The Darker Side of Inspection

Juha Iisakka and Ilkka Tervonen

of

University of Oulu

Department of Information Processing Science

The purpose of inspections is to intensify the quality of software development. In simple terms, the goal is to find errors in any sort of software development document. Although unlike testing, inspections are performed by groups of human beings – not computers – they are usually regarded as technical manoeuvres and the human or sociological aspect is forgotten.

When we began to consult with software development companies nearly ten years ago, with the aim (then and now) of improving software inspections, we soon noticed that inspections have some negative aspects. Some of these are inherent in the inspections themselves and some not, but all inspection instructors, especially software development managers, should work to mitigate these difficulties.

The goal of this paper is to report on some of the negative experiences and images related to inspections that we have observed among software engineers. These are based on our own experiences, since we have met numerous people who have felt this way, although none of them had by any means been conscious of all the negative aspects. Even though the subject of this paper might give that impression, we do not claim that inspections really possess all these faults. Many of them are a natural part of social interaction.

Frustration

Inspections involve a couple of problems related to frustration. These problems do not in fact arise out of the inspections themselves, but rather the basic reason for the frustration is that the organisation is unmotivated to perform inspections, even to the extent that an initially motivated organisation may eventually lose its motivation to do so. This is to a large degree why we have found that these problems as presented below are so common.

An inspection typically includes many people – inspectors and an author etc., and gathering them all together may be difficult, as people have different timetables. Consequently, inspection deadlines are relatively fixed, although really they should not be. Unfortunately, it is common for software development to be delayed, hence author of the document to be inspected may have difficulties in distributing it in time. The inspectors may have booked times for the checking work, but as they have their normal duties to perform, an overdue inspection may have to be done in overtime, which can become frustrating if it happens frequently.

Sometimes an inspection has to be delayed because of people's timetables. At the same time, software projects are typically obsessed with the time to market, so that a program may be in use before the inspection is completed. Naturally inspectors are unmotivated when they know that their findings will remain unheeded until the next version of the product appears, especially since the next version is always just a possibility, not a certainty, and the inspectors' findings may relate to a part that is not in the next version. Perhaps more importantly, projects with a prompt schedule have a bad habit of combining many document reviews into one inspection, despite the fact that the guidelines for inspections strictly forbid this merging, since no one is able to check excessive amounts material properly at one time. It is frustrating to try to find errors in huge piles of separate documents. In this case we propose that no inspections should be held at all, because they are a total waste of time under such conditions.

Virtual inspections are a modern way of performing the same function (see Harjumaa and Tervonen 2000), but these have their own way of frustrating inspectors, as they often imply a one-way process. Inspectors who send their comments to the author, for instance, may well not receive any feedback, even though it is well-known that humans do better work when feel they are useful. Even when the inspectors know that the authors will make good use of their comments, their motivation to check documents properly will diminish if they never see the fruits of their labours. Luckily this is easily taken care of. The author can reward the inspectors by sending them a list of the corrections made.

Just as inspectors become frustrated, so authors can also lose faith in the inspection process. Inspectors are far too often unprepared when they come to an inspection meeting. As Gilb and Graham (1993) state, unprepared inspectors should be sent away, although we have never heard of an organisation taking such action. A company with a practice that the most unprepared inspector is appointed scribe comes closest to this principle. An unprepared inspector can make only minor comments on matters such as punctuation rules, in which case the author understandably sees no use for inspection.

Attraction

Let us picture a scene from a recent action film. It is morning and there are two people present. One (a good-looking male, the hero) is sitting at a computer and discussing something with the other (an attractive female, the heroine). He tells her that last night he wrote a program to hack into the Pentagon computer systems. They desperately need some extremely tightly classified information in order to "save the universe". The scene is red-hot, for he has proved himself a real man with his code. But what if he tells her that he has coded the program and asks her to inspect the code? It is important for them that there should be no errors, because otherwise they will be caught. This does not sound manly at all, for in films the "real men", such as John Wayne, never ask for help. The sequence is believable in the eyes of an audience only if he is not the hero nor good-looking.

Inspections are not attractive for the author of the document to be inspected. He/she is confessing dependence upon other people's help and an inability to manage the task alone. Moreover, the author may think that the work is good and even be proud of it. The document is no longer the author's sole own once others have been able to influence its contents.

