**FISHEYE-CRUCIBLE**

**Fisheye**

* **FishEye** is a [revision-control](https://en.wikipedia.org/wiki/Revision_control) browser and search engine owned by [Atlassian](https://en.wikipedia.org/wiki/Atlassian), Inc.

### Add a repository

* Click the 'cog' menu in the FishEye header, and choose **Administration** (you'll need to be logged in as an [administrator](https://confluence.atlassian.com/display/FISHEYE/Administering+FishEye) to see this link).
* Click **Repositories** (under 'Repository Settings').
* Click  **Add repository**.
* Complete the wizard. See the following topics for more information:
  + 1. [CVS](https://confluence.atlassian.com/display/FISHEYE/CVS)
    2. [Git](https://confluence.atlassian.com/display/FISHEYE/Git)
    3. [Mercurial](https://confluence.atlassian.com/display/FISHEYE/Mercurial)
    4. [Perforce](https://confluence.atlassian.com/display/FISHEYE/Perforce)
    5. [Subversion](https://confluence.atlassian.com/display/FISHEYE/Subversion)
* Click **Add** to finish.
* You can Hide empty directories and deleted files.
* You can "watch" a repository in FishEye/Crucible. Watching the repository allows you to receive email notifications when changes are made to the repository. Navigate to the repository that you want to watch.

Choose Tools > Watch.

* FishEye allows you to add changesets, files, people and repositories as favourites.
* To see charts and activity of everyone who commits code to your FishEye repositories, click the People tab at the top of the screen

**Using smart commits**

* Smart commits allow repository committers to perform actions such as transitioning JIRA issues or creating Crucible code reviews by embedding specific commands into their commit messages.
* Smart Commit commands

The basic command line syntax for a Smart Commit message is:

**<ignored text> <ISSUE\_KEY> <ignored text> #<COMMAND> <optional COMMAND\_ARGUMENTS>**

Any text between the issue key and the Smart Commit command is ignored.

There are three Smart Commit commands you can use in your commit messages:

* + [comment](https://confluence.atlassian.com/fisheye/using-smart-commits-298976812.html#id-_smart_commits_commands-comment)
  + [time](https://confluence.atlassian.com/fisheye/using-smart-commits-298976812.html#id-_smart_commits_commands-time)
  + [transition](https://confluence.atlassian.com/fisheye/using-smart-commits-298976812.html#id-_smart_commits_commands-transition)

1. **Comment**

|  |  |
| --- | --- |
| * **Description** | * Adds a comment to a JIRA Software issue. |
| * **Syntax** | * <ignored text> ISSUE\_KEY <ignored text> #comment <comment\_string> |
| * **Example** | * JRA-34 #comment corrected indent issue |
| * **Notes** | * The committer's email address must match the email address of a single JIRA Software user with permission to comment on issues in that particular project. |

1. **Time**

|  |  |
| --- | --- |
| * **Description** | * Records [time tracking information](https://confluence.atlassian.com/jirasoftwareserver071/logging-work-on-issues-800707273.html) against an issue. |
| * **Syntax** | * <ignored text> ISSUE\_KEY <ignored text> #time <value>w <value>d <value>h <value>m <comment\_string> |
| * **Example** | * JRA-34 #time 1w 2d 4h 30m Total work logged |
| * **Notes** | * This example records 1 week, 2 days, 4 hours and 30 minutes against the issue, and adds the comment 'Total work logged' in the **Work Log** tab of the issue. * Each value for w, d, h and m can be a decimal number. * The committer's email address must match the email address of a single JIRA Software user with permission to log work on an issue. * Your system administrator must have [enabled time tracking](https://confluence.atlassian.com/adminjiraserver071/configuring-time-tracking-802593086.html) on your JIRA Software instance. |

1. **Workflow transitions**

|  |  |
| --- | --- |
| * **Description** | * Transitions a JIRA Software issue to a particular workflow state. |
| * **Syntax** | * <ignored text> ISSUE\_KEY <ignored text> #<transition\_name> <comment\_string> |
| * **Example** | * JRA-090 #close Fixed this today |

