Eclipse Contribution Use Cases

<table>
<thead>
<tr>
<th>Use Case : Getting a new project</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>How to do it :</strong></td>
</tr>
<tr>
<td>• go to the project page: <a href="http://projects.eclipse.org">http://projects.eclipse.org</a></td>
</tr>
<tr>
<td>• search for your project</td>
</tr>
<tr>
<td>• read the documentation (sometimes there are specific features to install)</td>
</tr>
<tr>
<td>• If you don't have the resources to download the code, find it on the git project page: <a href="http://git.eclipse.org">http://git.eclipse.org</a></td>
</tr>
</tbody>
</table>

Copyright © 2015, 2016 OPCoach and others, Made available under EPL licence v1.0

Hackathon ECNA 2016
Eclipse Contribution Use Cases

Use Case: Configure gerrit

How to do it:

- In the gerrit configuration settings, generate the https password: [https://git.eclipse.org/r/#/settings/http-password](https://git.eclipse.org/r/#/settings/http-password)

- In the git repository view open the gerrit configuration dialog and enter your gerrit user name:
Use Case: Starting to work on a single bug

How to do it:

- Check if the bug still exists!
- Take the bug on bugzilla (assigned status)
- Fix the code (it is not necessary to create a branch)
- When it is ok, commit with this message:

  Bug XXXXX – Title of the bug  (→ can be copied from bugzilla)
  Some explanations
  Signed-Off: your name  (→ set automatically with button)
  Change-ID: computed

![Commit Message Example](image)
## Use Case: Push to gerrit

You want to publish for a review after a commit

### How to do it:

- First ensure that the repository did not change since you started your modification. → Do a fetch
  - If there are changes, → Do a rebase
  - If there are any conflicts, fix them and commit if needed
  - Test again
- Push (to refs/for/master)

### Possible troubles:

- CLA is not signed
- Email address used for CLA must match the email address in commit message
- Git email address must match the email address in CLA
# Eclipse Contribution Use Cases

## Use Case: My review has been refused

**Cause**: your code is bad or failed

**Solution**: respond to the review comment, and try to fix or abandon the change.

### How to do it:

- On the gerrit web page answer to the reviewer
- Get the code of the commit:
  - if you still work on this commit, continue
  - if necessary fetch the review from gerrit (Team->Remote->Fetch from gerrit...) and get the latest patch using this dialog:

  ![Fetch a change from Gerrit](image)

  Enter the review ID and then 'ctrl Space' to get the patch

  to get the review ID check the review's URL:

  ```
  https://git.eclipse.org/r/#/c/46354/
  ```

- Fix the bug
- **Amend** the commit and push again *(don't change the commit ID)*
## Use Case: Testing the build

**Cause**: you have changed your architecture (features, plugins,...)

**Solution**: just launch a local build

### How to do it:

- launch the maven pom on the project with `mvn clean install`
- anyway gerrit will fail if there are building problems.
## Use Case: My review is accepted but can not be merged on repository

<table>
<thead>
<tr>
<th><strong>Cause</strong></th>
<th>The repository has advanced and the review is behind the current repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Solution</strong></td>
<td>rebase</td>
</tr>
</tbody>
</table>

### How to do it:

- On the gerrit page of the review, click on the 'Rebase button'

### Possible troubles:

Conflict → go back on Eclipse, resolve and push again the review (amend the commit with same commit ID).