Getting Started:

Installing Reviewboard Locally:
Assuming you are running *a clean install of ubuntu 14.04*
Mostly follow this link:
http://alephnullplex.appspot.com/blog/view/2009/07/09/tutorial-installing-review-board-on-ubuntu

Here is a condensed version of commands with some adjustments:

Setting Up the Server

sudo apt-get install apache2
sudo apt-get install libapache2-mod-python
~~sudo a2enmod python~~ You can use this command, but it is probably already enabled

Setting Up Storage

sudo apt-get install mysql-server
sudo apt-get install python-mysqldb
sudo apt-get install memcached
~~sudo apt-get install memcached-dev~~ I have always been unable to locate the pkg
sudo apt-get install python-memcache

Creating Database/User for ReviewBoard

mysql -uroot -p
mysql> create database reviewboard; *Query OK, 1 row affected (0.00 sec)*
mysql> create user 'rb'@'localhost' identified by 'rb_password'; *Query OK, 0 rows affected (0.00 sec)*
mysql> grant all on reviewboard.* to 'rb'@'localhost'; *Query OK, 0 rows affected (0.00 sec)*
mysql> exit *Bye*

Setting up Subversion

sudo apt-get install patch
sudo apt-get install subversion
sudo apt-get install python-svn

Getting ReviewBoard

sudo apt-get install python-setuptools
sudo easy_install reviewboard

Creating Site

sudo rb-site install /var/www/reviewboard the latter is just the directory to install in

The Site settings I used

- Domain Name = localhost
- Root Path = /
- Database Type = (1) mysql
- Database Name = reviewboard
- Database Server = localhost
- Database Username = rb whatever you put earier in the ‘creating database
- Database Password = rb_password whatever you put earier in the ‘creating database’
- Memcache Server = localhost:11211
- Admin Username Creation = admin whatever you want
- Admin Password Creation = password whatever you want
- Admin Email Address = YOUREMAIL@EMAIL.COM whatever you want

Configuring the Site

sudo chown -R www-data /var/www/reviewboard/htdocs/media/uploaded
sudo chown -R www-data /var/www/reviews.example.com/data
sudo cp /var/www/reviewboard/conf/apache-modpython.conf /etc/apache2/sites-available/
cd /etc/apache2/sites-available/
sudo a2dissite *
sudo a2ensite apache-modpython.conf
sudo apt-get install libapache2-mod-wsgi
sudo service apache2 reload
sudo chown -R www-data “/var/www/reviewboard/htdocs/media/ext”
sudo chown -R www-data “/var/www/reviewboard/htdocs/static/ext”

You now have a working site! Try opening a browser and typing in localhost to open the site

Uploading Review Requests

Download RBTools

sudo easy_install RBTools

Adding a repository

On the site, login into the admin account and head over to the admin page (/admin)
Go to Add Repository, fill it out. You will need some sort of repo filled with requests that you will change to upload diffs. I used git and GitHub
Install the corresponding revision control that you use locally (eg. sudo apt-get install git) and set it up
Clone the repo you indiciated on the website add repository page. Make some changes, add and commit them locally.
If you commit them locally, then you will be able to simply use 'rbt post' and it should upload a review request for you.
If you want multiple revisions of diffs on one review request, you will first have to publish the review request on the site, and make more changes to the repo and add/commit them locally. Then use 'rbt post -r=ID_NUM' command with the review request ID_NUM you want to add too. The number is the number after 'site/r/<ID_NUM>' for each review request

Note: It is suggested for development purposes to enable logging in reviewboard. You can enable logging through the admin settings page on the main left sidebar.

Developer Guide:

You do not really need to read it all but the Reviewboard's own official guide is a good reference (link).

File Layout:
Notice how it follows a Django filelayout

- setup.py
- cc_counter/__init__.py
- cc_counter/extension.py
- cc_counter/ccreader.py
- cc_counter/hfcca.py
- cc_counter/urls.py
- cc_counter/views.py (this is the main file you will work on)
- cc_counter/templates/cc_counter/download_ccdata.html
- cc_counter/templates/cc_counter/reviewrequest_cc.html

setup.py

This is the file that creates the python egg. The only two things you really need to ever change here are the *VERSION* values and the *package_data*.