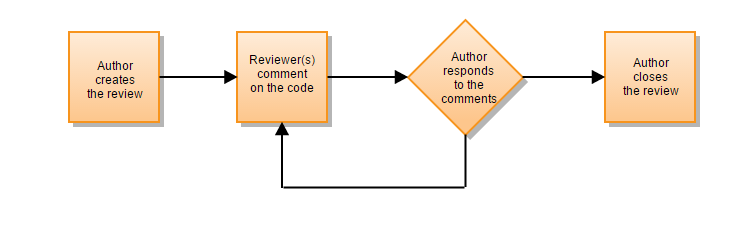
* Viewing an Issue's Changesets in JIRA
  1. Open the relevant issue in JIRA.
  2. Click the Source tab in the Activity section.
  3. The list of changesets related to the issue will display.
* Add the FishEye Charts Gadget to your JIRA dashboard
* Add the FishEye Recent Changesets Gadget to your JIRA dashboard

**CRUCIBLE**

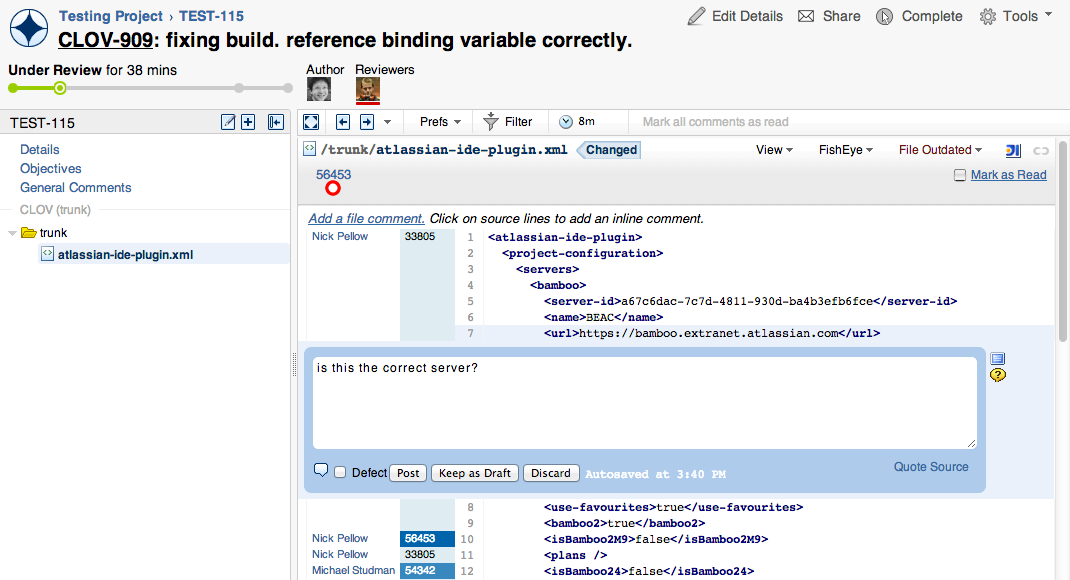
* Atlassian [Crucible](https://confluence.atlassian.com/www.atlassian.com/software/crucible/overview) is the on-premises code review solution for enterprise teams
* Collaborative peer code review tool.

