Google Gerrit

By - Rajesh Kumar

Twitter - RajeshKumarIN

Agenda – Introduction to Code Review

- Benefits of Code Review
- Overview of Gerrit
- Refresh of advanced Git commands needed for review
  - Exercise: advanced git commands

- Pushing changes for review
- Push validation (Jira, uploadvalidator, Jenkins)
- Review using Gerrit UX
  - Exercise: create a change and review workflow
  - Exercise: validation with Jenkins and submit

- Wrap-up: Code Review Etiquette, Q&A

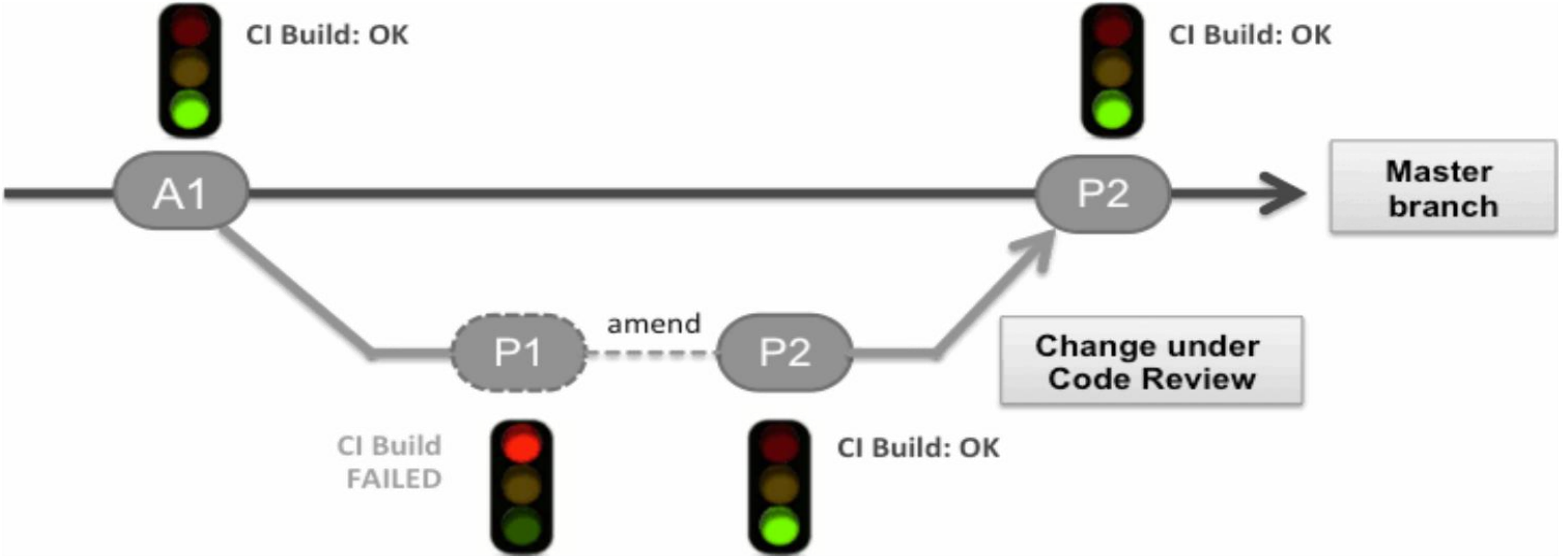
Agenda – Practicing Code Review workflow

- Exercise: adding a new patch-set to existing changes
- Exercise: multiple patch-sets reviews
- Exercise: rebase and conflicts resolution

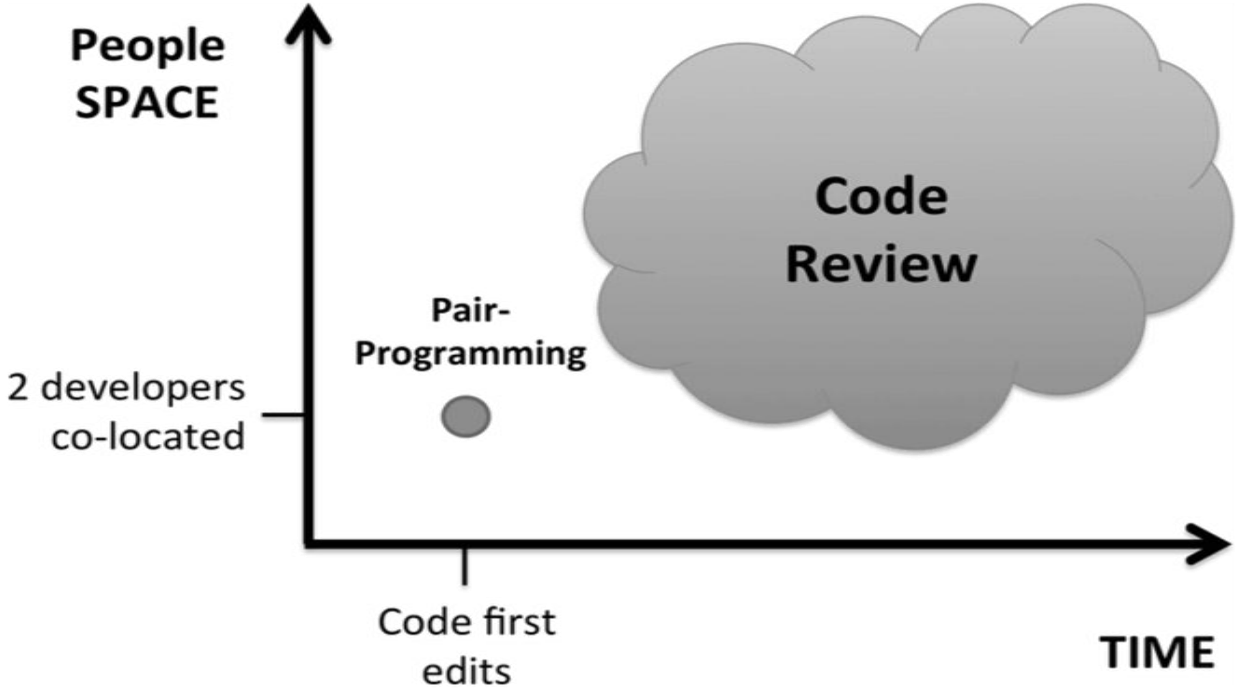
- Wrap-up, Q&A

Four Benefits of Code Review

1. Build Stability



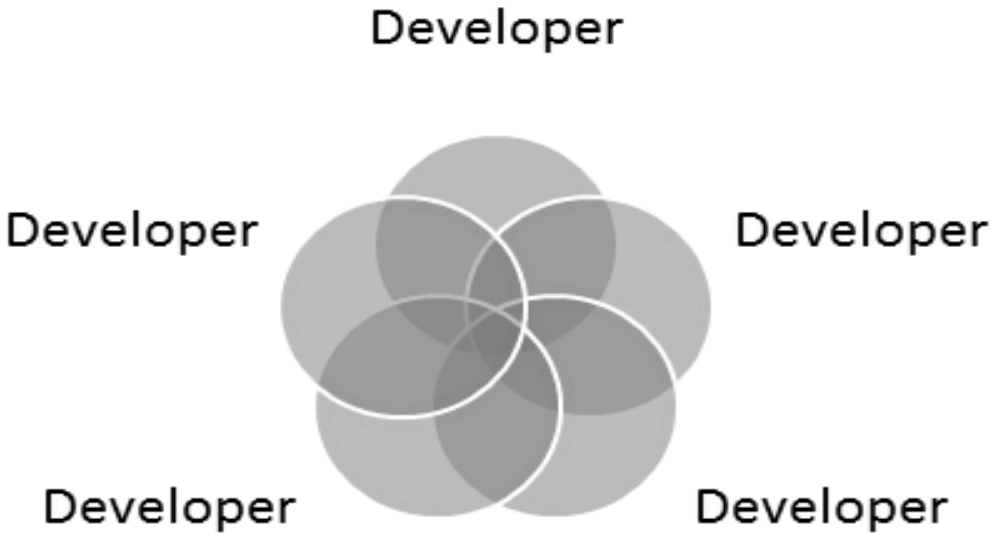
Four Benefits of Code Review



2. Knowledge Sharing

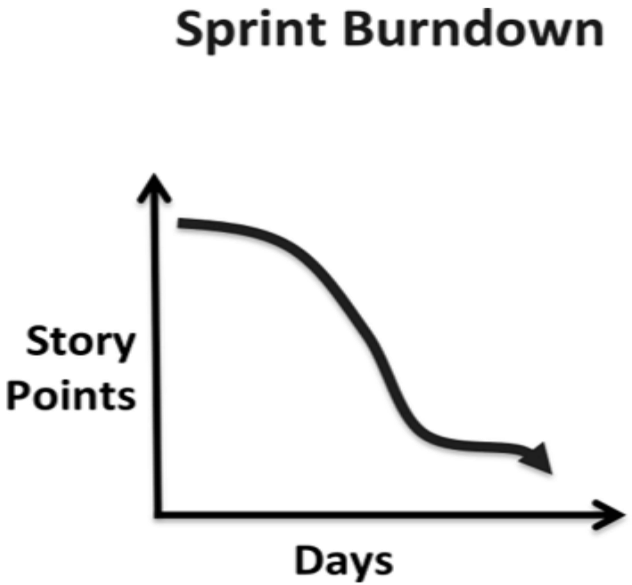
Four Benefits of Code Review

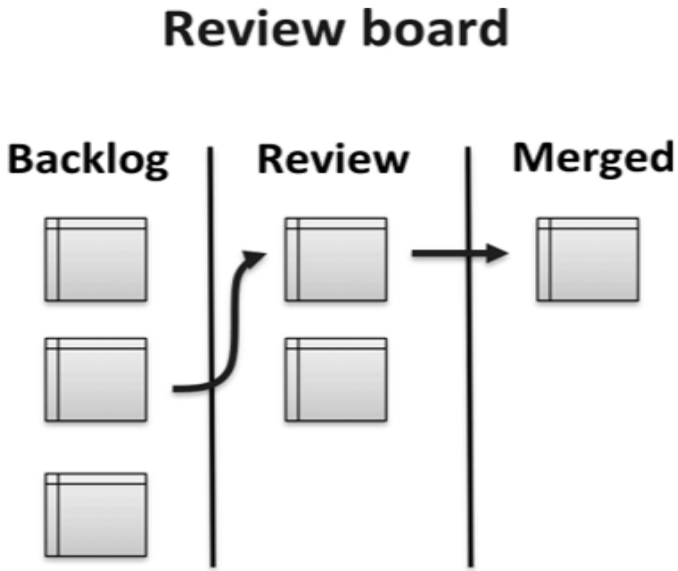
3. Team Dynamics – Collective Ownership



Four Benefits of Code Review

4. Qualitative Code Selection





Overview of Gerrit

Introducing Gerrit Code Review

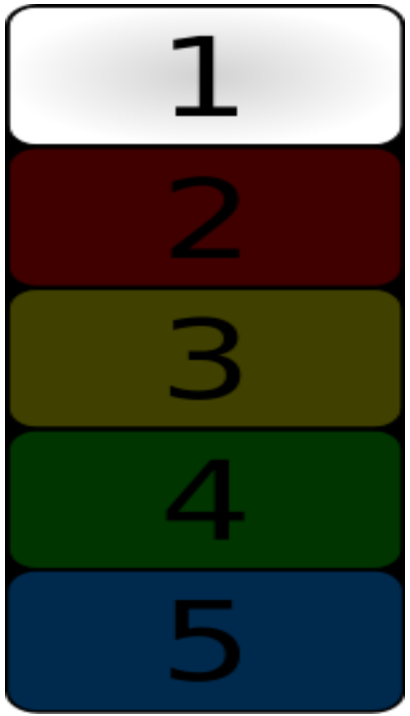
Continuous Integration is GOOD

- Live “weather forecast” of the project
- Detect and resolve conflicts earlier
- Bring TDD to life
- Enforce collective code ownership
- … and much more

