

Inspections in Small Projects

Juha Iisakka

University of Oulu
Department of Information Science
PL 3000
FIN 90014 OULU
Finland
juha.iisakka@oulu.fi

Abstract: Practically all inspection and review methods focus on projects for developing new software that involve numerous people, but software companies also have small projects on which only a few people are working, e.g. maintenance-oriented changes to existing software systems. Unfortunately, these small projects do not necessarily have the power to implement inspections efficiently. This paper raises certain problems that small projects have with inspections and discusses how different forms of inspection are suitable for small projects.

1 Introduction

This work is partly based on interviews with software companies regarding their inspection practices. My experiences with companies suggest that there are numerous practices connected with the performing of an inspection or review. The official inspection process as described in a quality book is typically copied directly from the literature, but the way in which it is actually performed may differ greatly from the official pattern. Traditions are usually important; and people endeavour to inspect in the manner in which they have always inspected. Orlikowski [Or96] uses the term organisational transformation over time to illustrate ongoing processes of improvisation enacted by organisational actors trying (tacitly or not so tacitly) to make sense of the world and act coherently in it. Organisational transformation over time also concerns inspections, of course, since the inspection process should correspond to the software development process. Nevertheless, people have not necessarily consciously tailored their inspection process, but rather it has ended its tradition in a somewhat haphazard manner. Unfortunately, not every transformation has been a wise one, and some companies may have poor inspection practices. Some have been fortunate, however, in tailoring the inspections proficiently to their needs and resources.

Small software projects are typically carried out by only a few persons, whereas a classical inspection assumes a logging meeting at which there will be a trained moderator, a scribe, an author and at least two inspectors present, i.e. at least five persons. (see [FW82]6)). What if a project cannot gather so many people together?

When I performed my interviews with software companies I acted as a consultant improving companies' quality processes. The purpose of interviews was to gather information of conditions. All the companies had formal inspection process definitions in their quality handbooks, but practices had diverged greatly from these. I endeavoured to find out why this was so.

The next section will focus on the inspection problems of small projects. The third section will suggest to give solutions to these problems and present some experiences of inspection experiments.

2 Small projects and inspection

When considering the problem of how to inspect software, Parnas [Pa94] noted that if one tries to inspect all but the smallest programs, the complexity would be overwhelming. In the commonly used forms of programming, interactions between code sections are unrestricted, and most often un-documented. There are many cases that must be considered, and it is hard to be confident that one has not overlooked something important. According to Parnas, experience has shown that people cannot easily understand long programs. When asked to study such programs, inspectors tend to focus on small details, while making inaccurate descriptions of the overall structure.

Small software projects in particular encounter difficulties with inspection. Inspection was originally designed for checking new software artefacts, and as small projects are very typically maintenance-oriented, the enormous amount of existing material involved is a common problem in inspecting them. In these cases the maintainers add something to the existing code. Normally one person has written or changed dozens of lines of code, but this new code is embedded into the existing software in a highly unmodular way (perhaps only a line has been added in one module). The inspectors have to check whether the new code has any influence on the existing code, which is very laborious.

As I have already stated there is at least five persons in any classical Fagan-style [see. Fa76] inspection meeting. Small projects cannot necessarily gather so many people together. Furthermore, small projects tend to be highly iterative, both because synchronising the work requires less effort and because the management structure is shallow, allowing quick feedback [RM00]. Traditional inspection is slow and does not tolerate well artefacts that are not finished.

A small project in a larger company can make use of inspectors from outside the project, and this is indeed advisable, since the project can make use of special expertise in that way. But what about small companies? Experience with both CMM and ISO 9001 has shown that software organisations with less than 15 developers have some difficulties in applying these models [DDDK98][Gr97]. Small companies are often subcontractors for larger ones, and in that case they typically can use their customers' staff in inspections. Conversely, software product companies are on their own and must organise inspections among their own staff.

Naturally there is no generally accepted definition of a small project. In general terms, it is smaller than an average project, just as a large project is larger than an average project. To extend this simple view, according to Russ & McGregor [RM00] it is possible to classify projects in terms of four environmental factors: the size of the development organisation, the complexity of the project, quality attributes and personnel interactions.

The size of the development organisation has a significant role when judging the resources available to small projects. If the company itself is large enough, a small project can receive more resources from outside than if it is taking place in a small company. Meanwhile a small company may not necessarily have enough people capable of taking part in an inspection. Moreover, large companies typically have separate quality organisations to support inspections.

An other reason why a small project may need outsiders to inspect it is complexity, which is based on the sophistication of the domain knowledge required for the project. This varies among projects, and some require very special expertise. In the case of a small but complex project, it can be hard to find inspectors with the necessary expertise to work alongside the author.

Small projects typically produce less material for inspectors, and consequently they can achieve quality attributes such as reliability, security and performance more easily than larger projects. Artefacts in small non-maintenance oriented projects are usually more easily inspectable.

Interaction between project members is an advantage that small projects possess, since communication is efficient and informal when the number of members is small. It has been shown that increasing the number of staff will lead to a proportionally greater increase in the number of interactions needed for communication. Pair programming (see [Be00]) is the epitome of efficient communication.

