

# TOWARD SOFTWARE GOVERNANCE IN R&D PROJECT

## *Géant Case Study*

Paweł Kędziora, Marek Lewandowski and Cezary Mazurek

*ICHB PAN, Poznań Supercomputing and Networking Center, Noskowskiego 12/14, 61, 704 Poznań, Poland*

Keywords: Software Governance, Quality Assurance, R&D Project.

Abstract: This paper presents motivation of introducing Software Governance activity into large R&D project, that has been already developed for several years. We present background of this activity, its structure and communication model. We will also present impact of the new activity on the whole project and describe expected results of this innovation. Introduction of the new activity has not been fully completed yet.

## 1 INTRODUCTION

Software governance is a crucial activity when organisation is focused on the efficient development and delivery of high quality IT products. Software governance lies in the scope of interest of numerous researches, groups of activists and large corporations. Because of complexity of the issue, and differences between organisation structures it is difficult to propose a generic and comprehensive model of software governance activity that would provide a set of mechanisms appropriate for majority of projects. Nevertheless, an attempt like this has been made. The paper at hand describes the approach of applying a customized software governance model, which uses the existing, well-known frameworks as a foundation.

### 1.1 Related Work

The standardisation efforts that were made over recent years, effected in creation of several well-known frameworks, like COBIT (Control Objectives for Information and related Technologies) ([www.isaca.org/cobit](http://www.isaca.org/cobit)) or ITIL (Information Technology Infrastructure Library) ([www.itil-officialsite.com](http://www.itil-officialsite.com)). Generality of these approaches is their most significant advance, since they allow to arrange and maintain the whole structure of the organisation and all processes related to the development and delivery of the software. However, the generic approach entails a lack of low-level procedures and best practices, that could be

introduced at the bottom of the organisation, especially among developers.

Global corporations benefitting their experience in development, deployment and maintenance of large scale IT projects, provide their methodologies and frameworks for software governance. These are usually delivered in a form of extensive packages containing not only knowledge ("best practices", or "know-how" procedures), but also products that enable to put it into practice. For example Oracle publishes on its web pages a number of "best practices" articles, e.g. "Continuous Integration with Hudson" (Stegeman, 2010), and delivers products (e.g., Oracle Enterprise Manager), that allow to fit the structure of the enterprise, project and the development process into selected methodology. It is a common practice, that large corporations deliver trainings and perform audits of enterprises and developed products; e.g., HP periodically organizes ITIL courses and exams (<http://www.hp.com/education/sections/itsm.html>).

Enterprises, that cannot, or are not interested in applying solutions delivered by external providers, should elaborate these on their own. Before developing own software governance policy or framework, current governance maturity must be defined. Craig Symons (Symons et al., 2005) refers to Forrester's Maturity Model and presents four stages of maturity: ad hoc, fragmented, consistent, best practices. Table 1 presents detailed descriptions of these stages.

Table 1: Four stages of Forrester's Maturity Model.

Ad hoc	There are no formal IT governance processes, procedures or mechanisms. IT investments are made on a completely ad hoc basis.
Fragmented	There has been some effort to formalize IT governance practices. These formalized processes may exist in some units and IT decisions within those units may be optimised, but there is no global effort to coordinate it within whole enterprise
Consistent	IT governance processes have been consistently applied across the enterprise.
Best practices	IT governance processes are fully evolved and optimized across the enterprise. A strong IT portfolio management process is in place to ensure that all IT investment decisions are themselves optimized. Executives are active participants in the governance process.

If an organisation wants to change its stage of maturity, a detailed roadmap (tasks and schedule) should be prepared. Anytime the maturity is increased, the probability of successful, operational deployment of the software augments (Michlmayr, 2005).

