

12 Best Code Review Tools for Developers

Code review is a part of the software development process which involves testing the source code to identify bugs at an early stage. A code review process is typically conducted before merging with the codebase.

An effective code review prevents bugs and errors from getting into your project by improving code quality at an early stage of the software development process.

In this post, we'll explain what code review is and explore popular code review tools that help organizations with the code review process.

What Is the Code Review Process?

The primary goal of the code review process is to assess any new code for bugs, errors, and quality standards set by the organization. The code review process should not just consist of one-sided feedback. Therefore, an intangible benefit of the code review process is the collective team's improved coding skills.

If you would like to initiate a code review process in your organization, you should first decide who would [review the code](#). If you belong to a small team, you may assign team leads to review all code. In a larger team size with multiple reviewers, you could enable a process in which every code review is assigned to an experienced developer based on their workload.

The next consideration for you is to decide on timelines, rounds, and minimal requirements for submitting code review requests.

The final consideration is about how feedback should be given in the code review process. Make sure you highlight the positive aspects of the code while suggesting alternatives for drawbacks.

[Your feedback](#) should be constructive enough to encourage the developer to understand your perspective and initiate a conversation when necessary.



— Keep your feedback informative

It is easy for code reviews to get stuck in limbo, leading to being less efficient and even counter-productive.

Why Is Code Review Critical?

The code review process is critical because it is never a part of the formal curriculum in schools. You may learn the nuances of a [programming language](#) and [project management](#), but code review is a process that evolves as an organization ages.

Code review is critical for the following reasons:

- Ensure that you have no bugs in code.
- Minimize your chances of having issues.
- Confirm new code adheres to guidelines.
- Increase the efficiency of new code.

Code reviews further lead to improving other team members' expertise. As a [senior developer](#) typically conducts a code review, a junior developer may use this feedback to improve their own coding.

How to Perform a Code Review?

There are four ways to conduct code reviews.

Over-the-Shoulder Code Reviews

Over-the-shoulder code reviews are done on the developer's workstation, where an experienced team member walks through the new code, providing suggestions through a conversation. It is the easiest approach to code reviews and does not require a pre-defined structure.

Such a code review may still be done informally today, along with a formal code review process that may be in place. Over-the-shoulder code reviews were traditionally done in person, while [distributed teams](#) can follow this method through [collaborative tools](#) as well.

Email Pass-Around

While over-the-shoulder code reviews are a great way to review new code, geographically distributed teams have traditionally relied on email for code reviews.

In this code review process, a developer emails a diff of changes to the whole development team, usually through [version control systems](#) that automate notifications. This email initiates a conversation on the changes, where team members may request further changes, point out errors, or ask for clarifications.

[notification]Channel-i project: branch develop updated. 5b76ed3f450cfb2683e58930ae1696038fcc6565 [IMG/IMG Developers](#)



channel-i.git@ via googlegroups.com [Unsubscribe](#)
to img-devel

Sun, Aug 11, 2013, 11:24 PM



Project Channel-i

The branch, develop has been updated

via 5b76ed3f450cfb2683e58930ae1696038fcc6565 (commit)
via f87361d4d7076144282fac5cef1f2949004ec610 (commit)
from ed3248eebf523bbbdd9f67363a75f97e8b38ea (commit)

Those revisions listed above that are new to this repository have not appeared on any other notification email; so we list those revisions in full, below.

----- Log -----
commit 5b76ed3f450cfb2683e58930ae1696038fcc6565
Merge: f87361d ed3248e
Author: Karan
Date: Sun Aug 11 23:17:20 2013 +0530

Merge remote-tracking branch 'apps/develop' into develop

commit f87361d4d7076144282fac5cef1f2949004ec610
Author: Karan
Date: Sun Aug 11 23:16:37 2013 +0530

Image aspect ratio corrected

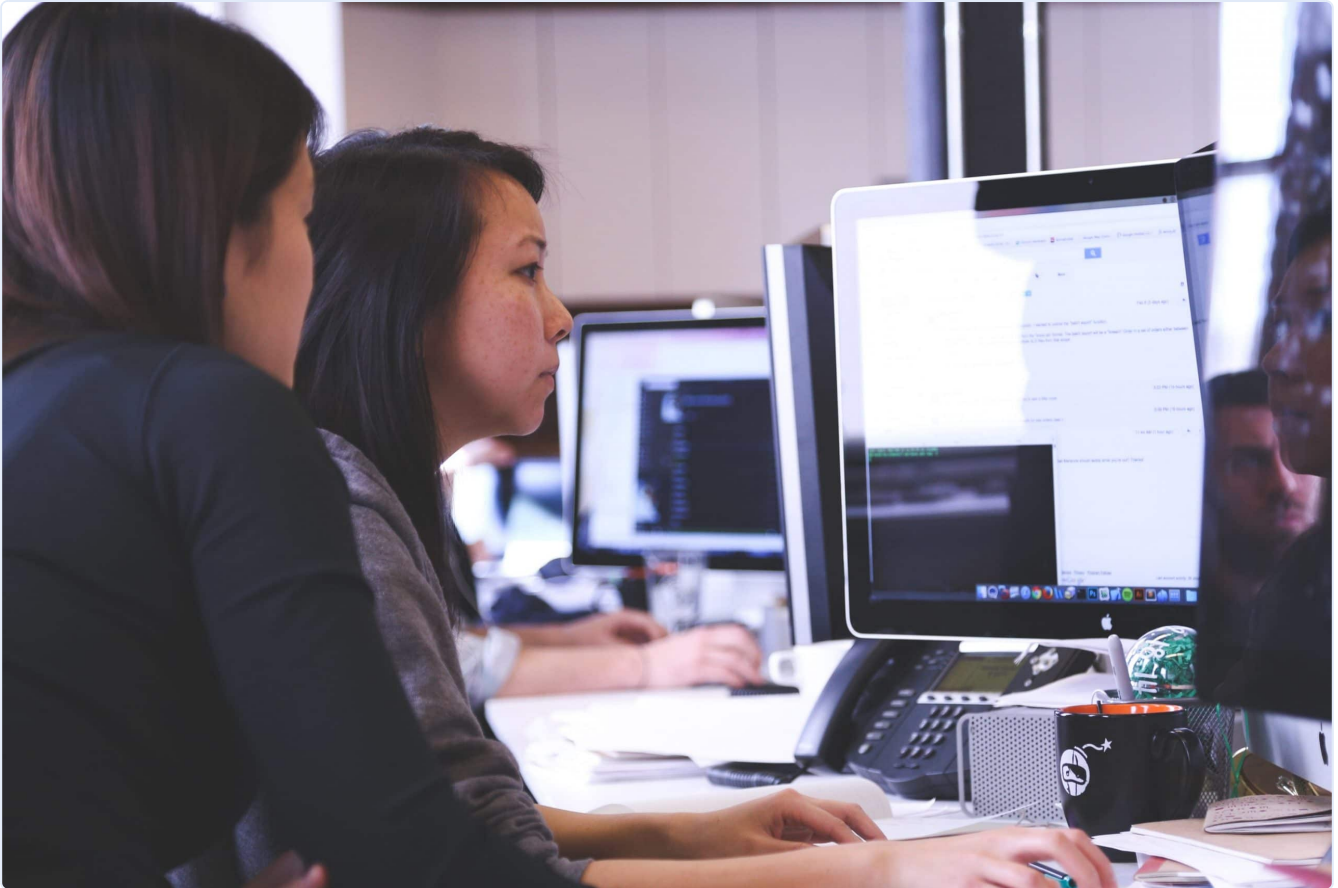
— Email Pass Around through Google Groups on each new push

In the early days, email was the primary means of communication because of [its versatility](#)

Open source organizations often maintained a public mailing list, which would also serve as a medium to discuss and provide feedback on code.

