Review Board

Adam Hornáček
What is Review Board?

- tool that helps with peer review process of source code and more
- web based
- open-source
- written in Python using Django
What is code review?

• process of making pieces of source code available to other developers to review

• intention to catch bugs and design errors early
Why code reviews?

• dramatically helps in quality of the product
• helps catching mistakes early
• which saves time and money
Code review types

• pre-commit review
• post-commit review
• both supported by Review Board
Pre-commit review

- code reviewed before going into the codebase
- diff file uploaded to Review Board
- once approved then committed and pushed
- used more than post-commit review
Pre-commit review best practises

- at least 2 approvals
- every issue resolved or dropped
- wait for approval of all reviewers
- close the review with commit log
Pre-commit workflow

• make local changes
• create review request with the changes
• publish the review request
• wait for feedback from reviewers
• if reviewers found issues
  • update the code and review request
  • jump to point 4 (wait for feedback)
• if enough approvals then push the changes
• close the review request
Post-commit review

- code reviewed after going into the codebase
- code is committed to the repository and reviewed later
- any fixes that need to be made have to be committed again
Supported version control systems

- CVS
- Git
- Mercurial
- Perforce
- Subversion
- Bazaar
- ClearCase
Rest API

- every user can generate a token (read only, full access)
- can be useful while committing – script for automatic commit messages
Usage
Dashboard

• outgoing
  • all
  • open
• incoming
  • open
  • to me
• starred
Review request

• contains
  • basic information
  • all issues
  • all diffs
  • bug number (URL to bug tracker)
  • reviewers
  • pictures or files
Draft

• newly created review request is marked as a draft
• it is not visible to anyone except author
• author has to publish it for it to become public
Reviewers

- review request can be targeted to
  - individual users
  - groups
- Review Board does not enforce that the people listed must review the change
• review request can have specified depends on field (another review request)

• newly created review request treats another review request as its parent

• requires some extensions to VCS (e.g. Mercurial Queues)
Issues

• types
  • info
  • minor
  • major

• anyone can comment on them

• can be marked as dropped or resolved
Diffs

• when updating the diff you need to upload a full diff between the upstream code (what’s present in the repository) and your change

• you can’t just upload a diff of the changes since the previous diff
Diff viewing

- side-by-side diff viewer
- can view changes between selected revisions
- there are many keyboard shortcuts
  - previous/next change
  - previous/next file
  - previous/next comment
  - add comment
Creating new review

- using web (user friendly)
- using ‘rbt post’ command from RBTools (set of command line tools for working with Review Board)
Markdown

- some fields (e.g. description) support rich text using the Markdown language
- below the field there is a link to a page which shows basic Markdown syntax
Editing review requests

• it is possible to change already created review requests (not only diffs)

• the changes are visible to everyone in a nice diff like manner
Not only code!

- it is possible to review file attachments as well
- it is possible to place a new comment on a part of the image
- also ordinary text files and Markdown files
Administration
Authentication

- registration
- LDAP
- NIS
- Active Directory
SSH

• we can specify SSH key which will be used to talk to SSH-backend repositories
Repository hooks

• it is possible to close review requests by tagging commit messages

• to enable this feature you need to add WebHook to the repository in the administration

• tags:
  • Reviewed at http://prak.mff.cuni.cz/reviews/r/123/
  • Review Request #123
Let’s have some fun
First steps

• turn on your computers
• go to http://prak.mff.cuni.cz/reviews/
• register and login
• add yourself to the advanced_sw_tools group
Making changes

- git clone https://github.com/Orviss/Kholinarium
  - quite large (cca 50MB), written in Java
- git clone https://github.com/Orviss/TFTP
  - written in C
- make some changes
- git diff —full-index > ~/my_changes.diff
Creating review request

- create a new review request with the changes
- add proper group or people
- try to upload some files (e.g. image)
- fill out appropriate fields
- try using Markdown
- publish it
Issues, everybody has some issues

• create some issues in your colleague’s requests

• wait for them to create issues in your request as well

• resolve your issues and close the review request
Installation

- straightforward and easy
- detailed guide on the webpage
- generates apache conf for you so you just need to copy it
Installation requirements

• PostgreSQL/MySQL
• Apache web server
• some additional Python libraries
Administration

• check out administration panel
  • http://prak.mff.cuni.cz/reviews/
  • admin/admin
Questions

- do you use in your work something similar? If so then what?

- do you think that code reviews are a good concept for improving code quality?
Thank you :)