

Code Review Practices

Prof. Dr. Dirk Riehle

Friedrich-Alexander University Erlangen-Nürnberg

AMOS F01

Licensed under CC BY 4.0 International

Traditional to Scrum Role Mapping (Recap)

Traditional

Scrum

Product Manager

Product Owner

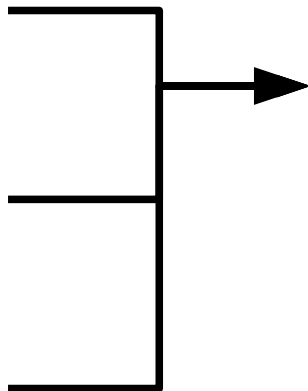
Engineering Manager

Software Developer

QA Engineer

Software Developer

Scrum Master



- Definition and purpose
 - “Code review is systematic examination [...] of computer source code.
 - It is intended to find and fix mistakes overlooked in the initial development phase,
 - improving both the overall quality of software
 - and the developers’ skills.
 - Reviews are done in various forms such as
 - pair programming,
 - informal walkthroughs, and
 - formal inspections.” [1]

[1] Adapted from https://en.wikipedia.org/wiki/Code_review [DR]

When to Review Code?

- 1. In the moment**
- 2. Before commit**
3. At another time
4. Before release

Pair Programming

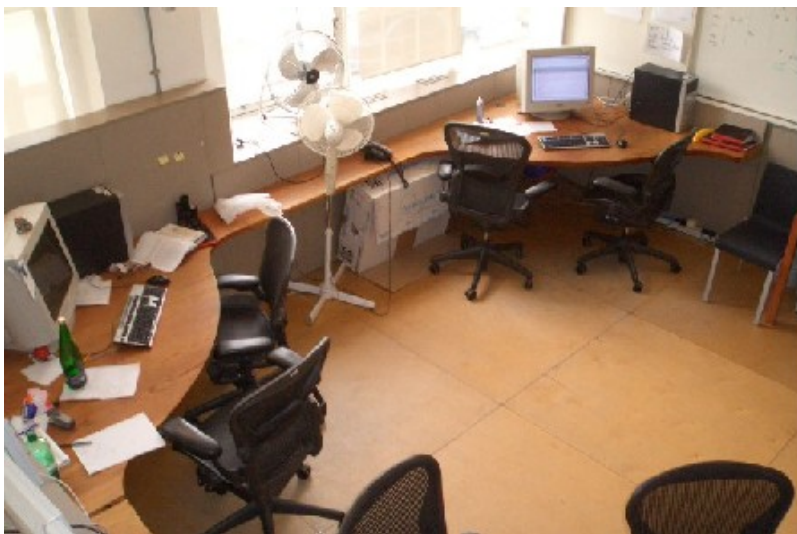
- Definition
 - Is programming carried out by pairs of programmers
 - One programmer implements, and the other programmer reviews
 - Effectiveness is debated; empirical studies show conflicting evidence
- Purpose
 - Quality assurance
 - Collaborative learning
 - Knowledge sharing
- Synonyms
 - Programmer and reviewer
 - Driver and co-driver
 - Pilot and navigator

- 1. Pair programming**
- 2. Pre-commit code review**

Pair Programming (Practices)