# Starting to use Crucible

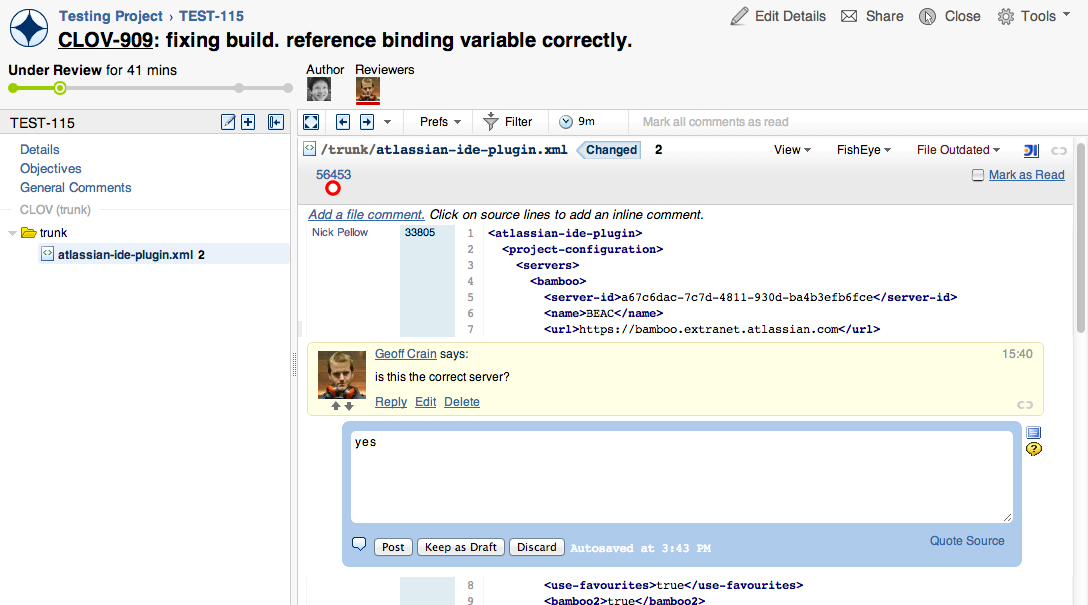
* Repository is already configured in Fisheye.
* Create a project in Crucible
* Click **Add a new project** in the **Projects** listing of the Administration area.(only user having admin access can add projects in Crucible)
* When you create a review you'll want to:
  + Add the files, changesets or other content that you want to be reviewed.
  + Choose the people who you want to be reviewers.
* **Crucible Workflow**



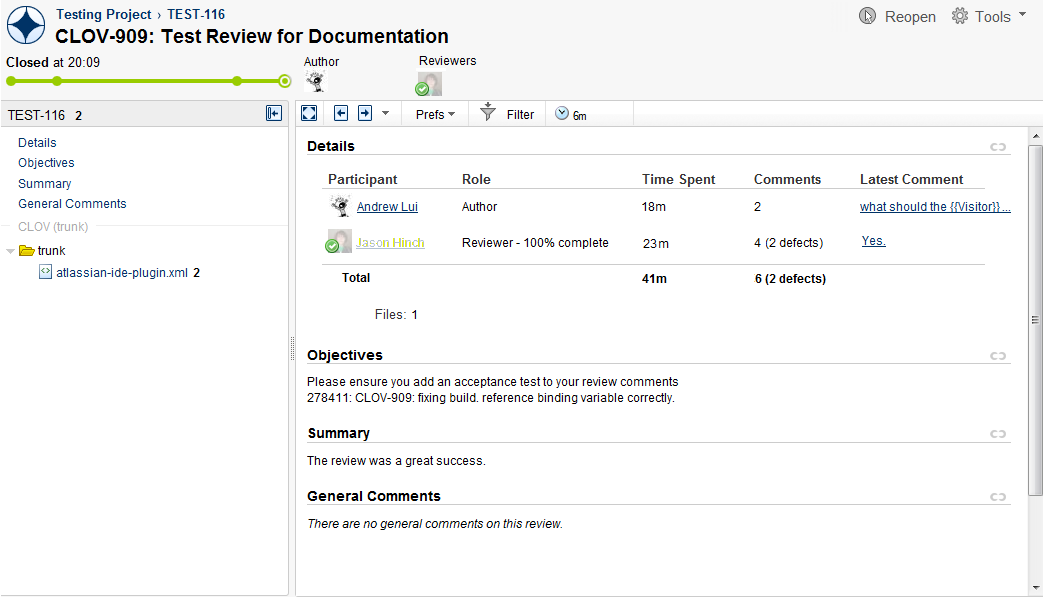
* + Example workflow: Two participant code review
  1. The code author starts the review
  2. The reviewer comments on the code



