THE STATE OF CODE REVIEW

Benchmarking Adoption of Code Review and its Impact on Software Quality

INTRODUCTION

Welcome to SmartBear Software's State of Code Review 2013, the first installment of our research series.

As the leader in code review tools and an advocate for improving software quality practices, SmartBear sought to find an answer to a seemingly simple question – **does code review matter?**

We launched an extensive study about code review practices, software quality, and how the two are related (or not!). We attracted respondents to our survey by placing ads in various developer-oriented online properties, as well as contacting practitioners who are on our mailing lists.

The response was overwhelming. Over **650 individuals participated** in the study, and 70% of them completed the extensive, 40-question survey entirely.

We found an audience that has a passion for development efficiency and quality.

From that audience, we learned that code review doesn't just matter, but serves as a critical development practice to the organizations with the greatest levels of software quality. In fact, over **70%** of the respondents in the survey participate in some kind of collaborative review, and those that do code review are **twice as likely** to be highly satisfied with their overall software quality than those that do not do code review.

Included in this report, you will receive code review and software quality performance benchmarks for participants in your industry, your organization size, and your development practices. You will learn what the most successful organizations are doing differently to get ahead, and which best practices you can apply for your own improvements in software quality.

For more reports and eBooks from SmartBear, visit www.smartbear.com.

Happy coding.

The SmartBear Software Practices Research Lab

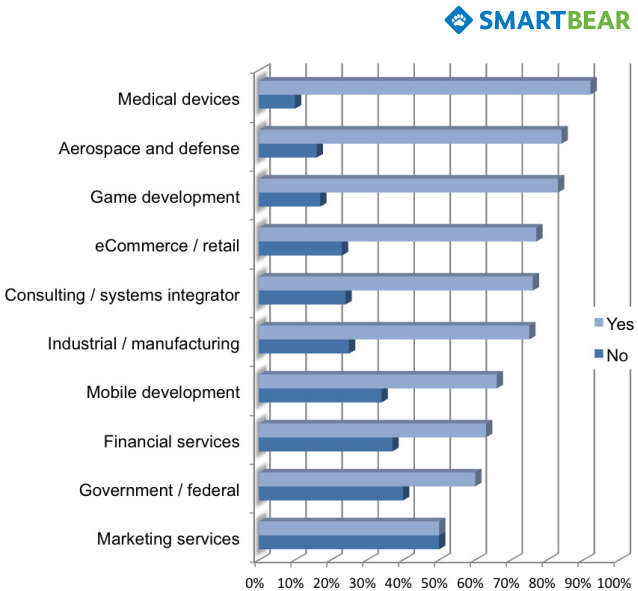
TABLE OF CONTENTS

WHO IS DOING CODE REVIEW?

This section will cover which industries, organization sizes, and types of development processes are participating in code reviews currently.

Use of code review by industry sector

Does your organization participate in code reviews or a peer-to-peer process of reviewing and improving source code?



SmartBear Software 2013 Software Quality Benchmark Survey
Fielded: January 2013 N=683 Customer N= 115

When it comes to software quality, code review is clearly a priority with 70% of all respondents indicating that they do some kind of code review.

Nearly all organizations in the medical devices sector participate, as well as those in aerospace and defense. What do these types of organizations have in common?

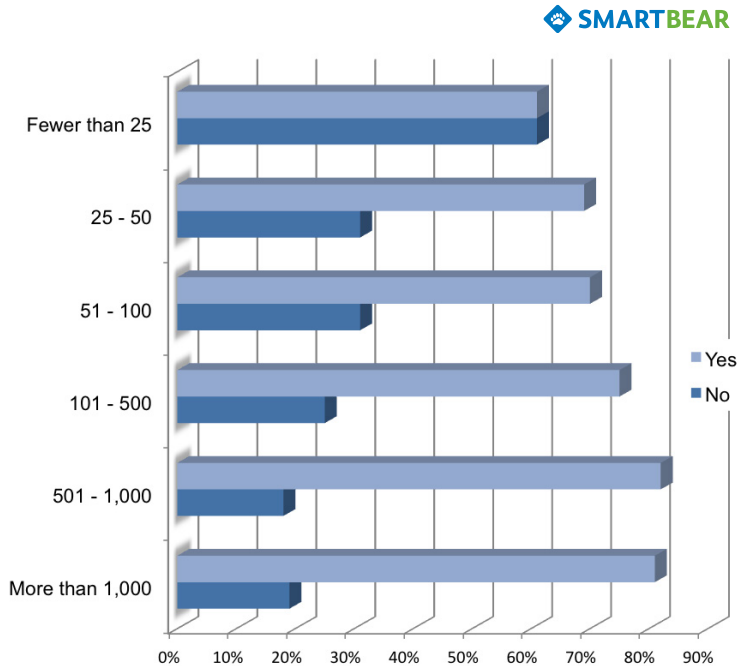
- The software is generally embedded in a physical product, making (bug fix) updates more difficult to accomplish.
- Both organizations are risk averse, since they serve markets where lives are at stake. A “small” error when controlling either the application of medical treatments, or when aiming missiles, is generally intolerable.
- They are both regulated industries and are generally forced to adhere to regulations or mandates regarding the assessment and verification of their quality.
- These organizations often require their developers to adhere to strict coding and architectural standards.