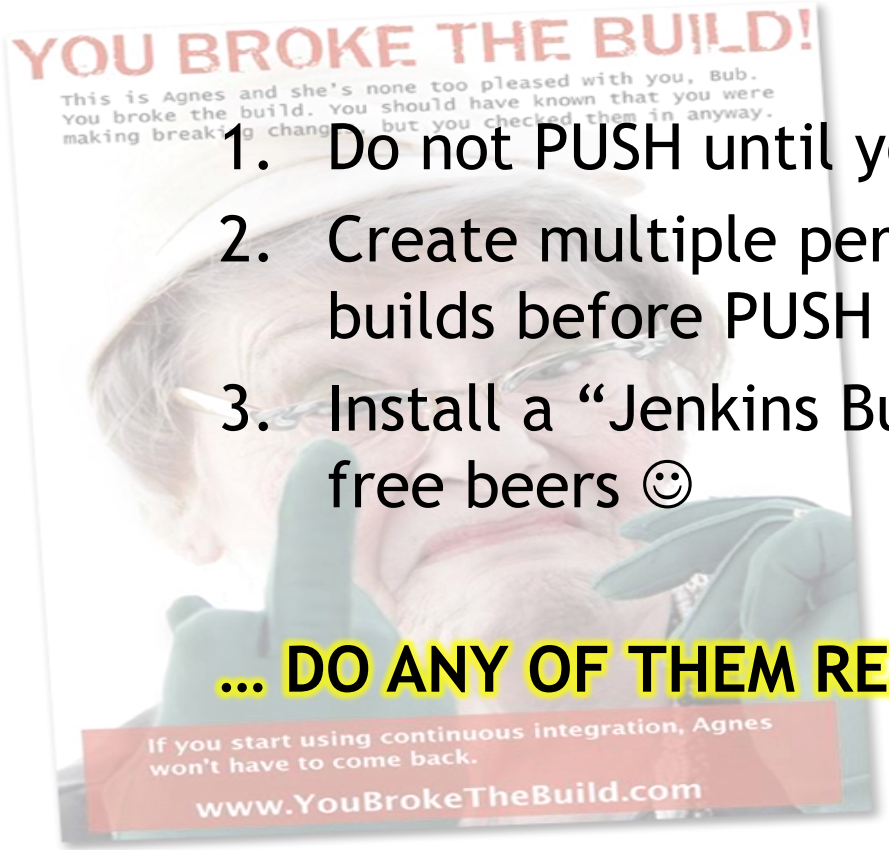
Breaking the build id BAD !

- All devs gets tests broken
- Build stops
- Test are NOT executed

… all team goes at “DEFCON1” to fix it ASAP !!



Broken build dilemma: how to avoid it ?



1. Do not PUSH until you're 100% sure of GREEN bar
2. Create multiple personal CI builds and validate builds before PUSH
3. Install a "Jenkins Build Game" plug-in and get free beers ☺

… DO ANY OF THEM REALLY WORK ?

What is Gerrit

- Earlier Name - Rietveld
- Web based code review tool
- Repository management for Git
- Workflow management
- Integrated access control
- Extend using server side plugins

Developement

- Rietveld written in Python
- Gerrit 2.x written in Java (Java EE Servlet) with SQL
- Gerrit uses Google Web Toolkit to generate front-end JavaScript code from Java source
- Gerrit 2.0-rc0 on January 13 2009

More

- 100% pure Java SSH and HTTP Git backend
  - Powered by JGit
- GWT Web-view administration
  - Users and Groups
  - Project and branch security
  - Git repository browsing
- Git repository replication engine
- Code collaboration and review
- Code validation through Jenkins Triggers

Gerrit, brief history

Google Mondrian

↓

Rietveld

↓

Gerrit Rietveld

↓

gerrit

The idea: Guido Van Rossum

- Code-review for Perforce
- Porting to SVN and OpenSourced
- Python-based

2008 - Project fork for AOSP (Shawn Pearce / Joe Onorato)

- Name changed to Gerrit Rietveld
- Based on Git
- Set of “patches” on original Guido’s Rietveld project

2009 - Gerrit 2, the Java + GWT rewriting (Shawn Pearce)

What is Jgit?

JGit is a lightweight, pure Java library implementation of the Git version control system – including repository access routines, network protocols, and core version control algorithms.

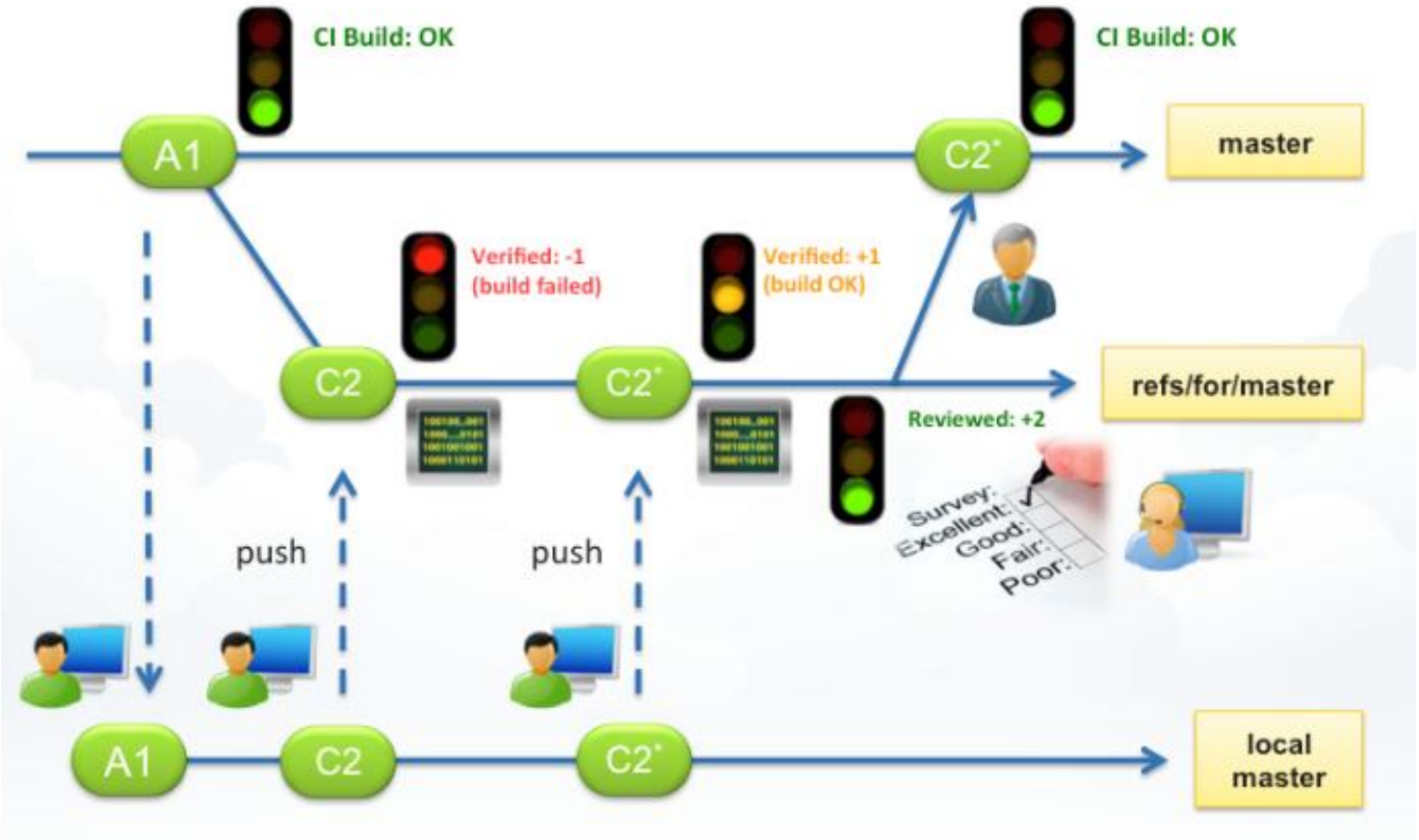
JGit is a relatively full-featured implementation of Git written in Java and is widely used in the Java community. The JGit project is under the Eclipse umbrella

Specifiations

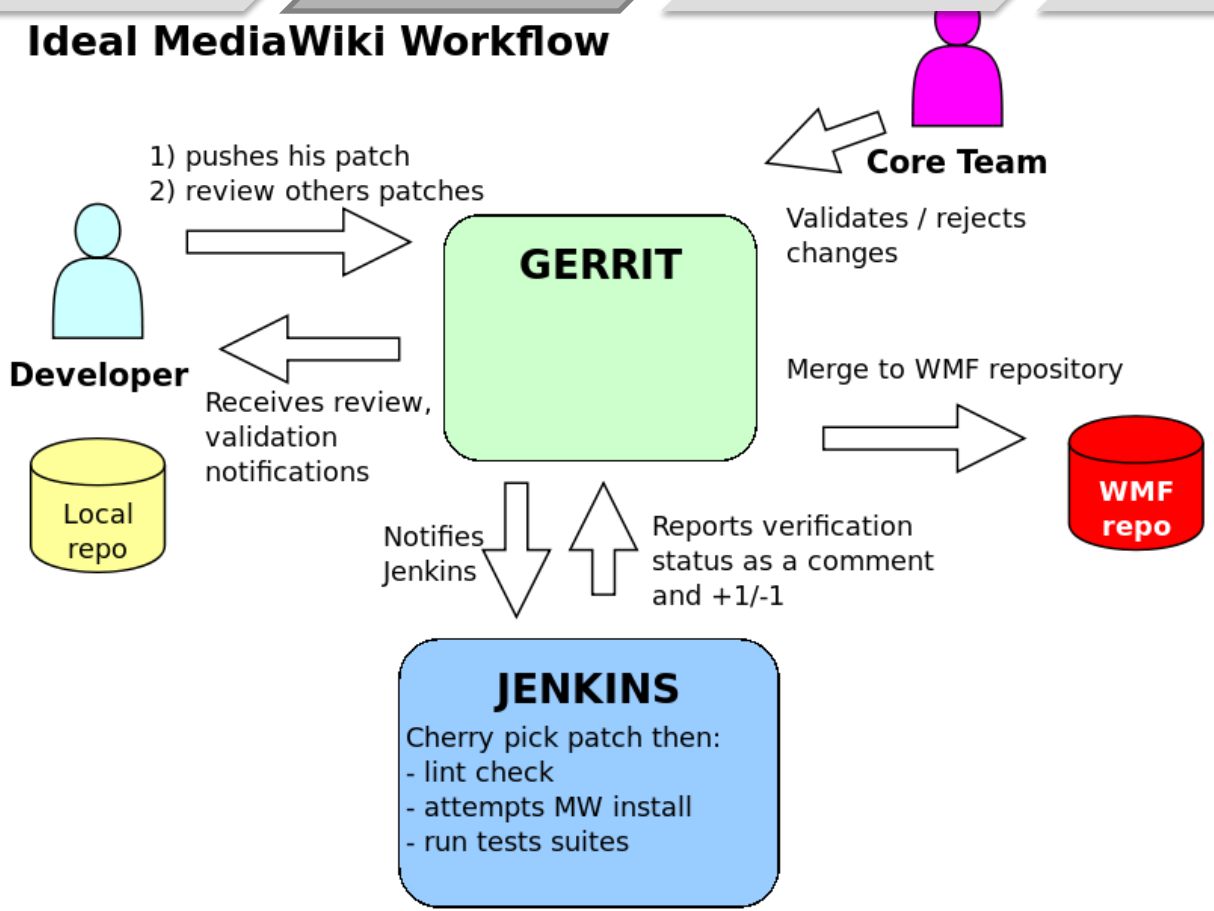
- Apache 2.0
- 93 included dependencies (Mostly Apache2.0, BSD, MPL1.1., EPL)
- Latest release 2.16.7
- 10 years old

