



Countering Gen Y Objections to Peer Reviews

A Layman & Layman White Paper



LAYMAN & LAYMAN

LAYMAN & LAYMAN

"The concept of counting the issues for an artifact under peer review is demoralizing and could ultimately be counterproductive. It essentially requires us to score one another's work. A review with zero issues found could mean either that the reviewers did a bad job or that the work was perfect. A review with many issues could mean the reviewers were too aggressive or that the work was lousy. This puts people at odds with one another. Meanwhile, the whole point of a peer review is to improve quality and foster better team communication. Also, keeping measurement data around would appear to be a mechanism for judging a developer's performance. What you actually want to encourage is good feedback. The numbers are meaningless, but serve as a frustrating factor that pits the producer against the reviewers and hurts team morale."

The paragraph above paraphrases a posted objection to a newly introduced peer review process in a software development house. How is it that, thirty years on from the widespread recognition that walkthroughs, inspections and reviews represent a highly effective and cost-efficient means of finding and removing defects at the earliest possible point, misconceptions about them remain so common? What can we glean from this developer's misconceptions about reviews and concerns about measurement? How should management and process improvement staff deal with this sort of pushback from developers?

First of all, we see from this developer's comments that it would be a mistake to assume that reviews are such a commonsense method that staff members intuitively understand how to go about conducting and participating in them. While they may have had a brief exposure to the concepts in college-level courses, these courses do not adequately prepare students for using the review class of methods in real world practice. Furthermore, it would be a mistake to assume they've learned the methods well during prior work experience. Although the methods may be widely recommended, they are less widely practiced, and even less widely practiced well.

Secondly, the developer expresses concern that reviewers might be "too aggressive." This suggests that he expects reviews to raise issues of a subjective nature. While it is not possible to eliminate subjectivity altogether from reviews, the purpose of these methods is not critique. The aim is to find non-conformance with objective criteria.

Our developer writes that the point of reviews is "to improve quality and foster better team communication," but he thinks of this as happening by means of "good feedback." He's worried that any recording of the outcome of reviews for measurement purposes would amount to scoring one another's work and be used to judge performance. All of his concerns highlight management challenges and failures we'll now consider.

The master key to success with reviews is management commitment. Lip service will not do. Watts Humphrey calls reviews "the most important step you can take to improve your software engineering performance." When management has a grasp of the enormous benefits of reviews, they become committed to the review process and communicate this through the importance they attach to it. Reviews provide data about the quality of work and the effectiveness of overall process, contributing both to project-level and bigger-picture business and process improvement goals. Management's attention to this information reinforces the value of reviews to staff and influences how they, in turn, think about the process. Unless management at all levels show real belief and interest in the review process, it will be a non-starter.

Training, coaching, and sound policies and procedures are essential if an organization is to institutionalize peer review methods and get good results from them. Think about college team sports as a parallel. Despite the considerable talent and experience level of the players, no one expects that a winning team can be fielded without training, coaching, and discipline.

Review methods cover a wide range of formality and focus, and are not limited to code reviews. Project plans, requirements, architecture, design, use cases, code, test plans, and other artifacts might be reviewed. Participants must understand the objective and scope of the review, roles and responsibilities, and the standards and conventions against which flaws and deficiencies can be determined. Participants must also understand that their role is limited to uncovering these issues – not to correcting them. Wherever possible, automated tools such as static analyzers should be used for detecting common, low-level flaws, allowing the human reviewers to concentrate on higher order issues. Templates should be provided to aid reviewers in assessing completeness, and checklists provided to help in spotting errors commonly found in the type of work product under review. These basic work aids are as necessary to reviews as checklists are to aircraft pilots, and any defined process that fails to include them is deficient.

Given the current trend toward agile methods, the adjective “lightweight” has become the buzzword *du jour*. For reviews, “lightweight” generally refers to buddy system or pass-around checks of a casual nature. These may be suitable when a developer merely wants feedback from peers, but unless the reviews 1) follow a defined process, 2) evaluate against objective criteria, and 3) generate measurement and follow-up data, they should be considered a superficial form of review. This is not to say that casual reviews aren’t sometimes appropriate and of benefit, but they are not suited for review of critical work products where the impact of undetected defects could be severe.

Our developer’s apparent phobia about measurement shows a lack of systems thinking, weak identification with the team, and a degree of distance from reality. Measurement gives visibility to the root causes of errors, patterns in where or how we make them, and a means of determining what the errors cost. This information, in turn, can help us to see (for example) where automated tools or further training might be beneficial. Simply put, measurement is essential to gauging our performance. Without it we have no way of assessing how well our processes and teams actually work, where they need improvement, and whether our efforts to improve them are effective.

We face some new and perplexing management challenges in an age when school children are given happy faces for homework just because it was handed in, or trophies just for being on the team. They played matches where score was not kept, and were forbidden the rough-and-tumble of traditional playground games like dodge ball and foursquare. Many of our young people have been conditioned to expect continual strokes and indiscriminate praise, and have little experience of demand, discipline, or reprimands. It is, perhaps, only to be expected that such people tend to focus on self rather than team. They think in terms of “my work” and “your work” rather than the “team’s work.” Reviews can help overcome this problem. Properly done, they can promote team formation and identity, and help to shift team members’ focus and sense of ownership beyond their individual handiwork to the unified whole.

Empirical studies have shown that removing defects with static methods reduces project cost and schedule delays by ferreting out quality problems earlier in the cycle when they are more easily corrected. Reviews also increase productivity by improving communication, accelerating learning, and cross-training. The benefits of reviews are so significant and so well documented that they need not be repeated here at length. They warrant a concentrated focus by management on putting sound review processes in place, and adequately training and coaching employees. Where management has struggled to introduce reviews effectively, it’s well worth considering outside mentoring to help in getting things right.