The next industry to have a high adoption rate for code review is game development. Those not familiar with the challenges in gaming may think this to be a peculiar revelation, but this industry isn’t just “fun and games.” A game’s graphics and paths can require complex coding, and code quality is imperative to their audience. This industry’s buyers (aka game players) invest an extraordinary amount of time to move from one level to the next. A bug that thwarts progress may not have a workaround, making all the time already invested in the game wasted time. This can be an infuriating circumstance for the game player. Since gamers have vast online communities; these types of defects can destroy a game’s reputation and adoption.

Another reason for the high adoption of code review in game development may be the nature of games themselves, which makes traditional, automated testing difficult. In order to create automated tests, the expected result of the test must be well understood. The tester sets up

the pre-conditions, runs the test and verifies that the post-conditions are what was expected. But in a game, it is nearly impossible to create enough of these test cases for adequate coverage. In this situation, assessing the code quality using code review is essential.

Use of code review by organization size



SmartBear Software 2013 Software Quality Benchmark Survey
Fielded: January 2013 N=683 Customer N= 115

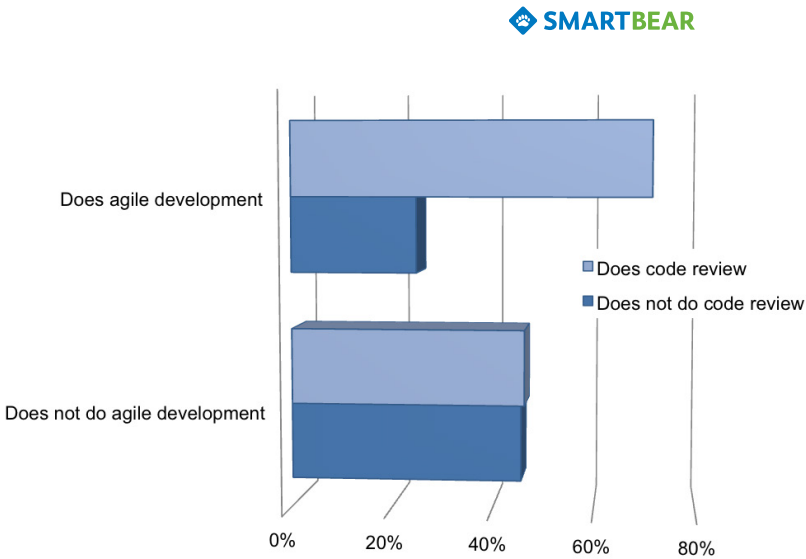
As organization size increases, so does the use of code review.

The larger an organization is, the larger their development team will be and the more likely they are to be geographically distributed. When the number of team members rises, so does the potential gain from collaboration and the need for formal processes.

For example, of the organizations that have more than 500, or 1,000 employees, 80% of them partake in some kind of code review practice.

Smaller organizations tend to have smaller development teams, and organizations with fewer than 25 employees may even have development teams as small as only two or three team members. Perhaps these teams feel that they can't afford to tie up their small development teams doing code reviews or, they may have constant communication about their code by virtue of their small team size.

Agile development by use of code review



SmartBear Software 2013 Software Quality Benchmark Survey
Fielded: January 2013 N=683 Customer N= 115

We wanted to learn more about the relationship between Agile organizations and the use of code review, so we asked participants to identify themselves as either using Agile methodologies, or not, and compared that to their participation in code review.

The vast majority of organizations who are Agile also participate in some kind of code review practice. Conversely, organizations that use other methodologies are evenly split in their adoption of code review. [Tweet this]

Some may find this result surprising, assuming that the fast-paced nature of agile development methodologies wouldn't allow for adoption of "time-consuming" processes like code review. But it's simply not true.