## 1.2 Motivation

Authors of this paper have been involved in the Géant project ([www.geant.net](http://www.geant.net)). In April 2009, Géant moved from its second (GN2) to third (GN3) phase, and thus faced a possibility of reorganization. Project Office had taken a proactive effort and had decided to change structures of involved teams, in order to minimize the possibility of occurrence of problems related to very individual activities of involved teams. It had been noticed, that projects, that finally would complement one another, are radically different, when methodologies, used tools and delivered quality are concerned. Project office decided to establish new, independent, activity, that was meant to provide development activities with "best practices" papers, coherent software infrastructure and support and audit mechanisms. The aim of the brand new activity was to provide uniform quality of products developed within a single consortium and thus move from "fragmented" towards "consistent" stage of maturity. According to ITIL and COBIT recommendations (Wallhoff,

2004), the new activity possessed a full autonomy and was completely separated from development activities.

In the following section we provide detailed information about software governance activity in R&D project: its internal structure, lifecycle and comparison of expected and achieved goals.

## 2 SOFTWARE GOVERNANCE IN GÉANT PROJECT

Software governance activity, had been split into three tasks:

- Best Practices,
- Development Infrastructure,
- Quality Assurance.

Division of the activity had taken into consideration individual characteristics of work in particular tasks (continuity versus single effort). The whole activity was intended to last for four years and end together with GN3 phase of the Géant project.

### 2.1 Best Practices

The first task, Best Practices, was supposed to prepare a set of procedures that would standardize development, deployment and maintenance of the software products. It was decided to poll all involved software development activities. The results of the survey were to be an introduction to a discussion about best practices that were to be delivered. After the poll, when its results were analyzed, it had been noticed, that projects, deployed within a single consortium, differ significantly. These differences could lead to incompatibility problems. The lack of common approach effected in unequal quality of delivered products. Results of the survey (Marović et al., 2009) revealed that all teams declared to software development use agile methodologies, but most of them introduced them inconsequently or selectively. Moreover, development infrastructure was unnecessarily multiplied, what led to ineffective usage of manpower, and inadequate dispersion of software source codes and documentation. Software governance leaders decided to prepare best practices on the basis of agile methodologies recommendations. Best practices were extended with recommendations related to the usage of coherent development infrastructure. Following register presents the principals of the agile methodologies (Beck et al., 2001):

- Individuals and interactions over processes and tools,
- Working software over comprehensive documentation,
- Customer collaboration over contract negotiation,
- Responding to change over following a plan.

Agile methodologies focus on individual developers (Abrahamsson et al., 2003), but when applied to multi-projects structure (like in a large consortium), have to be enclosed in specified frames. In case of Géant, these frames were Best Practices, that motivated developers and task leaders to follow a "common path" of the development. Project Office realized, that too much discipline would deprive development teams of initiative and flexibility (Nawrocki et al., 2005), so a common, but not limiting solution was needed.

Since OpenUP was the most popular agile methodology, it was selected to be the basis for best practices document. Later on, best practices have been extended with requirements suggested by basic and intermediate levels of Open Maturity Model (Petrinja et al., 2009).

Because of the range of the issues covered, best practices were published in three separate documents:

- "GN3 Software Developer Best Practice Guide" (Marović et al., 2009),
- "GN3 Software Architecture Strategy Best Practices Guide" (Marović et al., 2009),
- "GN3 Quality Assurance Best Practices Guide" (Marović et al., 2009).

The first paper ("Software Developer Best Practice Guide") introduced a common approach to development and deployment and provided principals for the most basic, source code level integration, since coding and releasing a software became a well-defined process. Developers were provided with unified procedures of setting up the development environment and exemplary project structure was published. The first paper explained how to use version control system in a secure and effective way. Detailed description of effective and "agile oriented" (with frequent releases and rapid changes of requirements) procedures related to building, integrating and releasing software were also provided. "Software Developer Best Practice Guide" put emphasis on providing meaningful and cohesive documentation.

