07-07-2017

Deliverable D5.3
Analysis of Requirements for Software Management

Deliverable D5.3

Contractual Date:	30-04-2017
Actual Date:	07-07-2017
Grant Agreement No.:	731122
Work Package/Activity:	5/SA2
Task Item:	Task 1
Nature of Deliverable:	R
Dissemination Level:	PU
Lead Partner:	PSNC
Document ID:	GN4-2-17-222699
Authors:	M. Wolski (PSNC); R. Martinovic (UoB/AMRES); S. Kupiński (PSNC); M. Łabędzki (PSNC); S. Javaid (LRZ); A. Todosijevic (UoB/AMRES); A. Radulovic (MREN); S. Visconti (GARR); C. Mazurek (PSNC); M. Adomeit (UoB/AMRES); I. Golub (PSNC).

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 731122 (GN4-2).

Abstract

This document presents the current state of the software management practices adopted by GÉANT development teams and highlights the scope for further work in selected areas of software infrastructure and management processes and practices.

Table of Contents

Table of Figures

Table of Tables

Executive Summary

One of the main objectives of SA2 Task 1 – Service Transition and Software management is to support the adoption of consistent software management practices across GÉANT services.

To this end, the team has carried out a thorough analysis of the current software management practices of GÉANT software development teams with the aim of defining a list of activities to manage and harmonise software management requirements for the project. This analysis was focused on the usage of the software development infrastructure to identify commons patterns and trends in the software development processes and practices. Information was gathered through surveys and interviews with the development teams, and the most important points that emerged are explored more in depth in this document.

The preliminary analysis carried out showed that there are about 30 software projects currently being developed by 20 teams in GÉANT. The majority of the development teams (60%) have between one and three developers. Almost 74% of the projects reported their maturity state as either pre-production or production. The analysis revealed that there might be an added benefit from having information about software projects and teams consolidated and presented uniformly in one place.

The first part of the survey that was carried out (Tools), investigated the usage of the various tools and services supporting software development in GÉANT, including GÉANT infrastructure and public cloud services. The analysis showed a high adoption level of basic software development tools. All development teams take advantage of the source code repository (100%), and the majority use the issue tracking system (84%), while the other supporting tools are only used sporadically.

Development teams use both GÉANT and publicly provided services and tools. As there are many benefits to using GÉANT tools, such as enabling to harmonise software development practices, consolidate available documentation etc., further work should be aimed at increasing their uptake where possible. To achieve this, it is necessary to raise awareness of the available tools and their benefits for software development, but also to define the requirements for GÉANT tools to ensure that they fulfil the needs of development teams.

The second part of the survey (Management) concerned several aspects of software development, including methodology, IPR management, documentation practices and federated security mechanisms. Software development methodology is implemented in a more structured way by larger development teams and in a less formalised way by smaller teams. IPR management and software licensing is the main area that needs to be attended to by all software development teams and the proactive support of the IPR coordinator is essential in this respect. The analysis also showed a high level of adoption of basic documentation practices as well as of federated AuthN services.

It is important to maintain momentum through constantly evolving and enhancing software management practices, which should be reflected in an increase in usage of GÉANT development tools, sustainable best practices in IPR management, increasing adoption of software development

methodologies, etc. To achieve these aims, the SA2/T1 team will establish closer collaborations with software development teams and the teams providing GÉANT development tools.

The first action item will be to establish a catalogue of software teams and the projects they are engaged in for the development of various software products or components. Furthermore, periodic monitoring and reporting may be implemented of software management practices. The SA2 T1 team will support these actions by offering the GÉANT community their guidance and expertise in the software management area.

1 Introduction

GÉANT pan-European network for research, education and innovation is Europe's leading collaboration on e-infrastructure. This infrastructure is enhanced and supported by a combined portfolio of various products and services that is constantly being assessed, updated and improved according to the needs of GÉANT users. These users are researchers, students and educators, many of whom are working on innovative and ground breaking projects or developing future technologies, and thus are demanding in terms of requiring high-quality and performant services and infrastructure. All products and services in GÉANT therefore must be developed to the highest levels of quality and security attainable if they are to both fulfil the demands of its users in the R&E community and complement the services provided by member NRENs.

To this end, GÉANT's product portfolio relies on a large number of software projects, which are developed in a distributed environment and involve the effort of many skilled associates coming from all member NRENs. A software project encompasses all elements of the delivery process, including persons and workflows, for a software product or component. More than one project may be working on the delivery of a single product and each team may be involved in more than one project.

One of the objectives of GN4-2 SA2 is to ensure that production services operated in this activity are of acceptable quality (including the software and security aspects) and that they are managed efficiently, with relevant procedures and processes in place and all relevant documentation available. Task 1 - Software Transition and Service Management is partially responsible for fulfilling those objectives by performing secure code and quality code audits of services and tools in transition and in production, as well as by providing software management support and dedicated training for software developers.

The software management (SWM) team in Task 1 was formed to support the adoption of consistent software management practices across GÉANT services. Building upon the experience and outcomes of the previous GÉANT projects, especially the best practices for software management introduced in the GN3 project, the SWM's activities include: software testing and release management infrastructure; support for user feedback; continuous software validation and improvement; software certification; and production of best practices and guidelines.

This document summarizes the work carried out by the SWM team in GN4-2 in the following areas:

- Collecting, analysing and presenting the software management practices that the GÉANT software development teams follow.
- Summarising the usage of the software development infrastructure.
- Identifying common patterns and trends in GÉANT software management practices and recognising their rationale.
- Providing grounds for determining the further scope of work for the team.