- Web based review
- Read code. Comment on code. Make better code

- Fine Grained ACLs
- Repository and branch level access controls.
- SSH, HTTP


CI Build: OK
CI Build: OK
A1
C2'
master
Verified: -1
(build failed)
Verified: +1
(build OK)
C2
C2'
refs/for/master
Reviewed: +2
Survey:
Excellent:
Good:
Fair:
Poor:
push
push
A1
C2
C2'
local master

Ideal MediaWiki Workflow



- System Groups
  - Anonymous Users
  - Project Owners
  - Change Owner
  - Registered Users
- Predefined Groups
  - Administrators
  - Non-Interactive Users
- Account Groups
- LDAP Groups

Examples of typical roles in a project

- Contributor
- Developer
- CI system
- Integrator
- Project owner
- Administrator

Access Categories

- Abandon
- Create Reference
- Delete Reference
- Forge Author
- Forge Committer
- Forge Server
- Owner
- Push
  - Direct Push
  - Upload To Code Review
- Add Patch Set
- Push Merge Commits
- Create Annotated Tag
- Create Signed Tag
- Read
- Rebase
- Remove Reviewer
- Review Labels
- Submit
- Submit (On Behalf Of)
- View Private Changes
- Delete Own Changes
- Edit Topic Name
- Edit Hashtags
- Edit Assignee

Gerrit Code Review can be extended and further customized by installing server-side plugins.

Core Plugins

- codemirror-editor
- commit-message-length-validator
- download-commands
- hooks
- replication
- reviewnotes
- singleusergroup

Other Plugins

- admin-console
- analytics
- avatars-external
- avatars-gravatar
- branch-network
- changemessage
- delete-project
- egit
- emoticons
- gitblit
- github
- gitiles
- healthcheck
- imagare
- imagare
- importer
- Issue Tracker System Plugins
- javamelody
- labelui
- menuextender
- metrics-reporter-elasticsearch
- metrics-reporter-graphite
- metrics-reporter-jmx
- metrics-reporter-prometheus
- motd
- OAuth authentication provider
- owners
- project-download-commands
- quota

Gerrit = pre-commit review

Review before push to target branch

- Pros:
  - Make sure every commit is good
  - Enforce company standards (e.g. Jira association)
  - Keep Build and Team code stability
  - Tightly integrated with Git

- Cons:
  - Slows down code integration
  - Needs tooling
  - Git proficiency

Pre-requisite

- JDK 1.8
- Gerrit uses NoteDB as the storage backend
- Can optionally use an external database such as MySQL or PostgreSQL.
- Gerrit is not yet compatible with Java 9 or newer at this time.

NoteDb

NoteDb is the next generation of Gerrit storage backend, which replaces the traditional SQL backend for change and account metadata with storing data in the same repository as code changes.

NoteDb will be the only database format supported by Gerrit 3.0.

How to Install Gerrit

- WAR file (Tomcat, Jetty etc)
- Self contained daemon (with embedded Jetty)

Download Gerrit

- Current and past binary releases of Gerrit can be obtained from the Gerrit Releases site.
- https://gerrit-releases.storage.googleapis.com/index.html

Database Setup

- If you choose NoteDB, Gerrit will automatically set up the embedded NoteDB database as backend so no set up and all configuration is necessary.
- Other Supported Database
  - H2
  - Apache Derby
  - PostgreSQL
  - MySQL
  - MariaDB
  - SAP MaxDB
  - Oracle
  - DB2
  - SAP HANA

MySQL

- https://gerrit-documentation.storage.googleapis.com/Documentation/2.16.7/install.html#createdb_mysql

Installation support

- Database Support
  - For accounts, groups, metadata in database
  - PostgreSQL, MySQL, or embedded H2
- Directory Support
  - OpenID authentication (google, yahoo etc)
  - LDAP (Users and Groups)
- HTTP
  - Embedded Jetty
  - Deployment of WAR to Tomcat, Jetty, etc.
- SSH
  - Embedded pure java SSH daemon on port 29418 (configurable)

What is GitHub?

GitHub is the best place to share code with friends, co-workers, classmates, and complete strangers. Over three million people use GitHub to build amazing things together.

What is Bitbucket?

Bitbucket gives teams one place to plan projects, collaborate on code, test and deploy, all with free private Git repositories. Teams choose Bitbucket because it has a superior Jira integration, built-in CI/CD, & is free for up to 5 users.

What is Gerrit Code Review?

Gerrit is a self-hosted pre-commit code review tool. It serves as a Git hosting server with option to comment incoming changes. It is highly configurable and extensible with default guarding policies, webhooks, project access control and more.

Why do developers choose GitHub?

- 1752 Open source friendly
- 1456 Easy source control
- 1238 Nice UI
- 1116 Great for team collaboration
- 851 Easy setup
- 492 Issue tracker
- 470 Remote team collaboration
- 463 Great community
- 443 Great way to share
- 434 Pull request and features planning

Why do you like GitHub?

Why do developers choose Bitbucket?

- 896 Free private repos
- 394 Simple setup
- 342 Nice ui and tools
- 336 Unlimited private repositories
- 237 Affordable git hosting
- 121 Integrates with many apis and services
- 119 Reliable uptime
- 83 Nice gui
- 81 Pull requests and code reviews
- 57 Very customisable

Why do you like Bitbucket?

Why do developers choose Gerrit Code Review?

- 7 Cleaner repository story
- 7 Code review
- 7 Good workflow
- 6 Good integration with Jenkins
- 6 Open source
- 4 Unlimited repo support

Why do you like Gerrit Code Review?

Why do you like Gerrit Code Revie

Submit

Lets get startted - Workflow

- Central git repository for entire team
- Code review are at commit level
- 5 New commits, 5 new reviews
- Search the dashboards show relevant review spanning repositories

Workflow cont..

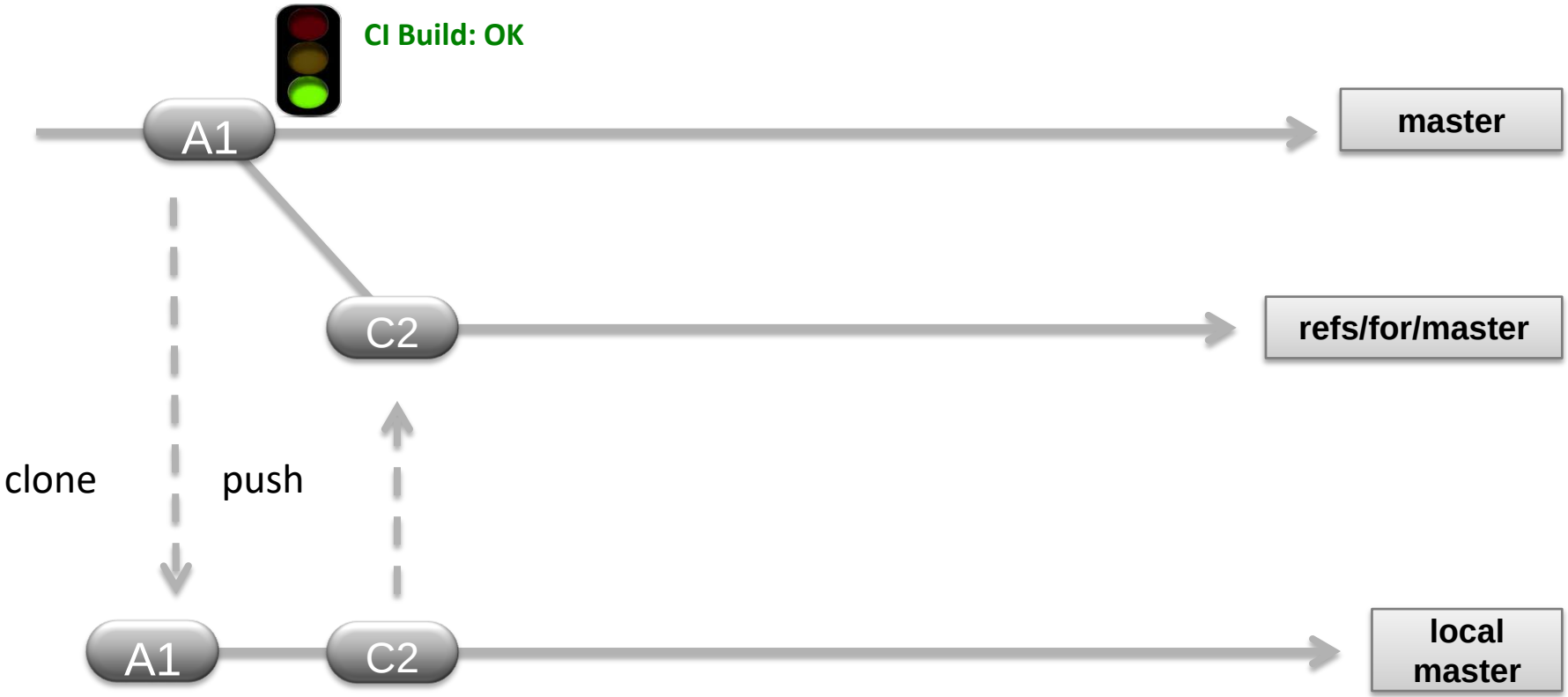
- git clone url
- git commit –m”this is fix”
- git push origin HEAD:refs/for/master

