OpenShift Contributor Guidelines

================================

[Summary](#summary)

[Communication](#communication)

\* [Google+](#google)

\* [IRC](#irc)

\* [Mailing list](#mailing-list)

\* [Twitter](#twitter)

\* [Trello](#trello)

[Using git and github](#using-git-and-github)

\* [Squash commits](#squash-commits)

\* [Enhancements and updates](#enhancements-and-updates)

\* [Bug fixes](#bug-fixes)

[Various expectations](#various-expectations)

\* [Fail gracefully](#fail-gracefully)

\* [Secrets in quickstarts](#secrets-in-quickstarts)

\* [Handling sensitive data from gears](#handling-sensitive-data-from-gears)

\* [Provide machine-readable script outputs](#provide-machine-readable-script-outputs)

\* [Hardcoding assumptions from the Online project](#hardcoding-assumptions-from-the-online-project)

Summary

=======

For any project with more than a handful of contributors, it is helpful to

agree on some guidelines for participation. This document walks through

various expectations that have developed for the OpenShift project. Some

may be reactions to mistakes that we are still working to correct, so

we request patience with past transgressions. With awareness that any

open source project guidelines must sometimes bend to allow specific

circumstances, we hope these will be useful guidelines for making this

project successful. That also means guidelines should be limited in order

to avoid becoming TL;DR.

Communication

=============

You do not work in a vacuum. OpenShift developers are happy to help and guide you.

### Google+ ###

The OpenShift Origin community central coordination point is our

[Google+ community](https://plus.google.com/communities/114361859072744017486). Join for news and Q/A.

### IRC ###

OpenShift developers discuss the project in real time on [#openshift-dev on Freenode](http://webchat.freenode.net/?randomnick=1&channels=openshift-dev&uio=d4).

### Mailing list ###

The OpenShift developer mailing list is <dev@lists.openshift.redhat.com> - you may join freely at

<https://lists.openshift.redhat.com/openshiftmm/listinfo/dev>.

### Twitter ###

Follow [@openshift](https://twitter.com/openshift) on Twitter.

### Trello ###

OpenShift Origin development is coordinated in [Trello](https://trello.com/atomicopenshift).

Using git and github

====================

### Squash commits ###

DO:

\* Use `git rebase -i` to combine multiple interim commits into single

coherent commits with helpful commit logs before submitting a pull

request. This keeps our commit logs readable.

AVOID:

\* Pull requests with lots of small commits for tweaks and interim

saves. This is just noise in the log.

\* Squashing commits from unrelated changes into one large commit -

this obscures the purpose and makes rollback of individual changes harder.

### Enhancements and updates ###

DO:

\* `git commit` without `-m` for multiline messages.

\* Format commit messages like this:

~~~

script/class/component: short description of the work done

Link to Trello card, PEP, mailing list, or other planning documentation

Detailed explanation

~~~

This makes later searches of the commit log much easier.

AVOID:

\* Non-descriptive one-line commit messages like this:

~~~

Changed foo to bar

~~~

(where, why?)

~~~

broker card 123

~~~

(You have the card open; at least cut and paste the description and link.)

### Bug fixes ###

DO:

\* `git commit` without `-m` for multiline messages.

\* Format commit messages like this:

~~~

script/class/component: short description of the fix, symptom, or bug

Bug <number>

Bugzilla link <https://bugzilla.redhat.com/show\_bug.cgi?id=number>

Detailed explanation

~~~

Some benefits from this:

1. Adding the short description helps readers search the commit log.

2. Seeing "bug `number`" (but not "BZ" or "Bugzilla"),

our GitHub detector will \*\*automatically\*\* add a comment on the bug

once your commit merges to master (which lets QE know that

your code is in). This can be anywhere in the message.

3. Including the BZ link (which you probably have handy) makes it

that much easier to get to it later (when you probably do not).

AVOID:

\* Non-descriptive commit messages like this:

~~~

Bug <number>

~~~

~~~

Tiny fix

~~~

Various expectations

====================

### Fail gracefully ###

DO:

\* Expect and check for failure conditions and raise or wrap as appropriate.

\* Set a timeout (preferably configurable) on any operation that could block forever.

\* If possible, provide error messages that help the user understand the

context of the error and what to do about it.

### Secrets in quickstarts ###

DO:

\* Salt or otherwise scramble secrets embedded in

quickstarts such that they are unique per application, not

universally shared. An obvious example needing protection would be [Rails

secret\_token.rb](http://www.phenoelit.org/blog/archives/2012/12/21/let\_me\_github\_that\_for\_you/index.html).

See [how to handle this

example](https://github.com/openshift/rails-example#security).

### Handling sensitive data from gears ###

Some environment variables and data from gears might be considered sensitive. Private keys, passwords, and tokens certainly are. Use your judgment about everything else.

AVOID:

\* Your cartridge storing any sensitive data in MongoDB. DB contents are too likely to be displayed indiscreetly.

\* Logging messages outside the gear (e.g. in mcollective.log) that contain sensitive data.

These messages should be scrubbed of sensitive data.

### Provide machine-readable script outputs ###

DO:

\* With any script, provide at least an option for machine-readable output (well-defined e.g. YAML, JSON, XML, a DSL, etc.). The default may be intended for human consumption but options should enable other scripts based on your script.

### Hardcoding assumptions from the Online project ###

AVOID:

\* Coding constants in origin-server for specific gear profiles,

cartridges, external integration points, or anything else we might

reasonably expect an OpenShift administrator to want to customize.

DO:

\* Use configuration files, cartridge manifests, and plugins to enable specific behavior.