To accomplish these goals, the team first gathered requirements for software management across GÉANT services and products by organising surveys and interviews with the development teams. The most important topics were selected from the wide range that were touched upon and explored more in depth. SA2 T1 analysts presented the status and proposed a desired outcome for each selected topics. Each desired outcome was supported through a description of the rationale for the expected improvements. Data gathered from the interviews and surveys are provided in the form of charts or tables to support the interpretation of trends and scenarios.

Providing specific solutions for the identified areas is outside the scope of this document and these will be outlined through further work and cross-activity collaboration at a later stage. However, some examples of actions that could address the issues noted are nevertheless included to give an indication of some of the possible ways in which the expected improvements could be achieved.

1.1 Methodology

The goal of the Software management team was to analyse the available information on the software tools and solutions adopted in the GN4-2 project and to provide a set of recommendations as well as propose future actions for their potential improvement. All data were gathered with the assistance of the leaders of the relevant Activities and development teams.

The sources of information used for this analysis were:

- A preliminary survey defined in order to create identity cards for software development projects and teams, gathering the following information: name and purpose of the project; key actors (service/product manager, lead developer or team lead); and, activity/task in which the projects are developed. The questionnaire was distributed to attendees at the School for Developers (S4D) 2016 that took place in Poznan, Poland on 17-19 October 2016.
- A two-part survey aimed at gathering information about the usage of software development tools (Tools survey) and the software management practices employed (Management survey). The survey consisted of 89 questions and its outline is presented in Appendix B: Software Tools and Management Survey.
- Known source code repositories, including GitHub.
- GÉANT Confluence instances used by some software development projects as a working area and collaboration space;
- Forge site containing partial information about the applications developed in GN3 and GN3plus.
- GÉANT Intranet and public website.

This entire combined effort resulted in identifying around 30 active software projects developed by 20 teams in GÉANT.

The Tools survey was conducted in December 2016-January 2017 and facilitated by NA3. The survey received responses from 19 of 20 software development teams representing 28 software projects (each team may be involved in more than one project). The teams are working in a number of activities in the GN4-2 project, including JRA1, JRA2, JRA3, JRA4 and SA2, and involving several NRENs in the software development effort.

The Management survey, which covered software development processes, testing practices, team roles, on-boarding procedures, etc. was presented to four selected development teams. In all four cases, the survey was completed during a video conference meeting.

All the responses from the aforementioned surveys and interviews were carefully analysed, both statistically and individually. The most important findings, i.e. those that were assessed as having a significant impact on software management practices, are presented in this document.

2 GÉANT Software Development Teams and Products

Information about individual participants in the GN4-2 project and their roles in particular activities and tasks is maintained via the internal COmanage Registry [GN-COmanage]. This type of information management is organised in a delegated manner is designed for virtual collaborations, and very well suited for large collaborations such as the GN4-2 project. Activity and Task Leaders are delegated with the responsibility of managing the information about their task members. Information from this registry is used for generating group mailing lists, making authorisation decisions when GÉANT participants access other services, etc. However, this registry does not contain information that is more granular, such as for example the software development teams in which a specific individual participates. In the GN3 project, the Forge service fulfilled the role of a hub for software efforts [GN-Forge]. At the peak in its use, the Forge contained information on 10 projects. However, the service has now been deprecated to the status of archive.

In-depth information about the project software development teams was gathered in the survey conducted by the SWM team. 30 software projects are being developed by about 20 teams of highly skilled professionals originating from different member NRENs. Information about the size of software development teams and software projects size was gathered to further analyse their and requirements for collaboration tools. This analysis does not present the amount of man effort involved, but rather the number of individuals engaged in each software development team. In addition, information was gathered on the maturity level of software developed in GÉANT, i.e. in pilot, pre-production or production, which is presented below.

Figure 2.1 shows the size of the software development teams according to the number of individuals involved. Most of the teams are rather small: twelve (60%) comprise between one and three developers; five teams (25%) are medium sized and have five or six developers, and the two largest teams (15%) have ten and 25 developers respectively. Looking at this aggregated data, the 12 smallest projects involve 25 people in total, while the remaining seven projects analysed involve 62 people. The two projects that reported having ten and 25 participants respectively are the GTS – GÉANT Testbed Service and the Pilot SDN/NFV Use Cases, the latter of which is a collection of several development sub-projects and exploratory tasks that are currently using ONOS technology.

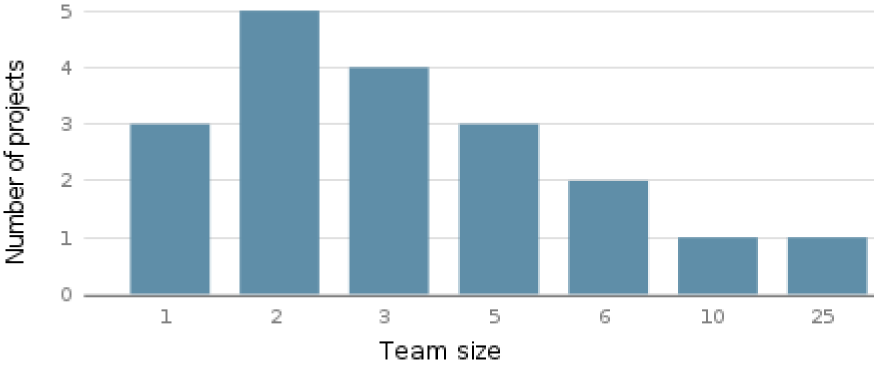

Figure 2.1: Number of software development projects by team size (number of persons)

It should be noted that almost all developers are engaged in GÉANT software projects only part time and many of these are working on several projects and for various teams at the same time. For this reason, calculating the total number of software developers in GÉANT is not a simple matter of summing all reported numbers of engaged persons. Most developers are multitasking and dedicating only a limited amount of time to each task they are entrusted with. If a developer's time is too thinly spread across many different software projects, this may negatively impact team productivity and performance. Mitigation of this risk is addressed as part of project management, usually during the project planning phase and the assignment of manpower.