Gerrit: high level workflow

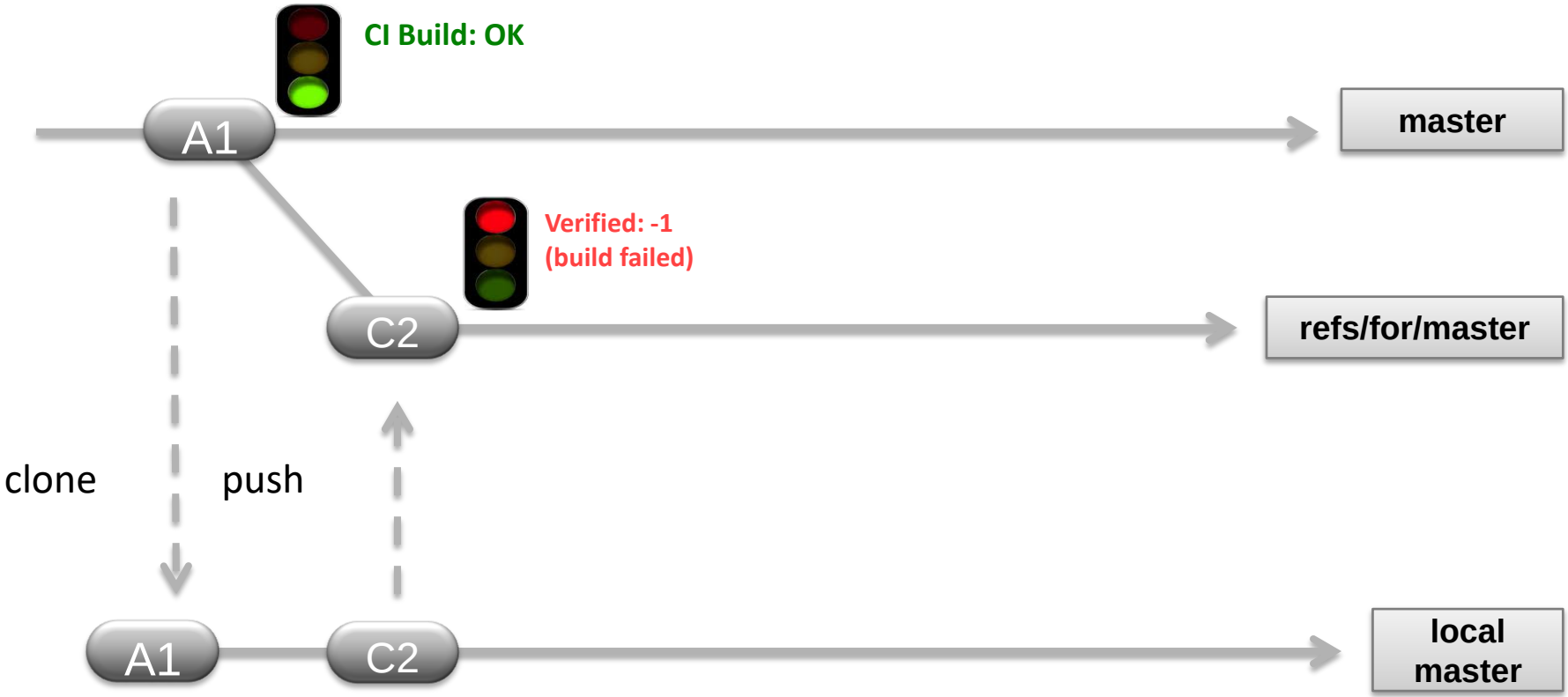
Gerrit: high level workflow

Gerrit: high level workflow

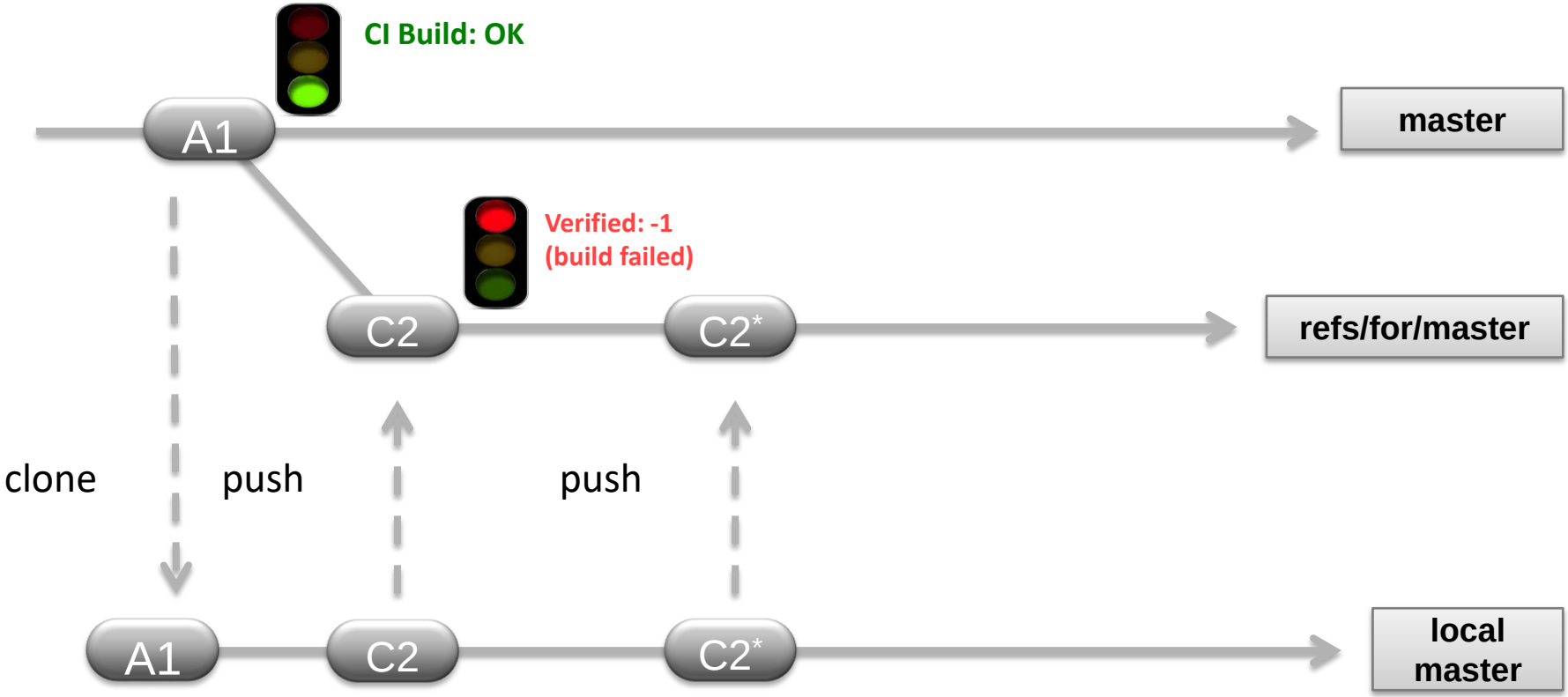
Gerrit: high level workflow



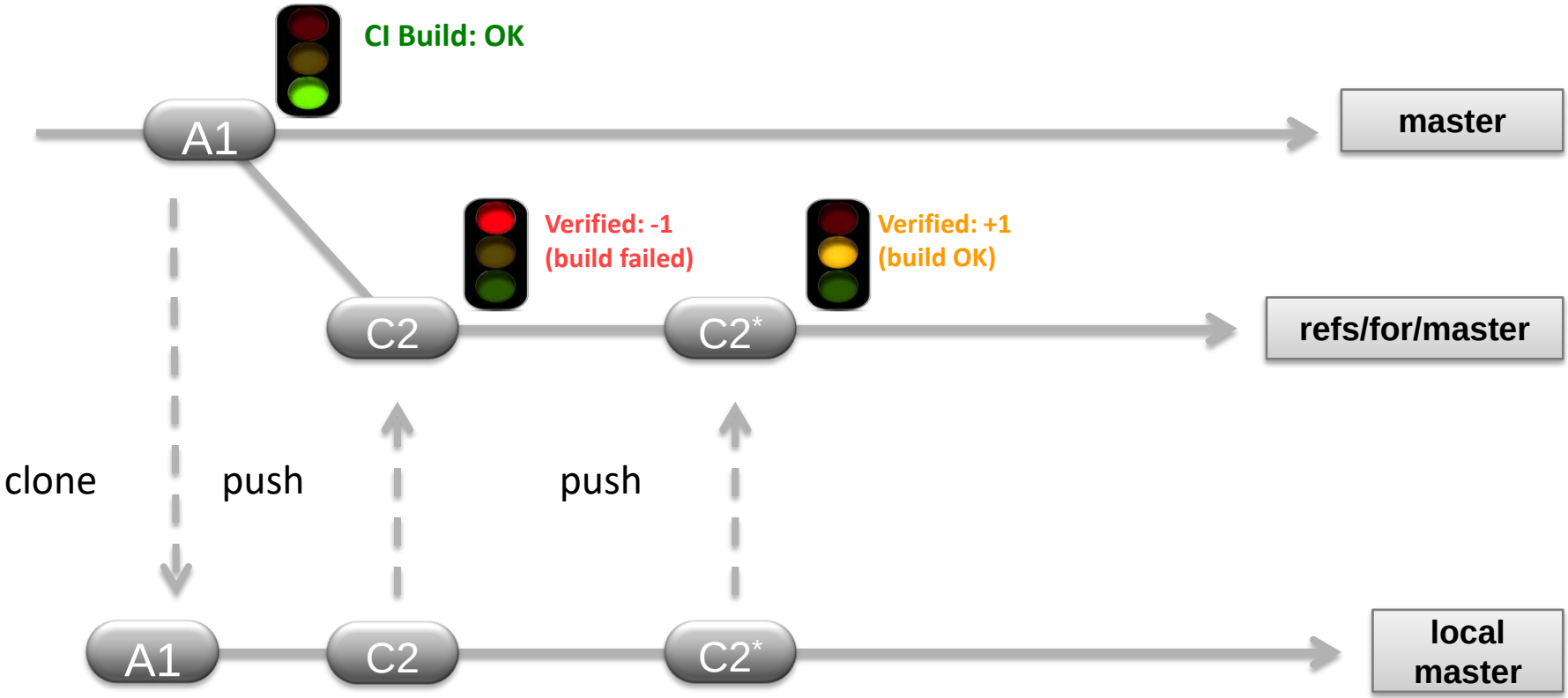
Gerrit: high level workflow



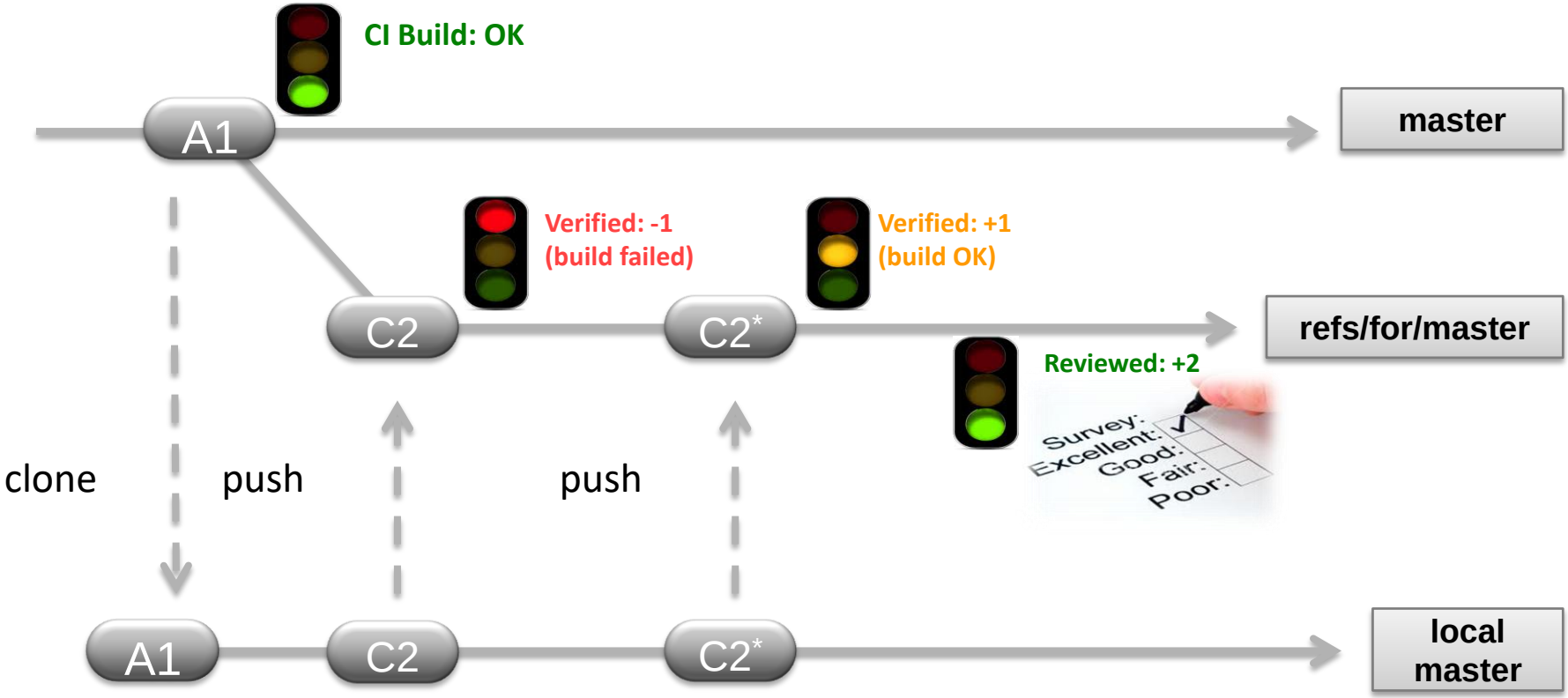