The second paper ("Software Architecture Strategy Best Practices") was intended for lead developers and project managers. It rephrased agile

development principals, adjusting them to Géant circumstances and placed emphasis on the issues that were applied differently by individual project leaders. This paper also described recommended approach to software design and explained complicated aspects of IPR policies and FOSS licences. The document provided cohesion and enabled integration between projects. The in depth analysis of IPR issues revealed conflicts between several licences assumptions and Géant principles. As a result, a new activity (IPR Management) was called into being.

The third paper ("Quality Assurance Best Practices") described in details issues related to testing and code quality checking. This paper was intended for developers and emphasised importance of test-driven development. It described testing approaches and tools required in successive development phases (e.g. unit, integration, functional tests). Properly defined, unified test approaches provide a uniform, high quality of delivered software. "Quality Assurance Best Practices" provided detailed procedures of bugs handling, requesting a new functionality and also described a lifecycle of the request ticked. Best practices, delivered in the third paper, provided independent quality measures and allowed project leaders to compare different projects.

Creation of aforementioned best practices documents was consulted with Development Infrastructure task. Best practices were intended to be not only abstract recommendations, but also guidelines to shared infrastructure, provided by the other task. The delivered infrastructure is described the in following section.

## 2.2 Development Infrastructure

The second task, "Development Infrastructure" focused on providing shared infrastructure for developed projects. Results of the survey, mentioned in the previous section of this paper, indicated, that numerous of development tools have been multiplied. This approach was not efficient, because effort of people responsible for providing and maintaining the infrastructure was also multiplied. The aim of this task was to provide a unified development infrastructure, before the actual GN3 development begins. Because of clearly specified requirements, relatively easy job (installation and configuration) and due to experience of involved employees, this task could deliver required solutions in a very short time. Development Infrastructure task delivered the following solutions:

Table 2: Products delivered by Development Infrastructure task (Żurowski and Kramer, 2009).

Version Control System	SVN
Build Management System	Maven
Continuous Integration Server	Hudson
Bug Tracking System	JIRA
Release Planning Tool	JIRA
Developer Portal System	TWiki
Wiki	TWiki
Mailing List System	Mailman

Centralized version control system provides a reliable and extremely safe environment for storing and maintaining the source code. SVN, the chosen system, not only allows to track applied changes, but also provides effective mechanisms for branching and merging the code. A common repository provides an easy access to all developed systems and thus allows audits, since all releases and up-to-date source code are available on demand.

Results of the survey indicated a lack of common approach to software build management. Particular projects built their software individually, what effected in different structures of resulting packages. Using a uniform build management system with properly defined tasks speeds up the software release process by automating the following routines:

- Building, testing and packaging.
- Releasing.
- Deployment of information pages.
- Software deployment.
- Dependencies resolving.
- Artefacts publishing.

For projects developed in Java, Maven has been suggested. The introduction of this tool turned out to be the most effective of all applied changes and enforced a “quality assurance oriented” way of management and development. OpenUP methodology assumes, that a build management system is tightly coupled with a continuous integration server. The continuous integration performs unit and integration tests when source code repository is changed and reports detected failures to particular developers. In case of GN3 development, Hudson has been used as a continuous integration server.

Finally, development infrastructure task focused on improving communication level. Géant, as a large scale project, developed by employees distributed across several European countries was exposed to vulnerabilities related to communication issues (Elliott and Scacchi, 2004). Introducing JIRA, a

browser-based issue tracking system significantly increased internal (among developers) and external (between developers and customers) communication level by providing well defined lifecycle of any project related issues. JIRA also enforced consistent approach to planning, since each project had to define its roadmap and formal release plan. Additionally, this tool provides advanced statistics, which allow to monitor current development progress and redefine release plan if needed. Common requesting and planning system allows to compare projects and estimate progress of the development of all products. Suggested developer portal, TWiki, provides a built-in versioning mechanism, so each modification of existing page or document, effects in a creation of a new version.

