

Robert L. Glass

# Inspections—Some Surprising Findings

*“You can see a lot just by observing”*—attributed to Yogi Berra

Someone recently asked me to name the three best software engineering practices. After mulling over the question, I came up with the equivalent of a realtor’s answer: “inspections, inspections, and inspections.” (Realtors are said to say that the three most important criteria for choosing a place to live are “location, location, and location.”)

What I meant by that only slightly facetious answer is that inspections, by all accounts, do a better job of error-removal than any competing technology (that is, inspections tend to find more errors), and they do it at lower cost (the cost per error found is lower). There are plenty of studies that keep coming up with the same findings—fully 90% of software errors can be found by inspections before the first test case is run.

Now don’t take this to mean that I’m a hot-blooded inspection zealot. I know inspections, if done correctly, are hard work.

They require many people to perform them, and in these days of schedule-driven projects, just finding those people is a hard task. Inspections require prepara-

tion, and where does that time come from? They require rigorous thinking, the kind that exhausts participants after only an hour or two of participation. And given the typical productivity figure of 100 lines of code per hour of successful inspection, they are extremely costly, all

claims that they are cheaper than the alternatives notwithstanding. In other words, inspection is a very bad form of error removal—but all the others are much worse.

Because of all that “hard work” stuff in the previous paragraph, most companies don’t do many inspections, and some do none at all. At best, the state of

the practice is “we inspect our key components.” At worst, it’s “we don’t have the time to do inspections at all.” And that’s too bad, because the value of inspections is probably the topic on which computing research agrees more consistently than any other. Look at all those so-called “breakthroughs”—things like the structured methods, object-orientation, CASE tools, 4GLs, and more. In spite of the outrageous claims all too often made for them, there are very few evaluative research findings to support their value, and those findings that do exist tend to be equivocal and to show modest benefits at best. On the other hand, studies of the value of inspections are fairly common, and they tend to speak with the same voice—inspections are the most useful, most cost-effective form of error removal. What a peculiar dichotomy our field has—we laud with our hearts, not with our heads.

There are really two things I would like to accomplish in this column. The first is to raise, yet again, the notion of inspections as an important tool in the software practitioner’s technologies kit. But the second is perhaps more important, and certainly a more interesting and more

unique contribution. Let's discuss what research tells us about how to conduct inspections. Now, don't quit reading here. I know what you're probably thinking—that Fagan inspections are the way we do inspections, and there's nothing new to say about that two-decade-old topic! Well, readers, nothing could be further from the truth. Research data says something quite the contrary. Get ready for some surprises.

First of all, let me tell you about my information sources. I try to keep abreast of the computing literature, but my notion of that literature is somewhat different from the academic norm. First of all, academic periodicals do have some interesting material relevant to practice, but it's all too infrequent. Practitioner-relevant findings can be found in periodicals like *Communications*, Ed Yourdon's *American Programmer*, and my own *Software Practitioner*, and at conferences like NASA-Goddard's Software Engineering Workshop (SEW). That's where real "been there, done that" practitioners tell their stories about lessons they've learned, and practice-relevant research they have conducted.

### What Research Has Learned about Conducting Inspections

Let's confront the Fagan issue right off the bat. Are formal inspections, with assigned roles and pre-inspection training in inspection process, the most effective way to go? No, say several studies.

The most interesting is a study by Rifkin and Deimel that

presented a new way of preparing for inspections. Instead of training the participants in inspection process, they trained them in code reading comprehension techniques, preparing them in a product-focused rather than a process-focused manner. And the findings were spectacular. The authors had been concerned with the value of the Fagan approach in eliminating post-release errors—the kind customers find—and they discovered there was a 90% reduction (as opposed to Fagan approaches) in the incidence of such errors when inspection participants used the new approach. There was a similar, non-Fagan finding by Porter and Votta (presented at the SEW in 1994), who used what they called "scenario-based" inspections in which participants each looked for certain classes of errors, and found that the results as measured in errors found outperformed Fagan.

Clearly, based on these studies, there are newer and better approaches than the Fagan. But all of this leads to an even more important question: Are meetings the best way to perform inspections? That is, do the inspection participants—whether they use Fagan or Rifkin or Porter approaches—need to get together in a meeting at all?

Several research studies find the answer is either "no" or "probably not." The most recent study resulted from a survey of the literature and was presented by Bruce C. Hungerford at the 1997 Association for Information Systems conference. He found studies showing the use of

inspection meetings tends to slow project progress by an average of two weeks (because of coordination problems among inspection participants), and that meetings produced none of the expected synergy (wherein more errors are found because of meeting participant interactions). Hungerford also reported that inspection meetings tend to find no more errors than the most competent participant, although he was concerned with the accuracy of that statement.

That finding echoed (and may have been based partly on) a study by Votta presented at the 1991 SEW, which found little meeting synergy—an 8% improvement in the number of errors found in meetings, which the author considered tiny. And Porter and Johnson, reporting in the June 1997 *IEEE*, found meetings "neither more effective nor less effective ..." (than inspections performed by a collection of individuals). What can we conclude? That there are better inspection approaches than Fagan, and that individual inspections may well be better than (and are certainly no worse than) inspection meetings.

What does research tell us about these individual approaches? Multiple individual readers are best, according to several research studies. One inspector tends to find a small percentage of the total errors (Basili, reporting at the 1990 SEW, suggested 26%, while Kelly, at the same meeting, said 33%). As a result, Basili suggests at least two independent inspectors be used, and Kelly suggests three or more. In a somewhat

contradictory finding, Porter et al. found (IEEE TSE, June 1997) that four participants were no more effective than two (the type of inspection analyzed here was slightly different, in that inspectors reviewed the material individually but then gathered in a meeting to discuss their findings, a technique they found to be 30% better than not using meetings).

HERE, THEN, IS THE BOTTOM line on inspections, as discovered by my reading (and interpretation) of the aforementioned research literature: They are extremely effective. There are better approaches than the commonly used Fagan method. Inspection meetings are of dubious value. The number of participants in a review process should probably be two or three. Meetings, if used, should be to report on the findings of the individual inspectors. Since all of this is very different from the state of the inspection practice, and even from the state of the inspection art as described in the advocacy research literature, I think there are some important lessons to be learned here. I hope you agree. **Q**

---

**ROBERT L. GLASS** (rglass@indiana.edu) is the publisher of *The Software Practitioner* newsletter and editor of Elsevier's *Journal of Systems and Software*.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.