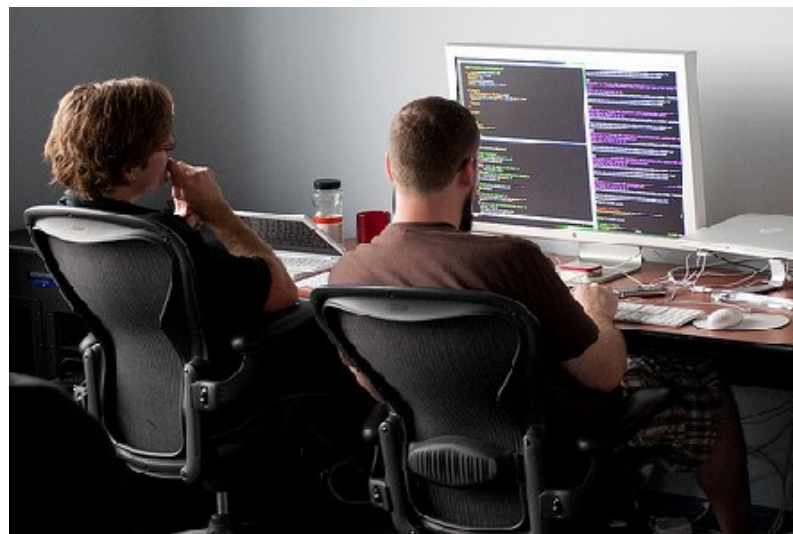
- **Process**

- Find comfortable partner
- Switch roles often
- Communicate regularly

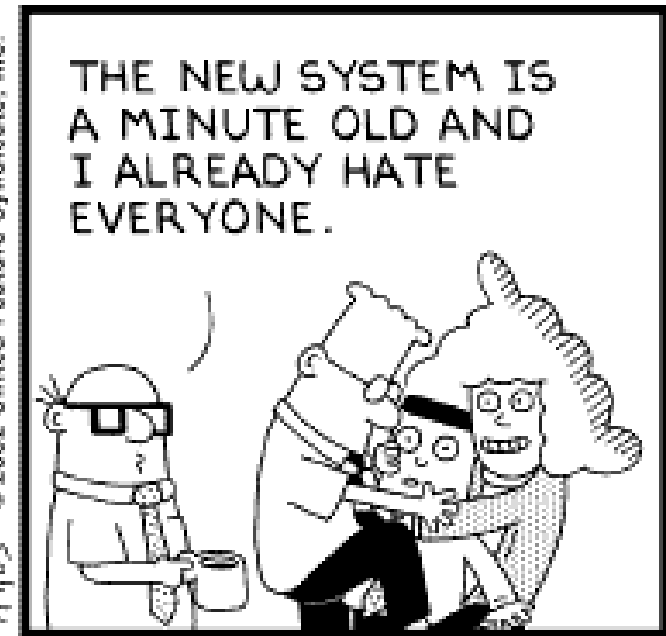
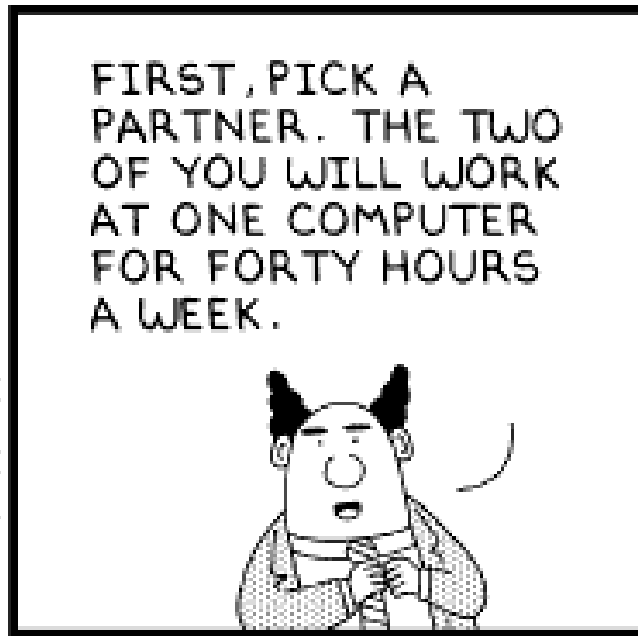


- **Advice**

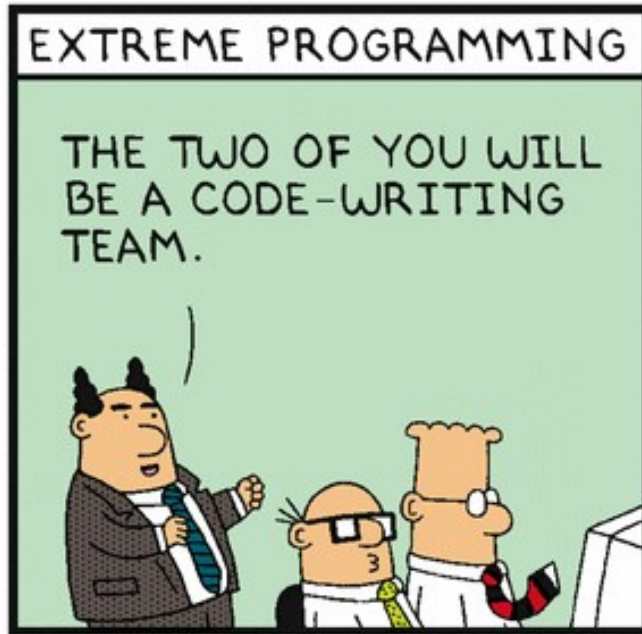
- Don't force it for small stuff
- Don't overheat, take a break
- Switch partners at times



