Code Reviews: The Gerrit Tool and Experience from Sierra

Pat Notz

Sandia National Laboratories

Site: https://sierra-trac.sandia.gov:4443/
I’m neither a Gerrit expert nor a code review expert.
I’m neither a Gerrit expert nor a code review expert.

This is about a code review tool, not policies – TF Policies:
http://sierra-trac.sandia.gov/trac/sierra/wiki/Modules/Aria/PeerReview
I’m neither a Gerrit expert nor a code review expert.

This is about a code review tool, **not policies** – TF Policies:
http://sierra-trac.sandia.gov/trac/sierra/wiki/Modules/Aria/PeerReview

Smart Bear has a free book on code review

There’s a copy in the TF team room
Motivation for Code Review

- **Make better software and teams** – Avoid duplication, share knowledge, encourage good design and consistent style
  - *Decide as a team what you want out of reviews...*


- **1540 Strategic Plan** “We will also incorporate peer review into both our code development and research activities. In the area of code development, this includes things like code and design reviews and software quality assessments.”

- **SNL ASC Software Quality Plan** – *ASC Software Quality Engineering Practices, Version 3.0*
  - PR28 – Conduct independent technical reviews to evaluate adequacy with respect to requirements
  - AR24 – Technical reviews (evidence that review occurred and review results) (PR28)
Motivation for Code Review

- **Make better software and teams** – Avoid duplication, share knowledge, encourage good design and consistent style
  - Decide as a team what you want out of reviews...

1540 Strategic Plan “We will also incorporate peer review into both our code development and research activities. In the area of code development, this includes things like code and design reviews and software quality assessments.”

SNL ASC Software Quality Plan – *ASC Software Quality Engineering Practices, Version 3.0*
- PR28 – Conduct independent technical reviews to evaluate adequacy with respect to requirements
- AR24 – Technical reviews (evidence that review occurred and review results) (PR28)
Motivation for Code Review

- **Make better software and teams** – Avoid duplication, share knowledge, encourage good design and consistent style
  - Decide as a team what you want out of reviews...


- **1540 Strategic Plan** “We will also incorporate peer review into both our code development and research activities. In the area of code development, this includes things like code and design reviews and software quality assessments.”

- **SNL ASC Software Quality Plan** – *ASC Software Quality Engineering Practices, Version 3.0*
  - PR28 – Conduct independent technical reviews to evaluate adequacy with respect to requirements
  - AR24 – Technical reviews (evidence that review occurred and review results) (PR28)
Motivation for Code Review

- **Make better software and teams** – Avoid duplication, share knowledge, encourage good design and consistent style
  - *Decide as a team what you want out of reviews...*


- **1540 Strategic Plan** “We will also incorporate peer review into both our code development and research activities. In the area of code development, this includes things like code and design reviews and software quality assessments.”

- **SNL ASC Software Quality Plan** – ASC Software Quality Engineering Practices, Version 3.0
  - **PR28** – Conduct independent technical reviews to evaluate adequacy with respect to requirements
  - **AR24** – Technical reviews (evidence that review occurred and review results) (PR28)
Sign In

Search for status:open

<table>
<thead>
<tr>
<th>ID</th>
<th>Subject</th>
<th>Owner</th>
<th>Project</th>
<th>Branch</th>
<th>Updated</th>
<th>V</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pushing new changes: `git push gerrit local-branch:refs/for/master/clever-name`
Learn more: Sierra Gerrit HOW-TO.

Sign In to Gerrit Code Review at sierra-trac.sandia.gov

Username: sknotz
Password: ********
Sign In | Close
<table>
<thead>
<tr>
<th>ID</th>
<th>Subject</th>
<th>Owner</th>
<th>Project</th>
<th>Branch</th>
<th>Updated</th>
<th>V</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>!ibdb9b832</td>
<td>Make specifying source terms an error if the SRC term is not active on the equal</td>
<td>Patrick K Notz</td>
<td>code</td>
<td>master (sanity-check-src)</td>
<td>Sep 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Pushing new changes:**

```bash
$ git push gerrit local-branch:refs/for/master/clever-name
```

Learn more: [Sierra Gerrit HOW-TO](https://example.com/sierra-gerrit-how-to)
Viewing a Change

Change ldb9b832: Make specifying source terms an error if the SRC term is not active on the equ...