In fact, central to the Agile Manifesto is collaboration. And code review is a highly collaborative technique. Code review is not JUST for finding defects, a key advantage is leveraging the reviews for knowledge transfer and team communication.

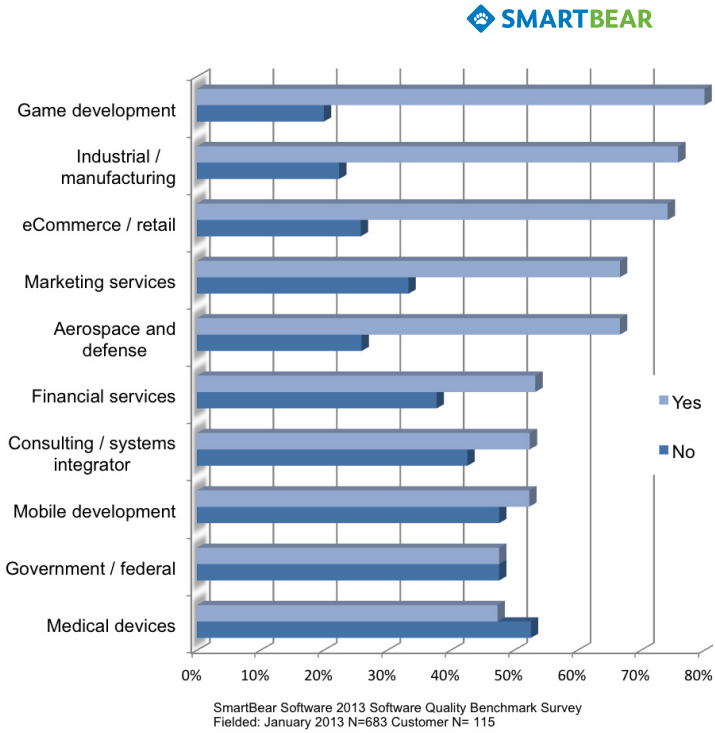
Furthermore, this debunks the myth that code reviews take too much time. Who has less time than organizations that adopt rapid development processes? In fact, code review reduces the total development lifecycle by eliminating defects early.

WHO IS USING A TOOL TO REVIEW CODE?

We know that 70% of organizations are doing code review. Now let's look at how many of them use a tool to facilitate this practice. In this section, we'll learn the industries, organization sizes, and types of development teams are using a tool for code review.

Tool-assisted review by industry sector

Do you, or have you ever, used any tools to assist you with your code review?

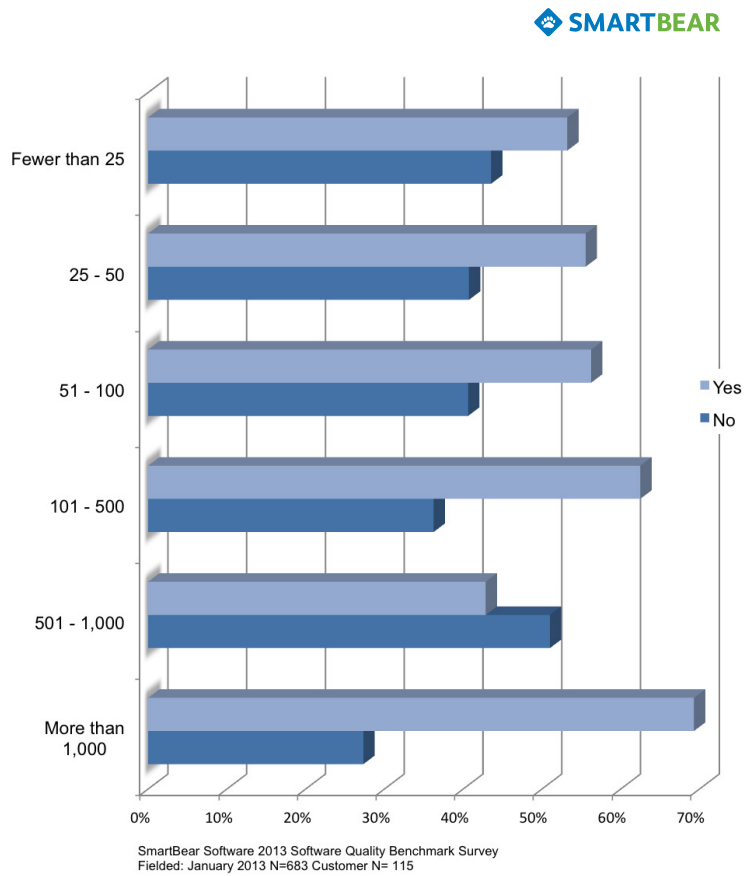


Although various methods for code review exist, a surprising majority of survey respondents - 61% of them - have used code review tools at one time or another.