Infrastructure, delivered by Development Infrastructure task, in conjunction with best practices provided by the previous one, introduced new standards and provided tools for high quality and effective development. Results of these tasks efforts have been received with interest by developers. Although interest had been demonstrated, a mechanism of control had to be provided. To comply with this need, Quality Assurance task had been launched. The following section provides detailed information about the third task of Géant Software Governance activity.

### 2.3 Quality Assurance

Quality Assurance, the third task of the Software Governance activity, was supposed to verify if projects developed in GN3 phase of Géant have been migrated to the delivered infrastructure and whether best practices, provided by a first task, have been put into life. This task was launched when best practices and development infrastructure were already in place.

Quality Assurance performs periodical (every four months) audits and, during this procedure, the auditor completes a questionnaire, that is homogenous for all projects. The form contains questions that are to verify not only the fact of following recommended best practices, but also the rate of indicators, that can be measured (e.g. code coverage - the percentage of source code verified by unit tests). After the internal review, the results of the audit are presented to lead developers and their feedback is collected. Lead developers cannot affect auditors and audit itself, but are to focus on issues that have been, in their opinion, missed or inadequately audited. Lead developers’ feedback is presented to the auditors. If the auditor accepts it, the particular audit is extended by lead developer’s



remarks. If the auditor does not agree, remarks are rejected. At the end of the iteration the results of audits are officially released. The released document contains not only results of the audit procedures, but also recommendations to eliminate weak points. Next iteration of audit focuses on indicated weak points and changes applied to the software.

GN3 phase of Géant began with the initial iteration of auditing. This iteration was supposed to provide primary status of the source code that was delivered in the previous phase of the project. Results of the initial audit have been delivered to lead developers, but these have not been officially published. Lead developers provided their feedback with suggestions of improving communication and audit itself.

Currently, the first iteration is in its final stage and the audit document is being prepared to be published. Quality Assurance task is going to perform six official audits and to provide advices and recommendations after each of them. After the sixth audit, a final deliverable document is to be published. In this document results of all audits will be provided. Quality Assurance will summarize its activity and describe changes that were applied to the projects to increase their quality and compatibility. We believe, that all tasks of Software Governance activity will succeed, and approach described in this paper will become a foundation in other, large scale R&D projects.

### 3 CONCLUSIONS

In this paper we have described efforts that were made in order to increase the level of quality and compatibility between different software projects developed in a distributed environment within a large scale R&D consortium.

This paper described Géant case. The previous phase of Géant, GN2, suffered from the lack of common methodologies, best practices and development infrastructure. The new Software Governance activity focused on these issues. Its internal structure reflects problems that it was meant to solve. Best Practices task provided set of documents with recommendations related to software development, software architecture and quality assurance. These best practices were tightly coupled with infrastructure delivered by Development Infrastructure task. To verify proper application of best practices and development infrastructure, another task, Quality Assurance, had been launched. This task has focused on series of audits, applied to all software projects that will

eventually be operationally deployed. The following audits focuses on changes applied in order to fulfil recommendations from the previous audit.

The described Software Governance activity has been launched at the very beginning of the GN3 phase, whereas best practices and development infrastructure have been provided in the first weeks of activity. The first iteration of audits is in the final stage, and recommendations are to be published within upcoming weeks. Nevertheless, even at this stage, compatibility and manageability of projects have significantly increased.

### ACKNOWLEDGEMENTS

The authors of this article are thankful to Mr. Anand Patil, initiator and the first leader of described Software Governance activity for his devotion and valuable comments when reviewing this article.