With the advent of code review tools, these mailing lists still exist, but primarily for announcements and discussion onward.

Pair Programming



— Pair Programming can be inefficient sometimes

Pair programming is a continuous code review process. Two developers sit at a workstation, but only one of them actively codes whereas the other provides real-time feedback.

While it may serve as a great tool to inspect new code and train developers, it could potentially prove to be inefficient due to its time-consuming nature. This process locks down the reviewer from doing any other productive work during the period.

Tool-Assisted

A tool-assisted code review process involves the use of a specialized tool to facilitate the process of code review. A tool generally helps you with the following tasks:

- Organize and display the updated files in a change.
- Facilitate a conversation between reviewers and developers.
- Assess the efficacy of the code review process with metrics.

While these are the broad requirements of a code review tool, modern tools may provide a handful of other functions. We'll explore a range of code review tools later in this post.

Why Should You Use Code Review Tools?

The main outcome of a code review process is to increase efficiency. While these traditional methods of code review have worked in the past, **you may be losing efficiency if you haven't switched to a code review tool**. A code review tool automates the process of code review so that a reviewer solely focuses on the code.

A code review tool integrates with your development cycle to initiate a code review before new code is merged into the main codebase. You can choose a tool that is compatible with your technology stack to seamlessly integrate it into your workflow.

For instance, if you use [Git for code management](#), TravisCI for continuous integration, ensure that you select a tool that supports these technologies to be able to fit into the development process.

There are two types of code testing in software development: dynamic and static.

Dynamic analysis involves checking if the code follows a set of rules and running unit tests, typically performed by a predefined script. Static code testing is done after a developer creates a new code to be merged into the current code.

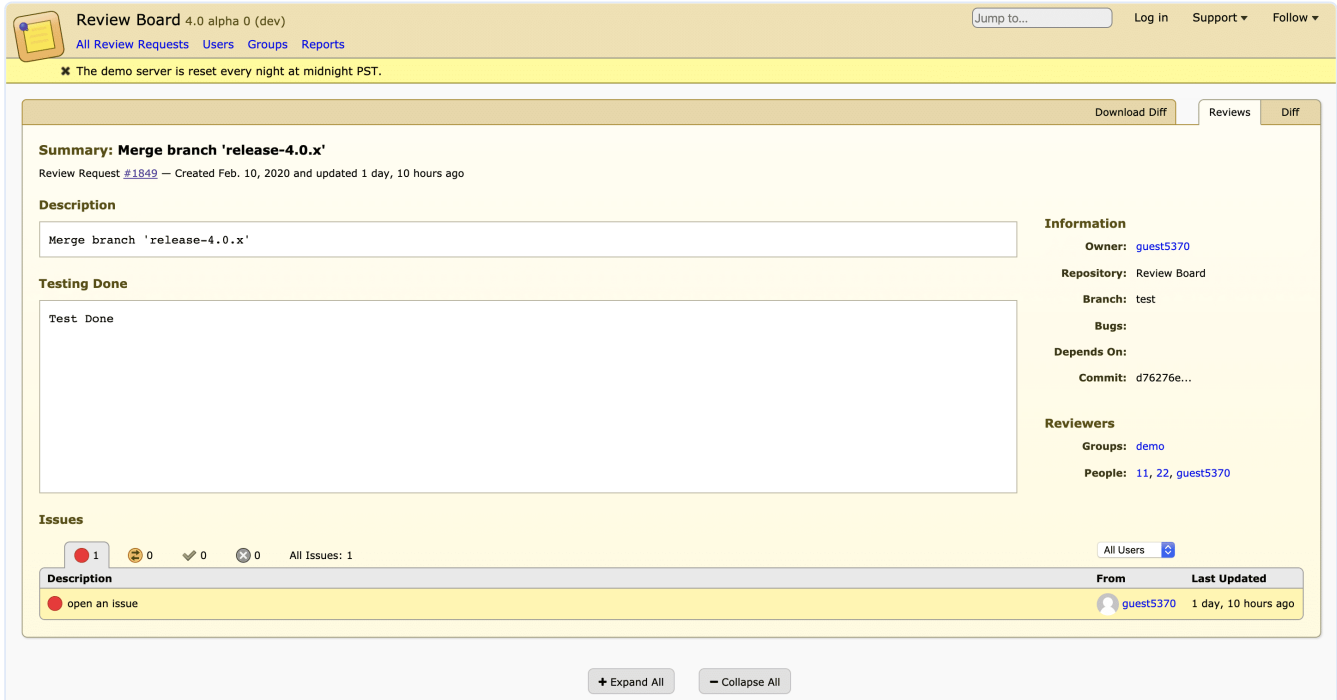
Now let's dive in some of the most popular code review tools!

A Closer Look at 12 Powerful Code Review Tools

In this section, we review the most popular static code review tools.

1. Review Board

[Review Board](#) is a web-based, open source tool for code review. To test this code review tool, you can either explore [the demo](#) on their website or [download](#) and [set up](#) the software on your server.

