

High Quality, Low Cost Software Inspections

Louis A. Poulin
GRafP Technologies

What do inspections, peer reviews, walk-throughs, and structured reviews have in common? These are all terms that are used interchangeably in software engineering. Yet, the activities that they entail are rarely carried out consistently in the course of developing an application. This article reviews this theme as Ronald A. Radice presents it in his new book.

I would bet that all articles submitted to CrossTalk, including this one, for which I can vouch, have been peer-reviewed. Why is it that peer reviews in the publishing industry are widely accepted, while they are the first items to be dropped off the priority list in the software industry?

According to Ronald A. Radice in his book "High Quality Low Cost Software Inspections" [1], there are several reasons why inspections are not more widely used in software development. First is the belief that inspections can only be done one way, a myth this book has all but obliterated.

Second is that inspections are not easy to do well, given the psychology that permeates them. Radice addresses this topic by discussing participant personalities such as aggressive inspectors, intimidating moderators, weak moderators, and defensive producers – whose products are being reviewed – and offers suggestions on how to deal with these situations.

Third, the perception that inspections represent an added cost to software development is still widely prevalent. The book certainly helps in countering this argument, with plenty of charts and data that demonstrate the added value of inspections. However, Radice, may very well have identified the fundamental underlying cause software inspections are not more widely used: inspections are low tech and are not the most enjoyable engineering tasks, especially when compared to design and coding.

But software inspections do work, and Radice's book contains 400 pages that not only demonstrate their value but also offer various approaches, techniques, and guidelines to conduct them. "High Quality Low Cost Software Inspections" is a must for anyone wishing to start inspections in their organization or to those who have performed inspections for some time and want to get better results. Radice describes the inspection process in detail, including the roles assumed by inspection participants and the type of data that should be collected, all the way to causal analysis of defects detected through such reviews.

Inspections also contribute to the culture change experienced by software companies that appreciate the value of data and allow the data to be used safely, in a nonthreatening way by the people who provide the data. However, this is easier said than done and does not happen overnight. The book includes a chapter on managing inspections and another on practical issues you can expect to deal with

"... inspections are low tech and are not the most enjoyable engineering tasks ..."

when introducing inspections. These chapters will prove helpful in preventing lukewarm reception by those who have been identified as participants, or downright failures.

The chapter on economics of inspections is particularly eloquent for anyone who needs to be convinced of their value. It references Infosys, where two teams were set up to assess inspections and unit testing. Inspections found 2.7 times more defects than did unit testing. According to Radice, another feature that differentiates inspections from unit testing is that when defects are found in inspections, the fix is often understood as soon as the defect is identified. Testing is characterized by a more serial approach: After a defect symptom has been observed, its cause must then be sought out and a fix devised.

Radice also takes a jab at the Software Engineering Institute's Capability Maturity Model® (CMM®) IntegrationSM (CMMISM) for diluting the value of inspections. Whereas peer reviews were deemed important enough to deserve a whole process area in the CMM for Software, they have now been reduced to a goal within the Verification Process Area in the CMMI. Implementation of inspections with the CMMI is now more a matter of choice than a requirement. Potentially,

organizations that do not see a need to perform inspections will now have a bigger hole to squeak through to prove their point that inspections are not required. We can only hope that it will not be the case.

Currently, software development has been hit hard in the technology sectors, which are early contributors to the current economic downturn. Inspections may be low tech, but they represent a sound investment to guarantee that products released by software companies operate as advertised. ♦

Reference

1. Radice, Ronald A. High Quality Low Cost Software Inspections. Andover, Mass.: Paradoxicon Publishing, Jan. 2002.

About the Author



Louis A. Poulin is president of GRafP Technologies. He has been involved in assessing the capability of information technology organizations and in developing hazard evaluation, hazard monitoring, and hazard prevention tools and methodologies applicable to various fields. Prior to this, Poulin served in the Canadian Navy as a combat systems engineering officer. He is a member of the Institute of Electrical and Electronics Engineers and a fellow of the Engineering Institute of Canada. Poulin has a bachelor's degree in engineering physics, a certificate in naval engineering, and a master's degree in electrical engineering.

550 Sherbrooke St. West
Suite 777
Montreal, Quebec
Canada H3A 1B9
Phone: (514) 847-0900
E-mail: lpoulin@grafp.com