Organizations in the game development industry are among the top three industries for both use of code reviews and a code review tool. [Tweet this] Considering the pressure on this sector to produce cutting edge technology, it is not surprising they would be at the top of the list for adoption of technology in their daily work.

We were surprised, however, to find that participants in the medical devices industry indicated that they were the most likely to do code reviews, however they are the least likely to use a tool. A tool would certainly help them to answer their auditing needs, both internally and externally, so this was an unexpected result for us to see.

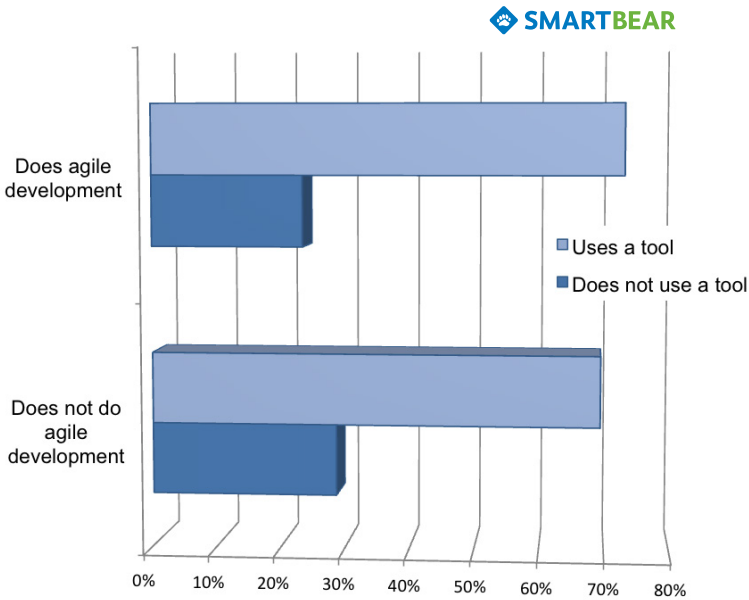
Use of a code review tool by organization size



SmartBear Software 2013 Software Quality Benchmark Survey
Fielded: January 2013 N=683 Customer N= 115

For use of code review, participation increased as organization size did, i.e., the more employees an organization had the greater the likelihood that they would participate in code reviews. When considering the use of a code review tool, however, the relationship between participation and organization size exists, though it is not as apparent as the trend with the use of code review. Organizations with 501 – 1,000 employees do not use a code review tool more often than they do. Still, the largest organizations with more than 1,000 employees are the most likely to use a code review tool. [Tweet this]

Agile development by tool-assisted review



SmartBear Software 2013 Software Quality Benchmark Survey
Fielded: January 2013 N=683 Customer N= 115

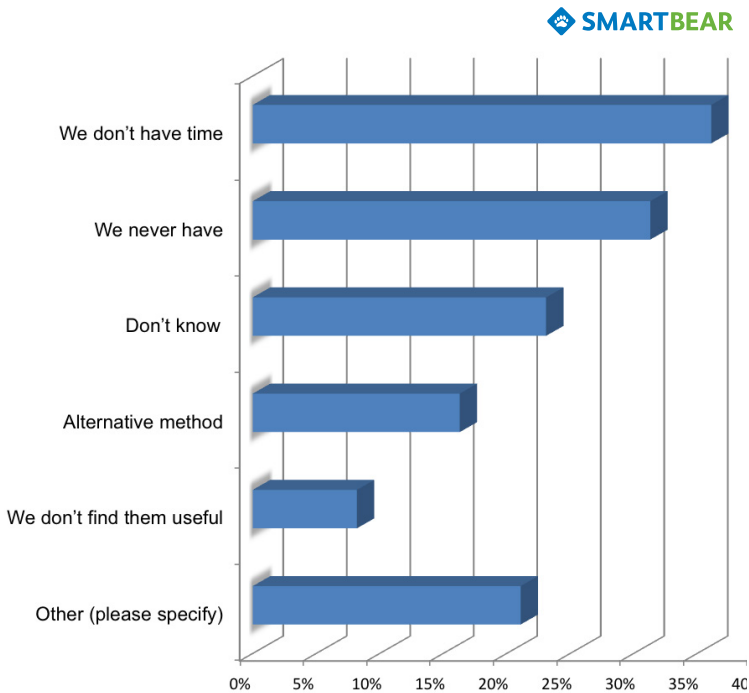
Earlier, we learned that agile teams are highly likely to do code reviews. In the above chart, we also learn that they are highly likely to use a tool. What's interesting about this dataset is that organizations that are not agile are not highly likely to do code reviews, but those that do are highly likely to use a tool.