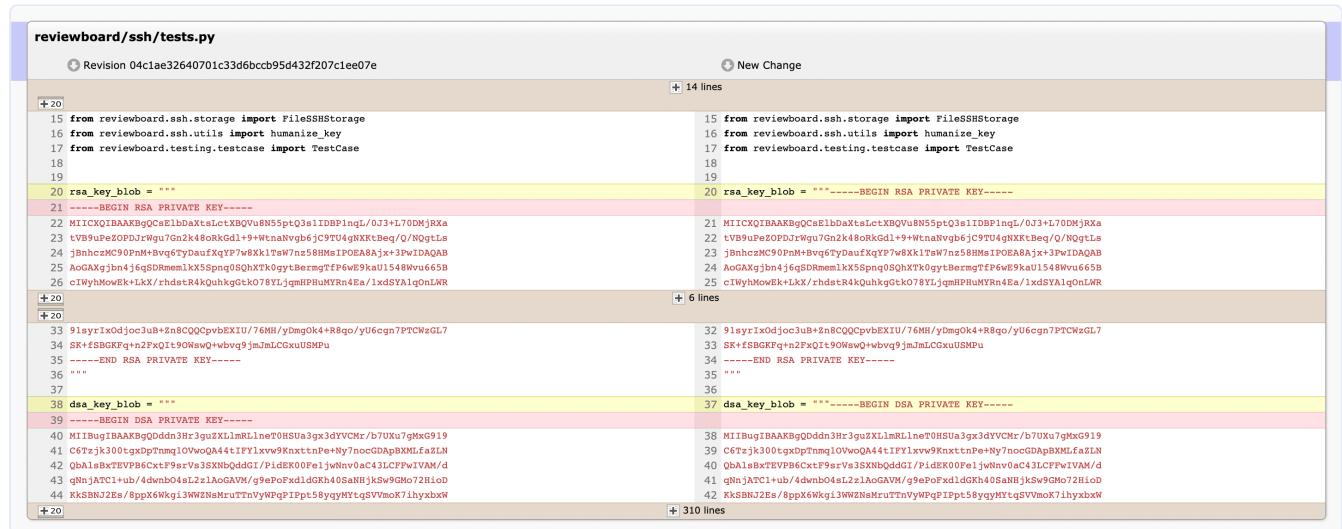


The screenshot displays the Review Board web interface. At the top, the header shows 'Review Board 4.0 alpha 0 (dev)' with navigation links for 'All Review Requests', 'Users', 'Groups', and 'Reports'. A yellow banner below the header states: 'The demo server is reset every night at midnight PST.' The main content area is titled 'Summary: Merge branch 'release-4.0.x'' and includes a 'Download Diff' button. The 'Description' section contains the text 'Merge branch 'release-4.0.x''. The 'Testing Done' section shows 'Test Done'. The 'Issues' section at the bottom indicates '1' issue with a red circle icon. On the right side, the 'Information' panel lists details: Owner: guest5370, Repository: Review Board, Branch: test, Bugs: (empty), Depends On: (empty), Commit: d76276e..., Reviewers: Groups: demo, People: 11, 22, guest5370. A table at the bottom shows the issue list with columns 'Description', 'From', and 'Last Updated'. The first row shows 'open an issue' from 'guest5370' updated '1 day, 10 hours ago'. At the bottom of the interface, there are buttons for '+ Expand All' and '- Collapse All'.

— Review Board Overview

The Python programming language and its installers, [MySQL](#) or [PostgreSQL](#) as a database, and a web server are the prerequisites to run Review Board on a server.

You can integrate Review Board with a wide range of version control systems — [Git](#), Mercurial, CVS, Subversion and Perforce. You can also link Review Board to [Amazon S3](#) for storing screenshots directly in the tool.



— Review Board Changes Overview

Review Board lets you perform both pre-commit and post-commit code reviews depending on your requirements. If you haven't integrated a [version control system](#), you can use a diff file to upload code changes to the tool for a review.

A graphical comparison of changes in your code is also provided. In addition to code reviews, Review Board lets you conduct document reviews too.

The first version of Review Board came out over a decade ago, but it's still in active development. Therefore, the community for Review Board has grown over the years and you will likely find support if you have any issues using the tool.

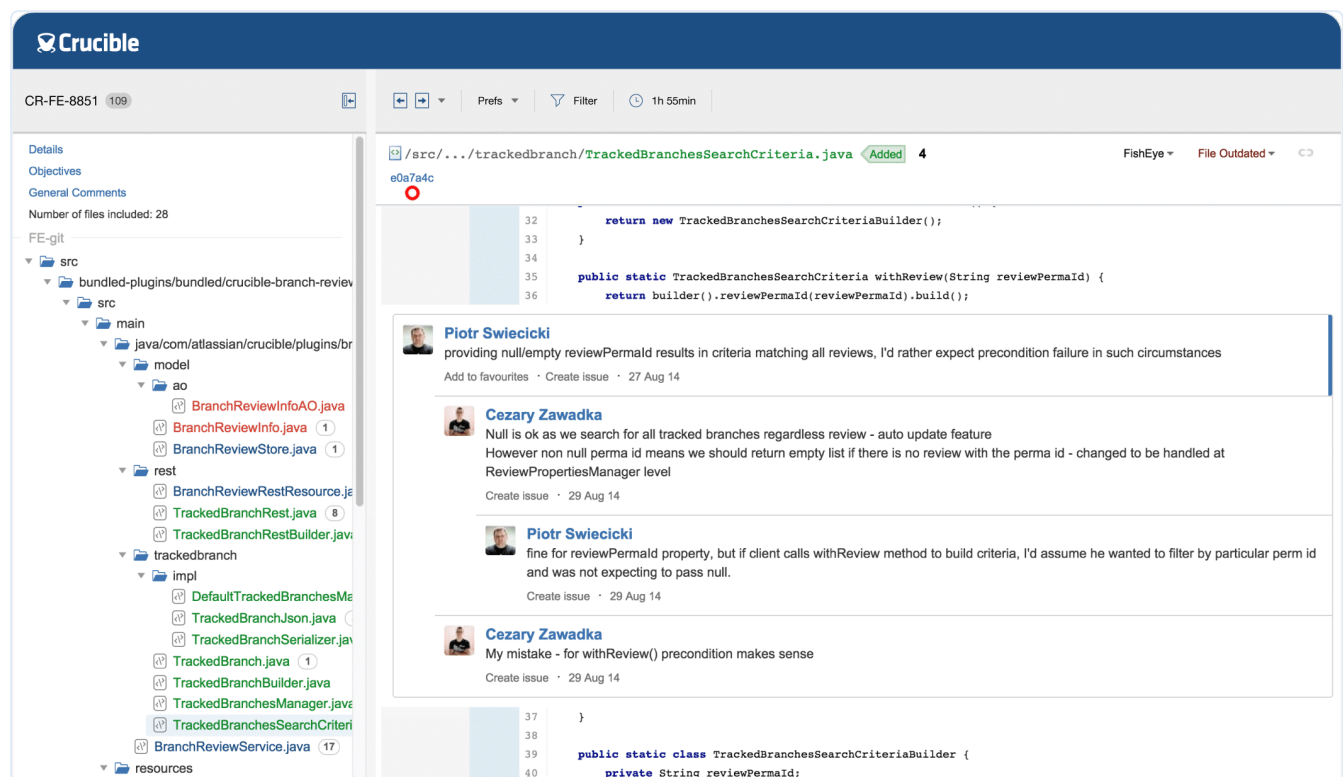
Review Board is a simple tool for code reviews, which you can host on your server. You should give it a try if you do not wish to host your code on a public website.

2. Crucible

[Crucible](#) is a collaborative code review tool by Atlassian. It is a commercial suite of tools that allows you to review code, discuss plans changes, and identify bugs across a host of version control systems.

Crucible provides two payment plans, one for small teams and while the other for enterprises. For a small team, you need to make a one-time payment of \$10 for unlimited repositories limited to five users. For large teams, the fees start at \$1100 for ten users and unlimited repositories.

Both these plans offer a 30-day free trial without the need for a credit card.