- Need Verified +1 (Verified)
- Need Code Review +2 (Looks good to me, approved)

Dependencies

Patch Set 1  ldb9b832bc9e46e55d9b57117e22b5ba8376201e

Author  Pat Notz <pknotz@sandia.gov>  Oct 12, 2009 11:48 AM
Committer  Pat Notz <pknotz@sandia.gov>  Jul 16, 2010 2:40 PM
Download  git fetch https://pknotz@sierra-trac.sandia.gov:4443/p/code refs/changes/01/1/1 & git checkout FETCH_HEAD

File Path  Comments  Size  Diff  Reviewed
Commit Message  
M  aria/src/Aria_Block_Description.C  +7, -3

Side-by-Side  Unified

Review  Abandon Change  Diff All Side-by-Side  Diff All Unified

All  My  Admin  Documentation
Viewing Diffs

Change ldbe8b9832: aria/src/Aria_Block_Description.C

Patch History

Old Version

(Download)

Up to change

Commit Message

Pushing new changes: `git push gerrit local-branch:refs/for/master/clever-name`

Learn more: Sierra Gerrit HOW-TO.
Leaving Comments
Keyboard Shortcuts
Publishing a Review

Verified:
- 1 Verified
- 0 No score
- 1 Fails

Code Review:
- 2 Looks good to me, approved
- 1 Looks good to me, but someone else must approve
- 0 No score
- -1 I would prefer that you didn't submit this
- -2 Do not submit

Cover Message:

It seems like point sources should receive similar treatment.

Patch Comments:

arla/src/Aria_Block_Description.C

Line 327:

Here's an inline comment.

Edit

Publish Comments  Cancel
Reviewed Change

Patch Set 1

File Path | Comments | Size | Diff | Reviewed
--- | --- | --- | --- | ---
Commit Message | 1 comment | Side-by-Side | Unified | Yes
aria/src/Aria_Block_Description.C | 1 comment | Side-by-Side | Unified | Yes

Comments

Patrick K Notz
Patch Set 1: Looks good to me, but someone else must approve
(1 inline comment)
It seems like point sources should receive similar treatment.
Partially Reviewed Changes

Search for status:open

<table>
<thead>
<tr>
<th>ID</th>
<th>Subject</th>
<th>Owner</th>
<th>Project</th>
<th>Branch</th>
<th>Updated</th>
<th>V</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ibdb9832</td>
<td>Make specifying source terms an error if the SRC term is not active on the equal</td>
<td>Patrick K Notz</td>
<td>code</td>
<td>master (sanity-check-ap)</td>
<td>2:23 PM</td>
<td>+1</td>
<td></td>
</tr>
</tbody>
</table>

Pushing new changes: 
```bash
$ git push gerrit local-branch:refs/for/master/clever-name
```
Learn more: [Sierra Gerrit HOW-TO](https://example.com/how-to)

Press '?' to view keyboard shortcuts

Powered by [Gerrit Code Review](https://example.com) (2.1.5) | [Report Bug](https://example.com/report-bug)
## Multiple Repositories

### Search for status:open

<table>
<thead>
<tr>
<th>ID</th>
<th>Subject</th>
<th>Owner</th>
<th>Project</th>
<th>Branch</th>
<th>Updated</th>
<th>V</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idb5a3e9</td>
<td>STK_MESH: Progress towards cleaning up unit-tests</td>
<td>James G Foucar</td>
<td>code</td>
<td>master (box-fixture-refactor)</td>
<td>Oct 11</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>I85e40f70</td>
<td>distest: add support for choosing num-procs when running locally</td>
<td>James G Foucar</td>
<td>tools</td>
<td>master (numprocs-for-distest)</td>
<td>Oct 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I6fad8f95</td>
<td>stk_syrix: clean up Transport algorithm, clean up various algorithm names...</td>
<td>Alan B Williams</td>
<td>code</td>
<td>master (stk_syrix-review)</td>
<td>Oct 8</td>
<td>✓</td>
<td>+1</td>
</tr>
<tr>
<td>Ieo317d8f</td>
<td>unit test for deformation gradients</td>
<td>Patrick K Notz</td>
<td>code</td>
<td>master (unit-test-improvements)</td>
<td>Oct 6</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>I8cfab39</td>
<td>improvements to unit test expression fixture</td>
<td>Patrick K Notz</td>
<td>code</td>
<td>master (unit-test-improvements)</td>
<td>Oct 6</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>I21101db9</td>
<td>Custom main() for aria unit tests</td>
<td>Patrick K Notz</td>
<td>code</td>
<td>master (unit-test-improvements)</td>
<td>Oct 6</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>I4cd086f8f</td>
<td>don’t call through the region to get the spatial dimension</td>
<td>Patrick K Notz</td>
<td>code</td>
<td>master (unit-test-improvements)</td>
<td>Oct 6</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>I4ef7ba6</td>
<td>STK_UTIL: Change ThrowsRequire/Assert to expect a message argument</td>
<td>James G Foucar</td>
<td>code</td>
<td>master (new-throw-require)</td>
<td>Oct 4</td>
<td>✓</td>
<td>+1</td>
</tr>
<tr>
<td>T9e10845f</td>
<td>Move a few Tau models to the new EQXModel (Equation Extension) factory</td>
<td>David Gregory Baur</td>
<td>code</td>
<td>master (TauFactory)</td>
<td>Sep 13</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Pushing new changes:**

```bash
$ git push gerrit local-branch:refs/for/master/clever-name
```