### REFERENCES

- Abrahamsson P., Warsta J., Siponen M.T., Ronkainen J., 2003. *New Directions on Agile Methods: A Comparative Analysis*. International Conference on Software Engineering, May 3-5, 2003, Portland, Oregon, USA.
- Beck K., Beedle M., van Bannekum A., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R., Melleor S., Schwaber K., Sutherland J., Thomas D., 2001. *Manifesto for Agile Software Development*. Available at: <http://agilemanifesto.org/> [Accessed 26 March 2010].
- Elliott M. S., Scacchi W., 2004. *Free Software Development: Cooperation and Conflict in A Virtual Organizational Culture*. IDEA Publishing
- Marović B., Kramer G., Wrzos M., Lewandowski M., Bobak A., Delvaux A., Żurowski W., Zuljevic T., 2009. *GN3 Software Architecture Strategy Best Practice Guide 1.0* Available at: [http://www.geant.net/Media\\_Centre/Media\\_Library/Media%20Library/GN3-09-185v2\\_Software\\_Architecture\\_Strategy\\_Best\\_Practice\\_Guide\\_1.0.pdf](http://www.geant.net/Media_Centre/Media_Library/Media%20Library/GN3-09-185v2_Software_Architecture_Strategy_Best_Practice_Guide_1.0.pdf) [Accessed 26 March 2010].
- Marović B., Kramer G., Wrzos M., Lewandowski M., Bobak A., Krysztofiak T., Martinovic R., Kostov S., Kupański S., 2009 *GN3 Quality Assurance Best Practice Guide 1.0* Available at: [http://www.geant.net/Media\\_Centre/Media\\_Library/Media%20Library/GN3-09-184v2\\_Quality\\_Assurance\\_Best\\_Practice\\_Guide\\_1.0.pdf](http://www.geant.net/Media_Centre/Media_Library/Media%20Library/GN3-09-184v2_Quality_Assurance_Best_Practice_Guide_1.0.pdf) [Accessed 26 March 2010].
- Marović B., Kramer G., Wrzos M., Lewandowski M., Delvaux A., Zuljevic T., Rodriguez C., 2009. *GN3*

- Software Developer Best Practice Guide 1.0*. Available at: [http://www.geant.net/Media\\_Centre/Media\\_Library/Media%20Library/GN3-09-186v2\\_Software\\_Developer\\_Best\\_Practices\\_Guide\\_1.0.pdf](http://www.geant.net/Media_Centre/Media_Library/Media%20Library/GN3-09-186v2_Software_Developer_Best_Practices_Guide_1.0.pdf) [Accessed 26 March 2010].
- Marović B., Mazurek C., Kramer G., 2009, *SA4 Software Developer Survey. Survey Specification v2.2*. Available at: <http://wiki.geant.net/pub/SA4/Admin/DocumentsMain/GN3-09-134-SA4-Software-Developer-Survey-Specification.pdf> [Accessed 26 March 2010]
- Michlmayr M., 2005. *Software Process Maturity and the Success of Free Software Projects*. Department of Computer Science and Software Engineering, University of Melbourne.
- Nawrocki J., Olek L., Jasiński M., Paliświat B., Walter B., Pietrzak B., Godek P., 2005. *Balancing Agility and Discipline with XPrince*. RISE 2005, Heraklion, Greece.
- Petrinja E., Nambakam R., Sitalini A., 2009. *Introducing the Open Maturity Model*. 2nd Emerging Trends in FLOSS Research and Development Workshop at ICSE 2009.
- Stegeman, J., 2010 *Continuous integration with Hudson*, Oracle publication
- Symons C., Cecere M., Oliver Young G., Lampert N., 2005. *IT Governance Framework Structures, Processes, And Communication*. Forrester Research
- Wallhoff J., 2004. *Combining ITIL with COBIT and ISO/IEC 17799:2000*. Scillani Information Whitepaper.
- Żurowski W., Kramer G., 2009. *Deliverable DS4.3.1: Specification of Software Development Infrastructure*. Available at: [http://www.geant.net/Media\\_Centre/Media\\_Library/Media%20Library/GN3-09-099-DS4-31\\_Specification\\_of\\_Software\\_Development\\_Infrastructure.pdf](http://www.geant.net/Media_Centre/Media_Library/Media%20Library/GN3-09-099-DS4-31_Specification_of_Software_Development_Infrastructure.pdf) [Accessed 26 March 2010].