— Crucible Code Review Tool ([Source](#))

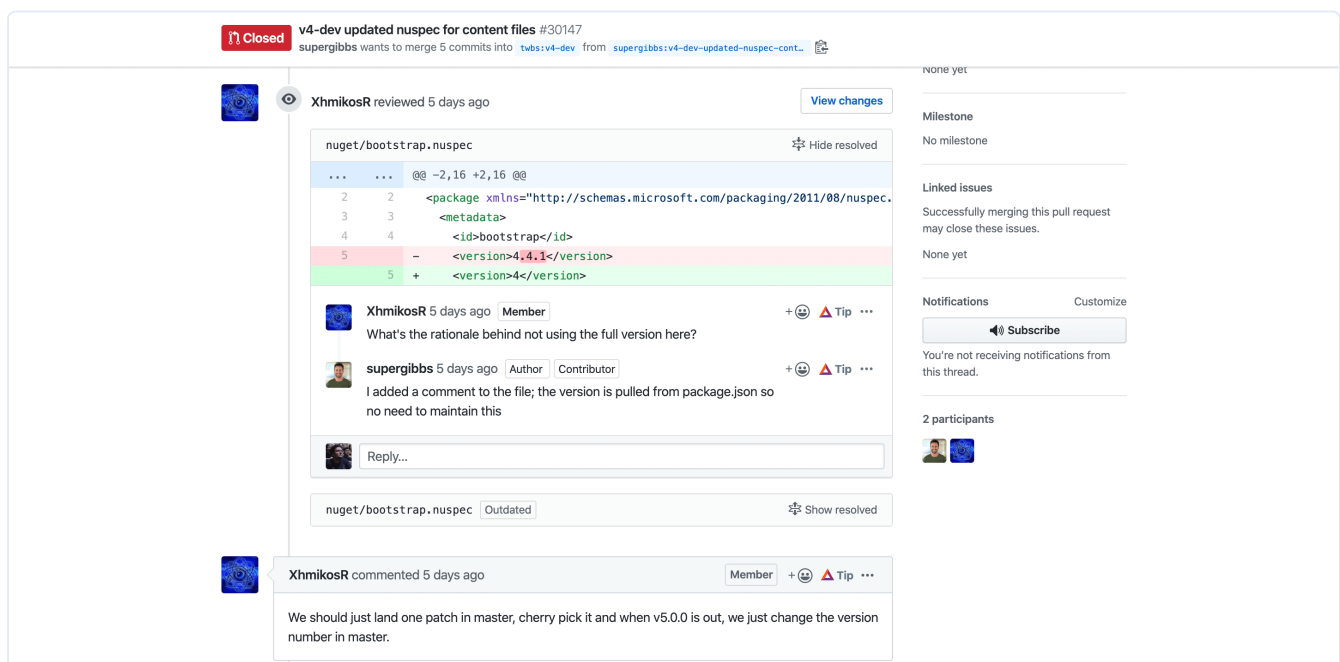
Similar to Review Board, Crucible supports a large number of version control systems — SVN, Git, Mercurial, CVS, and Perforce. Its primary function is to enable you to perform code

reviews. In addition to overall comments on the code, it allows you to comment inline within the diff view to pinpoint exactly what you're referring to specifically.

Crucible integrates well with Atlassian's other enterprise products like Confluence and Enterprise [BitBucket](#). However, you will possibly get the most benefits from Crucible by using it alongside [Jira](#), Atlassian's Issue, and Project Tracker. It allows you to perform pre-commit reviews and audits on merged code.

3. GitHub

If you use [GitHub](#) to maintain your Git repositories on the cloud, you may have already used forks and pull requests to review code. In case you have no idea of what GitHub is, here's a [beginner's guide to GitHub](#) and [the differences between Git and GitHub](#).



— GitHub Code Review Tool within a Pull Request

GitHub has an inbuilt [code review tool](#) in its pull requests. The code review tool is bundled with GitHub's core service, which provides [a free plan](#) for developers. GitHub's free plan limits

the number of users to three in private repositories. Paid plans start at \$7 per month.

GitHub allows a reviewer with access to the code repository to assign themselves to the pull request and complete a review. A developer who has submitted the pull request may also request a review from an administrator.

In addition to the discussion on the overall pull request, you are able to analyze the diff, comment inline, and check the history of changes. The code review tool also allows you to resolve simple Git conflicts through the web interface. GitHub even allows you to integrate with [additional review tools](#) through its marketplace to create a more robust process.

The GitHub code review tool is a great tool if you are already on the platform. It does not require any additional installation or configuration. The primary issue with the GitHub code review tool is that it supports only Git repositories hosted on GitHub. If you are looking for a similar code review tool that you can download and host on your server, you can [try GitLab](#).

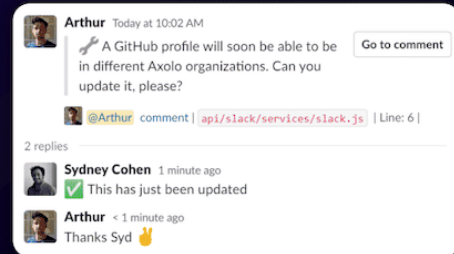
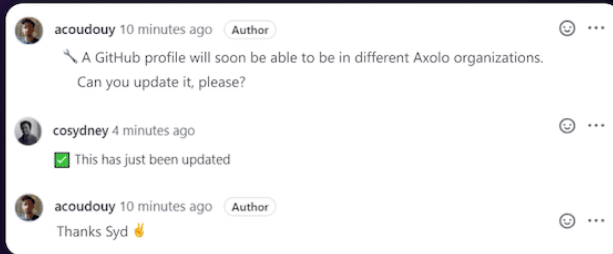
4. Axolo

[Axolo](#) is not what you expect when you read “code review tool”. You’re probably picturing a screen full of diffs. Not at all! Axolo is all about communication. Specifically, Axolo takes all of the usual back-and-forths on [GitHub or GitLab](#) and brings the discussion into Slack.

Handle code reviews from your Slack channels. Slack & GitHub/GitLab will be in total synch.



**CODE REVIEWS CREATE
NEW SLACK THREADS**



**SLACK ANSWERS ARE
PUSHED TO GITHUB/GITLAB**

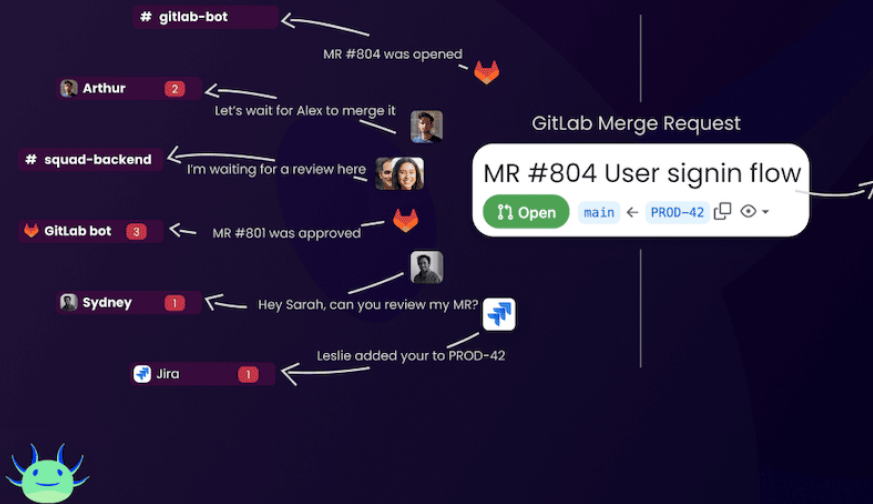


— Axolo Slack discussion

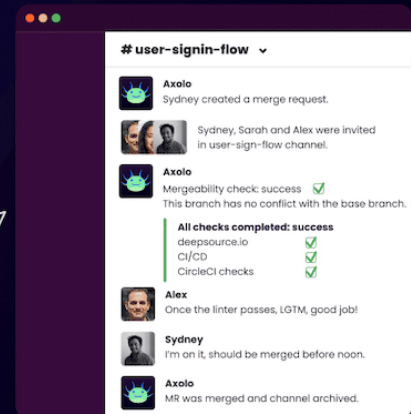
They do that by creating an ephemeral Slack channel for each code review, they invite only the people that should be (the code author, assignees, and reviewers), send only the required notifications in the channel (code comments, CI/CD, ...), and archive the channel once the branch is merged.