WHY DON'T YOU DO CODE REVIEWS?

Now that we have an understanding of the 70% who are doing code reviews, let's review the other 30% and why they aren't practicing code review.

Reasons organizations haven't adopted code review

Why don't you participate in code reviews? Check all that apply.



SmartBear Software 2013 Software Quality Benchmark Survey
Fielded: January 2013 N=683 Customer N= 115

The top reason for not participating in code review is lack of time. This is a common misconception of code review because it ties up development staff, especially senior developers, however, removing defects early in the process saves overall development time because the testing efforts are lower. The earlier you can catch defects, the cheaper and faster they are to fix.

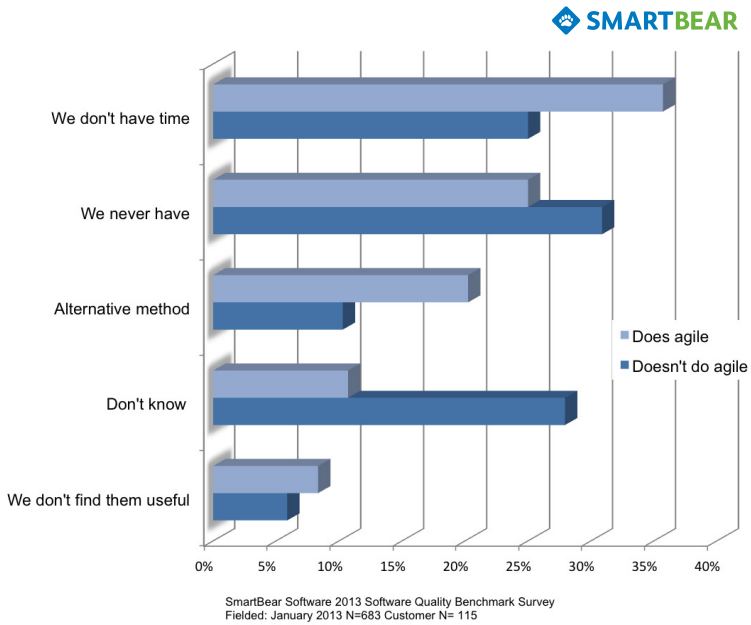
The second most popular response, “we never have” may indicate that the organizations have not had a “compelling event” – a situation where the lack of code reviews caused significant downstream difficulties.

Interestingly, organizations that have presumably attempted code reviews in the past and have not found them useful make up only 8% of total responses.

A significant percentage (21%) indicated some “other” reason for not participating in code reviews. For that choice, respondents were invited to enter open-ended responses so we could get a better understanding of their situation. The most popular responses include:

- having small / solo teams,
- issues getting management approval,
- not having a significant “culture of quality” in the organization,
- and our favorite: “we’re dumb.”

Reasons organizations haven't adopted code review, by Agile development



SmartBear Software 2013 Software Quality Benchmark Survey
Fielded: January 2013 N=683 Customer N= 115

Agile organizations that don't do code review (which is a low percentage of Agile teams) are the most likely to cite not having time as their primary reason for not participating in code review. Agile teams are also far more likely to use an alternative method to improving software quality than non-agile teams.

There are several aspects that might be influencing Agile teams who don't do code reviews:

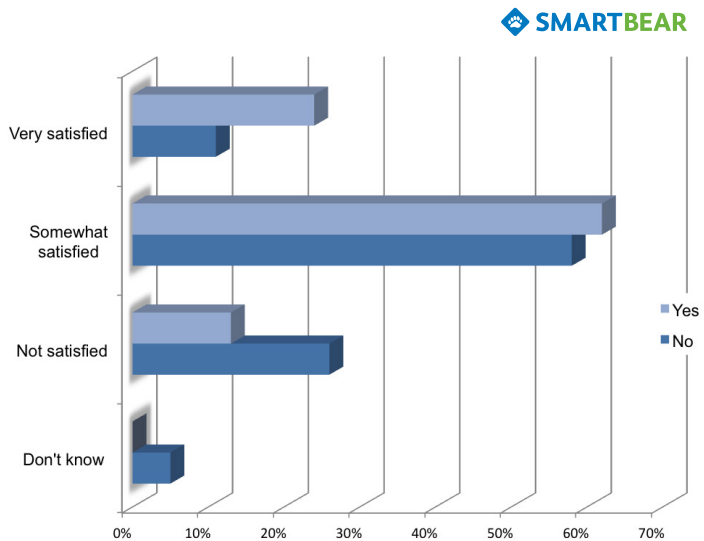
- Small, highly collaborative teams discussing implementations prior to coding
- Compressed sprint schedules that don't allow room for quality checks prior to deployment

