Improving Software Process Accountability with Spago4Q

Ernesto Damiani[1], Frati Fulvio[1], Sergio Oltolina[2], Gabriele Ruffatti[2]
[1]*Department of Information Technology*
University of Milan - Italy
{damiani,frati}@dti.unimi.it
[2]*Engineering Ingegneria Informatica - Italy*
{sergio.oltolina,gabriele.ruffatti}@eng.it

Abstract

Accountability in current development processes is achieved following invasive procedures that oblige developers to strictly follow organizations' policies, making such an important aspect for development process an onerous task. In this paper, we present the open source application Spago4Q, a flexible process monitoring tool designed for maturity assessment but adaptable for tasks like accountability monitoring.

1. Introduction

The concept of *accountability* has been treated in most of environments where the achieving of the results implies cooperative team work. In bibliography, the concept has been studied, as for instance, in the economics, engineering, and law fields, giving a definition of accountability as the acknowledgment and assumption of responsibility for actions, products, decisions, and policies including the administration, governance and implementation within the scope of the role or employment position, and encompassing the obligation to report, explain and be answerable for resulting consequences. This concept has been adopted and integrated by computer scientists, especially in case of large and scattered working teams. In that case, it is of paramount importance knowing *who* has done *what*. In her work [1], Kathy Parris describes the problems and the achievements of the implementation of an *accountability model* in a real working case study. In her model, accountability is seen as an exchange of accountings (a report of what you have done) and consequences (feedback on your actions), taking into consideration all the relationship between the development team and manager and stakeholders. Accounting may focus on end results, performance to plan, or planning, and accountability cycles are focused primarily on planning at the beginning, and then shifted more toward performance-to-plan and end results. While the Parris model is process-independent, some accountability-oriented processes have been proposed. Among them, the *Team Software Process* (TSP) [2], proposed by the Software Engineering Institute (SEI), stands out as an industrial strength, modern, large-scale, integrated framework that guides development teams in producing high-quality software-intensive systems. As for instance, TSP laid its basis on team work, where every activity of participants and the accomplishment of milestones are tracked through an extensive use of documents, and imply the complete respect of the engineering discipline it proposes. The two outlined approach, if adopted in all their aspects, result *invasive* for developers and require documentation effort. The open source tool Spago4Q, presented in Section 2, offers a *non-invasive* solution for that problem, in the sense that it requests an effort to developers that is as lower as possible and it do not implies any change to their typical work activities.

2. What is Spago4Q

Spago4Q (www.spago4q.org) [3] is an open source platform driving a new approach to software development process monitoring: it integrates an advanced meta-model representation for the process (the Spago4Q meta-model, developed in collaboration with the University of Milan [4][5]) which makes it fully independent from the specific development process, from the underlying data representation technique and from certification and measurement frameworks. It can be easily adapted to complex organizational contexts, independently from the adopted software development processes, infrastructure tools, measurement and assessment frameworks. Spago4Q is suitable for maturity assessment, effectiveness of development software process and quality inspection of the released software: this goal is achieved by evaluating data and measures collected from the project management and development tools with non-invasive techniques. It's a specialization of SpagoBI, the Business Intelligence Free Platform,

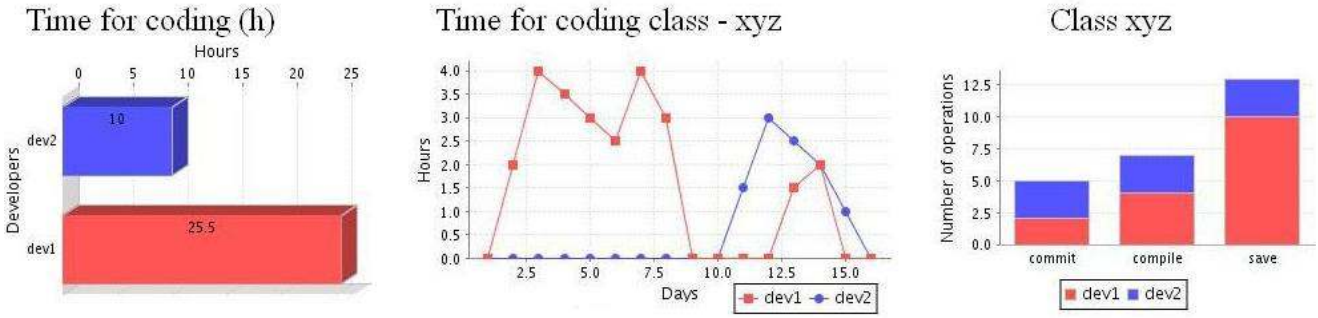

Fig. 1: Examples of Spago4Q Accountability dashboards

designed in order to be easily adapted to complex organizational contexts, independently from the adopted software development processes, infrastructure tools, measurement and assessment frameworks, supporting companies and organizations both in the certification process and, more in general, in monitoring a formalized development process.

3. Using Spago4Q to Improve Accountability

As seen before, Spago4Q has been design to support maturity assessments, effectiveness of development software process and quality inspection of the released software, thanks to its capability to automatically collect and evaluate data and measures from the project management and development tools. This makes Spago4Q a platform suitable for accountability too; as for instance, it offers functions allowing the tracking of developers access to software code, of the coding updates, or of the achieving for each developers of project's goals and milestones, and producing analysis and reporting (like *ad-hoc* accountability dashboards presented in Fig. 1) that give detailed information about accountability in a non-invasive fully-transparent way for developers. Looking at TSP example, Spago4Q could be configured to produce reports that combine data extracted from code repositories (such as Subversion) and planning tools (such as Polarion), to track the respect for each developers of milestones and goals, and the activity on specific classes or documents, and give an effective assessment of project risks basing on specific developers' activities. To achieve such results, Spago4Q could be easy integrated with specific monitoring applications, like the PROM tool (www.prom.case.unibz.it), that are able to track developer activities at a very fine-grained level of granularity, i.e. tracking the time spent on a single class or the interaction within the user and the IDE environment (save, compile, commit), in a fully-transparent manner. This approach is useful to acknowledge the source of each change or work on code, establishing a reliable non-invasive accountability system. Then, data collected by PROM could be combined with Spago4Q data and analyzed in specific reports.

4. Conclusions

The Spago4Q tool we presented, developed in particular for process maturity assessments, is a valid solution for organizations who want to monitor and report the accountability aspect of their process, inserting them in a wider solution that includes all the elements of the development process governance, monitor and analysis.

References

[1] K.V.C. Parris, "Implementing Accountability," in *IEEE Software*, vol. 13, no. 4, pp. 83-93, Jul., 1996

[2] W.S. Humphrey, "Introduction to the Team Software Process," *Addison-Wesley Professional*, 1999

[3] G. Ruffatti, E. Damiani, C. Bellettini, A. Colombo, F. Frati, S. Oltolina, and D. Tura, "New Trends towards Process Modeling: Spago4Q", in Proc. *2nd Knowledge Management in Organization Conference (KMO 2007)*, Lecce, Italy, pp. 219-222, 2007

[4] E. Damiani, A. Colombo, F. Frati, and C. Bellettini, "A Metamodel for Modeling and Measuring Scrum Development Process," Proc. *8th Int. Conf. on Agile Processes in Softw. Eng. (XP2007),* Como, Italy, 2007

[5] A. Colombo, E. Damiani, F. Frati, S. Oltolina, G. Ruffatti, and K. Reed, "The use of a meta-model to support multi-project process measurement," in *Proc. 15th Asia-Pacific Software Engineering Conference (APSEC 2008)*, Beijing, China, 2008