Contribute to Eclipse

Eclipse Con France 2014

18 - 19 June 2014
Table des matières

1 - Contribute to Eclipse  ? Yes you can  !
Contribute to Eclipse?
Yes you can!

Introduction
- Olivier Prouvost
- Eclipse trainer and expert in OPCoach
- Committer on E4 tools

How can I contribute?
Contributions can be code or course, but not only!
You can also:
- open bugs on the bugzilla: http://bugs.eclipse.org
- ask or answer questions in forums: http://www.eclipse.org/forums/
- write of fix errors in wiki: http://wiki.eclipse.org
- write comments on the different blogs listed on: http://planeteclipse.org
- watch, comment or provide videos on the youtube channel: https://www.youtube.com/user/EclipseFdn
- attend to Eclipse conferences: http://www.eclipsecon.org
- and of course provide code on git/gerrit system: http://git.eclipse.org

Code contribution... Who can contribute?
- The Eclipse foundation has no Eclipse developpers. They are in member companies.
- Actually anyone can contribute even if you are not in a member company
- You just need an account on eclipse.org and fill a CLA (Contributor License Agreement)
  if you contribute code and you are not a committer
- After some contributions, one committer can propose you as a new committer
- A vote will be organized and if everyone is ok, we become a committer.

What are the prerequisites to contribute to code?
You must have some knowledge on git!
You can get the pro git book for free here:
Read also this very good article (in french): http://www.git-attitude.fr/2014/05/04/bien-utiliser-git-merge-et-rebase/
You must have some knowledge of gerrit and the Eclipse process:
- read the new Lars Vogel’s book (only 6.79 € on amazon :))

1 - http://planeteclipse.org/
Contribute to Eclipse

How can I start to contribute code?

➢ Go to the git.eclipse.org² and choose the project you want
➢ clone the repository (URL are at the bottom):
  ➢ git clone git://git.eclipse.org/gitroot/e4/org.eclipse.e4.tools.git
Importing the project on git

➢ Once the code is cloned, import the projects in a new Eclipse workspace
➢ Set the target platform (depends on the project)
➢ Create a specific launch configuration that use the imported projects
The projects are organized as below:

**Gerrit : the code review process**

- Gerrit is used to review the code.
- It is an intermediate location where the code is pushed before the merge in the master branch.
- For a new commit, a review is created and reviewer can comment (-2,-1,0,+1,+2).
- The first review is an automatic review done by hudson.
- When all reviewers are Ok the code is merged in the master branch.
- Information on gerrit: [https://wiki.eclipse.org/Gerrit](https://wiki.eclipse.org/Gerrit)
A list of reviews (on e4.tools):

- Pushing a commit to gerrit

  For a contributor, pushing the code to gerrit needs to set the push URL for gerrit.

  **Push URL**

  ```
  git push gerrit@github.com:example-user/example-project.git
  ```

A typical review

- A review is created for each new commit pushed
A review contains comments and patches

The ID of the review is used to get the code in your repository

**Getting the code to make the review**

- Just use the ID of the review and select the patch number
- Open the dialog with the command: Team -> Remote -> Fetch from gerrit
- Enter the review ID and click Ctrl Space
- It will fetch the code and create a local specific branch
If the review of the code is not OK, the contributor can change the code and push it again. To do this he must work in the review branch and amend the commit. This will create a new patch in the review and add a comment.
Commit in staging view

**Rebuilding locally the project**
- Reviewing project on gerrit doesn't mean that the update site will be ok
- You must check it using a local build
- To do it, go in the local git repository and run: `mvn clean install`
- You will get the repository in the build/target/repository folder
- Try to install this repository in a new Eclipse instance
- If it is ok -> You can accept the patch and review with a +2

**Conclusion**
- Contributing to Eclipse is now easy and fun with git and gerrit
- Everybody can do it
- If you start with this process, start to fix easy bugs
- And attend to Hackathon to learn more about this!