HOW DOES CODE REVIEW AFFECT SOFTWARE QUALITY?

To determine code review's impact on software quality, we analyzed respondent's satisfaction with the overall quality of their software and broke it out by the use of code review as a practice, and the use of a tool as well.

Software quality by use of code review

How satisfied are you with the overall quality of your software, in terms of quantity and severity of bugs, customer satisfaction, etc.?



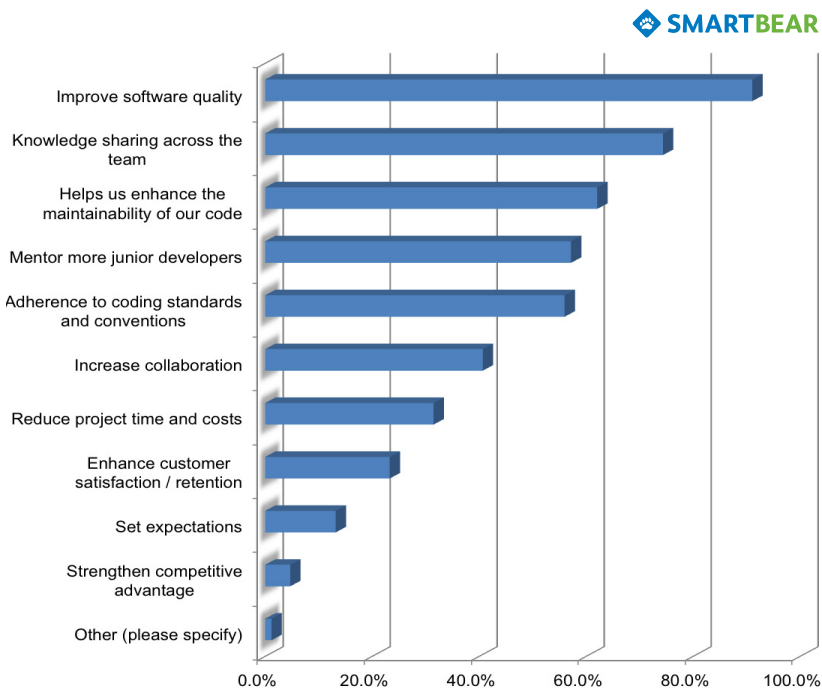
SmartBear Software 2013 Software Quality Benchmark Survey
Fielded: January 2013 N=683 Customer N= 115

In the above chart, the light blue bars represent organizations that do code reviews – with or without a tool. Those that do code reviews clearly have a greater level of satisfaction in the overall quality of their software – they were twice as likely to indicate that they were very satisfied with their software quality. [Tweet this]

Conversely, those that are not satisfied with the quality of their software, or don't even know if they're satisfied, are more likely to not participate in code reviews.

Top Benefits of Code Review

What do you think are the most important benefits of code review?

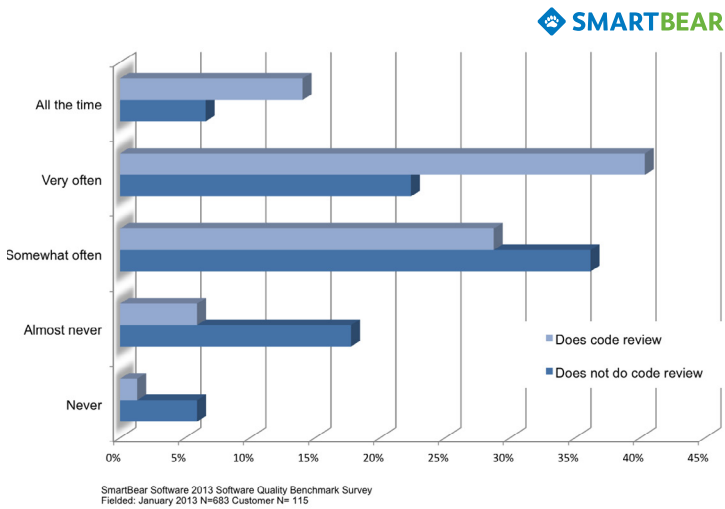


SmartBear Software 2013 Software Quality Benchmark Survey
Fielded: January 2013 N=683 Customer N= 115