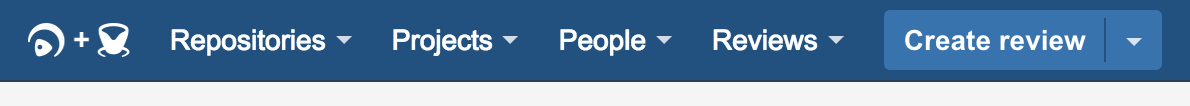
* 1. The author responds to the comments

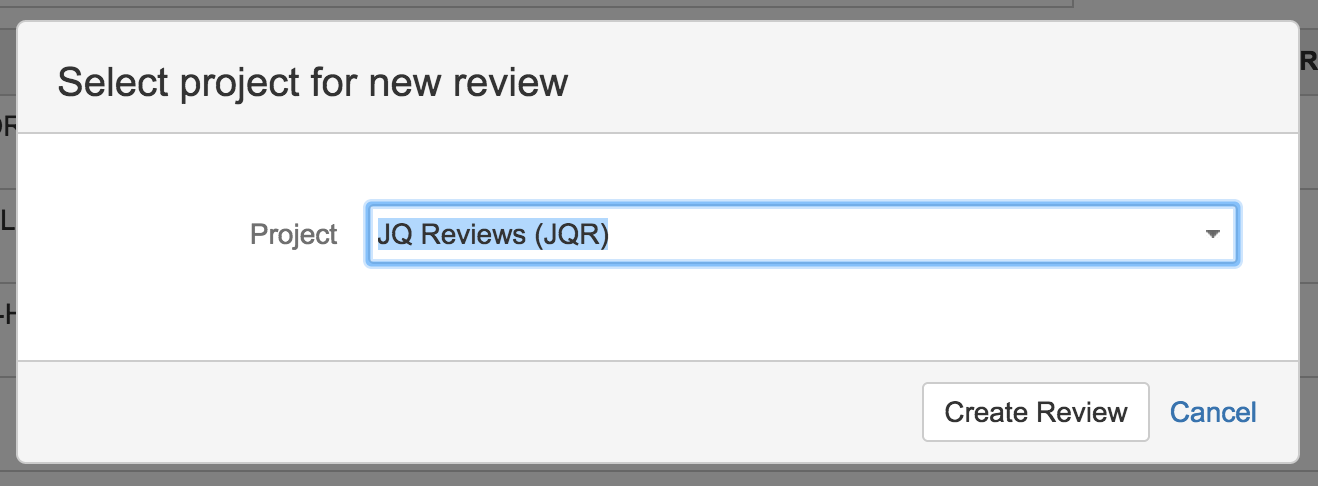


* 1. The author closes the review