Figure 2.2 shows the size of GÉANT software projects by a number of source codes lines (LOC – lines of code) or number of source files. Generally speaking the majority of applications (about 75%) are medium sized i.e. have between 5,000 and 50,000 LOC, or between 10 and 100 source files.

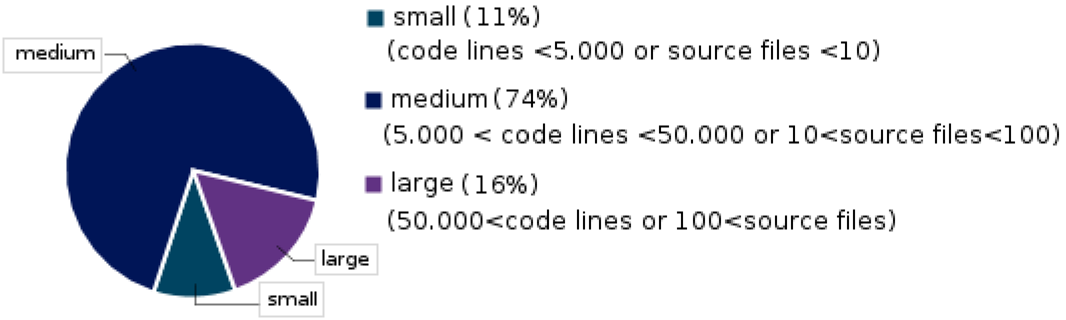

Figure 2.2: Software development projects size

Another aspect investigated in the survey is the maturity of developed software. The collected results indicate that 14 of 19 projects (74%) are either in the pre-production or production lifecycle phase, which shows a high level of maturity overall of software developed in GÉANT. Most of these projects are a continuation of the efforts started in previous GÉANT projects, which could partially explain their high level of maturity. However, GÉANT can also be seen to be investing in innovation, as shown by the five teams (26%) dedicated to new projects currently in pilot and development. Figure 2.3 gives an overall view of the maturity levels of software developed in GÉANT.

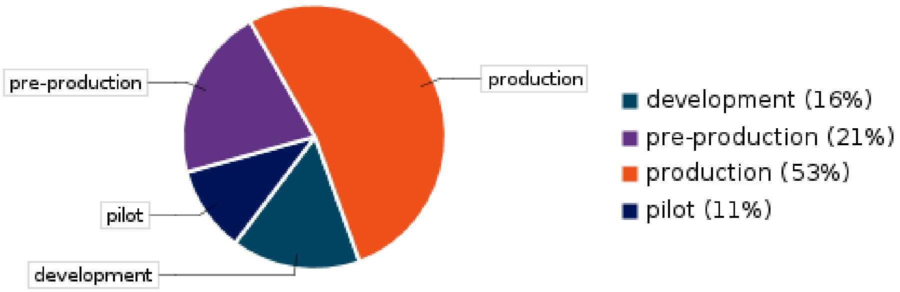

Figure 2.3: Software maturity level

2.1 Recommendations

Many software development teams in the GN4-2 project are working on products that have a high maturity level. The general recommendation is that information about software development teams and projects should be consolidated and presented uniformly and centrally in a single location, so as to simplify the management of the software projects. This could bring a multitude of benefits, including:

- It could provide a source of information about the technologies and programming languages used and thus promote knowledge sharing.
- It would enable developers could easily find other project participants with more experience in a given technology to ask them for advice.
- It would help training teams could plan the content of future workshops more accurately and efficiently, without the need to resort to questionnaires to gather information about what technologies or topics are of interest to GÉANT software developers.
- Auditors running various audits (security, quality code, etc.) could become acquainted with the projects more quickly if the basic information they require could all be accessed from a single location. In the longer term, this could also prove useful for developers, as they would not need to provide the same information each time their project is audited.
- It would facilitate the work of the software management team in SA2 T1 in analysing software management practices.
- Such an information repository could include license and intellectual property rights information, thus making it easier to monitor the state of IPR practices in GÉANT on an ongoing basis, and for IPR coordinators to align their activities within the GN4-2 project.
- Easily accessible information about individuals participating in software development teams would enable better human resource and effort planning.
- Users and stakeholders (either internal or external) may also find it helpful to be able to access software documentation (or related links), information about production instances and download packages, etc. in one place.

SA2 Task 1 therefore proposes that a software development information repository be established containing information about software development teams, licenses, intellectual properties rights, links to relevant documentation, frameworks and programming language used, etc. An example of such a repository is presented in Appendix A, where this information is organised as practical Software ID cards.

3 Software Development Infrastructure

Multiple international software development teams work together in the GN4-2 project to create and enhance software functionalities for services. For developers to be able to work efficiently and effectively within their own teams and with other teams in a multi domain environment, a software development infrastructure that provides the necessary tools and supports collaboration is a necessity. Some common services, such as issue tracker or source code repository, were made available for all software development activities during the GN3 project in 2009. The Software Developer Guide [SwDevBestPractices] developed during GN3 project, provides guidelines and recommendations on how to collaborate within GÉANT and use the tools involved in the development lifecycle. This document was released in 2012 and is still applicable as it provides general guidelines on the use software tools in the development lifecycle.