Of the 70% of organizations that do code review, their top reason is to improve software quality – **and it's working**. There are a number of additional benefits to code review, including knowledge sharing, maintainability and mentoring. While some organizations might opt for static analysis rather than code review, the results above point to some key benefits that static analysis can't provide: knowledge sharing, mentoring, and increased collaboration.

Milestones and Mandates by use of Code Review

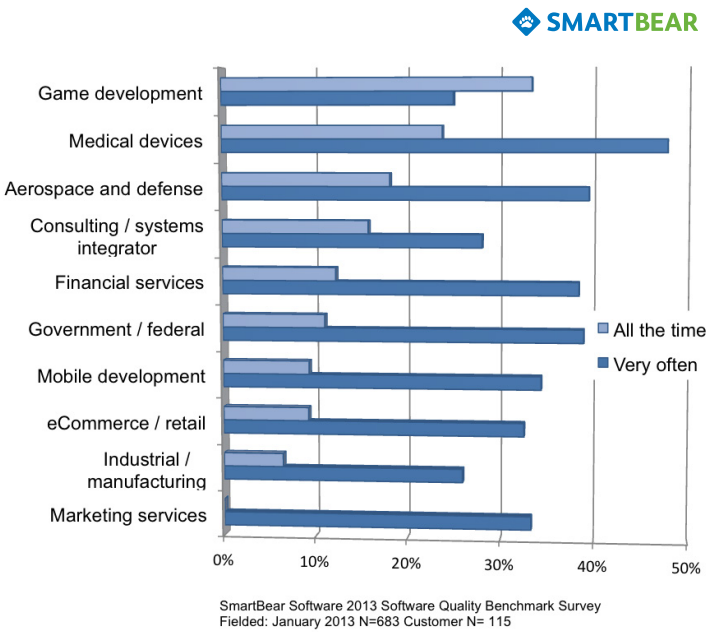
Are you able to meet milestones and mandates on time?



Organizations that don't do code review in order to "save time," take notice: organizations that do code reviews are far more likely to meet development compliance milestones and mandates all the time or very often. [Tweet this] Furthermore, those that do not do code review are far more likely to almost never, or never meet compliance milestones and mandates.

When we combine the responses for the last two questions, there is compelling evidence to suggest that adopting code review can save both time and increase software quality.

Milestones and mandates by industry

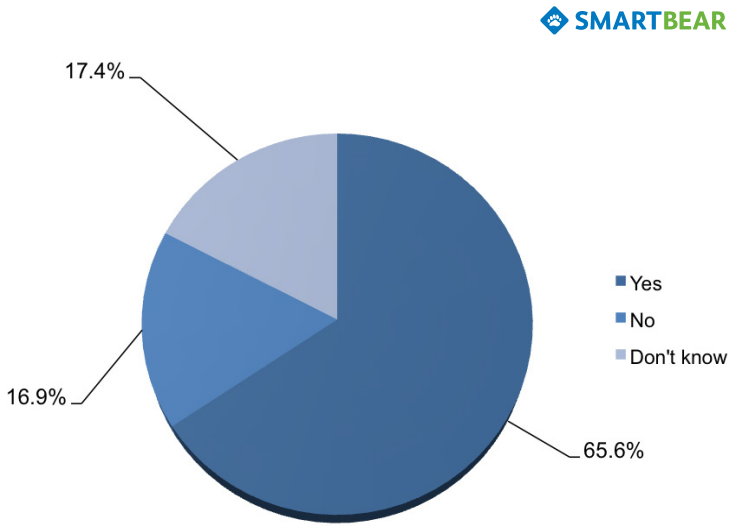


SmartBear Software 2013 Software Quality Benchmark Survey
Fielded: January 2013 N=683 Customer N= 115

Organizations in the game development, medical devices and aerospace / defense sectors are the top three industries to indicate that they meet compliance milestones and mandates either all the time or very often. These are the **same three industries** that are also the most likely to do code reviews.

Expense of Code Review

Do you believe that code review is less expensive than testing?