It is not attractive to call on some one to inspect a document, nor is being an inspector a very attractive duty. An inspector has to read carefully through a document that some one else has made, and the document is rarely interesting or entertainingly written. In addition, programming styles are highly individual, despite companies' efforts to standardise their coding guidelines. This means that the inspector typically has to show empathy with the author's way of programming.

Although one can make inspections more fascinating through the inspection tools, it is still relative common for inspectors to have to print out a document on paper in order to read it. If they find something, they must write it down on paper or in an e-mail. It is much more fun to read a document with a slick computer application and mark findings by mouse on the screen. The author, moreover, has more fun sending his/her document for inspection via a CSCW (computer supported collaborative work) inspection tool rather than just sending it by e-mail or, even worse, as a photocopy. Unfortunately, these tools lose their attraction very quickly and needs arise for newer ones (just as a sports car is no longer sexy if other people have newer ones).

On the other hand, an inspection meeting can present a very attractive situation for the inspector, The inspection meeting is a social occasion for showing off one's competence in the front of colleagues (i.e. the participants in the meeting: the author, team leader, scribe and other inspectors). Pointing out errors in the document proves superiority, as the implication is that the author has poor skills, as more capable persons (inspector that feels sexy) have to repair his/her work. Besides, and even worse, an inspection meeting can be a power game. In many cases there is more than one way to work out a problem, and inspectors can force the author to change a solution that works for another equivalent one.

Humiliation

A quality manager once had to deal with a very thin-skinned employee who had been in the company for many years and was responsible for software that no one else understood. In fact, he had always taken care of that software. According to that quality manager he was in practice indispensable to the company. Although his software was extremely important for the company, no one else understood it because he had his own unique way of coding. This sensitive person regarded his software as his own, so that finding fault with it meant finding fault with him. This meant that one had to be very careful when telling him about errors in it. The words "error" or "fault" were banned, since they were an insult to him. He had to be told in private that there were reports of problems with some part of the software, and then one had to just hope that he would find the error. Whether he found it or not, he never told anyone; the error just disappeared.

This case is not all that extraordinary. Programmers who are regarded as "gurus" are very sensitive about their "guru" status. It can be very difficult to inspect such a person's work, as he/she is doubtful about other people's skills or regards inspection as a joke or attempt at public humiliation. Gurus feel they are brilliant experts and look on an inspection meeting as a dangerous occasion, as they find it humiliating when other people (less brilliant and expert) reveal their mistakes.

It is not just gurus who can have feelings of shame as a result of inspections. It has been known for an author to burst into tears when an inspector mentioned that they

had found an error that might cause the system to crash. People handle criticism differently, some being able to tolerate much more frank comments than others. Newcomers and inexperienced employees in particular are often sensitive to criticism, as they are insecure about their place in the company. Also, in today's international workplaces there are often cultural differences that can create conflicts through an inspection, especially involving cultures that recognise the concept of "losing face". In situations where there is a danger of losing face it may perhaps be better if the author is not present at the inspection meeting. Inspectors can then send the minutes with very carefully phrased comments considering "topics" (not "errors") that were found and discussed. Regardless of the cultures involved, inspectors must behave in a businesslike manner, and above all be polite. Authors do not consider jokes about their best efforts to be funny.

Rivalry / Antagonism

In the current boom in the information industry even moderately experienced specialists have not found it difficult to secure a new job after having to leave an existing one. With a sort of recession possibly approaching, however, continued employment will not be as certain as it was a year ago.

An inspection is an occasion arranged in order to improve the quality of a software product. Unfortunately, many insecure employees consider this a way for a manager to separate the wheat from the chaff. Consequently, an inspection record includes two critical sets of data: the author of the inspected document and the errors found in the inspection meeting. If inspections are regarded as occasions for rating subordinates, employees will put too much effort into making their documents perfect before the inspection. While striving for perfection is admirable, it becomes uneconomic to continue trying to correct a document beyond a certain point. It is more efficient to let somebody else find any remaining errors.