Gerrit: high level workflow



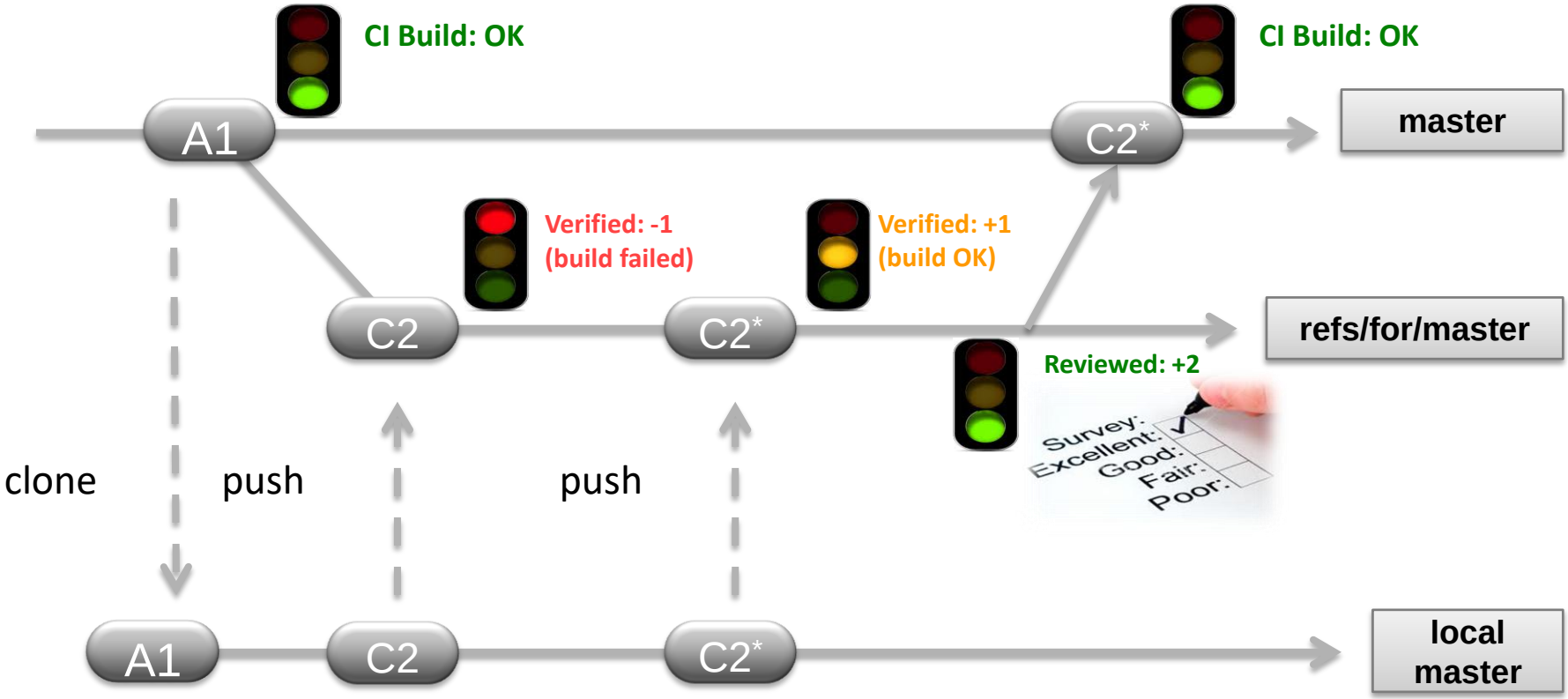
Gerrit: high level workflow



Gerrit: high level workflow



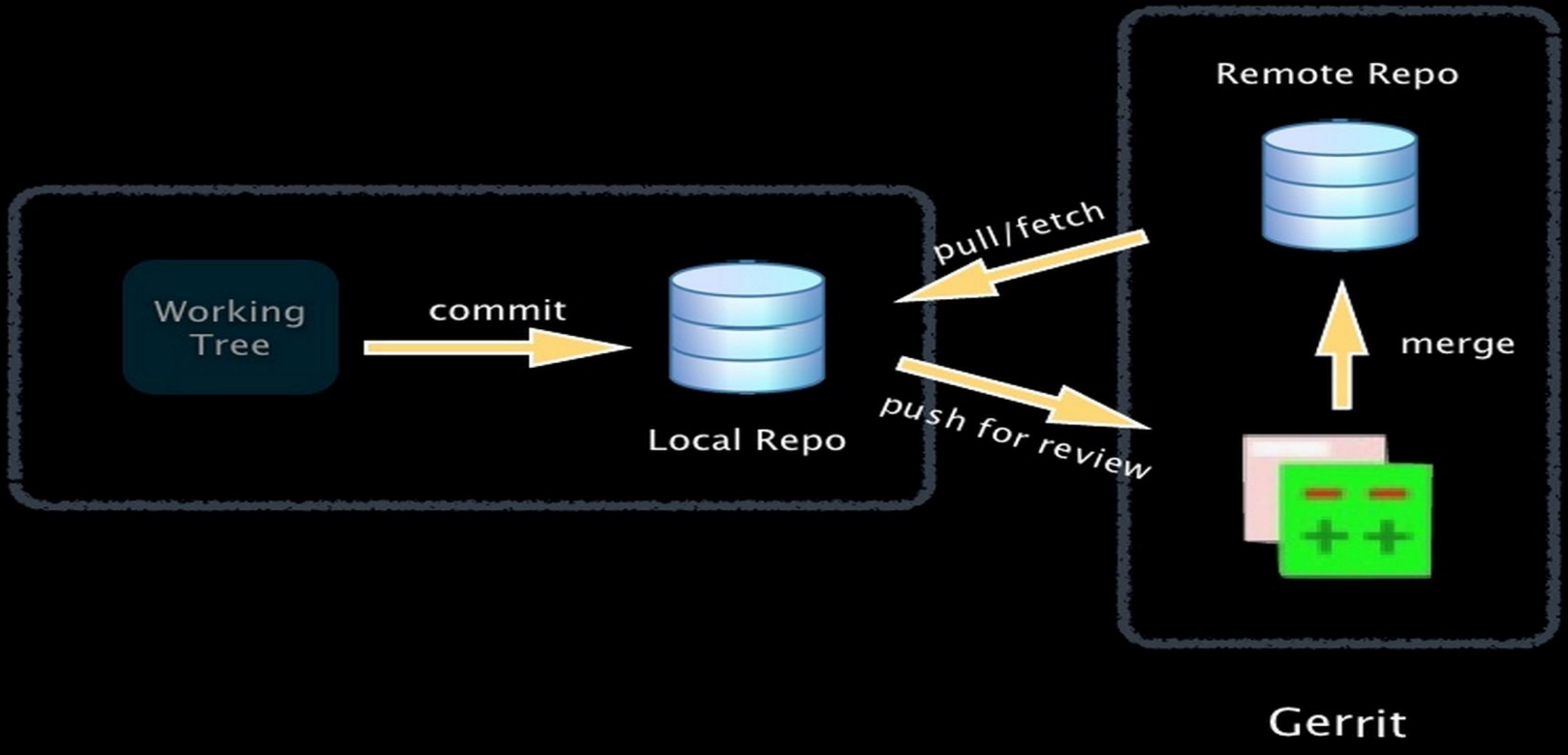
Gerrit: high level workflow



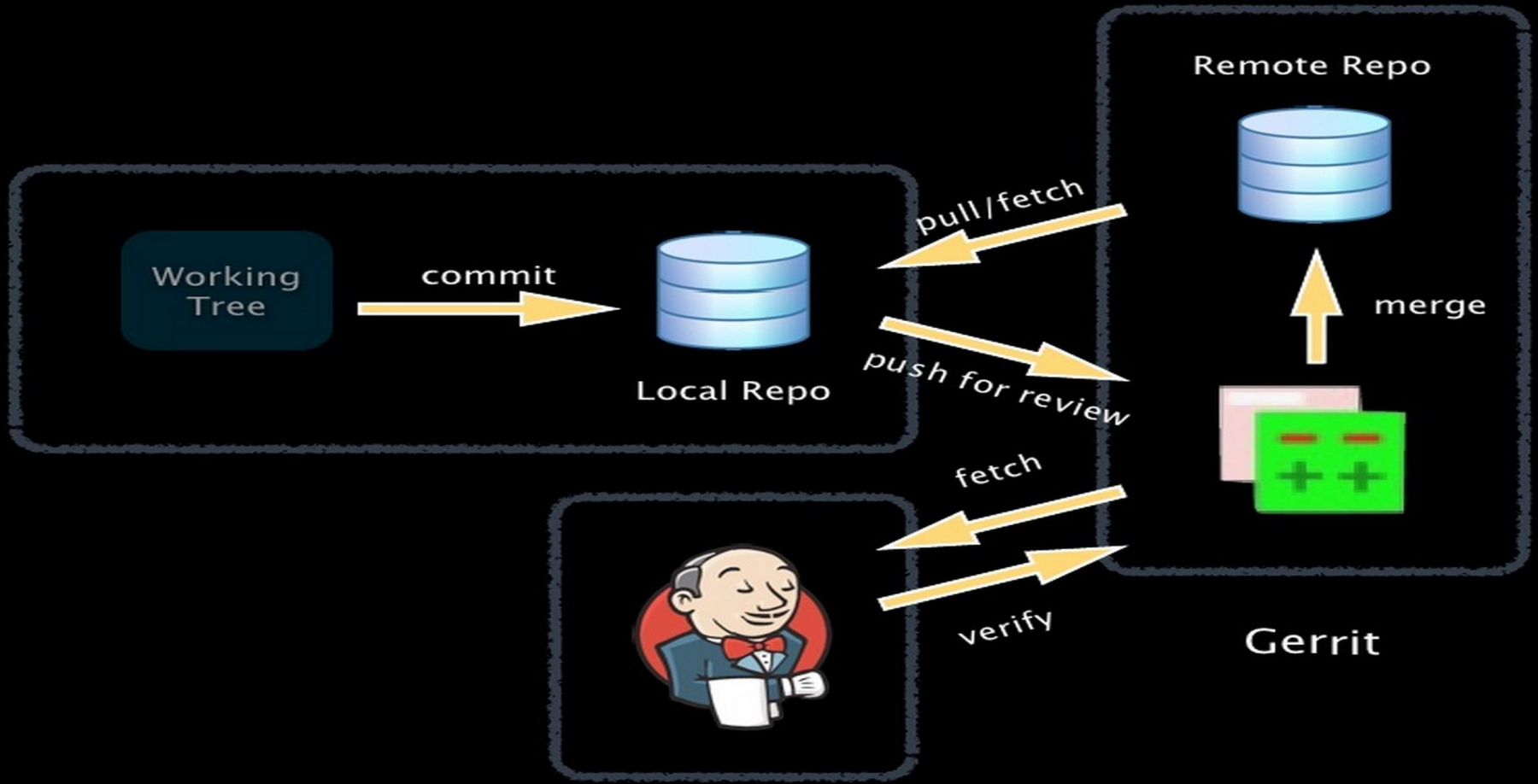

ONE
MORE
TIME

Git standalone

Git and Gerrit