The work on the software development tools infrastructure in GN4-2 is a continuation of the effort started during these previous projects. GÉANT currently provides a multitude of services and tools that support the whole product development process: from project planning and management, through software development, testing and delivery to user support. In this project phase, software development infrastructure is provided by SA3 T4. Currently 13 software development tools are available to GN4-2 development teams through the following types of software development support services:

- Issue Tracking / Project Management.
- Source Code Management repositories.
- Quality Assurance.
- Binaries Repositories.
- Continuous Integration.
- Identity management.
- Wiki pages.
- Virtual Machines Provisioning Infrastructure.

A complete list of all tools and services provided can be found in Appendix C. Figure 3.1 shows GÉANT software development tools usage, based on the findings of the survey conducted by the SWM team. The most popular tools are the Source Code Management repositories, Issue tracking/Project Management and Quality Assurance services.

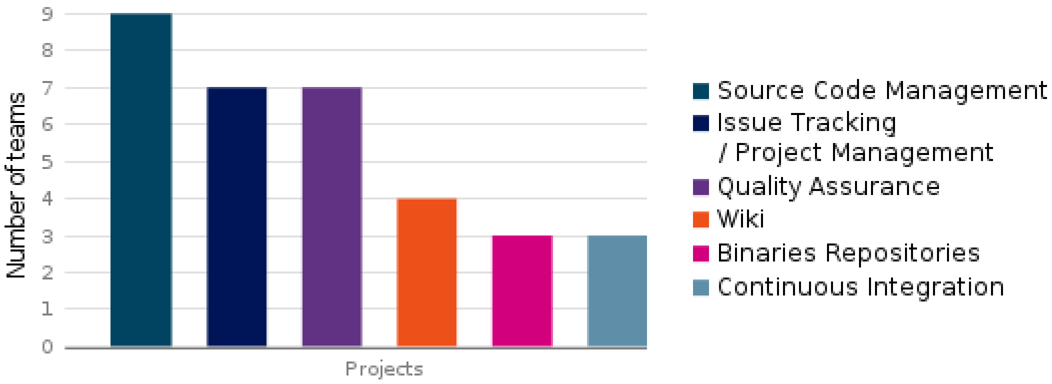

Figure 3.1: GÉANT software development tools usage

Usage of individual tools is further analysed below in this section. Some conclusions and recommendations for a further development roadmap and improvements of those tools are also provided.

3.1 Source Code Repositories

Version control management, also called source code management (SCM), is the process of change management of any type of source code or configuration files. The data store which is used for tracking all changes in the source code is called source code repository. The repository is crucial in the development lifecycle because it makes it possible to develop separate branches of the software simultaneously, which later can be merged and released or deployed as an integral software solution. Since bugs and other issues usually relate to specific versions, it is very important to be able to retrieve, compile and run different versions of the software. The Software Developer Guide [SwDevBestPractices] provides guidelines and recommendations on how to use the source code repositories as collaborative tools.

GÉANT runs two SCM repositories: GÉANT Git, based on Atlassian Bitbucket [Bitbucket], and GÉANT Subversion. Figure 3.2 shows usage of GÉANT and publicly provided SCM repositories based on the results of the survey. These results show that eight of 19 development teams (around 40%) use a public SCM such as GitHub or GitLab services, while another eight take advantage of GÉANT Git or GÉANT Subversion. Two teams utilise an NREN SCM repository and one team uses other SCM repository resources.

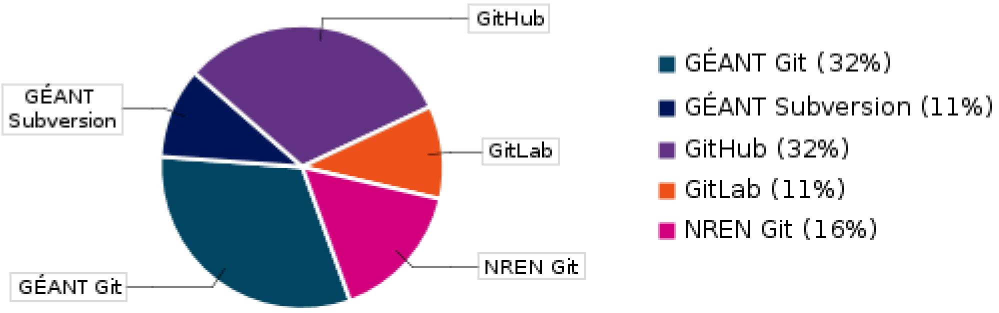

Figure 3.2: Adoption of SCM repositories

SCM repositories are the most often used GÉANT software development services. Since its introduction, the GÉANT Git repository has grown in popularity and is now mostly preferred over the GÉANT Subversion repository. This is likely because Atlassian Bitbucket, which is used in Git, is a more sophisticated tool than Subversion, offering not only basic source code management capabilities but also third party integrations, easier user management, git-flow [GIT-FLOW] and pull requests.

Figure 3.3 shows the usage of GÉANT SCM repositories as recorded in service log files, based on the number of code source syncs with the central repository per month, counting the number of commits for the Subversion repository and the number of code pushes for Git. The usage chart shows that most of the development effort migrated from the Subversion to the Git repository in GÉANT around mid-2016, while total activity on the two GÉANT repositories has remained about the same over the last two years.

