
Git and Gerrit For Working With and On Qt

Qt World Summit 2019

Presented by Kevin Funk



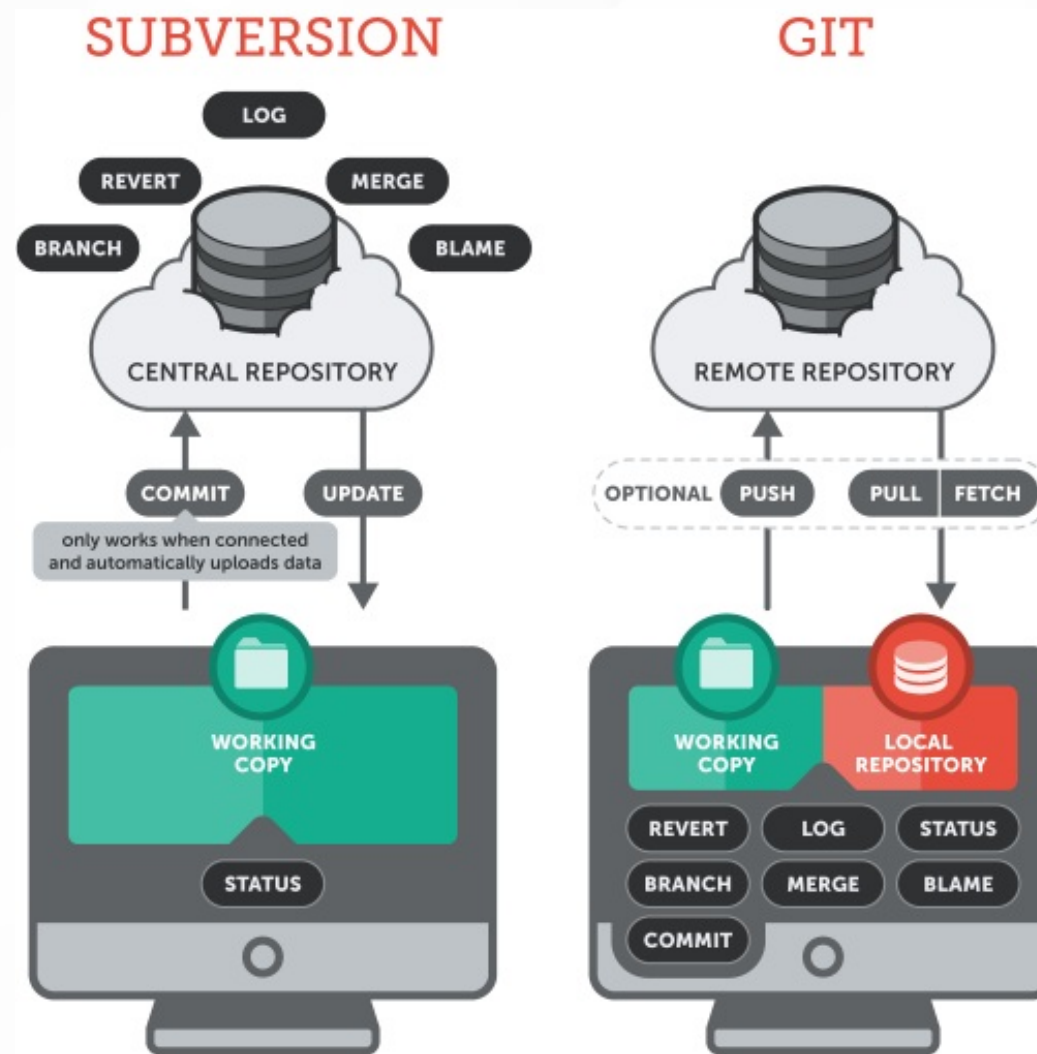
The Qt, OpenGL and C++ Experts

- **Intro to Git**
- Intro to Gerrit for Qt

- Started in 2005
- Created by Linus Torvalds to aid development of the Linux Kernel
- Primarily used for source code management in software development
- Available on all major platforms: Linux, Windows, macOS
- Free and Open Source

- Instant branching and merging
- Small and fast
- Fully distributed
- Data assurance/integrity
- Handles huge projects with ease (hint: Linux Kernel)
- Unique feature: *Staging area*