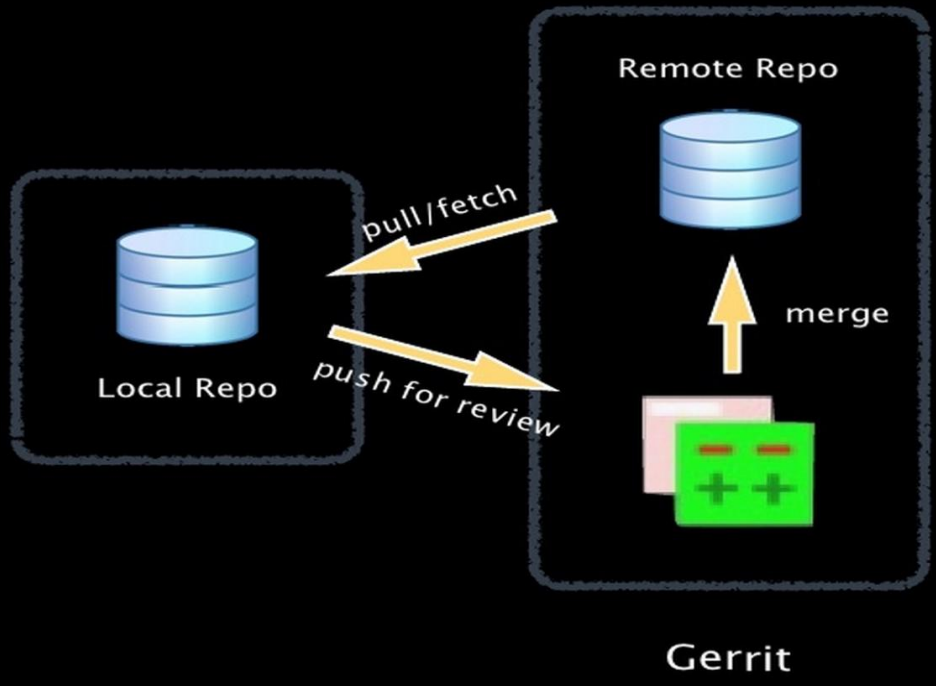


Git, Gerrit and CI



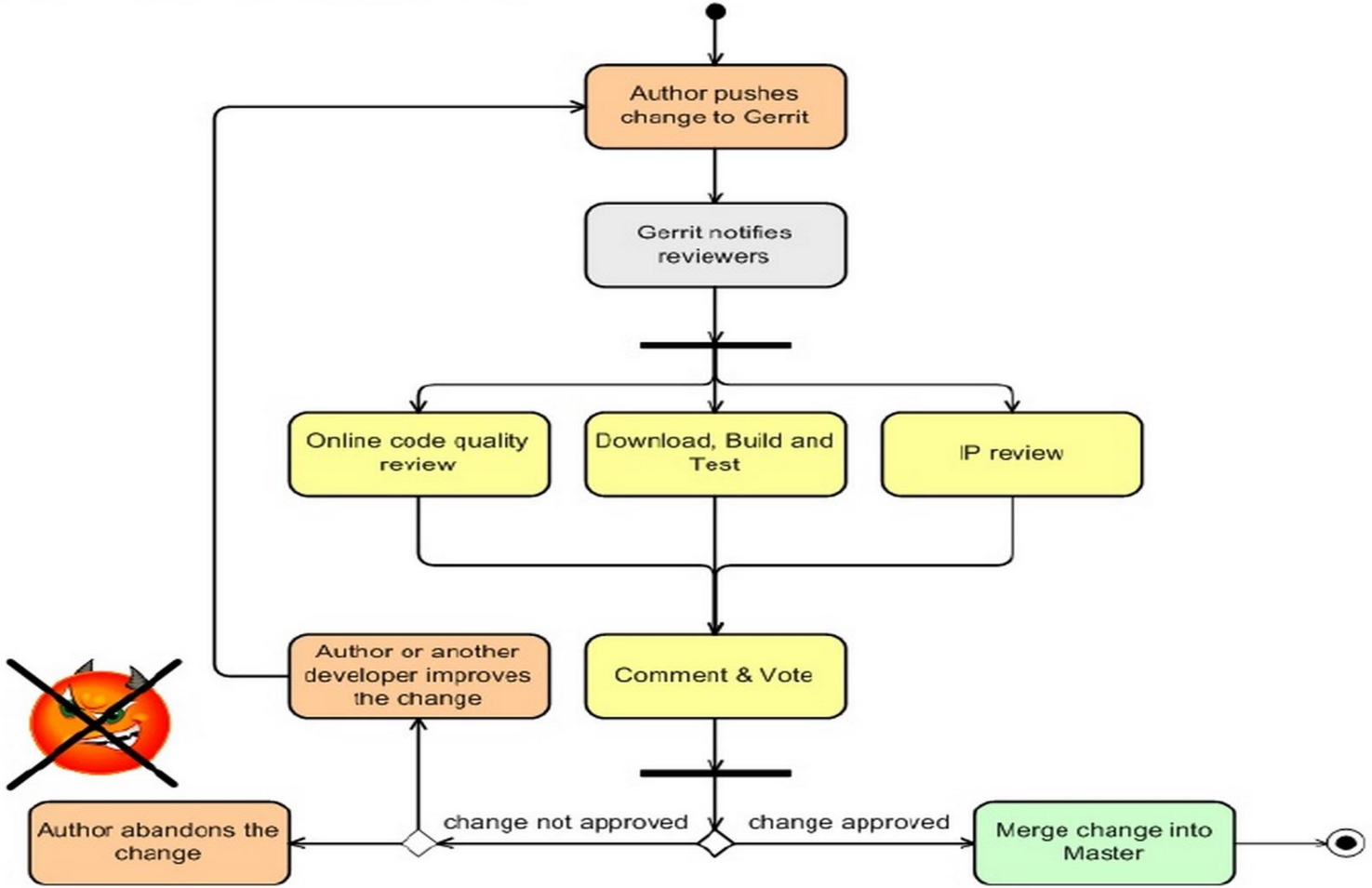
Request for Review

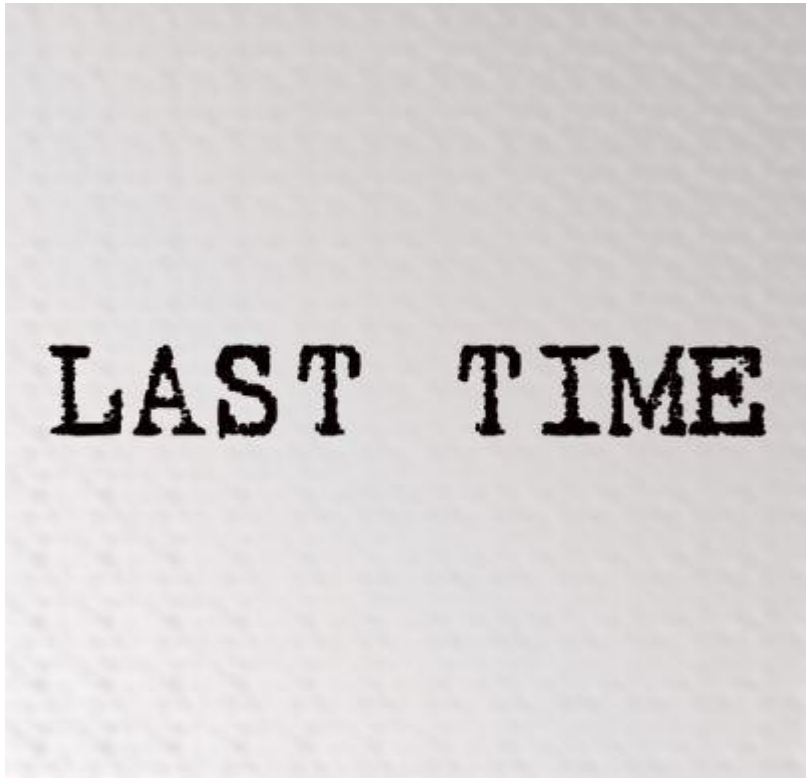
- Implementing a new feature
- Committing them to the repo
- Uploading changes to Gerrit

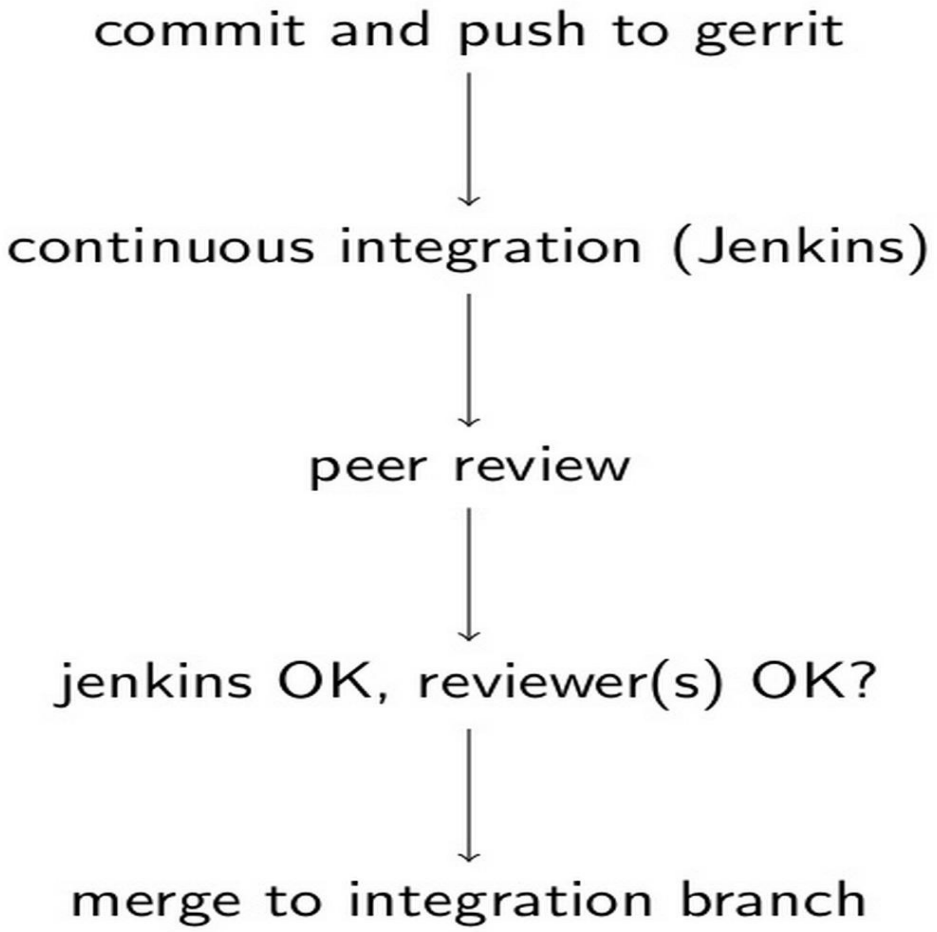



OMTP!