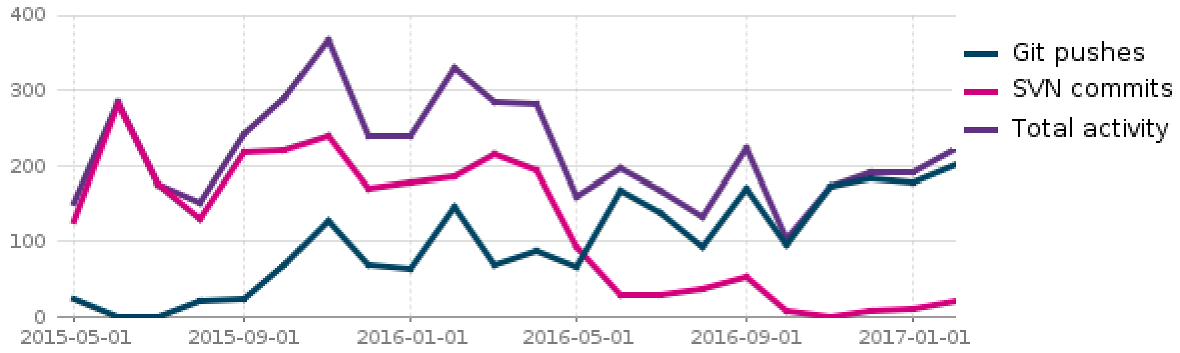

Figure 3.3: GÉANT Subversion and Git source code repositories usage

In the cases where software developers have chosen to use external public services over GÉANT tools, the survey responses did not clarify the reasons for this. One of the possible causes for their decision, which was in some cases highlighted by the development teams during the interviews, could be the wish to invite other members, outside of GÉANT, to collaborate in software projects. However, this would not appear to be a legitimate reason for using public SCM repositories, as this option is not much used in practice, as GÉANT software development teams predominantly share sources on GitHub in Read Only mode, which does not allow contribution to the code. For write access, teams rely on GitHub, and use the SSH-based authorisation method that is also available in the GÉANT Git and Subversion repositories. Other possible reasons that emerged during the interviews might be linked to the developers' habits inherited from their NRENs-specific projects, as well as a limited awareness of the GÉANT SWD tool.

As mentioned earlier, the argument for using public SCM repositories in order to reach larger community outside from GÉANT project participants seems not to be valid reason because of the pre-dominant Read Only policy. In order to decide whether to use a public or GÉANT SCM repository for a certain software project, software teams should determine two important facts beforehand. Firstly, each team considering working on a public SCM repository should be aware of the GÉANT IPR documentation, software license and related policies for importing external libraries into projects. Secondly, software development teams should carefully assess which set of tools is most suitable to their needs and will support their software development lifecycle in the most effective way. The advantage of using the GÉANT Git repository is out of the box integration with other SWD services such as issue tracker, continuous integration and quality assurance. It also simplifies access to source code for other GÉANT project participants, such as auditors who are testing the software during transition into production.

Additional effort should be employed to increase awareness of GÉANT SWD tools to promote their usage, which would benefit both developers and software auditors, and help harmonise GÉANT software development practices. An example of how this could be achieved are short how-to guides for developers about GÉANT Git (code.geant.net) including recipes on how to work with external contributors.

3.2 Issue Tracking Systems

Issue tracking systems are used to coordinate day-to-day software development work. This is one of the most important tools used by developers and software project managers, by the first to provide information about progress on the development of requested issues, and by the latter to create and maintains roadmaps and development plans. Usually, issue-tracking systems are realised as a collaborative platform, enabling the results of all activities to be immediately visible to other users.

One of the SWD tools provided by GÉANT is JIRA, a project management tool designed for Agile teams working in development process management, and includes Agile extensions such as Scrum and/or Kaban methodologies and predefined project templates. Besides its primary use for software development purposes, a Service Desk JIRA Application is also used in GÉANT for service desk and general project management.

Figure 3.4 shows the data collected by the survey on the usage of GÉANT JIRA and other issue tracking systems by GÉANT software development teams. Seven of 20 software development teams use the GÉANT JIRA issue tracker. Of the others, almost all teams that are using public GitHub or GitLab as SCM repository use their capabilities for issue tracking as well. One software development team uses and internal NREN JIRA Instance and one team uses an OTRS ticketing system.

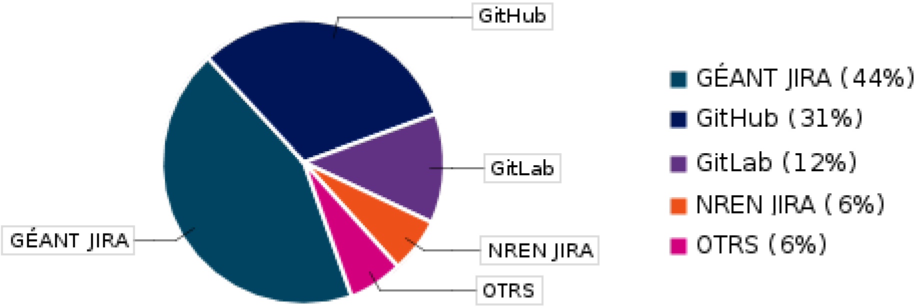

Figure 3.4 Usage of issue tracking systems

Issue tracking is crucial for every software product in GÉANT, not only for coding and testing but in all phases of the development lifecycle. Issue tracking systems are not limited to tracking bugs, but are also used for general issue tracking, providing support via helpdesk, tracking feature requests, improvements and inquiries, etc. These services can improve communication and customer satisfaction, raise software quality and development team productivity, and reduce software release time, and therefore costs [QABestPractices]. In addition, usage of issue tracking services by all development team members aids transparency, which is one of the pillars of the Agile software development framework Scrum.

Use of GÉANT JIRA is recommended because it is part of the GÉANT software development environment and provides integration with other services such as GÉANT Git (Bitbucket), GÉANT CI (Bamboo), GÉANT QA (SonarQube) and the GÉANT Binaries Repository (Artifactory). As such, it can be used to follow changes in software (or bug fix), create change requests to new binaries with modifications in implementations, etc. It also simplifies analysis of software management and development processes.

