

LOGIN | REGISTER NOW | SUBSCRIBE


**InstallAware Studio**

 Looking to replace  
your WISE installer?

**ComponentSource**  
The Definitive Source of Software Components

**DOWNLOAD A  
TRIAL NOW**

As of July 15, 2010 12:46 PM

 

HOME &gt;&gt; LATEST NEWS

## Making the case for code review

By [David Rubinstein](#)

April 20, 2010 — The practice of code review is a lot like going to the gym: You know it's good for you, but you don't always do it.

A study by Forrester Research released at the end of March found that code review processes are often ad hoc in nature and have not kept pace with the increasing complexity of software development. Further, the study found that organizations do not make the time to institute code review as a formal part of the development process.

Gwyn Fisher, CTO at code analysis software company Klocwork, which sponsored the Forrester study, cited a number of factors working against formalized code review. "It's hard to motivate people to do something that's not pleasant," he said. "It's potentially devastating for the person whose code is being reviewed, and it's annoying for an architect to get pulled away from his tasks to participate in a review. Even getting people to buy into a process that may or may not find something is hard to do."

The benefits of code review are well understood; finding bugs earlier in the life cycle was the top reason given by 84% of the 159 IT professionals surveyed who are directly involved in their organizations' development and code review processes. Other benefits included the sharing of best practices and highlighting new techniques, as well as the encouragement of refactoring and code simplification.

But the survey respondents said the biggest challenge they face is finding enough time to prepare for a review. Part of the reason is that in development, it's hard to predict when code will be ready to review, the study found.

"What we see is that there will always be hesitancy to believe in the idea that if you invest time now, you can save it later," said Gregg Sporar of code review software provider Smart Bear Software, now a division of testing company AutomatedQA. "People say they won't get any real value, that code review is too cursory and you won't find the really interesting stuff. Our response to them is, 'Maybe you're not doing it right.'"

Forrester recommended that code review should be made a formal part of the development process. If the organization is doing agile development, code review should occur as part of the "done" criteria for a story, the report said.

In waterfall development, a build should not be considered finished until code is reviewed. "By making code reviews a milestone, it's possible to report on the status of the milestone. This helps ensure that code review, although a very technical activity, has the same level of visibility of other activities in the process," the report stated.

The report then urged development shops to put any issues found with the code into the backlog, so they are treated as any other bug—assigned as tasks to specific developers and monitored by management.

Code reviews and their results often don't match up to organizational structures, with about 70% of respondents saying that QA is not represented in the review process, and 55% saying that the software architect is also not involved.

Yet Smart Bear's Sporar said it's more important to have reviews done by the right people at the appropriate point of the project's development. "Which artifacts are you creating in addition to the code, and who is reviewing those? Are you doing unit tests? Who's reviewing those, the subject matter experts?"

Klocwork's Fisher sees the review becoming more use-case oriented when more people are brought in. "The number of stakeholders is so much greater outside the chain of command...so you get more 'Did you think of X?' 'Are we delivering on that?'"

Social-media tools make it easier to bring in these additional stakeholders, and they are helping to advance code review, even if it's just informally for now. "Code review traditionally had five guys around a table. Now, you put it out there and see what happens. Chances are someone will be watching," Fisher said.

Social tools, he added, "form engagements in a more democratized way, rather than specifically saying, 'You're invited to a review.' This brings code review to the people."

Fisher went on to say that once an organization finds success with code review, "it becomes self-sustaining. The typical tradeoff has been time, features and quality. Pick any two. Quality traditionally has been the last one considered."

Sporar agreed, saying, "The connectivity explosion and sharing of code has been a huge boon not only to code review, but any kind of collaborative activity. The highest bandwidth still is face-to-face, but schedule and timing conflicts come up, so maybe these [social network] tools can help us out.

"Folks who get onto the correct track, you'll never get them off of it."

**Related Search Term(s):** [code review](#)

**Share this link:** <http://www.sdtimes.com/link/34294>

### Comments

04/21/2010 11:15:44 AM EST

Second the point on making it a milestone. If it's not a part of an acceptance process, it will invariably get pushed to the side. Engineers have a lot on their plate and the pressure to deliver more in less time will force them to focus just on urgent issues.

Andrew Yang

United States