Dilbert on Pair Programming 1 / 2

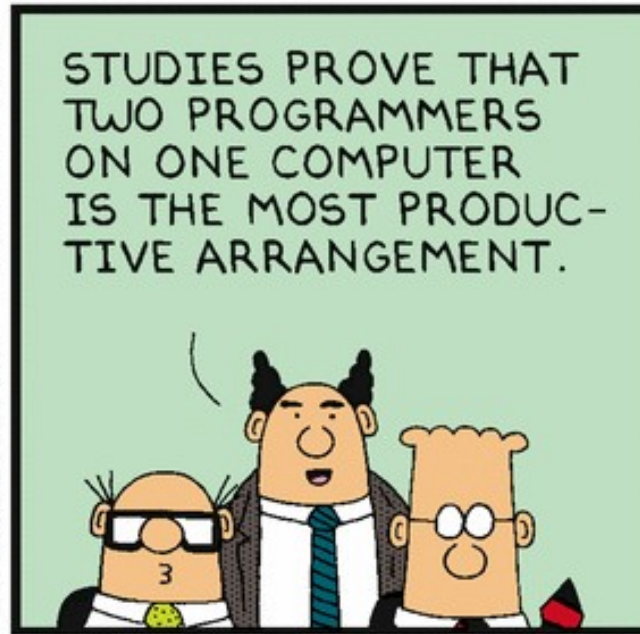


Dilbert on Pair Programming 2 / 2



scottadams@aol.com

www.dilbert.com



1/11/03 © 2002 United Feature Syndicate, Inc.



Pre-Commit Code Review

- Code review
 - Is the (peer) review of source code for quality criteria
 - Reviewer has accept or reject responsibility
 - Cf. “Vier-Augen-Prinzip” (in German)
- Pre-commit code review
 - Is the review of source code before it gets committed to a team repository
 - Typically facilitated by a software tool, e.g. Gerrit
 - May lead to back and forth between developers until “LGTM”

Example Code Review with Gerrit 1 / 3

The screenshot displays the Wikimedia Code Review (Gerrit) interface for a change with ID 11f962956. The page is titled "Change 11f962956: Added" and shows the change details, commit message, and a table of file paths with comments.

Change Details:

- Change-Id: 11f962956e9cf9c404c2fc685963964978ef52516
- Owner: preilly
- Project: test/mediawiki/extensions/examples
- Branch: master
- Topic: 2012/bug12345
- Uploaded: May 29, 2012 3:25 PM
- Updated: Jun 13, 2012 12:42 AM
- Status: Abandoned

Commit Message:

Added get version method to extension

Change-Id: 1584255f80c7e2634617a803e0a387c1bfa6400ee

Added get version method to extension

Change-Id: 11f962956e9cf9c404c2fc685963964978ef52516

Dependencies:

- Need Rebase or Has Dependency

Reference Version: Base

Patch Set 1: fee9e992a68d285a727e39608cb08e535bcc10a (gitblit)

Patch Set 2: 7b7840b470961405b70560d545fe6b39d84601c (gitblit)

Author: preilly <preilly@wikimedia.org> May 29, 2012 3:13 PM

Committer: preilly <preilly@wikimedia.org> May 29, 2012 3:31 PM

Parent(s): 7cf0d68d9553c16d3e426de213e7db11d14ca5ea Merge "Small change for the sake of review test"

Download: [checkout](#) | [pull](#) | [cherry-pick](#) | [patch](#) | [Anonymous HTTP](#)

Diff: git fetch https://gerrit.wikimedia.org/r/test/mediawiki/extensions/examples refs/changes/32/9332/2 && git checkout FETCH_HEAD