From Subversion to Git



Source: git-tower.com

- Most powerful: git via command-line
 - But also has the most-steep learning curve
- Graphical (standalone):
 - gitk (cross-platform; 'default')
 - Git Extensions (cross-platform; free, GPL)
 - SourceTree (Windows, macOS; free, proprietary)
 - *many others!* (<https://git-scm.com/downloads/guis/>)
- Graphical (integrated):
 - Git plugins for most Integrated Development Environments

Note that in this guide we'll focus on interaction with the command-line client

- Reference manual: <https://git-scm.com/doc>
- Or for a more brief version:
 - `man git`, OR
 - `git --help` (equivalent to above), OR
 - `git COMMAND --help`

- Reference manual: <https://git-scm.com/doc>
- Or for a more brief version:
 - `man git`, OR
 - `git --help` (equivalent to above), OR
 - `git COMMAND --help`
- Another indispensable source for help:
 - stackoverflow.com

- Tell Git who you are:

```
git config --global user.name "Your Name"  
git config --global user.email "your@email.com"
```

- Global settings file: `~/.gitconfig`
- Project settings file: `$PROJECTDIR/.git/config`

Windows users

Git looks for the `.gitconfig` file in the `%HOME%` directory:
`C:\Users\%USER%` for most people

USE CASE: Initial Check-out

- Example (checking out from a remote location):
`git clone git://code.qt.io/qt/qt5.git`
- Example (... and a specific branch):
`git clone git://code.qt.io/qt/qt5.git -b 5.12`

- See your changes:
`git status` OR
`git diff` OR
`gitk`
- Mark the files you want to check in:
`git add <files>`
- Check in the marked files:
`git commit` OR
`git commit -m"checkin message"`
- Fetch latest version from remote server:
`git pull`
- Push the changes to the server:
`git push`
- You may commit all files using **-a** option:
`git commit -a`

Demo

- Intro to Git
- **Intro to Gerrit for Qt**

How to use Gerrit by example of contributing to Qt

- Qt's development occurs in the open in repositories at <https://codereview.qt-project.org> (a Gerrit code review instance).
- They are mirrored read-only at code.qt.io
- The Qt project provides free hosting to projects related to Qt which have been incubated and accepted by the community
- Cloning Qt:

```
git clone git://code.qt.io/qt/qt5.git
```
- See http://wiki.qt.io/Building_Qt_5_from_Git for build instructions

- Prerequisites
 - Qt Account (SSO)
 - <https://login.qt.io/register>
 - Access to Gerrit
 - <https://codereview.qt-project.org/>
 - Access to JIRA (for closing bugs, etc.)
 - <https://bugreports.qt.io/>
 - Signed Contributor Agreement
 - Can be done in Gerrit
 - More info: <https://www.qt.io/legal-contribution-agreement-qt>

Contributing to Qt via Gerrit (cont'd)

- You might want to submit a patch back to Qt's repositories.
- To do this you need to sign up for a Gerrit account at <https://codereview.qt-project.org/>.
- Set up for using Gerrit on your local clone.
 - `git remote add gerrit GERRIT_URL; ...`
- Commit locally as usual.
- Push local commits to gerrit.

```
git push gerrit HEAD:refs/for/5.9
```

or

```
git push gerrit HEAD:refs/for/dev
```

- A remote hook from Gerrit will print an URL.
- Open the resulting URL(s) and follow the review procedures.

Gerrit Setup for Qt development

- Configure SSH:

```
1 Host codereview.qt-project.org
2     Port 29418
3     User
```

- Add a commit hook:

```
scp -p codereview.qt-project.org:hooks/commit-msg .git/hooks
```

- Or copy from <http://codereview.qt-project.org/tools/hooks/commit-msg>
- Add a new "remote" to your clone:

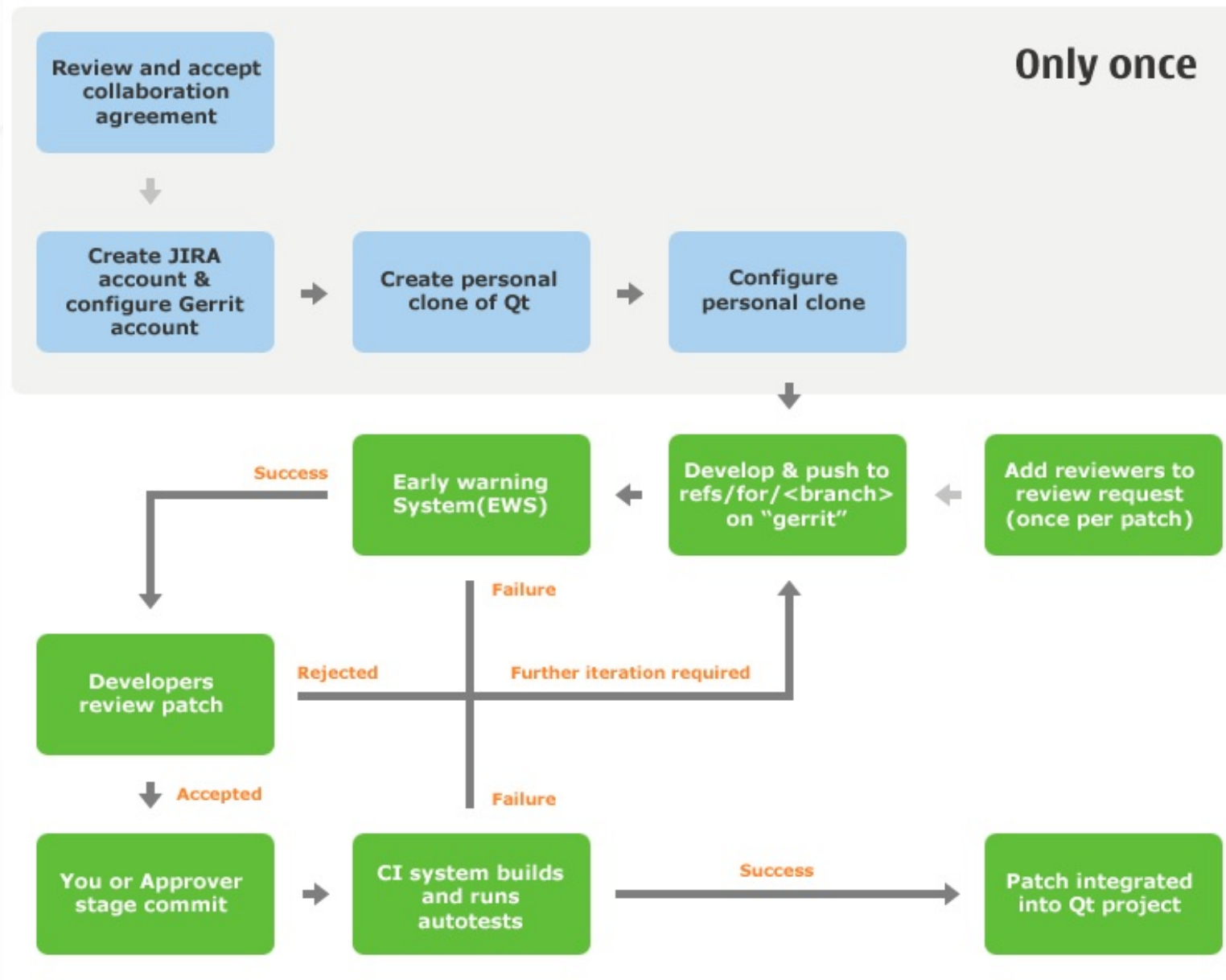
```
git remote add gerrit ssh://codereview.qt-project.org/qt/qt5
```

- See <http://wiki.qt.io/Setting-up-Gerrit>

Adding reviewers to your Change Request

- Find out who worked on the source file(s) you are changing
 - `git log [path]`
 - `git blame [path]`
- Or at least add the module maintainer
 - <https://wiki.qt.io/Maintainers>

Change Request: Life Cycle



- Entry point: <https://www.qt.io/contribute-to-qt>
 - https://wiki.qt.io/Setting_up_Gerrit
 - https://wiki.qt.io/Gerrit_Introduction
- Helpers
 - Git gpush script: <https://wiki.qt.io/Git-gpush-scripts>

Questions?

www.kdab.com
kevin.funk@kdab.com

KDAB is now offering Git trainings and workshops!

