ISPIS: A Framework Supporting Software Inspection Processes.

Conference Paper · January 2004

DOI: 10.1109/ASE.2004.1342772 · Source: DBLP

Marcos Kalinowski
Universidade Federal Fluminense

Guilherme Horta Travassos
Federal University of Rio de Janeiro

ISPIS: A Framework Supporting Software Inspection Processes

Marcos Kalinowski
Systems Engineering and Computer Science Program – COPPE/UFRJ
mkali@cos.ufrj.br

Guilherme Horta Travassos
Systems Engineering and Computer Science Program – COPPE/UFRJ
ght@cos.ufrj.br

Abstract

This paper describes ISPIS, a computational framework for supporting the software inspection process whose requirements set was derived from knowledge acquired by empirical studies. ISPIS allows the inspection of all artifact types by geographically distributed teams. Specific defect detection support is provided by the integration of external tools. A case study has shown the feasibility of using ISPIS to support real inspections.

1. Introduction

Over the years, many theories and techniques regarding software inspections have been proposed. Some of them have been evaluated by empirical studies, and can be considered a body of knowledge in the software inspection field. For instance, in [1], a reorganization of the software inspection process, based on results of empirical studies has been described.

Moreover, results of a survey [2] show that although many software companies perform reviews they do it unsystematically and few knowledge about software inspections is used. Thus the full potential of reviews is seldom exploited.

Many tool support proposals sprouted to address problems concerning software inspections [3][4][5], most of these tools focus on defect detection or the inspection of specific artifact types. However, analyzing research knowledge concerning software inspections, it seems possible to explore empirical evaluated information to offer more support to some decision points of the inspection process. For instance, knowledge described in [6][7][8][9] and [10] has not been used in any of these approaches. Moreover, integration with other defect detection tools to provide appropriate support for the inspection of different artifact types is also not considered by these approaches.

Based on this scenario, we introduce ISPIS, a computational framework for supporting the software inspection process, which can be used by geographically distributed teams to inspect different artifacts produced throughout the software development life cycle. Many of the requirements for this framework are derived from knowledge acquired by empirical studies.

2. ISPIS

ISPIS supports the reorganization of the inspection process presented by Sauer et al. [1], which introduces changes to reduce the cost and total time for the accomplishment of a particular inspection type, the inspections with asynchronous meetings performed by geographically distributed teams. To provide appropriate support for the use of specific defect detection techniques ISPIS uses an integration mechanism, which generates XML transformation drivers, to integrate external tools. Thus the data produced by these tools can be shared with ISPIS. Figure 1 shows ISPIS's integration philosophy, used to deploy the complete computational framework described in [11].

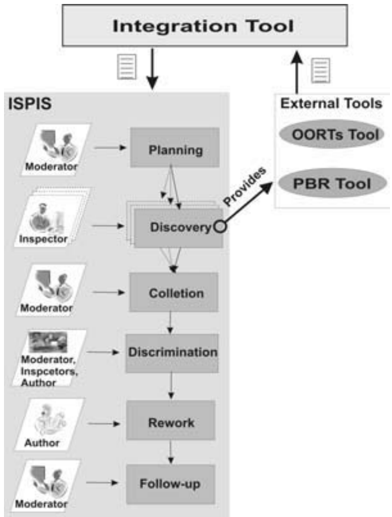

Figure 1. Software Inspection Support Framework.

The external tools currently available are PBR Tool [12] and OORTs Tool (prototype), which aims to support the inspection of object oriented high level designs.

To achieve inspection process coordination ISPIS was implemented as an extension of a workflow tool. It allows ISPIS to dynamically change the process definition when decisions are made during the inspection's enactment. Moreover, notifications are sent by e-mail to participants when activities need to be accomplished.

The inspection process management is supported allowing the moderator to monitor the inspections, and using historical data to support decision making tasks. Below the support provided to each inspection process activity is described.

In the planning activity, inspector performance data and a sorted list of the most indicated inspectors for the inspection being planed are provided. The sorted list could be achieved by using knowledge described in [7].