**Learn more:** [Sierra Gerrit HOW-TO](https://example.com)
1. Screenshots

2. About Gerrit

3. Set Up & Usage

4. Change-Ids

5. Sierra Experience
The Gerrit Project

About the project

- Designed for Git
- Primary developers at Google, BDFL is #2 contributor to Git
- Both Git server and code review tool
  - Lots of access controls, Contributor Agreements
- Open Source – Apache License 2.0
- Notable public projects using Gerrit:
  - Android: [https://review.source.android.com/](https://review.source.android.com/)
- Full Java stack (web server, Git, SSH)
- Can install/run as ordinary user
- Code submissions stored in Git repo, other data in Gerrit db
Gerrit is designed to be the sole interface to the Git repository:

1. Developer uploads proposed changes
2. Developers conduct review
   - Acceptance defined as cumulative scores of +1 Verified and +2 Code Review
3. If not accepted: Developer fixes and re-uploads
4. If accepted, code incorporated into target branch via Submit button
   - Fast-forward, merge, cherry-pick
5. If changes are undesired, submission is “abandoned” via Abandon button

Note: Submissions/reviews are kept forever, even abandoned ones
Gerrit’s Intended Workflow

Gerrit is designed to be the sole interface to the Git repository:

1. **Developer uploads proposed changes**
2. **Developers conduct review**
   - Acceptance defined as cumulative scores of +1 Verified and +2 Code Review
3. **If not accepted**: Developer fixes and re-uploads
4. **If accepted**, code incorporated into target branch via **Submit** button
   - Fast-forward, merge, cherry-pick
5. **If changes are undesired**, submission is “abandoned” via **Abandon** button

**Note**: Submissions/reviews are kept forever, even abandoned ones
Gerrit’s Intended Workflow

Gerrit is designed to be the sole interface to the Git repository:

1. Developer uploads proposed changes
2. Developers conduct review
   - Acceptance defined as cumulative scores of +1 Verified and +2 Code Review
3. If not accepted: Developer fixes and re-uploads
4. If accepted, code incorporated into target branch via Submit button
   - Fast-forward, merge, cherry-pick
5. If changes are undesired, submission is “abandoned” via Abandon button

Note: Submissions/reviews are kept forever, even abandoned ones
Gerrit’s Intended Workflow

Gerrit is designed to be the sole interface to the Git repository:

1. Developer uploads proposed changes
2. Developers conduct review
   - Acceptance defined as cumulative scores of +1 Verified and +2 Code Review
3. If not accepted: Developer fixes and re-uploads
4. If accepted, code incorporated into target branch via Submit button
   - Fast-forward, merge, cherry-pick
5. If changes are undesired, submission is “abandoned” via Abandon button

Note: Submissions/reviews are kept forever, even abandoned ones
Gerrit's Intended Workflow

Gerrit is designed to be the sole interface to the Git repository:

1. Developer uploads proposed changes
2. Developers conduct review
   - Acceptance defined as cumulative scores of +1 Verified and +2 Code Review
3. If not accepted: Developer fixes and re-uploads
4. If accepted, code incorporated into target branch via Submit button
   - Fast-forward, merge, cherry-pick
5. If changes are undesired, submission is “abandoned” via Abandon button

Note: Submissions/reviews are kept forever, even abandoned ones
Gerrit’s Intended Workflow

Gerrit is designed to be the sole interface to the Git repository:

1. Developer uploads proposed changes
2. Developers conduct review
   - Acceptance defined as cumulative scores of +1 Verified and +2 Code Review
3. If not accepted: Developer fixes and re-uploads
4. If accepted, code incorporated into target branch via Submit button
   - Fast-forward, merge, cherry-pick
5. If changes are undesired, submission is “abandoned” via Abandon button