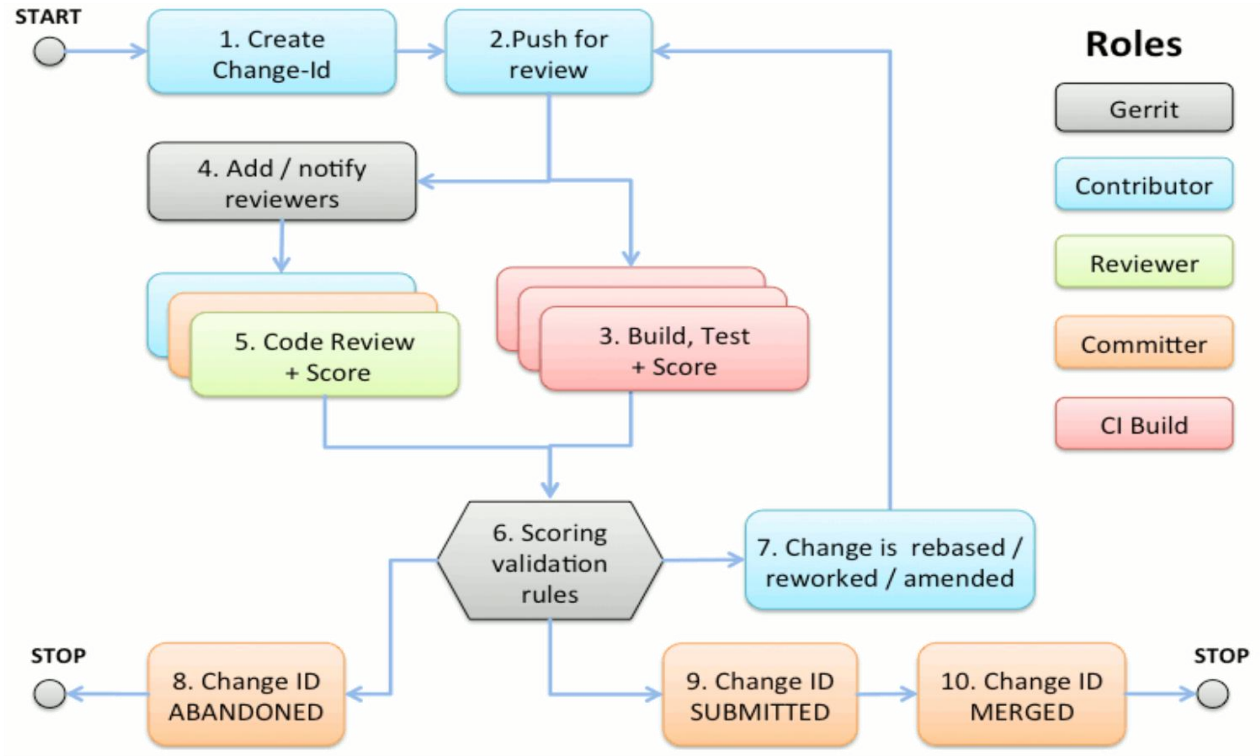
Gerrit - Workflow




LAST TIME


commit and push to gerrit
continuous integration (Jenkins)
peer review
jenkins OK, reviewer(s) OK?
merge to integration branch

Gerrit High Level Review Workflow



Cool, isn’t it ?

- Gerrit allow “automatic” topic-branches
- Triggers with Jenkins branch validation
- Enforce collective code-ownership
  - People “interacts” with the code changes
  - Discussion on style and architecture
  - Democratic voting (+1 / -1)
- Project history
  - Reason behind changes is recorded in code-review

See Gerrit “live” in action

1 . Create a User for host the Gerrit Service

- $ sudo adduser gerrit2
- $ sudo su gerrit2

2. Set-up

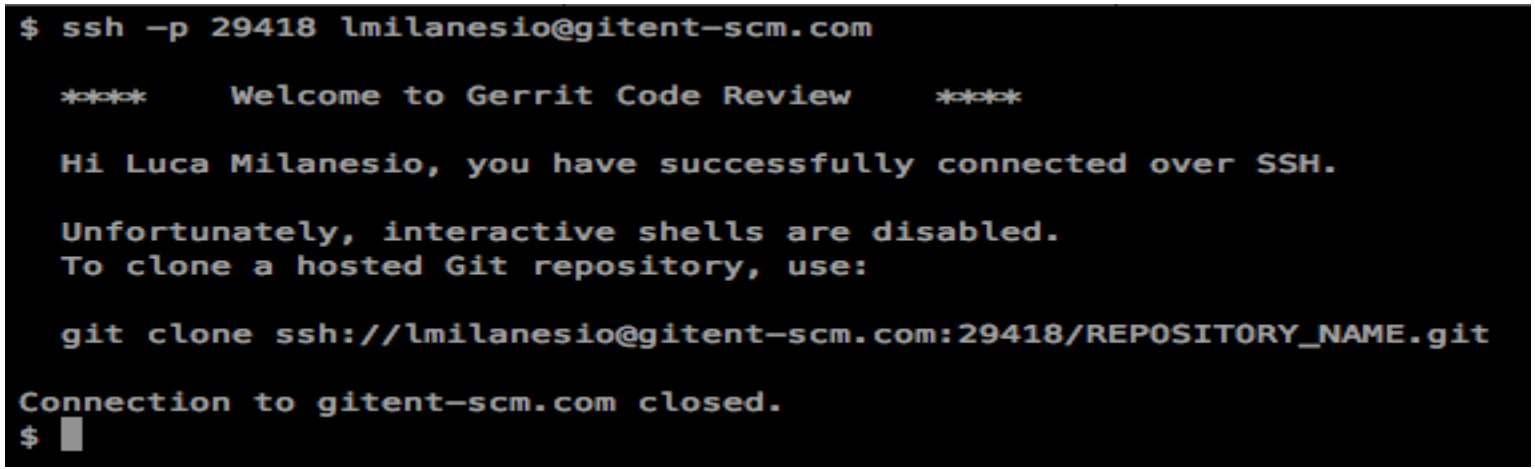
- Option-A: download and install
  1. Download from http://code.google.com/p/gerrit/downloads/list
  2. Gerrit install wizard:
     ```
     java –jar gerrit-2.2.2.1.war init -d ~/gerrit
     ```
  3. Run Gerrit:
     ```
     ~/gerrit/bin/gerrit.sh start
     ```
- Option-B: Gerrit as hosted service
  - Assembla.com (free for OpenSource projects)
  - GitEnterprise.com (free up to 10 users)
  - Gerrit Config - $site_path/etc/gerrit.config

3. Add SSH keys

- Option-A: not easy ☹
  –
- Option-B: use the provider user registration
  – Assembla: https://www.assembla.com/signup
  – GitEnterprise: https://gitent-scm.com/signup

3. Talk to Gerrit via SSH

- Gerrit SSH console
  - Listen at 29418 port



4. Create a Gerrit project

- Gerrit Project is:
  - Git repository (use “path notation” to organise projects)
  - Access permissions
  - Code-review and change-sets
- Option-A: use Gerrit SSH command

```
$ ssh -p 29418 lmilanesio@localhost gerrit create-project lmit/33degree
```

- Option-B: user the provider-specific page
  - Assembla: N/A (only 1 project associated to your “space”)
  - GitEnterprise: https://gitent-scm.com/newrepo

5. Clone repo from Gerrit

- Git SSH repository URL is:
  ssh://<user>@<host>:29418/<Gerrit project>.git

```
$ git clone ssh://lmilanesio@gitent-scm.com:29418/lmit/33degree.git
Cloning into 33degree...
remote: Counting objects: 2, done
remote: Finding sources: 100% (2/2)
remote: Total 2 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (2/2), done.
$
```

- **http://<Gerrit URL>/p/<Gerrit project>.git**

```
$ git clone http://review.gitent-scm.com/p/lmit/33degree.git
Cloning into 33degree...
remote: Counting objects: 28, done
remote: Finding sources: 100% (28/28)
remote: Total 28 (delta 0), reused 26 (delta 0)
Unpacking objects: 100% (28/28), done.
$
```

NO

machine <hostname> login <username> password <password>

6. Gerrit Change-Id

- SHA-1 GUID of a change-set under review
- MUST be last line of commit msg

Hint: install Gerrit post-commit hook for auto-generating Change-Id after each Git commit

```
$ scp -p -P 29418 lmilanesio@review.gitent-scm.com:hooks/commit-msg .git/hooks/
commit-msg
$
```

Now all Git commit will auto-generate a Change-Id !