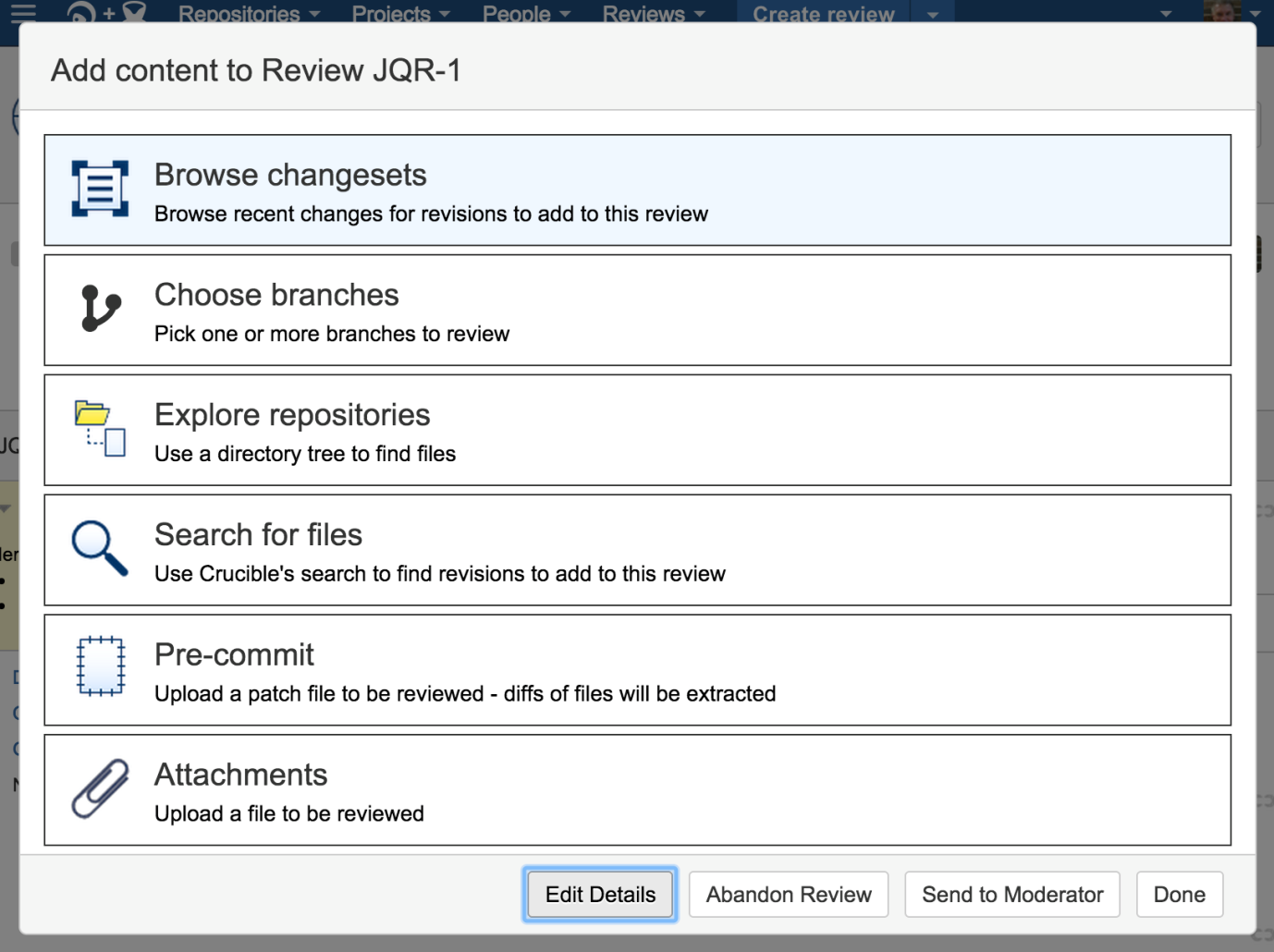


* Click **Create review** to open the review creation form.
* Choose the project in which you want to create the review:

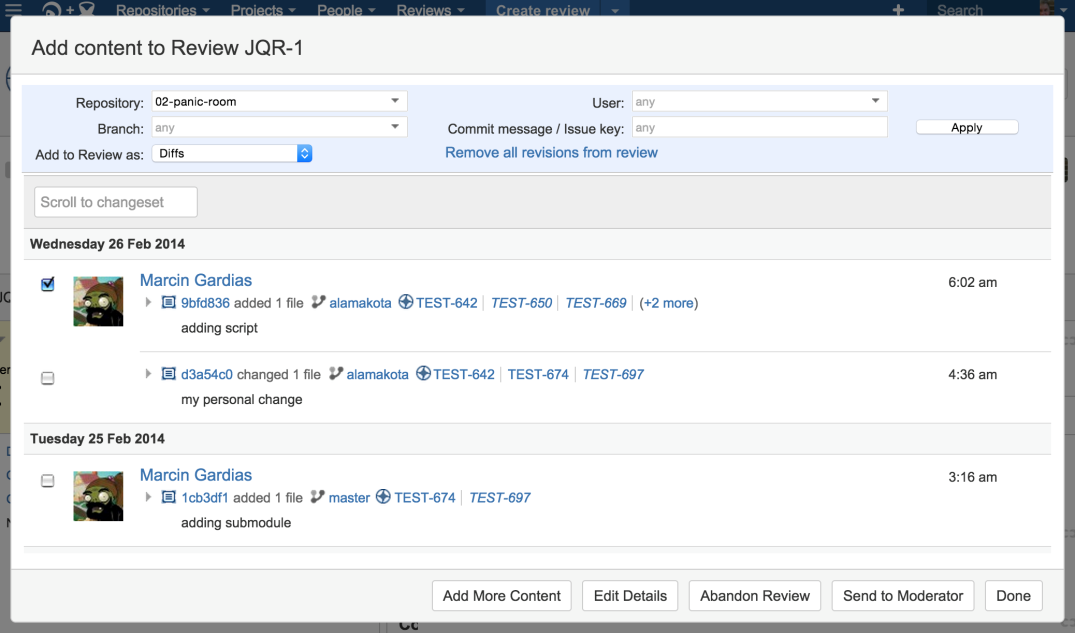




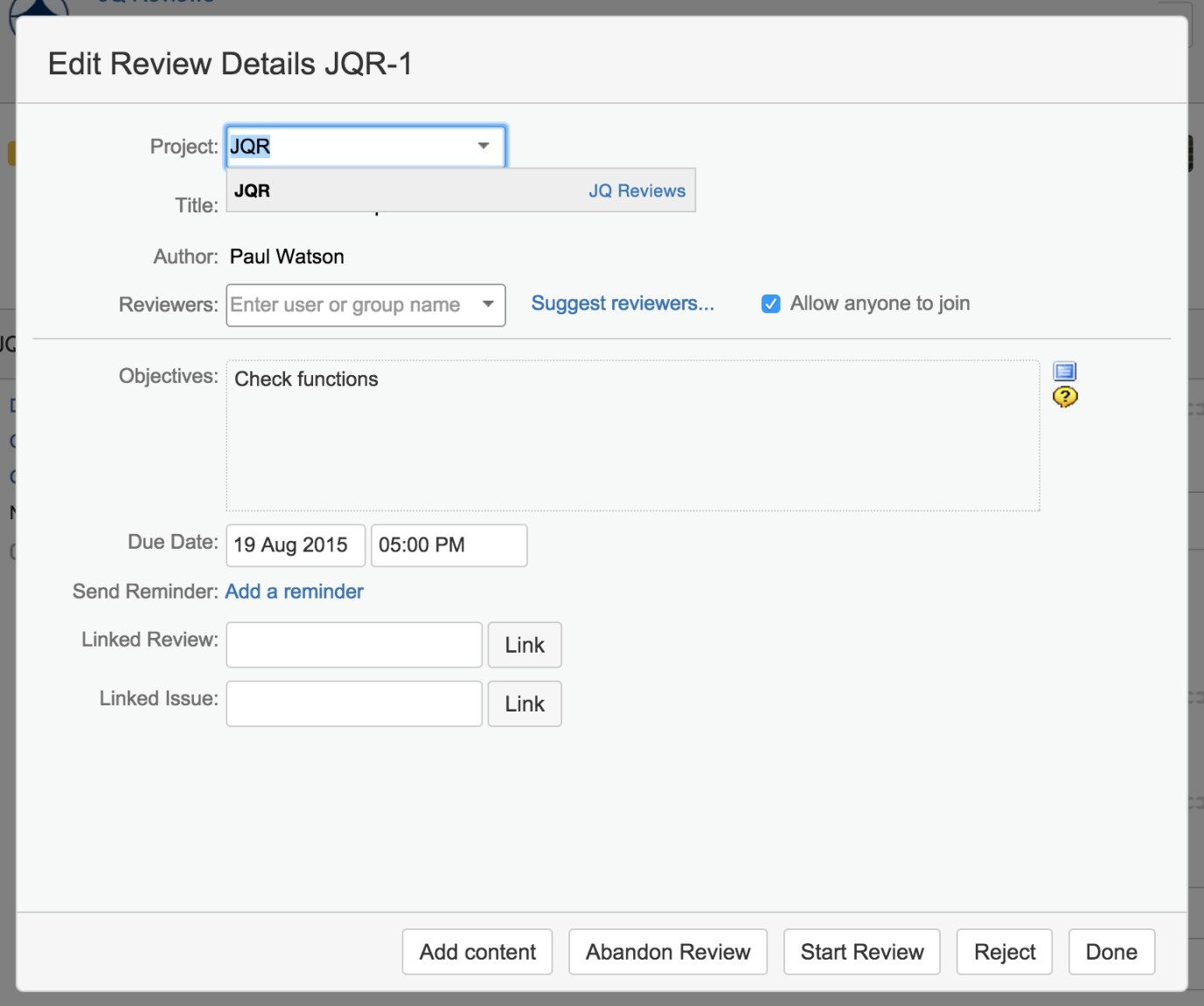
* In the next screen, click **Browse changesets** to see the list of changesets available for the review.