Note: Submissions/reviews are kept forever, even abandoned ones
Gerrit is designed to be the sole interface to the Git repository:

1. Developer uploads proposed changes
2. Developers conduct review
   - Acceptance defined as cumulative scores of +1 Verified and +2 Code Review
3. If not accepted: Developer fixes and re-uploads
4. If accepted, code incorporated into target branch via Submit button
   - Fast-forward, merge, cherry-pick
5. If changes are undesired, submission is “abandoned” via Abandon button

Note: Submissions/reviews are kept forever, even abandoned ones
Notes About Sierra’s Gerrit Configuration

- Sign in using Kerberos username/password (uses LDAP)
- Must be in `wg-sierra-dev` meta-group to sign in and see code
- Gerrit coexists with Git — looking at same repository
- Hosting code, docs and toolset repos repositories are available
- Gerrit will not merge/ff/cherry-pick to target branches
  - Submit button will never appear
  - User must still use standard `git push` to push changes to master etc. branches.
  - After you’re done with the submission, Abandon it – just note in the abandon comments that it was accepted
Outline

1. Screenshots
2. About Gerrit
3. Set Up & Usage
4. Change-Ids
5. Sierra Expreience
One Time Set Up

1. Create an SSH key – different from Kerberos ticket (sorry)
2. Set Up Gerrit Account
   1. Sign in to Gerrit
   2. Add SSH public key to Gerrit
   3. Gerrit Settings: Watched Projects
3. Verify SSH connection
4. Add a Git remote for Gerrit
   
   `git remote add gerrit ssh://sierra-trac.sandia.gov:5915/code`

Details also at

One Time Set Up

1. Create an SSH key – different from Kerberos ticket (sorry)
2. Set Up Gerrit Account
   1. Sign in to Gerrit
   2. Add SSH public key to Gerrit
   3. Gerrit Settings: Watched Projects
3. Verify SSH connection
4. Add a Git remote for Gerrit
   
   `git remote add gerrit ssh://sierra-trac.sandia.gov:5915/code`

Details also at

Submitting Changes for Review

Changes are submitted using standard Git but with a special ref-spec:

$ git push gerrit master:refs/for/master/sanity-check-src

- gerrit is the Git remote we configured earlier
- Git ref-spec has the syntax: my-ref:your-ref
- The left-side master refers to the master branch in my project
- The refs/for/master tells Gerrit that the submission is destined for the master branch in the main Git repository
- The optional /sanity-check-src is an called a “topic name”. Topic names are displayed in Gerrit change lists – they do not affect Git
Submitting Changes for Review

Changes are submitted using standard Git but with a special ref-spec:

$ git push gerrit master:refs/for/master/sanity-check-src

- **gerrit** is the Git remote we configured earlier
- Git ref-spec has the syntax: \textit{my-ref:your-ref}
- The left-side \textit{master} refers to the \textit{master} branch in \textit{my project}
- The \textit{refs/for/master} tells Gerrit that the submission is destined for the \textit{master} branch in the main Git repository
- The \textit{optional /sanity-check-src} is an called a “topic name”. Topic names are displayed in Gerrit change lists – they do not affect Git
Submitting Changes for Review

Changes are submitted using standard Git but with a special ref-spec:

$ git push gerrit master:refs/for/master/sanity-check-src

- **gerrit** is the Git remote we configured earlier
- Git ref-spec has the syntax: `my-ref:your-ref`
- The left-side **master** refers to the **master** branch in my **project**
  - The **refs/for/master** tells Gerrit that the submission is destined for the **master** branch in the main Git repository
  - The **optional** `/sanity-check-src` is an called a “topic name”. Topic names are displayed in Gerrit change lists – they do not affect Git
Submitting Changes for Review

Changes are submitted using standard Git but with a special ref-spec:

$ git push gerrit master:refs/for/master/sanity-check-src

- `gerrit` is the Git remote we configured earlier
- Git ref-spec has the syntax: `my-ref:your-ref`
- The left-side `master` refers to the `master` branch in *my project*
- The `refs/for/master` tells Gerrit that the submission is destined for the `master` branch in the main Git repository
- The *optional* `/sanity-check-src` is an called a “topic name”. Topic names are displayed in Gerrit change lists – they do not affect Git
Submitting Changes for Review

Changes are submitted using standard Git but with a special ref-spec:

```bash
$ git push gerrit master:refs/for/master/sanity-check-src
```