3.3 Continuous Integration Systems

Continuous integration (CI) systems present aggregated information on the status and working progress of software projects. They can be used to verify software quality, as they can execute test cases and automatically publish statistics of failures and successes, without requiring the developers' attention. Another highly valuable functionality of these systems is static analysis of the source code, which allows for easier discovery of possible problems in code base and automatisation of measurement of source code quality. All time-consuming operations are executed in the background, providing users with lists of warnings and vulnerabilities.

There are two CI systems available in the GÉANT SWD environment: Bamboo (introduced during the GN4-1 project) and JenkinsCI (introduced during the GN3 project). Bamboo is meant to replace JenkinsCI. However, JenkinsCI is still used to archive instances and thus provide access to the build history of projects and result data. All projects but one have been migrated to the new Bamboo CI tool. Six (32%) of the 20 projects surveyed use a continuous integration service for product development. As shown in Figure 3.5, half of these use GÉANT services, one uses a public service, and two use NREN services.

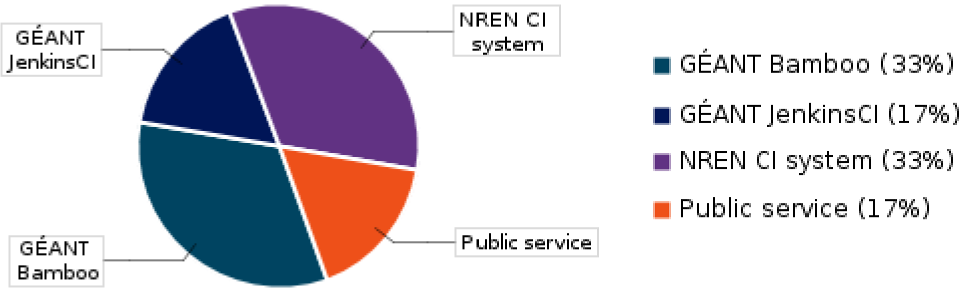

Figure 3.5 Usage of Continuous Integration Systems in GN4-2

Figure 3.6 below shows what CI systems are being used for by software development teams in GN4-2. In 83% of the projects, CI systems are used for performing unit tests and deployment of development and test environments. 66% use them for compiling code sources and 50% for validation of software quality. Only 33% of responders use CI for integration tests and for providing dependencies. Only one project team declared that they use CI for production environment deployment

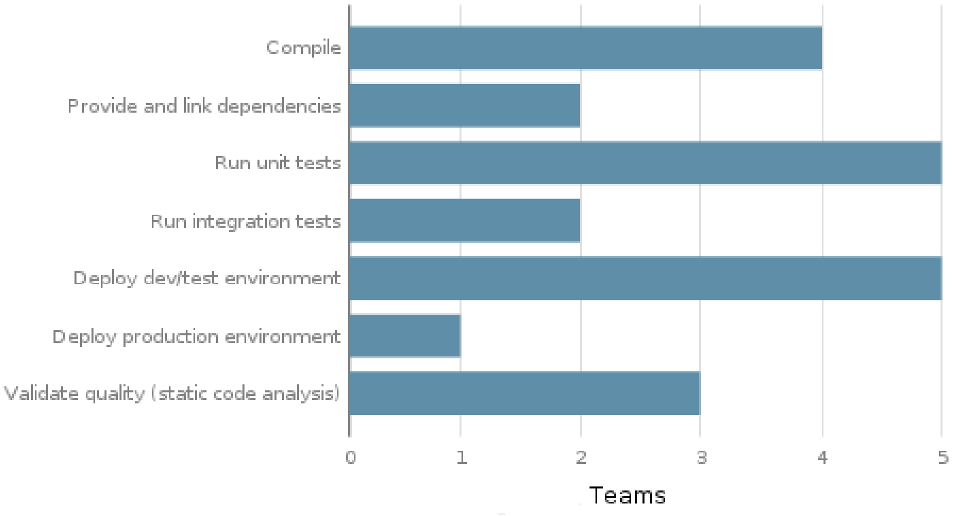

Figure 3.6 Usage strategies for CI tools

Further progress in the adoption of continuous integration services will be beneficial to the quality of software products. As regards the maturity of projects, survey results indicated that 74% of these were either in pre-production or in production. This indicates a high level of maturity of the software developed in GÉANT overall, which should result in high adoption of CI services according to approved recommendations [SwDevBestPractices]. Six projects have declared that they currently use CI and one of these has also introduced DevOps practices for deployment in the production environment. Considerable effort should be devoted to promoting the GÉANT Bamboo service, which could be used together with public SCM services such as GitHub or GitLab, making the appropriate guides and manuals available.

3.4 Recommendations

Software development in GÉANT is supported by a wealth of tools facilitating software planning, management and documentation. As the general approach to software development is evolving, influenced among other factors by the impact of cloud technologies, the social environment, having geographically dispersed teams, the infrastructure and accompanying tools should take into account these changes in the environment. In view of this, the following actions might be considered as next steps:

- The existing toolset should be reviewed against its initial purpose, current usage and future actions, for example regarding federated authentication. These actions may result in tools consolidation (e.g. Git and SVN source code repositories, Wikis) or retirement (e.g. Forge), or deployment of new tools (e.g. private GitLab).

- The GÉANT community needs a clear definition of the purpose of tools and their recommended usage supported by practical use cases and testimonials. This requires an effort to establish a satisfying level of consistency and utility of these tools and for their dissemination, as well as to provide the relevant training. Ideally, GÉANT software development tools should be promoted and introduced in the School For Developers (S4D) training series in GN4-2.