- VERSION - just for self consistency checker
- package_data - Any of the static files that the extension utilizes are indicated here. Ex. the html/js/css files for the views. More reading here.

cc_counter/__init__.py

Generic Python package indicator

cc_counter/extension.py

The “main module” of the extension. It extends the extension class of reviewboard and is where we link up our django files into the reviewboard. This is mainly what Reviewboard’s main guide explains.

The main things we need for the reviewboard extension, however, are simple.

We need to extend the site’s URLs to take in our extension’s django urls/templates. We also need to extend to inject our code into the existing review templates.

To do this we use two hooks in the *__init__* function of our extension: *URLHook* and *TemplateHook*

- URLHook - extends the urls the site takes. Here we have it adding the *urlpatterns* we have in the accompanying cc_counter/urls.py file.
- TemplateHook - injects html to the existing templates of reviewboard.
    - This is the list of all the places we can inject to within the templates
    - You do not need to change this but the urls we apply it to (currently diffview_url_names) are from the reviewboard’s url.py file.
    - Notice that “cc_counter/templates/cc_counter/reviewrequest_cc.html” is going to be our generic

cc_counter/ccreader.py

This is the main Cyclomatic Complexity analyze function. It will wrap all the analyze functions for the extension.

There is one main function: *analyze_file*

- analyze_file(filename) - takes in a file and outputs a list of the dictionary entries listing all the functions and their cyclomatic complexity

The remaining functions are just helper functions

- hfcca_analyze(filename) - wraps the hfcca cyclomotic complexity function
- func_identifier(filename) - takes in the a filename and lists the file’s function names
- output_txt(filename) - outputs a list of strings and outputs it to a text file.

cc_counter/hfcca.py

This is an open source Cyclomatric Complexity analyze function. Takes java, c, and c++ files. You do not really need to but more detail here

cc_counter/urls.py

This is the url file for our django views. Note a few urls

- ‘cc/<review_request_id>/<revision>/<filediff_id>’ - a page that will gives a cyclomatic data for a specific file. Only kept for quick testing reasons right now. Not actually used
- ‘r/<review_request_id>/diff/cc/<revision>/<filediff_id>’ - a page that also gives the cyclomatic data for a specified file. The ‘r/<review_request_id>/diff’ is so that the ajax function for the injected html from TemplateHook will work (It seems to need to be within the same domain to be able to ajax).
- ‘r/<review_request_id>/diff/cc/’ - this is meant to be the generic view that will be ajaxed into the

extension

cc_counter/views.py

This is the real heart of the extension. It is where all the Django views are. There is one main function: *_download_ccdata*

- _download_ccdata(request, review_request_id, revision, filediff_id, ...) - this will output the cyclomatic complexity of the specified revision of a file from the repository.

There are two main views currently: *download_ccdata* and *reviewrequest_cc*:

- download_ccdata(...) - packages _donwload_ccdata's output into a view
- reviewrequest_cc(...) - the supposed eventual generic view of the extension

Code to look at:
c:\projects\reviewboard\reviewboard\diffviewer\chunk_generator.py
    class DiffChunkGenerator(object):
c:\projects\reviewboard\reviewboard\diffviewer\diffutils.py
    def get_original_file(filediff, request, encoding_list):
    def patch(diff, file, filename, request=None):

Path

- reviewboard.diffviewer.diffutils.get_original_file
- reviewboard.reviews.views._download_diff_file
- reviewboard.reviews.views.download_modified_file | reviewboard.reviews.views.download_orig_file
- reviewboard.reviews.urls.download_diff_urls (line 13)
- reviewboard.reviews.urls.diffviewer_revision_urls (line 17)
- reviewboard.reviews.urls.diffviewer_urls (It is on line 51)
- reviewboard.reviews.urls.review_request_urls (line 69)
- reviewboard.reviews.urls.urlpatterns (line 112)
- reviewboard.urls.localsite_urlpatterns (line 77)
- /r/<id_num>/diff/<id_num>/raw