It's time to stop slacking around, end context-switching today.

Old Way



Axolo way



— The Axolo way

With daily reminders, pull request recap notifications for stand-ups, and dedicated code review timeslots, engineering teams can seamlessly review code without any stale pull requests.

5. Collaborator

Collaborator by SmartBear is a peer code and document review tool for development teams. In addition to source code review, Collaborator enables teams to review design documents too. A 5-user license pack is priced at \$535 a year. A free trial is available depending on your business requirements.

Review Detail Report

URL: <https://mail361.smartbeartrial.com/go?page=ReviewDetailReport&formSubmittedreportConfig=0&reviewId=6>

Navigation: HOME REVIEW **REPORTS** ADMIN

Search: []

Results [Printable] [PDF]

Overview

Base Information		Defect Information		Time Information	
ID	6	Number of Defects	3	Created On	2018-11-27 21:21:32
Status	Inspection	Number of Open Defects	3	Finished On	2018-11-28 16:46:16
Title	Github pull request #2, my-initial-feedback	Number of Fixed Defects	0	Total Person Time	00:28:43
Creator	Administrator	Number of Tracked Externally Defects	0	Wall-Clock Time	19:24:45
		Number of Comments	2	Total Reviewer Time	00:00:00

User Defined Custom Fields

Overview: [] I changed some copy, made an unordered list ordered, and got rid of some random code. #chaos

Checklist

Title	Status	User	Date
Check for typos in the copy.	Checked	Jack Swigert	2018-11-28 15:35:57
Ensure that links are correct.	Unchecked		
Ensure that the messaging aligns with SmartBear Brand Guidelines	Checked	Jack Swigert	2018-11-28 16:45:11
Test on different mobile devices with CBT: https://crossbowtesting.com/	Unchecked		

Participants

Name	Role	State	Person-Hours
Fred Haise	Product Manager	Active	00:00:00
Jim Lovell	Developer	Active	00:00:00
Glynn Lunney	Tester	Active	00:00:00
Jack Swigert	Security Team Rep	Active	00:28:08

Remote System Links

Remote System	Linked Item	Current Status
plonda/marketing-training	PR#2: my-initial-feedback	open

Defect Log

ID	Creator	File	Location	Severity	Type	Defect State	Text
1	Jack Swigert	email-template	Line 15	Major	Style	Open	This is a bad idea.
2	Jack Swigert	email-template	Line 247	Minor	Performance	Open	I think we need this actually. Why did you remove?
4	Jack Swigert	email-template	Line 469	Minor	Documentation	Open	It was actually just 1100.

Jack Swigert | Support until 2018-12-31 | System | 11.4.11402 | © 2005-2018 SmartBear Software, Inc.

— Collaborator Review Source

Collaborator supports a large number of version control systems like Subversion, [Git](#), CVS, Mercurial, Perforce, and TFS. It does a good job of integrating with popular project management tools and IDEs like Jira, Eclipse, and Visual Studio.

This tool also enables reporting and analysis of key metrics related to your code review process. Moreover, Collaborator helps in audit management and bug tracking as well. If your tech stack involves enterprise software and you need support to set up your code review process, you should give Collaborator a try.

6. CodeScene

[CodeScene](#) is a code review tool that goes beyond traditional static code analysis. It performs behavioral code analysis by including a temporal dimension to analyze the evolution of your codebase. CodeScene is available in two forms: a [cloud-based solution](#) and an [on-premise solution](#).

CodeScene's cloud-based plans start free for public repositories hosted on [GitHub](#). For up to ten private repositories and a team of ten members, CodeScene costs €99 (about \$115) per month. An on-premise installation of CodeScene costs €15 (about \$17) per developer per month.

CodeScene™
Powered by Empear

</> Projects 🔍 Plans 📄 Documentation 👤 sdaityari@gmail.com [sdaityari] ▾

ANALYSIS RESULTS
TO-DO-LIST ⚙️

Dashboard
Scope ▾
Analysis Data
System Trends
Technical Debt
Architecture
Team Dynamics
System

Analysis Scope

Project information, analysis scope, evolutionary data, and file content metrics.

Project	to-do-list	Commits	4
Includes History From	2013-12-01 12:04:26 (UTC)	Entities	6
Analyzed At	2020-03-09 21:00:19 (UTC)	Changed entities	9
Duration	53 seconds	Authors	3
		Active Authors	1

File Content

Language	Files	Code	Comment	Blank
JavaScript	4	162	51	46
CSS	3	68	24	14
HTML	1	47	5	13
Markdown	1	1	0	0