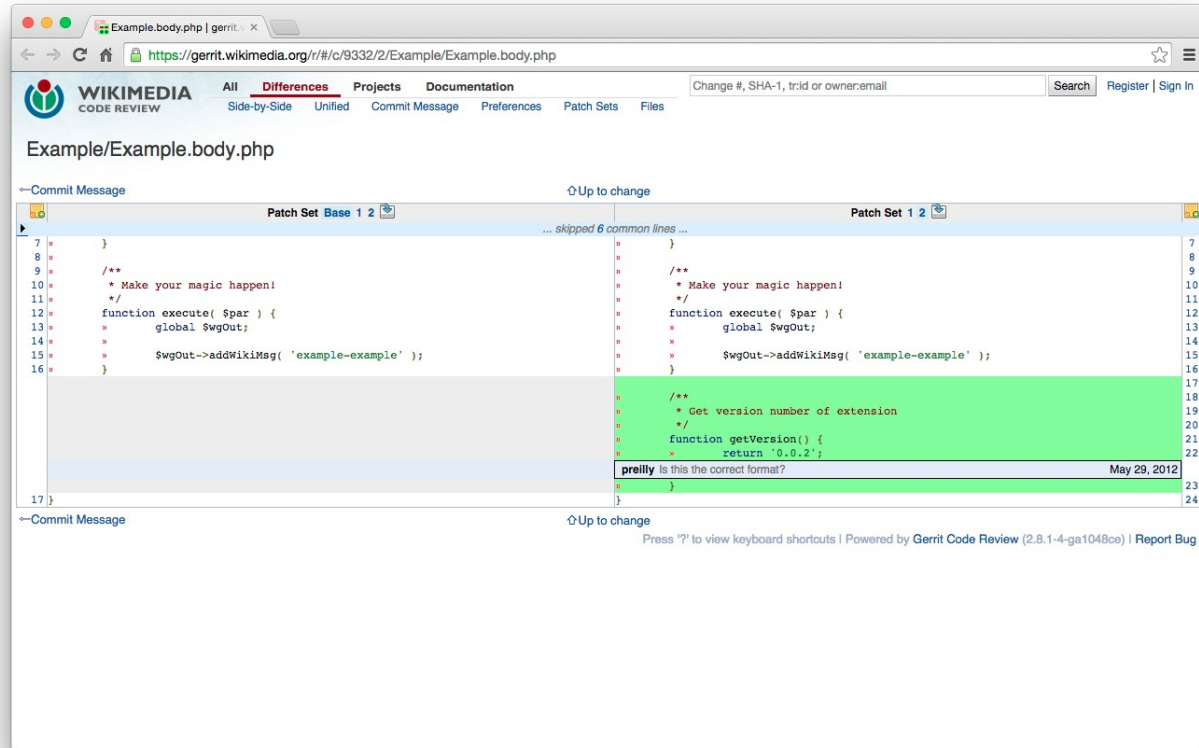
File Path	Comments	Size	Diff
Commit Message			Side-by-Side Unified
M Example/Example.body.php	1 comment	+7, -0	Side-by-Side Unified
		+7, -0	All Side-by-Side All Unified

Comments: [Expand Recent](#) | [Expand All](#) | [Collapse All](#)

- preilly Uploaded patch set 2. May 29, 2012
- preilly Patch Set 2: (1 inline comment) May 29, 2012
- Catrope Jun 13, 2012