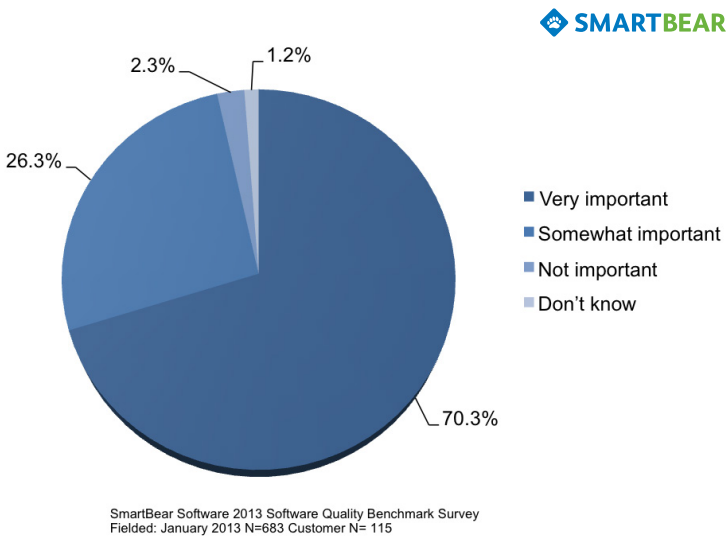
SmartBear Software 2013 Software Quality Benchmark Survey
Fielded: January 2013 N=683 Customer N= 115

By now, we understand the value code reviews bring to software quality and project milestones, but there's another important benefit we have not yet discussed… the bottom line.

When asked if code review was less expensive than testing, **66%** of organizations indicated that it was. Since code reviews improve development efficiency, and the quality of the end product, it has proven to be an essential component to reducing development costs thus increasing profitability.

Overall Importance of Code Review

Overall, how important do you feel conducting code reviews are?



The impact of code reviews is further demonstrated in the above chart. **70%** of organizations indicated that code reviews are **very important**. [Tweet this]

Overall, code reviews have proved to increase software quality, save time, and increase development efficiency. Those organizations that feel they don't have time to do code reviews should know that over 90% of organizations find code reviews at least somewhat effective in meeting compliance milestones and mandates. The result? Better software, on time, with happy customers.

Simply put, not only does code review matter, but it is rated by the development community as *highly important*.

APPENDIX

Our respondent pool

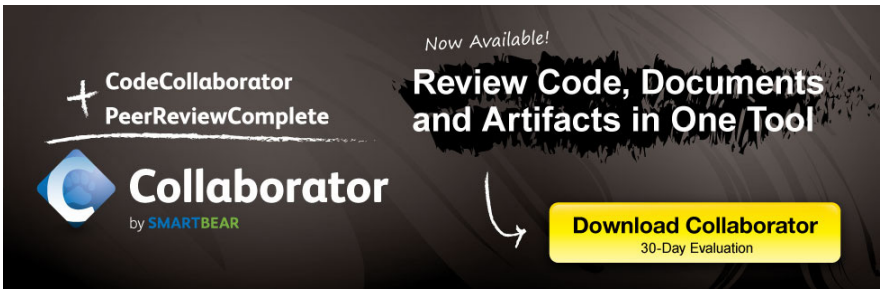
The State of Code Review Survey received 683 respondents, 70% of which completed the 40-question survey to 100% complete. Our respondent pool consisted of primarily Software / SaaS Developers (49%), Architects (14%) and Directors / Managers (13%). The remaining respondents were a mix of Project Managers, QA Engineers, Website Developers, Business Analysts, Product Managers and Hardware Developers.

We had a wide variety of industries participate. The consulting, Industrial / manufacturing, financial services and eCommerce / retail sectors were the most common. Another popular response included “Other” (31%), where respondents primarily indicated they were in “software development.”

A large portion of respondents included enterprise-level organizations with more than 1,000 employees (40.7%).

For any questions regarding this study or our response pool, contact us at research@smartbear.com.

If you like what you’ve read, then share this eBook.



About SmartBear Software

More than one million developers, testers and operations professionals use SmartBear tools to ensure the quality and performance of their APIs, desktop, mobile, Web and cloud-based applications. SmartBear products are easy to use and deploy, are affordable and available for trial at the website. Learn more about SmartBear, the company's award-winning tools or join the active user community at www.smartbear.com, on Facebook or follow us on Twitter @smartbear and Google+.

SMARTBEAR

SmartBear Software, Inc. 100 Cummings Center, Suite 234N Beverly, MA 01915
+1 978.236.7900 www.smartbear.com
 SB-C-041713-WEB