

Issues in Software Inspection

A forum for exchanging ideas, philosophy, and experience.

In manufacturing, agriculture, and transportation, inspections are a routine part of business, making our lives safer and more comfortable. Yet in software, we generally leave inspections to companies that are quite advanced in their process maturity. Why do most companies skip inspections when there is so much to gain? In 1991, Don O'Neill began the National Software Quality Experiment, an ongoing database of software defects and inspections practice. In his study, participants from dozens of organizations describe defects and enter them into the experiment database. Here, O'Neill makes a case for implementing inspections at all levels of software development practice, based on what he has learned thus far.—Tomoo Matsubara

SOFTWARE INSPECTIONS ARE GENERALLY viewed as promising much but delivering little. Although recognized as a technical best practice, inspection adoption has been slow: the Software Engineering Institute reported that only 22 percent of the organizations it assessed practiced software inspections.

Some factors inhibiting more rapid adoption include the following:

- ◆ Many managers see only the added cost of inspections, not the benefits of greatly reduced defect leakage.
- ◆ Practitioners already faced with tight schedules are reluctant to take on additional work. They are also concerned that publicly detecting defects in their work may affect their performance evaluation.
- ◆ Because SEI assigns peer reviews to level 3 of the CMM, most organizations postpone the adoption of software inspections.

Technical best practices are often viewed as too advanced for low-maturity organizations. But the enterprise that becomes globally competitive is one that operates on a longer planning horizon, sets a competitive improvement agenda, and selects the best industry practices to achieve its goal. This enterprise will roll out the software inspections process because it delivers value.

A CLOSER LOOK. The purpose of software inspection is to detect and correct defects before they leak

through subsequent development phases and into the field. To accomplish this, inspectors closely and carefully examine specifications, design, code, test, and other artifacts and compare them with

Implementing inspections can reduce cycle time and lower your costs.

leading indicators of excellence for each artifact type. Among the indicators are the following:

- ◆ completeness, based on traceability of the requirements to the code, which is essential for maintainability;
- ◆ correctness, based on the clear specification of intended function and its faithful elaboration in code, which is essential for reliability and availability; and
- ◆ style, based on consistency of recording, which is essential for maintainability.

These and other factors constitute the exit criteria for activities within the software life cycle. By detecting defects early and preventing their leakage into subsequent activities, you eliminate rework and its associated costs. Thus, in addition to improving maintainability, reliability and availability, implementing software inspections can reduce cycle time and lower your costs.

WHO'S LOOKING? Software inspectors are practitioners who play defined roles: manager, moderator, producer, recorder, reviewer, and reader. Multiple roles may be played by an individual, and each role carries with it specific behaviors, skills, and knowledge.

- ◆ The manager plans inspections, assigns resources, and reviews the results without attending inspection sessions.
- ◆ The moderator directs software inspection activities and facilitates interaction among inspec-

Editor:

Tomoo Matsubara
Matsubara Consulting
1-9-6 Fujimigaoka,
Ninomiya Naka-gun,
Kanagawa 259-01
Japan
matsu@sran125.sra.co.jp

tion team members. Moderators intervene as little as possible and as much as necessary to ensure inspections are effective and efficient.

- ♦ The producer creates the materials to be inspected and addresses issues raised during the inspection.

- ♦ The recorder describes each issue the producer raises, noting defect category, defect severity, defect type, and defect origin.

- ♦ The reviewer raises issues and concerns about the software product without proposing solutions.

- ♦ The reader reads parts of the product aloud to focus attention on a particular trouble spot.

WHAT'S IT WORTH? When a code defect leaks into testing, you may have to execute multiple tests to confirm the error and gather information about it. This is time consuming and takes you away from the task at hand. In addition, a defect that leaks to the next phase will likely cost at least 10 times more to detect and correct than if it is found and fixed in the phase of origin. In fact, IBM Rochester reported that defects leaking from code to test cost nine times more to detect and correct, and defects leaking from test to the field cost 13 times more (Richard Lindler, "Software Development at a Baldrige Winner, IBM Rochester," *Proceedings of the National Institute for Software Quality and Productivity Conference*, Nov. 1991).

To determine the return on investment for software inspections, you divide your net savings by your detection cost. In 1991, I initiated The National Software Quality Experiment, a study gathering data on defects and inspection practice in dozens of companies organized by software process maturity level, organization type, product type, programming language, and global region. Thus far, the experiment has shown that companies who use inspections have had a return on investment ranging from four to eight dollars saved for every dollar spent (Don O'Neill, "National Software Quality Experiment: Results 1992-1995," *Proceedings of the 8th Annual Software Technology Conference*, Software Technology Center, 1996).

HOW GOOD IS IT? Although software inspections originated and evolved in new development, their usefulness in maintenance is now well established. For example, in NSQE, the lines of code inspected per hour ranged from 250 to 500 for new development and from 1,000 to 1,500 for maintenance.

Once you adopt software inspections and complete training, your organization can expect to detect 50 percent of the defects present. To achieve expert practice (a detection rate of 60 to 90 percent) can take from 12 to 18 months. After 10 years of use, IBM reported 83 percent and AT&T reported 92 percent for defect detection resulting from software inspections practice (NSQE).

WHERE DO WE START? Each inspection participant must be trained in the structured review process, the roles of each practitioner, process and product checklists, and forms and reports. The cost to acquire the knowledge, skills, and behaviors is about 12 hours per practitioner (Don O'Neill, *Software Engineering Course and Lab*, SEI, 1989). Managers are also trained in how to initiate inspections and interpret the data they produce. This takes about four hours.

You should also plan for the costs of preparing and conducting inspections and managing and using the data they produce. A typical inspection includes five practitioners who spend one to two hours each preparing for the inspection and one to two hours each conducting it. My study has shown that for each session, 10 to 20 hours of total effort are expended, resulting in the detection of 5 to 10 defects. To manage and use the data collected, you establish a database of measurement results from which you can generate reports and graphs. The cost to establish a functioning database is on average two person-months. The cost for entering data is included in the time practitioners spend conducting the inspection.

WHAT'S THE HOLDUP? Defects are a fault line in the software crisis. Forty-two percent of all defects result from lack of traceability from the code to the design

to the requirements or business case. A critical legacy system that lacks traceability places its enterprise on an iceberg floating south. Where traceability is lacking, code rules.

Software inspections clearly provide value in the form of shortened schedules, lower production costs, smoother operation, and improved user satisfaction. Once in use, software inspections contribute to higher employee morale as practitioners sharpen their standard of excellence.

COMPETING FOR THE FUTURE. Software inspections deliver measured results that can be an advantage in meeting goals of global competitiveness. This advantage is rooted in the learning and discovery you need to raise your organization's core competence by setting the standards of excellence, measuring the shortfall of current practice, pinpointing areas of neglect, and selecting the most promising changes.

By gaining control of its software resources, your organization can field trustworthy software and thus distinguish

A defect that
leaks to the next
phase will likely
cost at least
10 times more to
detect and correct.

itself from the competition. Just as people are known by the company they keep, a company will be known by its software and the problems it causes. ♦

Don O'Neill is an independent consultant who helps organizations manage strategic software improvement. He previously worked for IBM's Federal Systems Division and served a three-year residency at SEI. O'Neill served on the executive board of the IEEE Software Engineering Technical Committee and as an IEEE Distinguished Visitor. He can be reached at O'NeillDon@aol.com.