For defect discovery ISPIS by itself supports only ad-hoc inspections, which can be used to inspect all artifact types. More specific support can be provided by registering external tools in ISPIS. Once registered, the tool is provided for inspections that should use the technique it supports.

In the defect collection activity, following the empirically evaluated suggestion of [10], duplicated discrepancies identified by the moderator are directly classified as defects and forwarded to the document author's rework activity.

Defect discrimination proceeds as an asynchronous meeting with discrepancies being used as discussion topics. According to the results of an empirical study [9] the anonymity of the participants helps to correctly classify discrepancies. Thus in ISPIS participants are anonymous and only their roles are displayed.

During rework the author can attach the artifacts's corrected version and produce a defect correction form.

Finally, in the follow-up activity the moderator's decision to re-inspect (or not) the artifact is supported by estimating the current and the next inspection's defect detection effectiveness. To estimate the current defect detection effectiveness the *Weighed Average of Individual Offsets* (WAO), presented and empirically evaluated in [8] is used. To estimate the next inspection's defect detection effectiveness the ILM model, described in [6], is used. Moreover, a comparison with the number of defects found on the same artifact type in historical data is provided.

A case study [11] showed the feasibility of using ISPIS in real inspections (using ad-hoc support and using external tools). After the case study some adjustments were made and ISPIS has been adopted for real inspections at COPPE/UFRJ. Moreover, an experimental study [11] was conducted to evaluate the support provided to the planning activity and showed very positive results.

3. Conclusions

We believe that making ISPIS available at COPPE/UFRJ could allow its use by research partners and industry. Our assumption is that making more systematic inspections in practice can be possible through the appropriate use of historical data and empirically evaluated knowledge regarding inspections.

Acknowledgements

The authors would like to thank CAPES, CNPq, and the CNPq-NSF READERS project for financial support.

References

[1] Sauer, C., Jeffery, D.R., Land, L., Yetton, P., "The Effectiveness of Software Development Technical Review: A Behaviorally Motivated Program of Research", IEEE Transactions on Software Engineering, 26 (1): 1-14, Jan. 2000.

[2] Ciolkowski, M., Laitenberger, O., Biffl, S., 2003, "Software Reviews: The State of the Practice", IEEE Software 20 (6): 46-51.

[3] MacDonald, F., Miller, J., "A Comparison of Computer Supported Systems for Software Inspection", Automated Software Engineering, An International Journal, Volume 6, No. 3, pp.:219-313, July 1999.

[4] Lanubile, F., Mallardo, T., "Tool support for distributed inspection", Proc. of the 26th Annual International Computer Software & Applications Conference (COMPSAC 2002), Oxford, England, August 2002.

[5] Grünbacher, P., Halling, M., Biffl, S., "An Empirical Study on Groupware Support for Software Inspection Meetings", Proceedings of the 18th IEEE International Conference on Automated Software Engineering, 2003.

[6] Adams, T., "A formula for the re-inspection decision", Software Engineering Notes 24(3): 80, 1999.

[7] Carver, J., "The Impact of Background and Experience on Software Inspections", PhD Thesis, University of Maryland, USA, 2003.

[8] Biffl, S., Halling, M., Koeszegi, S., "Investigating the Accuracy of Defect Estimation Models for Individuals and Teams Based on Inspection Data", Proc. of the 2nd ACM-IEEE ISESE, Italy, 2003.

[9] Vitharana, P., Ramamurthy, K., "Computer Mediated Group Support, Anonymity, and the Software Inspection Process: An Empirical Investigation", IEEE Transactions on Software Engineering, vol. 29, number 2, Feb. 2003.

[10] Launubile, F., Mallardo, T., "An Empirical Study of Web-Based Inspection Meetings", Proc. of the 2nd ACM-IEEE ISESE, Italy, 2003.

[11] Kalinowski, M., Travassos, G.H., "A Computational Framework for Supporting Software Inspections", Proceedings of the 19th IEEE Conference on Automated Software Engineering, September 2004

[12] Silva, L.F.S, Travassos, G.H., "Tool-Supported Unobtrusive Evaluation of Software Engineering Process Conformance", Proc. of the 3rd ACM-IEEE ISESE, USA, 2004.