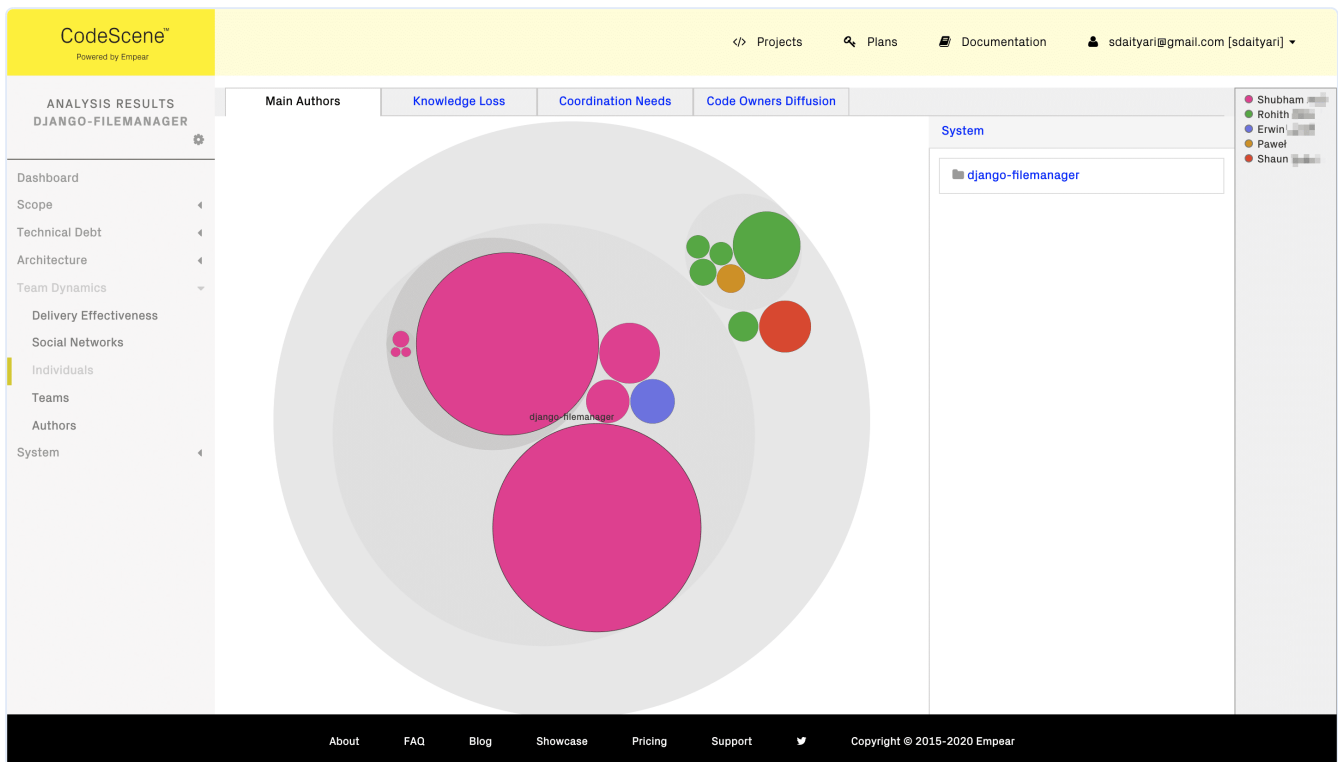
AboutFAQBlogShowcasePricingSupport🐦

Copyright © 2015-2020 Empear

— CodeScene Code Review Tool Analysis

CodeScene processes your version control history to provide code visualizations. In addition to this, it applies machine learning algorithms to identify social patterns and hidden risks in code.

Through the version control history, CodeScene profiles every team member to map out their knowledge base and create inter-team dependencies. It also introduces the concept of hotspots in your repository by identifying files that undergo the most development activity. These hotspots require the highest attention going forward.



— CodeScene Knowledge Maps

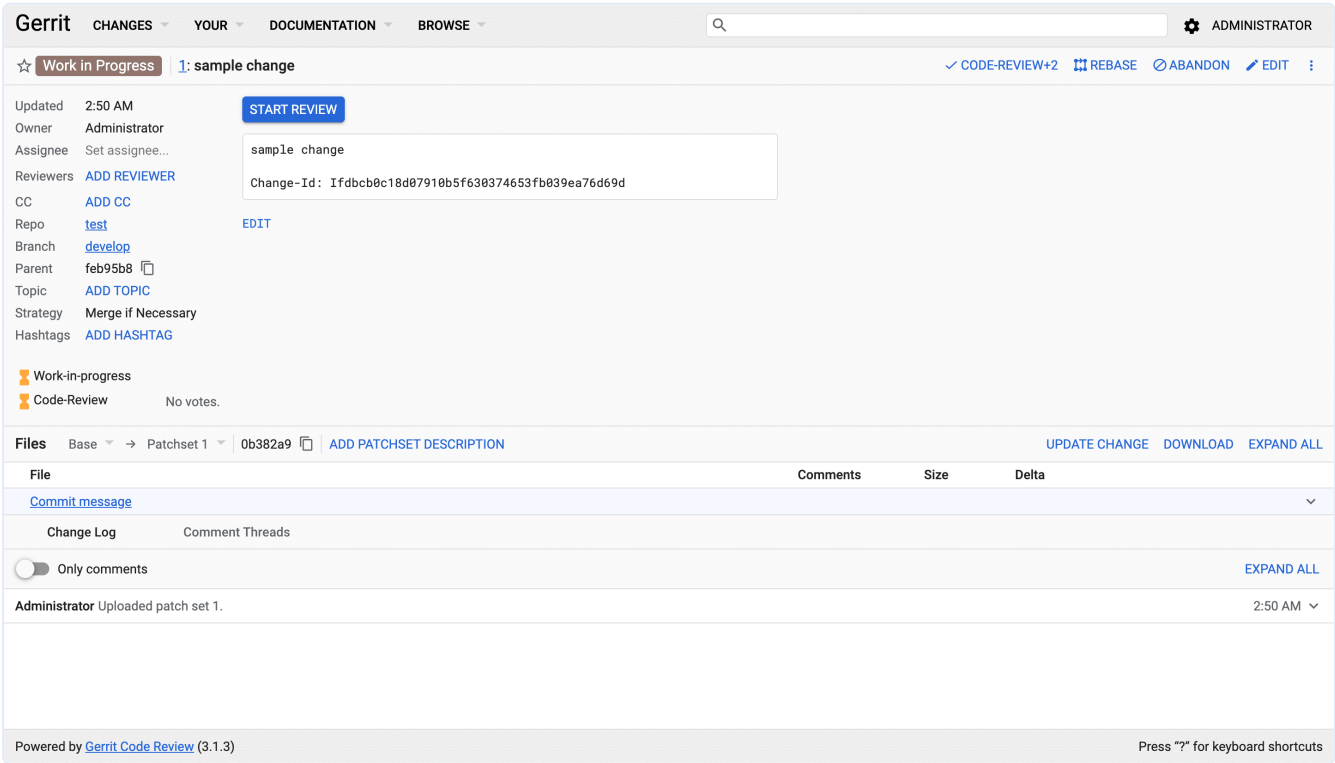
If you are looking for a tool that goes beyond a traditional, conversational code review tool, make sure to check out the free trial of CodeScene. To learn more about the underlying logic behind CodeScene's behavioral code analysis, check out this white paper on [CodeScene's use cases and roles](#).

7. Visual Expert

[Visual Expert](#) is an enterprise solution for code review specializing in [database code](#). It has support for three platforms only: PowerBuilder, [SQL Server](#), and [Oracle PL/SQL](#). If you are using any other DBMS, you will not be able to integrate Visual Expert for code review.

A free trial is available, but you need to send a request to get a quote on its pricing.

installation process for a standalone version of Gerrit.