7. Submit a change for review

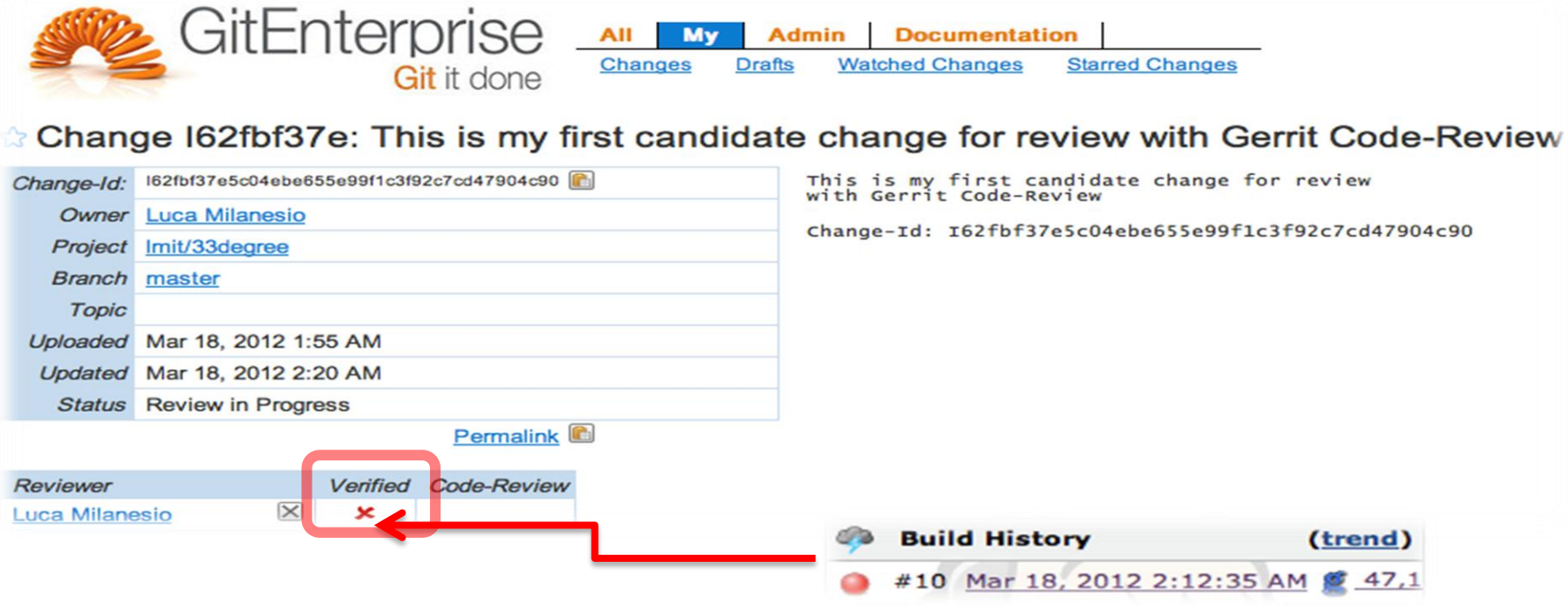
- Change for review committed locally
- Push to **`refs/for/<branch>`** for submitting local Git

```
$ git commit -a
[master d290762] This is my first candidate change for review with Gerrit Code-Review
 1 files changed, 1 insertions(+), 0 deletions(-)
$ git log
commit d290762166045bf0f1bb5ea97c3942aef3ebd1f1
Author: Luca Milanesio <luca@milanesio.org>
Date:   Sun Mar 18 01:46:34 2012 +0000

    This is my first candidate change for review
    with Gerrit Code-Review

    Change-Id: I62fbf37e5c04ebe655e99f1c3f92c7cd47904c90
$ git push origin HEAD:refs/for/master
Counting objects: 15, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (8/8), 702 bytes, done.
Total 8 (delta 1), reused 0 (delta 0)
```

8. Verify change against CI

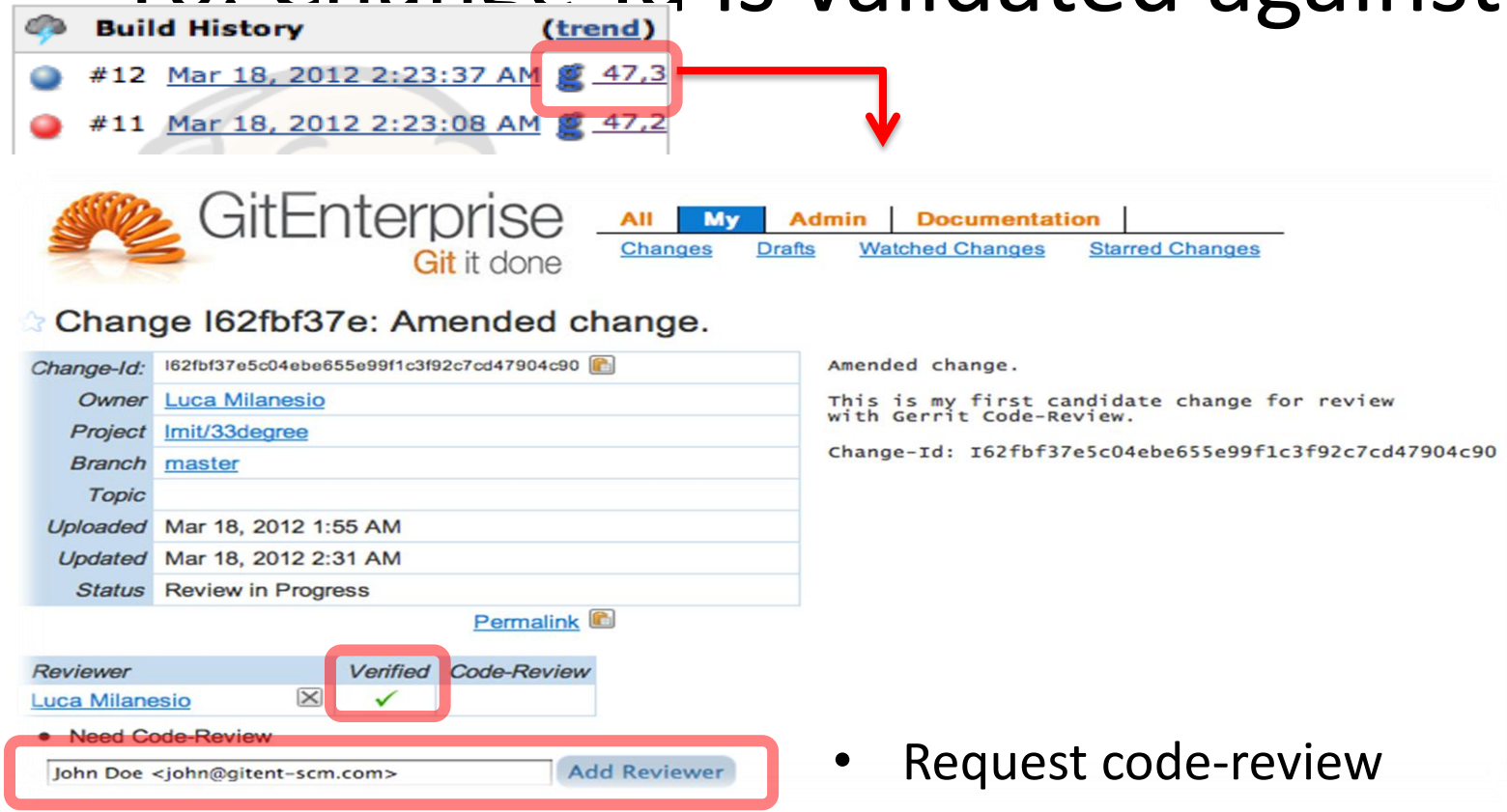


9. Additional patch-set to Change-Id

- Fix the problem locally
- Amend the commit (same Change-Id)
- Push again to **`refs/for/<branch>`** for adding one extra

```
$ git commit -a --amend
[master df28be2] Amended change.
 1 files changed, 1 insertions(+), 0 deletions(-)
$ git push origin HEAD:refs/for/master
Counting objects: 15, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (8/8), 724 bytes, done.
Total 8 (delta 1), reused 0 (delta 0)
remote: Resolving deltas:   0% (0/1)
To http://review.gitent-scm.com/p/lmit/33degree.git
 * [new branch]      HEAD -> refs/for/master
$
```

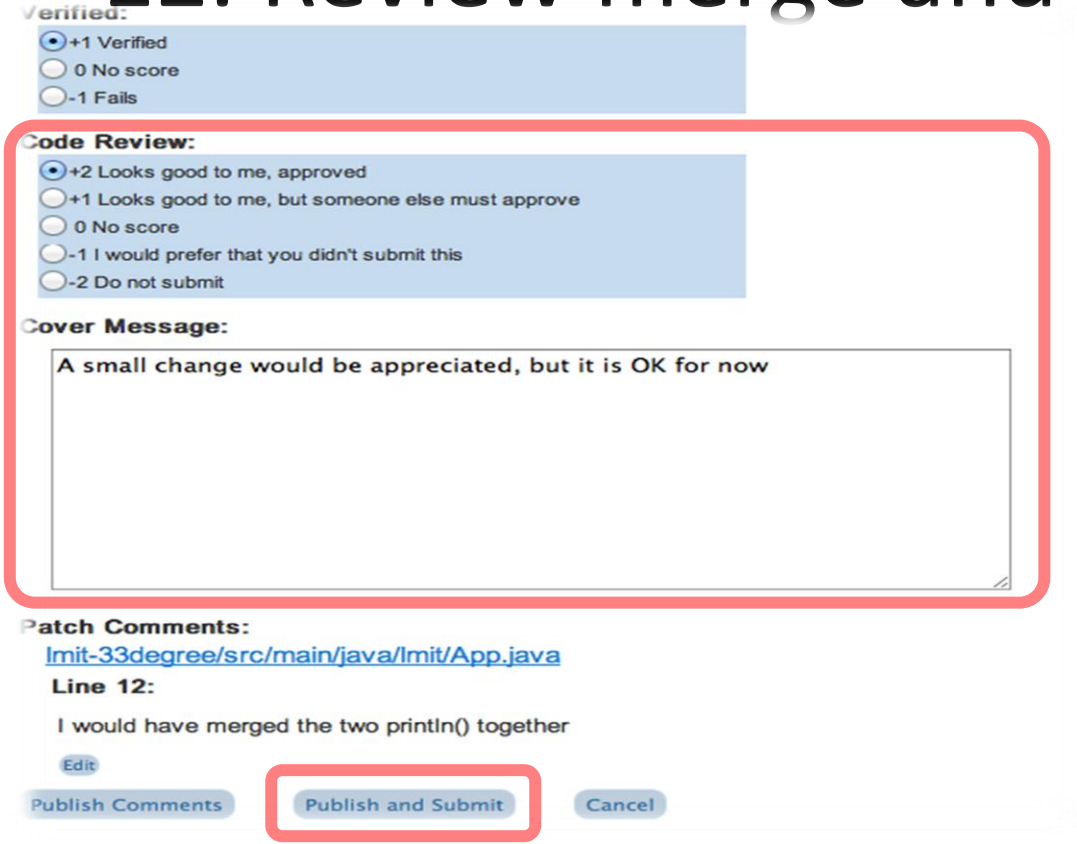
10. Change-Id is validated against CI



- Request code-review

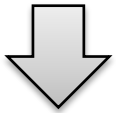
11. Review and comment changes

12. Review merge and submit change



- One or more reviewers can “vote” for the change
- Authorised users can then “Submit” the change.
- Change is automatically merged to master

13. Master build is triggered … SUCCESS !!



GitHub

Code Collaboration & Version Control

Follow

Stacks
33.3K

I Use This

Fans	Jobs	Votes
29.6K	3.81K	10K

734K	128K	32.5K

Bitbucket

Code Collaboration & Version Control

Follow

Stacks
9.99K

I Use This

Fans	Jobs	Votes
8.6K	458	2.77K

3.72K	5.82K	5.74K

Gerrit Code Review

Code Review

Follow

Stacks
59

I Use This

Fans	Jobs	Votes
60	0	37

-	-	0

Credits and resources

- Many thanks to

  Shawn Pearce, father of Gerrit

  Its contributors and Google Inc.

- Google Gerrit code-review
  http://code.google.com/p/gerrit/

gerrit
Gerrit Code Review

- Assembla Gerrit
  http://review.assembla.com

assembla

- GitEnterprise
  http://review.gitent-scm.com
  @gitenterprise

GitEnterprise
Git it done