Patch Set 2: Abandoned

Example Code Review with Gerrit 2 / 3

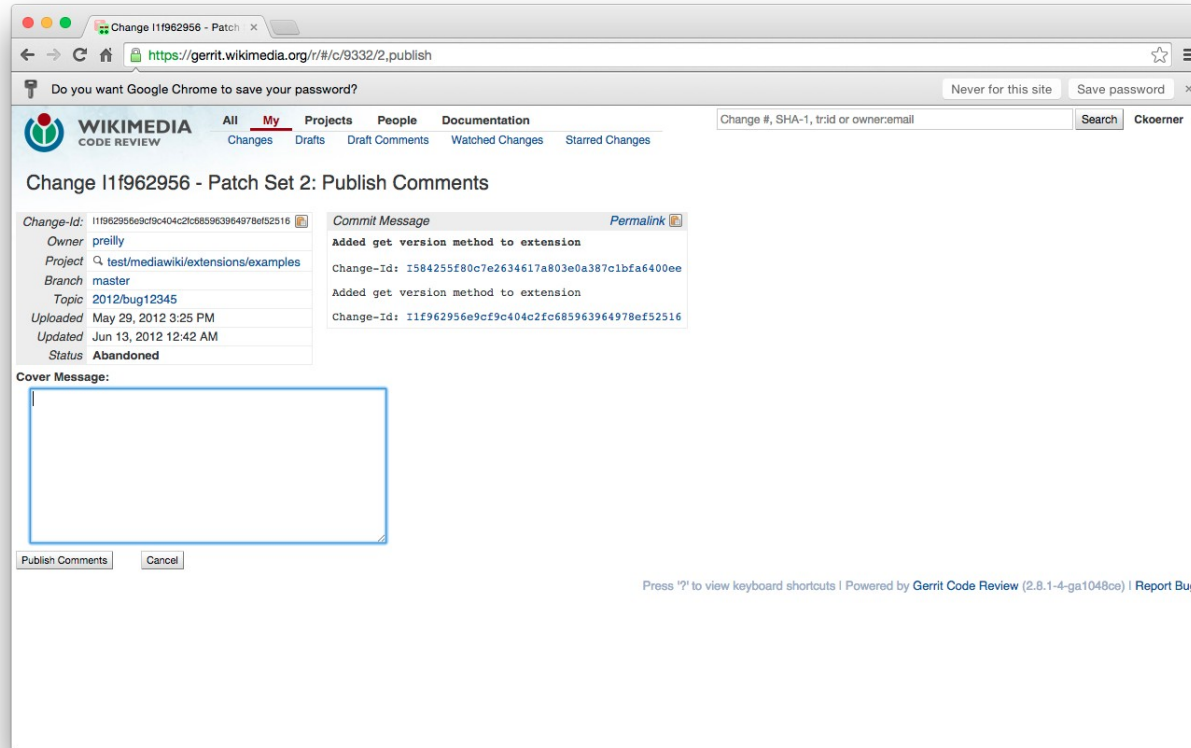


The screenshot displays the Wikimedia Gerrit Code Review interface for a file named `Example.body.php`. The browser address bar shows the URL `https://gerrit.wikimedia.org/r/#/c/9332/2/Example/Example.body.php`. The page features a navigation bar with tabs for **All**, **Differences** (selected), **Projects**, and **Documentation**. Below the navigation bar, there are links for **Side-by-Side**, **Unified**, **Commit Message**, **Preferences**, **Patch Sets**, and **Files**. A search bar is located on the right, with a **Search** button and links for **Register** and **Sign In**.

The main content area shows a side-by-side comparison of the code. The left pane is labeled **Patch Set Base 1 2** and the right pane is labeled **Patch Set 1 2**. The code is displayed in a light blue background with line numbers on the left and right. The diff highlights changes in green. The code includes a function `execute` and a `getVersion` function. A comment `prelilly` is visible in the right pane, dated **May 29, 2012**.

At the bottom of the interface, there is a footer with the text: **Press '?' to view keyboard shortcuts | Powered by Gerrit Code Review (2.8.1-4-ga1048ce) | Report Bug**.

Example Code Review with Gerrit 3 / 3



Benefits of Pre-commit Code Review

- Collaboration
 - Improves knowledge sharing and teamwork
 - Makes it easier to establish topics like security
- Quality assurance
 - Leads to more disciplined developers
 - Prevents (some) errors before they happen
 - Raises overall quality standards
- Feeling of responsibility
 - Specifically, supports collective code ownership
 - Strengthens overall feeling of responsibility

Agile vs. Open Source Code Review

- **Agile methods**

- Programming guidelines
 - Code reading >> writing
 - Make it easy to get acquainted
- Collective code ownership
 - Feature-oriented development
 - Typically co-located development
 - Everyone has write access
- Pair programming
 - Changes are reviewed directly
 - Everyone is a peer

- **Open source**

- Programming guidelines
 - Code reading >> writing
 - Showing respect for project
- Individual code ownership
 - Component-oriented development
 - Typically distributed development
 - Strictly regulated write access
- Patch review
 - Changes are submitted for review
 - Two-class reviewing hierarchy

Review / Summary of Session

- Code review practices
 - Pair programming
 - Pre-commit code review
- Agile vs. open source approach

Thank you! Questions?

dirk.riehle@fau.de – <http://osr.cs.fau.de>

dirk@riehle.org – <http://dirkriehle.com> – [@dirkriehle](#)

Credits and License

- Original version
 - © 2020 Dirk Riehle, some rights reserved
 - Licensed under [Creative Commons Attribution 4.0 International License](#)
- Contributions
 - None yet