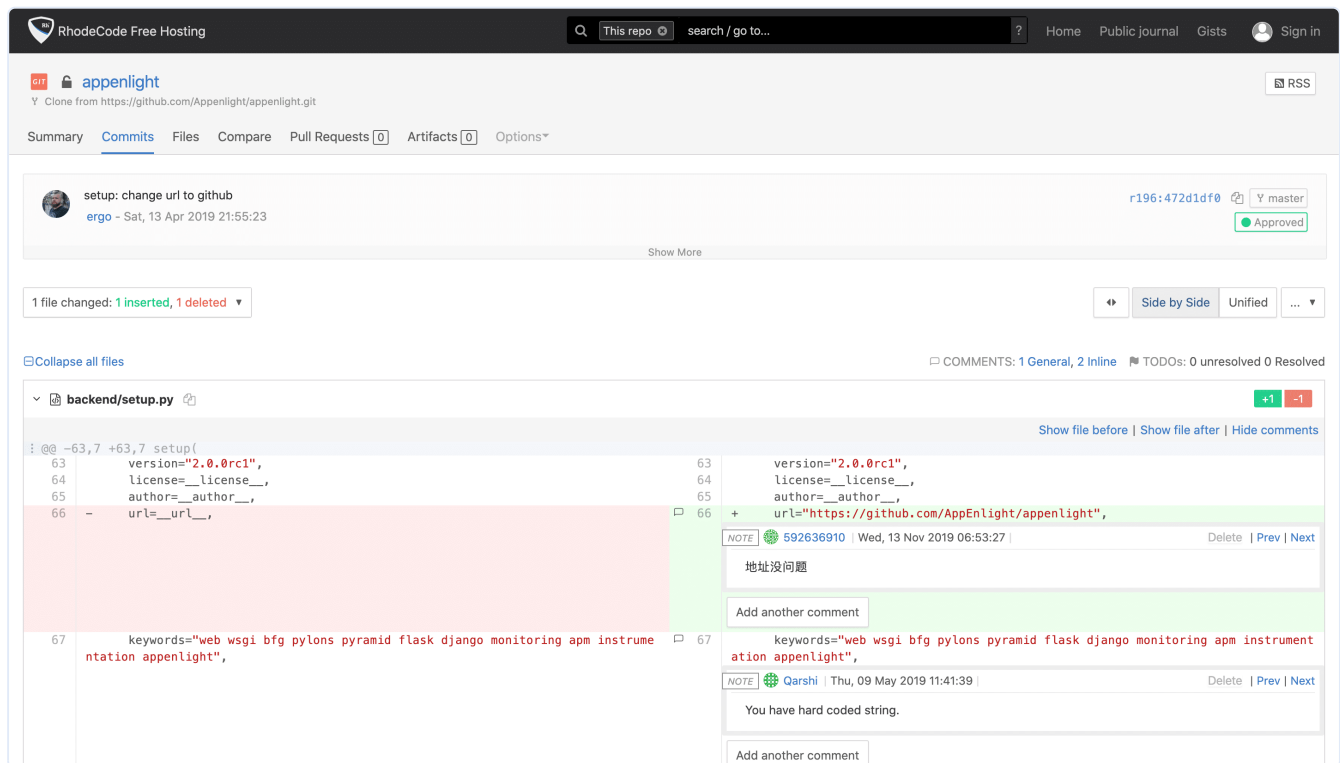
— Gerrit Code Review Tool

Gerrit combines the functionality of a bug tracker and a review tool into one. During a review, changes are displayed side by side in a unified diff, with the possibility to initiate a conversation for every line of code added. This tool works as an intermediate step between a developer and the central repository. Additionally, Gerrit also incorporates a voting system.

If you possess the technical expertise to install and configure Gerrit, and you are looking for a free code review tool, it should serve as an ideal solution for your projects.

9. Rhodocode

[Rhodocode](#) is a web-based tool that assists you in performing code reviews. It supports three version control systems: Mercurial, Git, and Subversion. A cloud-based version of Rhodocode starts at \$8 per user per month, whereas an on-premise solution costs \$75 per user per year. While it is enterprise software, its [community edition](#), which is free and open source, can be downloaded and compiled free of charge.



— Rhodocode

Rhodocode enables a team to collaborate effectively through iterative, conversational code reviews to improve code quality. This tool additionally provides a layer of permission management for [secure development](#).

In addition, a visual changelog helps you navigate the history of your project across various branches. An [online code editor](#) is also provided for small changes through the web interface.

Rhodecode integrates seamlessly with your existing projects, which makes it a great choice for someone looking for a web-based code review tool. Therefore, the community edition is ideal for those with technical expertise looking for a free and dependable code review tool.

10. Veracode

[Veracode](#) provides a suite of code review tools that let you automate testing, accelerate development, integrate a remediation process, and improve the efficiency of your project. The suite of code review tools by Veracode is marketed as [a security solution](#) that searches for vulnerability in your systems. They provide a set of two code review tools:

- [Static Analysis](#): A tool that enables developers to identify and fix security flaws in their code.
- [Software Composition Analysis](#): A tool that manages the remediation and mitigation process of flaws in code.

The screenshot displays the Veracode web interface for a project named 'Webgoat_.NET'. At the top, there's a navigation bar with links like 'MY PORTFOLIO', 'SCANS & ANALYSIS', 'REPORTS', 'POLICIES', 'eLEARNING', and 'DIRECTORY'. Below this, the application name 'Webgoat_.NET' is shown. The main area is divided into two sections. The top section shows a code scan for 'Static: 8_10_17 | Dynamic' with a 'Schedule a Consultation' button. It displays a snippet of C# code from 'Util.cs'. The bottom section, titled 'Flaws: 2 of 65 | Filtered by: Flaw Source = user', shows a table of detected vulnerabilities. Two flaws are listed, both with a 'Very High' severity and a 'V.Likely' exploitability. Both flaws are related to 'Improper Neutralization of Special Elements used in an OS Command (OS Command Injection)' in the 'dotnetgoat.dll' module. The table columns include ID, Severity, Exploitability, CWE ID & Name, Publish Time, Module, Source, Paths, Status, and Mitigation Status.

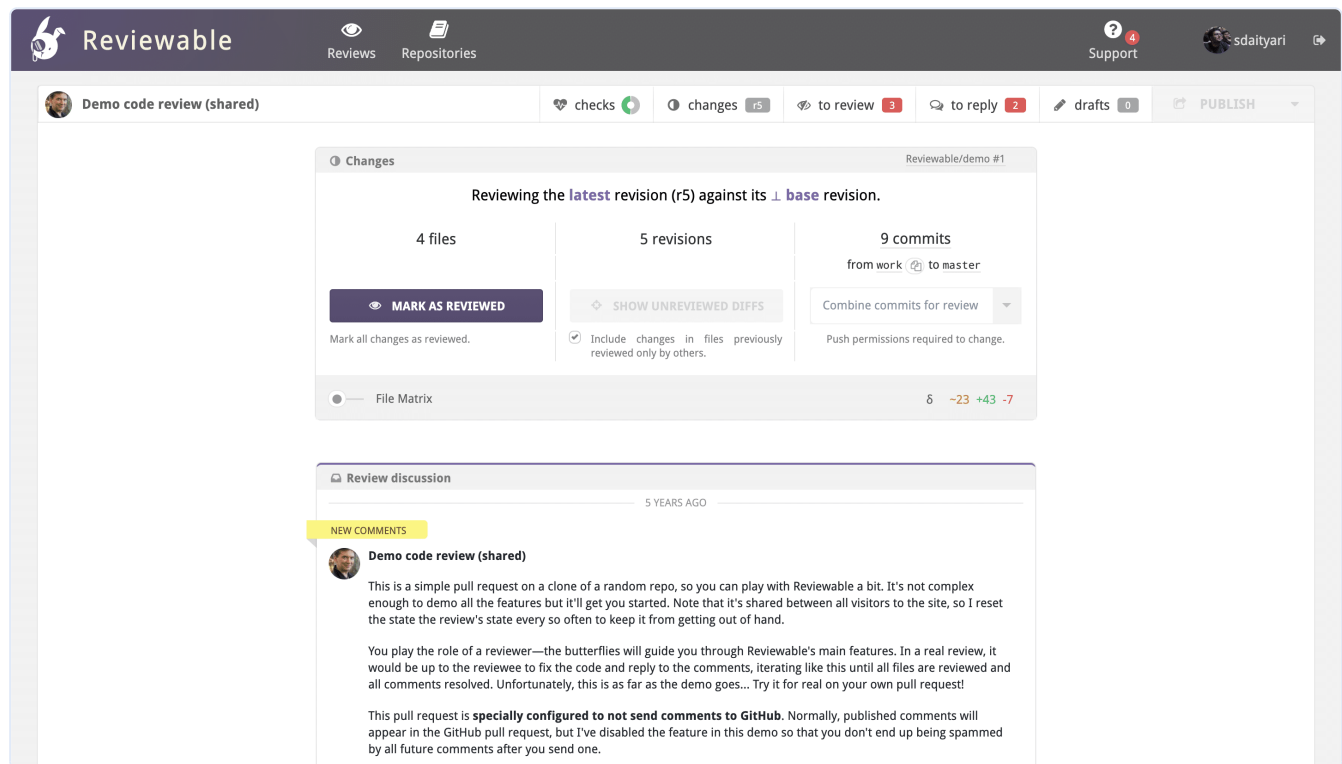
ID	Sev	Exp	CWE ID & Name	Publish Time	Module	Source	Paths	Status	Mitigation Status
63	Very High	V.Likely	Improper Neutralization of Special Elements used in an OS Command (OS Command Injection)	8/16/2017 3:46 PM EDT	dotnetgoat.dll	util.cs: 52	1 Path	Reopened	None
113	Very High	V.Likely	Improper Neutralization of Special Elements used in an OS Command (OS Command Injection)	8/16/2017 3:46 PM EDT	dotnetgoat.dll	util.cs: 30	1 Path	Reopened	None