As stated above, inspection encounters certain problems in small projects:

1. Small projects have too little staff for conventional inspections.
2. The staff do not necessarily have enough expertise for all inspections.
3. A small project typically has a very informal organisational structure, whereas adequate inspections need a certain level of formality, e.g. the existence of pre-planning, inspection invitations and a follow-up.

3 Some experiences with inspection

The inspection types are linked to the problems encountered in small project inspections. If a small project is carried out by a large organisation, it can easily borrow inspectors from outside the project. In that case, its limited staff will not be a barrier to conventional inspection. A no logging meeting style of inspection can be difficult, however, if the inspectors do not know the circumstances under which the project is taking place. Conversely, pair inspection is perfect for a small project, as it requires only two persons.

When a small project lacks expertise it has to hire experts from outside. The best inspection types in such cases are walkthrough, if the external experts are not technically competent. By comparison, technically competent inspectors can send their comments autonomously to projects members, that participate in the inspection meeting, check the issues and decide what to do.

We have experimented in our university with small projects and inspections focused on finding ways to improve companies' quality processes. This has meant that students have participated in pilot projects in which they have tried out inspection practices that might be useful for companies' purposes.

The fact that these trials were performed in companies and not in a university laboratory naturally both improved and detracted from their value. It is difficult to decide what aspects of the resulting experiences are based on the types of inspection –employed rather than on the companies' circumstances, but the motivation for carrying out such a project is probably better in a company, as one is doing a "real job." The students Pekka Vuorio and Susanna Oravainen, who were just finishing their studies, were regular employees doing exactly the same software development work as before the projects. The two companies (who wish to remain unnamed) involved in these trials both had small projects, and one of them also had a larger one with pair inspections.

A small project seems to be very sensitive to personality changes. Students felt that a newcomer slowed down a small project much more than an ordinary one, which is natural, since personal relationships are emphasised when there are less people working together. A newcomer always needs time to blend in with the team.

A small project will typically have only a few documents, and this can be a problem, as it may be hard to remember afterwards what was agreed. Consequently the students found it useful to record items (errors and improvements) for pair inspection meetings. A suitable tool was needed, however, to ensure that people recorded items properly, As it is tedious to be an inspector when one needs to write everything down by hand.

A pair inspection needs checklists and software development rules in the same way as a larger inspection. Otherwise it can easily become merely a chat about the artefact under inspection. When pairs performed inspections alone (without the authors) they were tempted to correct small errors straight away, as this can be easier than recording them and reporting them to an author later. On the other hand, when an inspector worked together with an author, with the situation was comparable to a pair programming of extreme programming. Surprisingly, pair inspections were usually regarded as efficient when the participants were well trained, in the same way as conventional inspections, in spite of the fact that there were no strict procedures to follow.

4 Conclusion

One well-recognised and efficient way to improve quality in software development is inspection, but conventional inspection can be poorly suited to small projects. This study has endeavoured to show that inspection can be tailored in the organising sense and software companies can define their own type of inspection. It is essential to plan inspection actions beforehand for fear of haphazardness.

The inspection process is also very sensitive to organisational changes, and apparently calls for a stable organisation. Some companies have a notorious habit of perpetually re-organising themselves that makes it almost impossible to implement inspections. Similarly, I have seen many cases in which new managers have ruined all inspection activities. They feel they need evidence of productivity; and there is more time to spend on coding when there are no quality activities. There is a term, "PowerPoint commitment", for the attitude of managers who are spuriously interested in quality activities [Ab00]. As [Lu04] shows the motivation to perform inspections is the best way to have efficient inspections.

References

- [Ab00] Abrahamsson, P., Is management commitment a necessity after all in software process improvement? Proceedings of the 26th EUROMICRO conference, Maastricht, Holland (2000).
- [Be00] Beck K., Extreme programming explained, Boston: Addison-Wesley, 2000.
- [DDDK98] Demirörs E., Demirörs O., Dikenelli O. & Keskin B., Process improvement towards ISO 9001 certification in a small software organisation, Proceedings of the 20th ICSE, Kyoto, Japan (1998).
- [Fa76] Fagan M. E., Design and code inspections to reduce errors in program development, IBM System Journal Vol 15 No 3, (1976) 182–211.
- [FW82] Freedman D. P. & Weinberg G. M., Handbook of walkthroughs, inspections and technical reviews, Boston: Little, Brown and company (1982).
- [GG93] Gilb T., & Graham D., Software Inspection, Wokingham: Addison-Wesley (1993).
- [Gr97] Grünbacher P., A software assessment process for small software enterprises, In New Frontiers of Information Technology, Proceedings of the 23rd EUROMICRO Conference, Budapest, Hungary (1997).

- [Lu04] Lussier S., New tricks: How open source changed the way my team works, IEEE Software, January/February (2004).
- [[Or96] Orlikowski W., Organizational transformation over time, Information systems research, Vol. 7, No. 1 (1996).
- [Pa94] Parnas D. L., Inspection of safety critical software using function tables, Proceedings of IFIP World Congress (1994).
- [RM00] Russ M. L. & McGregor J. D., A software development process for small projects, IEEE Software, September (2000).