Software Governance in a Large European Project - GÉANT Case Study

Marcin Wolski[1](✉) and Toby Rodwell[2]

[1] Poznań Supercomputing and Networking Center, Poznań, Poland
`marcin.wolski@man.poznan.pl`
[2] GÉANT Association, Cambridge, UK
`toby.rodwell@geant.org`

Abstract. Sustainable software development is an essential part of GN4, the project co-funded by Europe's NRENs and the EU. This article presents how software governance has been applied in GN4 during its iterations.

Keywords: Software governance · Software processes · Software improvements

1 Introduction

GÉANT refers both to the research and innovation community of European NRENs (operators of national networks for science and education), and also a sequence of network-related projects co-funded by the EC and the European NRENs. The latest such project is GN4, a truly pan-European collaboration between 39 partners i.e. 37 European NRENs, NORDUnet (representing the five Nordic countries) and GÉANT Association[1]. The third and final phase of GN4, GN4-3, started in January 2019 and will last 48 months.

The NREN community is involved in collaborative software development activities, focused on delivering software products that provide advanced services. The GÉANT product portfolio contains software with different levels of maturity, size and target domains: starting from prototype solutions (proof of concept), through pilot applications that usually target a closed group of users, to production versions supporting the delivery of operational services. Their users mostly comprise of GÉANT partners and their member institutions, researchers,

---

[1] https://www.geant.org/.

students and educators, who expect high-quality, reliable services and infrastructure to support their work or studies.

Proper support of the software governance has had significant attention in GÉANT since before GN4 began, and this grew even stronger with the introduction of service transition and operation processes in the first two phases of GN4 (GN4-1 and GN4-2) [6]. This work has been continued and it is currently the responsibility of GN4-3 WP9 Task 2: Software Governance and Support. This task provides comprehensive governance and support for software development within the project, to guarantee a consistent level of software product reliability and resilience, and ensure the overall quality level of the GÉANT services that rely on these products.

The paper is organized as follows. Section 2 describes in more detail the software development effort in GÉANT and presents the main features of the software projects and teams. Section 3 describes the major outcomes of software governance in GÉANT and outlines the future work in GN4-3 WP9 Task 2. Section 4 explains the interest in participation in the Profes 2019 event and the expected collaboration opportunities this will afford.

2 Software Development in GÉANT

Most software teams in GÉANT (SW teams) are distributed and involve engineers from different NRENs and the GÉANT organization. The teams have autonomy in adopting a specific methodology for software development and choosing associated tools [1]. This results in a wide range of processes and approaches to software development [4].

In Table 1 we present selected information for GÉANT software projects:

- Codebase age – time between the first and last commit in the project,
- Codebase size – Source Lines of Code (SLOC),
- Languages – number of languages used in a project,
- Team size – number of GN4-3 contributors for the project in all roles (tester, developer, manager etc.),
- Team size (software developers) – number of GN4-3 software developers contributing to the project,
- Projects per person – a number of projects a single person contributes to.

Currently, there are 29 projects in the GÉANT software portfolio. They are usually developed by the GÉANT and NREN community, but in a few cases they have been inherited from external open source initiatives, and are currently being developed globally (e.g., COTURN[2]).

The GÉANT product portfolio contains software projects with a wide range of sizes (from less than a thousand to nearly one million lines of code) and system age (from 92 days up to more than 17 years), and all of them use more than one language (on average, 9).

---

[2] https://github.com/coturn/coturn/blob/master/README.md.

Table 1. GÉANT software projects and teams as of July, 2019 (source: https://sc.geant.org)

Statistic name	Items	Min	Max	Median
Software projects	29			
Codebase age		92 days	17 years	3 years
Codebase size		858	965989	35864
Languages		4	23	9
Software teams	29			
Team size		1	10	2
Team size (Software developers)		1	6	2
Project per person		1	4	1
Software contributors	63			
Software developers	47			

Moreover, there are 63 GN4-3 software contributors, which is to say someone who commits source codes and/or reports issues. 47 of these people are software developers, who have made at least 1 commit. Notably, GÉANT software teams are usually small, with just two members on average - the largest team has 10 members, of whom 6 are developers. In most cases people contribute to only one project.

3   Outcomes and Planned Work

3.1   Software Catalogue

The GÉANT Software Catalogue (GSC) addresses the recognized need for a unified repository of software teams and projects in GÉANT, which will lead to improvements in software development by facilitating knowledge exchange and fostering opportunities for cooperation. GSC provides a global view of the whole of software development in GÉANT, automatically aggregating and presenting information from different data sources with software artefacts such as source code repositories and issue tracker systems [2]. The tool has successfully accomplished the transition to production phase and it currently holds consistent information on about 30 software projects and over 300 individuals from GÉANT, who have contributed to the software development (i.e. throughout all phases of GN4).

Further development of the GSC is planned for the remaining years of the GN4-3 project. The roadmap includes improvements and new features requested by the software community, for example support for complementary data source with software artefacts (like SonarQube), and extended reporting capabilities.