— Veracode Overview ([Source](#))

Code review is a part of the Software Composition Analysis and you can opt for [a demo of Veracode](#) before committing fully to it. Here is the link to request [a quote](#).

11. Reviewable

[Reviewable](#) is a code review tool for [GitHub pull requests](#). It is free for open source repositories, with plans for private repositories starting at \$39 per month for ten users. Since the tool is integrated with GitHub, you can sign in using your GitHub account and get started.



— Reviewable Code Review Tool Overview

If you would like to check out a typical review on Reviewable, you can head over to [a demo review](#).

One interesting thing about Reviewable is that it overcomes a few drawbacks of the code review in GitHub's pull requests feature. For instance, a comment on a line of code is

automatically hidden by GitHub once a developer changes the line because GitHub assumes that the issue has been fixed. But, in reality, things may be different.

Also, GitHub has relatively small line limits for displaying file diffs.

If you are looking for a tool tightly coherent with GitHub but would like more features than pull requests, Reviewable should be your go-to tool.

12. Peer Review for Trac

If you use Subversion, the [Peer Review Plugin for Trac](#) provides a free and open source option to conduct code reviews on your projects. The Peer Review Plugin integrates into the [Trac open source project](#), which is a wiki and issue tracking system for development projects.

My Code Review - Review Administration: Accounts ... Code Review Details - Rev... +

localhost:8000/Review/peerReviewMain

logged in as Tester Logout Preferences Help/Guide About Trac

Projektliste Wiki Timeline Roadmap Browse Source View Tickets New Ticket Search Tags Admin Beenden SQL Peer Review

My Code Reviews Create a Code Review Report Search Code Reviews

My Code Reviews

This is a list of open code reviews currently assigned to you.

☐ Show my finished reviews. Update

When selected the list will include reviews already finished and marked as **reviewed** by the reviewer. Note that reviews which are marked as finished by the author will not be shown.


Review	Author	Name	Review Status	My Status	# Comments	# Not Read	Date Opened	Files
2	Tester	Review of Initial checkin	new	new	3	2	06.02.2016	/peerreview/trunk/codereview/templates/peerReviewBrowser.html @ 1 /peerreview/trunk/codereview/templates/peerReviewNew.html @ 1 /peerreview/trunk/codereview/peerReviewMain.py @ 1 /peerreview/trunk/codereview/peerReviewNew.py @ 1
3	Reviewer	Followup of 'Review of Initial checkin'	new	new	1	1	06.02.2016	/peerreview/trunk/codereview/peerReviewNew.py @ 4 /peerreview/trunk/codereview/templates/peerReviewBrowser.html @ 4 /peerreview/trunk/codereview/templates/peerReviewNew.html @ 4

This is a list of code reviews created by you.

☐ Show closed reviews. Update

When selected the list will include reviews marked as **closed**. These are obsolete reviews.

Review	Name	Review Status	# Comments	# Not Read	Date Opened	Files
2	Review of Initial checkin	new	3	2	06.02.2016	/peerreview/trunk/codereview/templates/peerReviewBrowser.html @ 1 /peerreview/trunk/codereview/templates/peerReviewNew.html @ 1 /peerreview/trunk/codereview/peerReviewMain.py @ 1 /peerreview/trunk/codereview/peerReviewNew.py @ 1


 Powered by Trac 1.0.9
 By Edgewall Software.

— Peer Review Plugin for Trac Overview ([Source](#))

Trac integrates the wiki and issue tracker with your reviews to provide an end-to-end solution. While the basic functionality of comparing changes and conversation is available, the plugin lets you design [customized workflows](#) for your projects.

For example, you could decide tasks to be done on triggers like the submission of a change or approval in a code review. You can also create [custom reports](#) on your projects.

If you are also looking for a wiki for documentation and an issue tracker to manage your project's roadmap, Trac should provide a good option for you.

Summary

The code review process plays a key role when it comes to boosting the efficiency of your organization. Specifically, taking advantage of the right code review tool is what helps you to remove redundancy in your development cycle.

We looked closer to the most popular code review tools available and here's what we found:

- For a small team just starting out, [Review Board](#) is a good choice to initiate the code review process.
- If you are looking for an open-source code review tool, give [Gerrit](#), [Peer Review for Trac](#), or [community edition of Rhodocode](#) a try.
- Are you looking for a fairly easy to use code review tool with support? You should try out [Rhodecode](#).
- If you use Git and GitHub to manage your codebase, give [GitHub](#)'s inbuilt code review editor a try. If you want to go beyond the basic features of pull requests, you should check out [Reviewable](#).
- Do you belong to a team that uses Oracle, SQL Server, or PowerBuilder for your database code management? You can try out [Visual Expert](#), a code review tool that specializes in database code.
- If you are looking for an enterprise solution, try out Atlassian's [Crucible](#), SmartBear's [Collaborator](#), or, [Veracode](#).
- In case you want to use ML and AI to go beyond code review into the behavioral analysis, you should check out [CodeScene](#).

Now it's your turn: what code review tool are you using? Why? Tell us in the comments!

Suggested reading:

- [Top 13 Scripting Languages You Should Pay Attention to](#)
- [What Is the Best Programming Language to Learn](#)