- **gerrit** is the Git remote we configured earlier
- Git ref-spec has the syntax: `my-ref:your-ref`
- The left-side **master** refers to the **master** branch in **my project**
- The **refs/for/master** tells Gerrit that the submission is destined for the **master** branch in the main Git repository
- The **optional** `/sanity-check-src` is an called a “topic name”. Topic names are displayed in Gerrit change lists – they do not affect Git
Open Reviews

Search for status:open

<table>
<thead>
<tr>
<th>ID</th>
<th>Subject</th>
<th>Owner</th>
<th>Project</th>
<th>Branch</th>
<th>Updated</th>
<th>V</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>!ibdb9b832</td>
<td>Make specifying source terms an error if the SRC term is not active on the equal</td>
<td>Patrick K Notz</td>
<td>code</td>
<td>master (sanity-check-src)</td>
<td>Sep 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pushing new changes: $ git push gerrit local-branch:refs/for/master/clever-name
Learn more: Sierra Gerrit HOW-TO

Press '?' to view keyboard shortcuts
Powered by Gerrit Code Review (2.1.5) | Report Bug
Example Push to Gerrit

$ git push gerrit bc-fix:refs/for/master/bc-fix

Warning: the RSA host key for 'sierra-trac.sandia.gov' differs from the key
Offending key for IP in /asclldap/users/pknotz/.ssh/known_hosts:90
Matching host key in /asclldap/users/pknotz/.ssh/known_hosts:152
Are you sure you want to continue connecting (yes/no)? yes
Enter passphrase for key '/asclldap/users/pknotz/.ssh/id_rsa':

Counting objects: 29, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (16/16), done.
Writing objects: 100% (16/16), 8.97 KiB, done.
Total 16 (delta 15), reused 0 (delta 0)
remote:
remote: New Changes:
remote:  https://sierra-trac.sandia.gov:4443/2
remote:
To ssh://sierra-trac.sandia.gov:5915/code
 * [new branch] bc-fix -> refs/for/master/bc-fix
Reviewing Code

- Note, some submissions may include multiple commits (topic names are useful)
- Browse the diffs, leave comments
- You’re not done until you press **Review** and then **Publish Comments**
- Comments recognize `ticket:1234` (Trac links)
- Convention is that a change needs a +1 Verified score and a +2 Code Review score for acceptance
- “Verified” is undefined by Gerrit – what it means is up to users
Changes vs. Patches

**Change**

A "change" is a change to the code that attempts to do one thing. Ultimately, it will be committed as a single commit. However, it may go through multiple iterations (patches).

Changes are identified by Gerrit specific hashes prefixed by ’I’.

**Patch**

An iteration to a change. A change may go through many versions before it is approved and allowed to be submitted.

Patches are identified by Git commit hashes.
## Changes vs. Patches

<table>
<thead>
<tr>
<th>Change</th>
<th>A &quot;change&quot; is a change to the code that attempts to do one thing. Ultimately, it will be committed as a single commit. However, it may go through multiple iterations (patches).</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Changes are identified by Gerrit specific hashes prefixed by ’I’</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Patch</th>
<th>An iteration to a change. A change may go through many versions before it is approved and allowed to be submitted.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Patches are identified by Git commit hashes.</strong></td>
</tr>
</tbody>
</table>
Change View with Multiple Patches

Change lea58f1e5: BCs shouldn't need to provide DOF expressions

BCs shouldn't need to provide DOF expressions

The EQ_Description classes should take care of calling provide_field_exprs while we're setting up BCs.

Change-Id: lea58f1e52735b4c556ac4fde3870ea65b3228e1c

Permalink

Veriﬁed Code Review

Patrick K Notz

☑

+1

Verified; Looks good to me, but someone else must approve

David Jason Glaze

☑

Looks good to me, approved

Dependencies

Patch Set 1

9114aade9b03061eaa113eab0039420bbecb130

Patch Set 2

f03212471a6121077675cd09bec7825de212 CCD

Author

Pat Notz <pknotz@sandia.gov> Sep 9, 2010 2:04 PM

Committer

Pat Notz <pknotz@sandia.gov> Sep 9, 2010 2:27 PM

Download

git fetch ssh://pknotz@sierra-trac.sandia.gov:3915/code refs/changes/03/3/2 & git cherry-pick FETCH_HEAD

Review

Restore Change Diff All Side-by-Side Diff All Unified

File Path Comments Size Diff Reviewed

commit Message

M ana/src/Ana_BC_Description.C

+11, -26 Side-by-Side

+11, -26 Unified

✓
Diff View for Multiple Patches
You can re-use the Change-Id when you re-submit changes. Just put a Change-Id line at the bottom of your commit message.

commit b408dc9f00eb6bc0d64f593c469b31d9549fb364
Author: Pat Notz <pknotz@sandia.gov>
Date: Mon Sep 13 11:33:14 2010 -0600

Add a viscous traction post-processor

This post-processor omits the pressure contribution to the stress when computing the traction vector.

