SQUALE – Software QUALity Enhancement

Alexandre Bergel, Simon Denier, Stéphane Ducasse, Jannik Laval,
Fabrice Bellingard, Philippe vaillergues , Françoise Balmas,
Karine Mordal-Manet, ...

November 14, 2008

DON NOT EDIT THIS VERSION ANYMORE svn+ssh://ducasse@scm.gforge.inria.fr/svn/squale
Repository Root: svn+ssh://ducasse@scm.gforge.inria.fr/svn/squale

The Squale project was born from industrial effort to control software quality. Its goals are to refine and enhance Qualixo, a software-metric based quality model already used by large companies in France (AirFrance, PSA) and to support the estimation of return on investment produced by software quality. Qualixo is a software quality model based on the aggregation of software metrics in higher level indicators called practices, criterias and factors.

1 Funding and Global Effort

Squale stands for Software QUALity Enhancement. It is a national project, funded by the french Inter-ministerial R&D project 2006 - 2008 (“Projet R&D du Fonds Unique Interministériel”) in the context of the "Pôle de compétititivité Systematic - PARIS Region". The total cost of the project is 3100 K Euros for a two year duration. The project started administratively in June 2008, and concretely in September 2008. The total effort on the project is about 25 year man.

2 Project Goals

Over the last decade, the need for quality in software has increased. Several quality models have been proposed [1, 6, 7, 3, 4]. These models emphasize the need to have quality checks while developing a software program. As far as we are aware of, no model to assess quality of existing software programs have reached a significant acceptance. The project stems from the experience gained by the SME Qualixo with the assessment of software quality for PSA and Air France. The goals of Squale is to evaluate and enhance the existing software quality approach in the following areas:

- evaluate and enhance the current software-metric quality model, including current practices and current metrics,

- define dashboards related to software quality,
- provide means to assess evolution of software quality over time,
- provide economical indicators to assess added value (ROI) of software quality measurements, and
- disseminate acquired knowledge through an open-source platform supporting the model and a community of users

In particular, Squale aims at identifying models for software quality and return on investment and developing open-source tools to aggregate data from external quality measurement tools. Finally one goal is to build an anonymous database of project data to build a reference in the domain.

3 Detailed Description

Software quality measurement is the key to software quality improvement. New trends in software development –outsourcing, merge of information systems, maintenance of "new" legacy systems in Java/C++ as well as a general awareness of the high costs of maintenance– have reinforced the need for software quality measurement. However, quality measurement is still not applied in a systematic way because of the following factors: it lacks standardization, it is difficult to represent into business terms and added values, and lacks proof of its profitability.

Most analysis tools provide low-level software metrics, which are difficult to understand for non-technical people. Software metrics target methods, classes, or packages of a software. Such metrics are syntax-driven (style checking), structural (cyclomatic complexity, etc.), object-oriented (depth of inheritance, etc.), architectural (layered model), model-based (coupling, etc.), test-driven (code coverage, etc.), or generic (performance, security, etc.).

In the Qualixo model, metrics are combined to compute marks for practice between 0 and 3. Practices are then combined to quantify, for the whole or parts of software, quality factors such as maintainability, evolutivity, or reusability. Squale aims at reporting quality on specific dashboards and views, each tailored for a specific user: programmer, project manager, maintenance manager, sale manager, team leader, quality specialist. Software quality has to be monitored throughout the software life-cycle, in order to measure and follow progression or deterioration of quality.

The technical-economical aspect of the project targets the assessment of refactoring costs as well as the benefits of a quality-driven approach. First a technical-economical model should provide a cost for creation and modification of components based on software metrics. Then the cost of refactoring can then be predicted following a quality diagnosis. Second the Squale project aims at computing the profits of a quality-driven approach, *i.e.*, the costs of non-quality. The goal is to assess the benefits of quality measurement and also to find its limits.