A clear and coherent policy stating conditions of use of GÉANT software development infrastructure versus public cloud services (such as GitHub) should be defined and implemented in practice. This policy should consider various factors, such as project size, maturity, target domain, etc. and recommend a number of options, listing pros and cons of the different approaches. Moreover, the core tools and services containing software artefacts and documentation (e.g. source code repository or issue tracker system), should be based on internal GÉANT software development infrastructure, as this is necessary to enable coherent permission and access management to be used for whole NRENs community (through eduGAIN Authentication Services). Furthermore, the internal infrastructure can be customised to fully support GÉANT processes such as PLM, IPR and others.

4 Software Governance and Development Processes

Software governance comprises a set of structures, processes and policies that regulate software development and deployment within an organisation. Usually, governance defines procedures such as workflows i.e. step-by-step guidelines, which help development teams establish a common working process. While procedures are not so relevant to those working independently, they reveal their value in collaborations, especially in the case of activities that are rarely performed. As proven recipes, they ensure that work is carried out to an accepted level of quality. Several pre-defined procedures already exist in various areas, especially in project management (PRINCE2 or PMBOK being prime examples [PMMCA]). Oftentimes, those procedures become worldwide standards and best practices accepted and verified by numerous adopters in various environments. In particular, software development is an area that is very well suited to the application of standardised procedures. There are multiple proven good practices and efficient methodologies routinely employed with satisfactory results by software development teams. Some of these are Scrum (an agile way of organizing teams and work), Test-driven development (code development approach), Specification by Example (requirements management), User Story Mapping (requirements analysis), etc.

Following well-defined methodologies and processes produces multiple benefits in terms of the efficiency of development team and the quality of products, in particular:

- Greater transparency resulting in less misunderstandings and conflicts.
- Greater efficiency and efficacy as team members better understand each other’s intentions and actions as they are following a single work concept.
- Increased quality as teams are enabled to focus on the product rather than on finding ways to cooperate;
- Increased team satisfaction deriving from efficiency, efficacy and product quality – is higher in teams following a clear definition of "work".

Software governance practices were first introduced in the GÉANT community during the GN3 project and are being successfully continued in GN4-2. The effort dedicated to software governance, throughout subsequent projects introduce an overarching software architecture strategy, incorporated a software process framework for service activity products and a software sustainability plan. Some of these strategies are set out in best practice documents covering software documentation [SDBestPracticesGuide], architecture [SwASBestPractices] and guidelines for using tools within the integrated software development lifecycle [SwDevBestPractices]. Moreover, as part of this effort, software development support infrastructure and tools, such as version control systems, configuration control and management systems, build management systems and wiki pages are set up. Additionally, periodical software audits, including documentation review, source code inspection, testing procedures, security audits and others, are conducted to check that the software development process had followed the defined guidelines.

SA2 is extending this effort with the aim of applying an enterprise-style approach to software development, which is required for the software products operated in SA2 production. The annual training events – School for Developers and Secure Code Training – aim, through collaborative sessions, to strengthen the knowledge and skills of GN4-2 development teams in quality and secure coding by sharing technical knowledge and software development approaches as well as providing hands-on coding experience. The service validation and testing process adopted in the SA2 environment [D8.1], includes examining source code to search for potential bugs, bad code architecture, duplicated code and similar coding irregularities to ensure that only quality-tested products reach production. This process is supported by the software quality measurement framework, which uses recognised standards for system and software quality, supported by evidence from the analysed software management practices [D8.1-support].

This section further investigates selected aspects of software governance and development processes, and presents some of the findings in this area of the survey conducted with GÉANT software development teams.

4.1 Software Development Methodology

A software development methodology is a framework used to structure, plan, and control the process of developing software. When choosing a methodology, proper consideration should be given to team size, especially when considering highly formalised practices such as Agile or Waterfall. Smaller teams may also use looser, self-defined processes, but for larger teams it is recommended to have stricter and well-defined methodologies in place. The choice of methodology should also depend on project characteristics such as clarity of initial requirements, requirements stability, and system complexity, among others [IFFCSDM], [SADA], [SSDMOC].

According to the results of both parts (Tools and Management) of the survey, large software development teams in GN4-2 are using well-defined software development methodologies, in particular Extreme Programming, customised Agile, and other approaches defined in-house. Smaller teams work in a closer collaboration, which allows them to be organised in a less-formalised manner.

The Management survey also showed that Agile methodology is the best known among the interviewed software developers. This was also confirmed during the 2016 School 4 Developers that focused on Agile methodology. Input gathered from the participants implies that GÉANT developers are well versed in Agile practices. These practices describe a set of principles for software development under which requirements and solutions evolve through the collaborative effort of self-organising, cross-functional teams. Therefore, they align well with GÉANT's goals of providing its users with high quality services in an open, transparent and collaborative manner that is quickly responsive to changes in users' needs. Teams that employ Agile practices reported satisfaction with this methodology and the organisation of their work. On the other hand, teams that do not employ Agile practices on average reported that the work processes and organisation for their projects could be improved.

This leads to conclude that further effort should be directed into expanding the usage of Agile methodologies. In particular, the Agile training for GÉANT the software developers' community should continue both to support the further adoption of Agile practices and taking into account the constant movement of developers participating in GÉANT projects. Future training could also specifically address smaller software development teams that could still benefit from selective implementation of Agile methodologies.

Alignment of Agile with ITIL [ITIL] and GÉANT Product Lifecycle Management (PLM) should also be considered. This would help bridge the gap between the managers and developers and connect business strategy/orientation/product features with development goals. It will also help the developers clearly map long-term goals and short-term development roadmaps.