Naturally, a manager is being unfair if he rates employees over errors found at inspections. It is not always certain who is primarily at fault in the case of a bug. It is probable, of course, that the author may be "guilty", but it is nonetheless plausible that the customer may have changed his/her mind. Correspondingly, the primary error may exist in the requirements laid down for the document. In any case, the author may only have been doing maintenance work on an existing document, or else the inspectors may have misunderstood a certain point in the material. In addition, it is easy to find errors in a clear, well organised document, so that easy detection of defects may be seen as a virtue in an author's work and not a fault.

Likewise, employees may in theory expend too much effort on finding bugs when they are inspectors in order to show their antagonist's incompetence. We havenever met this situation in practice, however, or at least no one has ever confessed to it.

Involuntary Good Samaritan

The staff of software companies mostly negotiate their pay individually with their employer, their salaries normally being dependent on their working performance. Naturally, key duties or important tasks that have been done well also pay well. Many employees have their own duties, so that someone is responsible for maintenance of a

particular application, for instance, and someone else takes care of user interface development. Many companies have personal merit bonuses that are tied to each employee's performance of his/her primarily duties. If employees only perform their main duties, they will do a "better" job, since they will have more time to accomplish their primary goals.

The effort required to inspect other peoples' documents nevertheless reduces the time that the employees have for their main duties. A manager does not necessary see inspection efforts, although a good one should be able to value inspections and recognise hardworking inspectors. Inspecting is a mutual procedure, of course, so that if one inspects other people's documents, they will be inspecting one's own documents in turn. It is hard to refuse when asked to be an inspector. The problem is that inspecting duties tend to accumulate with the same persons. There are always certain colleagues whose comments are more valuable than others', and skilful employees have to inspect more often than inexperienced or less competent employees. This can have an effect of evening out the differences between the more skilful employees and the less skilful ones, since the latter will have more time to perform their main duties.

A person whose duties are not critical to the company's success will be in trouble when the employer runs into financial problems. Inspecting and all other quality-related tasks can become a secondary activity in a software company, for in principle, the only important task in the area of software development is coding. Thus it is not necessary wise to focus solely on inspection duties, although we have met people whose only task has been to perform inspections.

Discipline

Traditional Fagan-style inspection is a highly disciplined process, and many people consider it too rigid. Inspection meetings in particular are not fun; rather they are boring and tedious events – close to two hours just booking issues not over one minute per each. Inspections are never pictured as fun. They may be important, compulsory, efficient or useful, but no one has ever said to us that they enjoy inspections.

New types of inspection such as virtual inspection can reduce the degree of tedium, but inspections need order, for otherwise people just make small-talk. All meetings have a habit of dissolving into disorder without discipline, and an inspection is much more efficient when everyone knows what to do.

What is the solution?

We admit that inspections are often tedious to perform, but efficient inspections are very useful and they can improve quality much more than testing or other methods, and therefore there is no reason to abandon them. Authors should feel that their companions are endeavouring to make their work better. Moreover, they have a shared responsibility, for if there are difficulties with an unfound error, no one is solely responsible any longer. Of course, only an incompetent management will find a particular employee "guilty" of an undisclosed error

The management of the company is the key to the success of inspections. If the side-effects listed above are very common in an organisation, there is something wrong with the management. In our experience, inspections are not efficient or useful without the full support of the management. Correspondingly, inspections do not improve quality if the employees know that the management is using them as meeans of monitoring performance

Highly individualistic people are not necessary well suited for inspections, whereas co-operative people with good teamwork skills are typically effective in such a context. Employees who feel that they are working for the collective good (i.e. for the good of their company) and not for themselves tend to improve inspections, for they are the first to realise that they are not inspecting the work of others but that it is in fact their own shared work.

Acknowledgements

We thank Mr Mark Lawford for his valuable comments to this work.

References

Gilb T.and Graham D. "Software Inspection", Wokingham, Addison-Wesley, 1993.

Harjumaa L. and Tervonen I. "Virtual Software Inspections over the Internet", Third Workshop on Software Engineering over the Internet, ICSE 2000, pp. 30-40, 2000.

Mr. Juha Iisakka is an associate professor at the University of Oulu, Finland

Mr. Ilkka Tervonen is a professor at the University of Oulu, Finland.

Both have acted as inspection consultants for many companies.