Change-Id: Ic4352c0e1c7c948ff78926acd60910cb90b227ab
Gerrit comes with a Git hook that can automatically create Change-Ids in your commit messages for you.

Having Change-Ids in commit messages makes it easy to keep the same ones across rebase, amend, etc.

If Gerrit hasn’t seen the Change-Id before, it will create a new Change using the provided Change-Id.
Experience To Date

Gerrit available since mid September 2010

- 59 changes submitted (excluding ‘test’ submissions)
- 77 patches reviewed
- 30 registered users
- 11 users submitted code for review
- 21 users provided feedback on changes
Studies show that reviews should cover \( \lesssim 400 \) LOC and review rates are typically between 200-400 LOC/hr [Smart Bear book].
Developer Reactions

Usage among code teams

- Framework/STK team – heavy usage
- Thermal/Fluid team – moderate usage
- Infrastructure team – beginning to use it
- Solid Mechanics team – experimenting

- Multi-commit changes are clumsy
- “Gerrit removes all of the annoying work of sharing code and collecting feedback”
- “Anyone who is serious about peer review no longer has an excuse!”
- “Sometimes in-person reviews are faster and easier”
- “…if it is not in your culture, no matter how easy you make it, peer review is not going to happen”
Usage among code teams
- Framework/STK team – heavy usage
- Thermal/Fluid team – moderate usage
- Infrastructure team – beginning to use it
- Solid Mechanics team – experimenting

Multi-commit changes are clumsy
- “Gerrit removes all of the annoying work of sharing code and collecting feedback”
- “Anyone who is serious about peer review no longer has an excuse!”
- “Sometimes in-person reviews are faster and easier”
- “…if it is not in your culture, no matter how easy you make it, peer review is not going to happen”
Developer Reactions

Usage among code teams
- Framework/STK team – heavy usage
- Thermal/Fluid team – moderate usage
- Infrastructure team – beginning to use it
- Solid Mechanics team – experimenting

Multi-commit changes are clumsy
- “Gerrit removes all of the annoying work of sharing code and collecting feedback”
- “Anyone who is serious about peer review no longer has an excuse!”
- “Sometimes in-person reviews are faster and easier”
- “…if it is not in your culture, no matter how easy you make it, peer review is not going to happen”
Usage among code teams
- Framework/STK team – heavy usage
- Thermal/Fluid team – moderate usage
- Infrastructure team – beginning to use it
- Solid Mechanics team – experimenting

Multi-commit changes are clumsy

“Gerrit removes all of the annoying work of sharing code and collecting feedback”

“Anyone who is serious about peer review no longer has an excuse!”

“Sometimes in-person reviews are faster and easier”

“...if it is not in your culture, no matter how easy you make it, peer review is not going to happen”
Developer Reactions

Usage among code teams
- Framework/STK team – heavy usage
- Thermal/Fluid team – moderate usage
- Infrastructure team – beginning to use it
- Solid Mechanics team – experimenting

Multi-commit changes are clumsy
- “Gerrit removes all of the annoying work of sharing code and collecting feedback”
- “Anyone who is serious about peer review no longer has an excuse!”
- “Sometimes in-person reviews are faster and easier”
- “…if it is not in your culture, no matter how easy you make it, peer review is not going to happen”
Developer Reactions

Usage among code teams
- Framework/STK team – heavy usage
- Thermal/Fluid team – moderate usage
- Infrastructure team – beginning to use it
- Solid Mechanics team – experimenting

Multi-commit changes are clumsy
“Gerrit removes all of the annoying work of sharing code and collecting feedback”
“Anyone who is serious about peer review no longer has an excuse!”
“Sometimes in-person reviews are faster and easier”
“...if it is not in your culture, no matter how easy you make it, peer review is not going to happen”
— Questions? —

- RTFM
- Contributor Agreements; Single Sign-On
- Dependencies
- Download submissions to your project
- Search Operators (see TFM)
- Dig around the UI
- Restore Change (i.e., undo Abandon)
- Starred changes
- Many preferences for viewing diffs

http://code.google.com/p/gerrit/