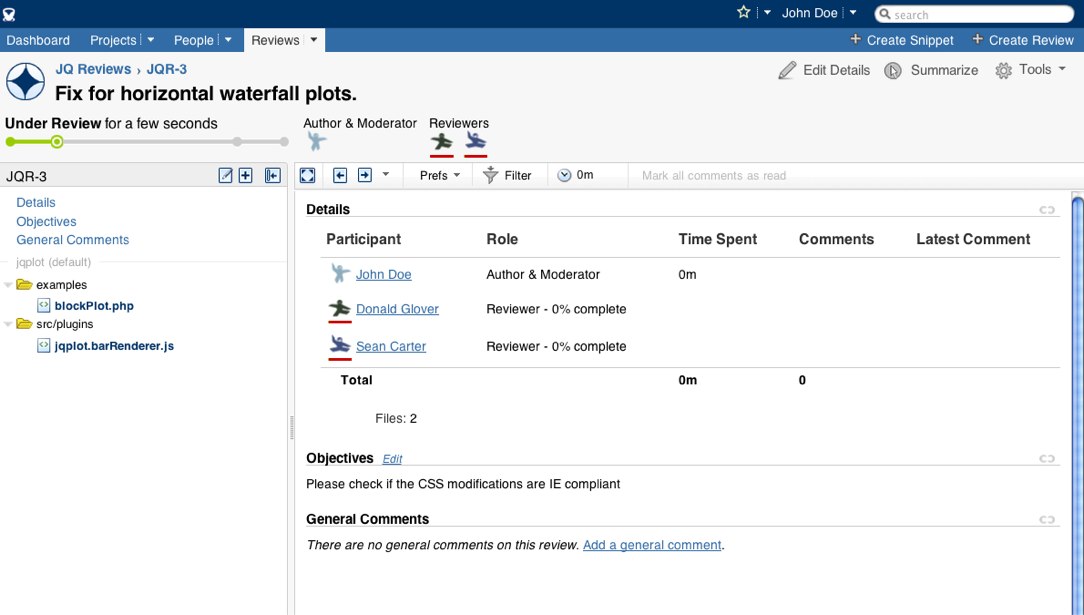
* Select the changesets that you want to be reviewed, then click **Edit Details**:



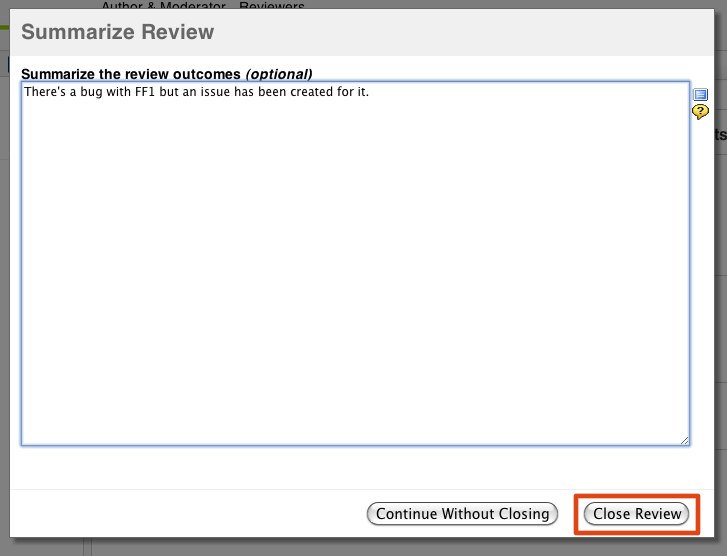
* Now add reviewers and update the review information, then click **Start Review**:



* The review is now created and the reviewers will have been notified that a review is pending.



* In order to close a review, when you are the moderator, you need to click on **Summarize** at the top right and then close the review from the dialog:



# Creating a review from JIRA

* JIRA must be integrated with Crucible before you can do this. We have integrated (172.20.19.28) instance of JIRA wih Fisheye-Crucible.
* Go to the JIRA issue that relates to the work to be reviewed.
* Click **Commits** in the Development panel.
* Click the **FishEye / Crucible** tab.
* Start the process for creating a review for either a single commit, or for all the commits related to the JIRA issue.
* In Crucible, the new review is in edit mode:  
  + The content of the changeset becomes the content (i.e. files) to be reviewed.
  + The author of the commit becomes the [author](https://confluence.atlassian.com/crucible/authors-in-crucible-298977326.html) of the review, if Crucible is aware of this user. Otherwise the [creator](https://confluence.atlassian.com/crucible/creator-298977327.html) of the review becomes the author.
  + The [creator](https://confluence.atlassian.com/crucible/creator-298977327.html) of the review becomes the [moderator](https://confluence.atlassian.com/crucible/moderator-298977339.html).
  + The commit log message is used as both the Title and [Statement of Objective](https://confluence.atlassian.com/crucible/statement-of-objective-298977330.html).
* Choose **Tools** > **Start Review**, in Crucible, when you are ready.
* As you work your way through the files in a review, Crucible tracks the ones you have viewed. Whenever you open a file for review, Crucible will automatically mark it as read



* Crucible will automatically track the time you spend in a Crucible review. When you open a file for review, a counter in the Review Details panel starts. The time is added to your total when you leave the review screen.
* The People tab in Crucible to see charts and activity from people with accounts on the system.

## Considerations when creating reviews

### Review effectiveness

Based on our own experience of over 13000 reviews, we have found that reviews with fewer files and reviewers are more effective. We have seen effects such as:

* Time spent reviewing each file decreases as the number of files increases.
* Reviewers spend less time reviewing as the number of reviewers increases.
* Reviewers find fewer defects as the number of reviewers increases.

We recommend that reviews be created with care to get the best value from them:

* Avoid overloading the review. Reviews should be focused on just a few necessary files.
* Avoid overcrowding the review. Reviewers should be selected with care, and should be guided individually on what to look for.

### Crucible performance

The performance of a Crucible instance can be seriously degraded if very large reviews are created.

To prevent a user from accidentally causing this, Crucible has a limit on the review content size when creating reviews. The limit is 800 file revisions.

Each version of a file in a review counts as one revision – so when a review is created for a single modified file, that is two revisions. Each subsequent change to the file you add to the review is one more revision. A 'whole file' in a review is only one revision.

If you really need to create a larger review, you can get your system administrator to set the crucible.review.content.size.limit property as described on the [JVM system properties](https://confluence.atlassian.com/fisheye/jvm-system-properties-298976942.html) page, but remember that performance will be poor when creating and viewing very large reviews.