</u> | <u>151</u> | <u>158</u> | <u>66</u> | <u>68</u> | <u>27</u> | <u>643</u> |
|          | minor       | .        | <u>1</u>  | <u>1</u>   | .          | <u>4</u>   | <u>1</u>  | .         | .         | <u>7</u>   |
|          | trivial     | .        | .         | <u>2</u>   | .          | <u>2</u>   | <u>1</u>  | <u>2</u>  | .         | <u>7</u>   |
|          | enhancement | .        | <u>3</u>  | <u>9</u>   | <u>8</u>   | <u>7</u>   | <u>3</u>  | .         | .         | <u>30</u>  |
|          | Total       | <u>1</u> | <u>65</u> | <u>135</u> | <u>166</u> | <u>188</u> | <u>78</u> | <u>76</u> | <u>30</u> | <u>739</u> |

# Fixed bugs – 0.2



# Standards



- Client components conforming to CommonJS Asynchronous Module Definition (AMD) format
- HTML5 standard source code
- Participating in specification of CrossFire browser debug protocol
  - <http://getfirebug.com/doc/crossfire/crossfire-splash-2011.pdf>

# UI Usability



- Eclipse User Interface Guidelines followed where applicable to a web UI
- Orion has a completely new icon set with consistent color themes
- Went through several design iterations with graphic designer to come up with Orion 0.2 look
- Emphasis on allowing browser functionality to show through: using normal links, not hijacking the browser context menu, letting browser manage tabs

# Schedule



- Milestones every 6 weeks, 8 cycle duration
  - Became open source shortly after M3
- Tracked schedule
  - All milestones delivered as planned
- End game (release candidate) milestones for 3 cycles
  - Duration reduced from 2-week to 1-week cycles at RC2
  - Relatively short end-game due to Orion being a very new project

# Process



- Orion is developed using an open, transparent, and inclusive process
- Teams rely on Bugzilla, mailing lists and newsgroups for input
- Weekly planning calls conducted with all contributors welcome
- Self-hosted development on <http://orion.eclipse.org>
  - Meeting minutes posted to the orion-dev mailing list
- Publicly available milestone plans:
  - [http://wiki.eclipse.org/Eclipse/Indigo\\_Plan](http://wiki.eclipse.org/Eclipse/Indigo_Plan)
  - [http://wiki.eclipse.org/Orion/Milestone\\_Plan](http://wiki.eclipse.org/Orion/Milestone_Plan)

# Community



- Orion team members are active in Bugzilla, newsgroups, and mailing lists
- Held an Orion Planning Summit in Palo Alto, CA (March 2011)
- Blogs started by Orion committers are active:
  - <http://planetorion.org>
  - <http://planetecclipse.org>
  - Team is using the eclipse-orion IRC channel
  - [irc.freenode.net/#eclipse-orion](http://irc.freenode.net/#eclipse-orion)
  - also see: <http://wiki.eclipse.org/index.php/IRC>
- The Orion team participates in code camps, conference presentations, and tutorials, including
  - EclipseCon 2011, What's Next Conference, jsconf, Eclipse Indigo Demo Camps
- The Orion team interacts with other open source projects, standards bodies, and other projects on eclipse.org, including
  - Mozilla, Dojo Foundation (Dojo framework, Maqetta), jGit, Mylyn, jsdoc, Firebug



# 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.
- Client code (JavaScript) is dual-licensed with EPL and EDL
- Approved 0.2 release log:
  - <http://www.eclipse.org/eclipse/development/project-log-files/eclipse-orion-creation-release-0-2-log.pdf>

# Future Plans



- Shift to 3 or 4 month release cycle to keep up with pace of development in the web community. For example with new browsers shipping every 3 months we can't ship Orion once a year and expect to be current
- Aiming for a 0.3 incubating release in October 2011
- Focus on enabling richer integration, filling in gaps in self-hosting usability