3.2 Software Maturity Model

The GÉANT Software Maturity Model (SMM) has been designed to achieve two primary goals: to identify and define the key best practices that help to successfully deliver software; and to highlight areas for improvement by teams. Unlike software methodologies, which specify how to undertake certain actions, maturity models propose a framework that simply specifies the goals to be achieved. Therefore it is a good fit for the GÉANT project, where planning and evaluating the implementation of software products within a distributed environment, with only partial and ever changing involvement of team members, remains exceedingly challenging.

The concept of the SMM was presented and evaluated with the help of selected GÉANT software teams [3]. Based on the initial feedback, a revised version of SMM has been prepared. The current effort is concentrated on producing a policy for software best practices in GN4-3 based on the SMM framework [5]. The outcome of this work will be presented in a GÉANT project deliverable.

3.3 Software Developer Training

For almost a decade, the GÉANT project has organised two types of software development training for the project participants, namely Secure Code Training (SCT) and School of Software Engineers (SSE). SCT focuses on secure programming with the aim of minimizing the number of security bugs in the source code. SSE focus on code quality and management. As with SCT, the topics differ from one training to another to address contemporary GÉANT project requirements, and software methodologies and techniques in common current use in the community. Since 2010 SCT and SSE have respectively trained 107 and 73 developers from over 20 NRENs [7].

Both training events are going to be continued and delivered annually to the GÉANT software community.

3.4 Software Code Reviews

Software code reviews are offered as a supporting activity performed by an independent testing team. An independent review is much more likely to detect errors that the code authors are themselves blind to, as the reviewers, being unfamiliar with the code, do not focus on functionality. Typically code reviews are conducted as part of system testing during the transition of a service to production, but they can be requested by any GÉANT software team at any time, even during an early stage of the Software Development Life Cycle.

Currently the reviews are predominantly manual (i.e. an expert developer reads through the code) but it is planned to extend the use of automated tests in the code review process, using the customized and supported SonarQube tool[3].

---

[3] https://ci.geant.net/sonar/.

3.5   Software Tools

GÉANT currently provides a range of services and tools to support software development[4]. These tools are now accessible to the whole GÉANT community through federated authentication and authorisation[5].

Together the tools form a technology stack which supports the full development life-cycle from requirements management via issue/task management to source code repository, and through continuous integration and deployment service to binaries repository and production deployment. Software projects developed with the use of common GÉANT SWD tools set, policies and best practices will benefit from increased transparency and maintainability of the GÉANT projects. Compliance with Intellectual Property Rights (IPR), which is overseen by a different team within the project, can also be facilitated through the GÉANT Software Development Infrastructure.

4   Collaboration

We expect to meeting a variety of different representatives of the software engineering community attending the conference and would be glad to have the opportunity to speak about GÉANT's experience concerning software governance processes. Equally, hearing lessons learned from other projects could provide further refinements to the existing software development practices in GÉANT. Finally, learning about the tools, policies and best practices used by practitioners and researchers will be valuable for influencing the GÉANT software governance roadmap.

The presentation will include the current status and future work in the area of software governance in the GÉANT project. In addition, there may be a demo of the GÉANT Software Catalogue shown to the audience.

References

1. Bilicki, V., Golub, I., Vuletic, P., Wolski, M.: Failure and success - how to move toward successful software development in Networking. In: Terena Networking Conference (2014)
2. Łabędzki, M., Wolski, M.: GÉANT Software Catalogue as a Code Portfolio. GÉANT Connect, p. 58 (2018). https://www.geant.org/News_and_Events/CONNECT/Documents/CONNECT29FINAL(1).pdf
3. Stanisavljevic, Z., Walter, B., Vukasovic, M., Todosijevic, A., Łabędzki, M., Wolski, M.: Géant software maturity model. In: 2018 26th Telecommunications Forum (TELFOR), pp. 420–425, November 2018
4. Stanisavljevic, Z., Wolski, M., Labedzki, M., Kupiński, S., Todosijevic, A., Adomeit, M.: Harmonising the software development process in the Géant community. GÉANT Connect, pp. 50–51 (2018). https://www.geant.org/News_and_Events/CONNECT/Documents/connect_28_web.pdf

---

4 https://software.geant.org.
5 https://edugain.org/.

5. Todosijevic, A., et al.: Ensuring best practice For Géant software development. GÉANT Connect, p. 29 (2019). https://www.geant.org/News_and_Events/CONNECT/Documents/CONNECT_31.pdf
6. Wolski, M., et al.: Deliverable D5.3 - analysis of requirements for software management. Technical report (2017). https://www.geant.org/Projects/GEANT_Project_GN4/deliverables/D5-3_Analysis-of-Requirements-for-Software-Management.pdf
7. Wolski, M., Labedzki, M., Frankowski, G., Berus, P., Mazurek, C., Adomeit, M.: Delivering software training to the Géant community. GÉANT Connect, pp. 10–11 (2017). https://www.geant.org/News_and_Events/CONNECT/Documents/CONNECT_24.pdf