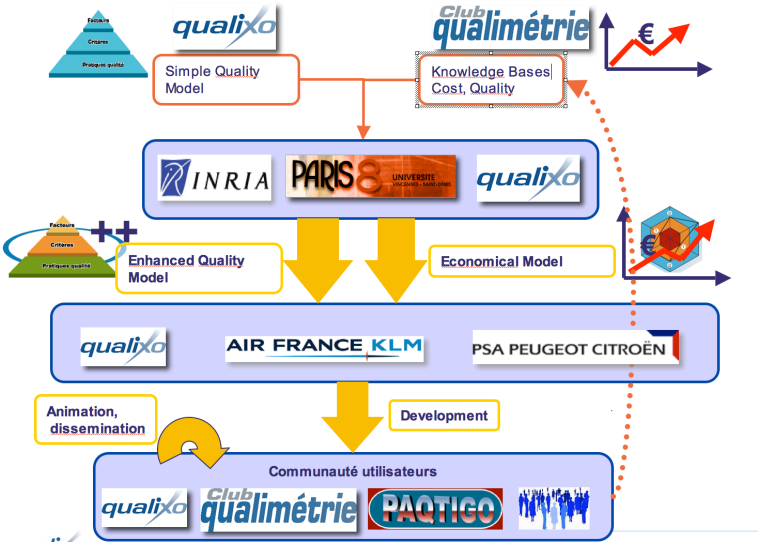

Figure 1: Structure of the Squale Project

In a first stage, Squale will target the Java and C++ programming languages. However, the model and the tool will be adaptable to other languages and data sources through the development of specific tools and the parameterization of the practices-metrics model. The Cobol and ABAP programming languages will be the next targets. The goal is to provide homogeneous dashboards for any software pool. Finally, the Squale project will set up a software forge for the open source tools and launch a user community around quality measurement.

4 Project Participants

The project is structured around three kinds of partners: two scientific partners (INRIA and LIASD), two SMEs experts in software analysis and development, two large companies (PSA and AirFrance). Figure 1 describes the interactions between the participants which are detailed hereafter.

Here is the list of the participants and the main actors.

- RMoD Team of INRIA Lille Nord-Europe: Alexandre Bergel, Simon Denier, Stéphane Ducasse, Jannik Laval.
- LIASD University Paris 8: Françoise Balmas, Karine Mordal-Manet.
- Qualixo: Fabrice Bellingard, Philippe Vaillergues, Florent Zara.
- Paqtigo: Hervé Crespel.

- PSA Peugeot Citroën: Thierry Bey. PSA is one of the early adopters of the Qualixo model.

- Air France DSI, Arnaud Poivre. AirFrance was a co-developer of the Qualixo quality model since a couple of years.

5 Achievements

According to Marinescu and Ratiu [6], Qualixo can be classified as a Factor-Criteria-Metrics quality model. Qualixo is being applied in large companies such as AirFrance or PSA. It uses measurements to assess software quality. These measurements cover a number of different aspects of a software, including specification accuracy, programming rules, and test coverage.

While the project is still young, several promising works are on progress.

- Squale the platform implementing the Squale model will be released officially as open-source software early 2009. A major effort to document and clean it is under development.

- Qualixo has been originally implemented on top of Eclipse. An implementation of Qualixo, named MoQam (Moose Quality Assessment Model) is under development in the Moose open-source and free reengineering environment. A first experiment has been conducted [5]. Exporters from Moose to the Squale software are under development.

- We are assessing the metrics and practice used originally in the Qualixo model. We are also compiling of a number of metrics for cohesion and coupling assessment. We want to assess for each of these metrics whether their relevance in software quality setting.

- Dependency Structure Matrix (DSM), an approach developed in the context of process optimization, has been successfully applied to identify software dependencies among packages and subsystems. A number of algorithms helps organizing the matrix in a form that reflects the architecture and highlights patterns and problematic dependencies between subsystems. However, the existing DSM implementations often miss important information in their visualization to fully support a reengineering effort. We distinguish independent cycles and stress cycles using coloring information.This work has been implemented on top of the *Moose* open-source reengineering environment and the Mondrian visualization framework. It has been applied to non-trivial case studies such as the *Morphic UI* framework available in open-source Smalltalk *Squeak* and *Pharo*. Results have been implemented in the *Pharo* programming environment. A first experiment has already been conducted [2].

References

[1] Jagdish Bansiya and Carl Davis. A hierarchical model for object-oriented design quality assessment. *IEEE Transactions on Software Engineering*, 28(1):4–17, January 2002.

[2] Alexandre Bergel, Stéphane Ducasse, Jannik Laval, and Romain Peirs. Enhanced dependency structure matrix for moose. In *FAMOOSr, 2nd Workshop on FAMIX and Moose in Reengineering*, 2008.

[3] Robert L. Glass. *Building Quality Software*. Prentice-Hall, 1997.

[4] Ho-Won Jung, Seung-Gweon Kim, and Chang-Shin Chung. Measuring software product quality: A survey of iso/iec 9126. *IEEE Softw.*, 21(5):88–92, 2004.

[5] Jannik Laval, Alexandre Bergel, and Stéphane Ducasse. Assessing the quality of your software with moqam. In *FAMOOSr, 2nd Workshop on FAMIX and Moose in Reengineering*, 2008.

[6] Radu Marinescu and Daniel Raţiu. Quantifying the quality of object-oriented design: the factor-strategy model. In *Proceedings 11th Working Conference on Reverse Engineering (WCRE'04)*, pages 192–201, Los Alamitos CA, 2004. IEEE Computer Society Press.

[7] Diomidis Spinellis. *Code Reading The Open Source Perspective*. Addison-Wesley, 2003.