4.2 Intellectual Property Rights

Intellectual property rights (IPR) comprise a vast and complex subject area and need to be carefully managed to reduce risks and protect the project's interests while still respecting the open source philosophy. The Intellectual Property Framework defined for the GN4-2 project enables participants to handle intellectual property in ways that are consistent with GÉANT's principles. It specifies the practices under which GÉANT operates with regard to intellectual property created or developed during the course of a GÉANT Activity.

The Intellectual Property Rights Coordinator in NA1 T2 administers all aspects of IPR that are applicable in the context of the GN4-2 project. Detailed contact information as well as information and guidelines on IPR policies applicable to GN4-2 are made available to the community on the NA1 T2 intranet page, including the following documents:

- IPR Glossary [IPRG]
- Your Guide to IP in Horizon 2020 [YGTIP]
- Fact Sheet - IPR management in software development [FSIPR]
- GÉANT IPR Policy [GN-IPR]
- GÉANT Standard Open Source Software Outward Licence [GSOSSOL]

The GÉANT IPR policy comprises a default policy for software developers and other IPR creators. It indicates that three "green" licenses only (Apache, BSD, MIT) are allowed when incorporating another library or framework into a project.

While the licensing policy for the software developed may not be so important when it is for the purpose of internal deployments, in cases where code-bases are publicly available, licensing needs to be applied that intellectual property is protected. Projects that do not have a defined IPR are vulnerable to the following risks [FTESDSKAIP]:

- The intellectual property developed may not be sufficiently protected.
- The product cannot be properly validated in terms of compatibility with 3rd party products (usually code/libraries and graphics) used for implementation.
- Software cannot be legally distributed and/or used.
- Deployment of the developed products may on the one hand be very limited, while also potentially cause legal issues for both the developer and the deploying site.

Developers often need to incorporate third party code, libraries, products or frameworks into the software they are developing. In that process, it is crucial to consider the IPR licences of the 3rd-party resources used and any applicable GÉANT or other IPR policies wherever possible. However, developers are primarily focused on programming challenges, and in some cases are not able or do not have the competencies to analyse the complex terms of non-green 3rd party licenses and make the necessary IPR-related decisions.

For this reason, developers must maintain close cooperation with the IPR Coordinator. They should also use tools that perform automated check of licenses compatibility, such as the GÉANT Binaries Repository tool provided in the GÉANT SWD environment. While it is important to keep developers updated and raise their awareness of IPR, the IPR burden on development teams should at the same time be reduced by providing them with the appropriate support.

4.3 Documentation Practices

An important part of software development is producing adequate documentation that targets different user and stakeholder groups. A set of best practices for documentation management [SDBestPracticesGuide] were defined during GN3 for use in GÉANT projects and are still applicable in GN4-2. These provide recommendations about writing and managing software documentation, including defining the documentation's purpose (e.g. internal or public facing) and document type (e.g. architecture, design specification etc.). Moreover, it is very important to have proper software documentation in place when a service relying on that software enters the production environment. As defined in the GÉANT PLM, during the transition to production, all of the documentation needed by the users, and by operational and support teams is validated before the service can enter the production stage.

As regards internal documentation, the above-mentioned GÉANT best practices guide advises creating and maintaining the following types of documents throughout the software's development lifecycle: software requirements specifications, architecture / design specifications, and internal technical documentation. The best practices indicate that internal technical documentation can be created using the same tool used for regular source code, so that the programmer can directly refer to his/her code. It is suggested to document the following source code elements: classes and interfaces (author, date, aim), fields, method and all implemented algorithms, and non-standard data structures. This to enable the information to be retrieved and easily published when needed.

As regards documentation to be made publicly available for released software, the GÉANT best practices guide recommends the following types: deliverables, ReadMe, release notes, user guides, admin guides, API Guides, help material, web content, and marketing material.

According to the survey results, GN4-2 software development teams give great attention to documentation. The great majority of software projects maintains the three main types of documentation, i.e. development, user and installation guides. Different locations are used for storing documentation: while some of the teams are using GÉANT repositories to store internal and public documentation, others are also using other locations as this is often mandated by the software management tools used. In that regard, the way software documentation is managed is usually consistent with the tools in use, for example, if a team uses GitHub as an issue management system, it also takes advantage of this tool for storing documentation.

The GÉANT best practices guide recommends that all software documentation should be stored in a single location. For every software product in GÉANT, a list of the available current documentation should be easily available, without the need for any extra effort to search for the most up-to-date version of a document. On the other hand, imposing strict rules for storing the documents in one place is not a viable solution while there are a variety of applications developed in GÉANT, with different target domains, maturity etc.

The approach that this team recommends is implementation of a software information repository, as introduced in Section 2.1, which would also contain the list of existing available documentation attached to the Software ID card.

4.4 Usage of Federated Authentication Services

The services and products that GÉANT provides exist in open multi-domain environment easily accessible from the Internet. At the same time, the data generated by users of these services and products represents valuable intellectual property and the safety and privacy of this data must be handled with the utmost care. To this end, it is very important to enable proper authentication and authorisation mechanisms for all software projects in GN4-2 that require them.

While securing software from unauthorised usage, implementing authentication does not have to hinder users' experience. Federated authentication services exist to enable users to gain easy access to a wide array of services provided by GÉANT and its partner NRENs. Benefits that federated authentication services provide include, but are not limited to single sign-on, simplified authentication compared to "standard" authentication methods, and delegation of management of credentials within a user's home institution.

16 (88.89%) of the 18 projects surveyed use some form of authentication in their products, showing that GÉANT development teams approach the topic of security very seriously. The remaining two projects that are not applying any authentication involve internal tools that have implemented access authorisation in other ways.

As regards the types of authentication applied in software projects, the majority reported using eduGAIN service as the authentication provider. The complete results on this area are presented in Figure 3.5.

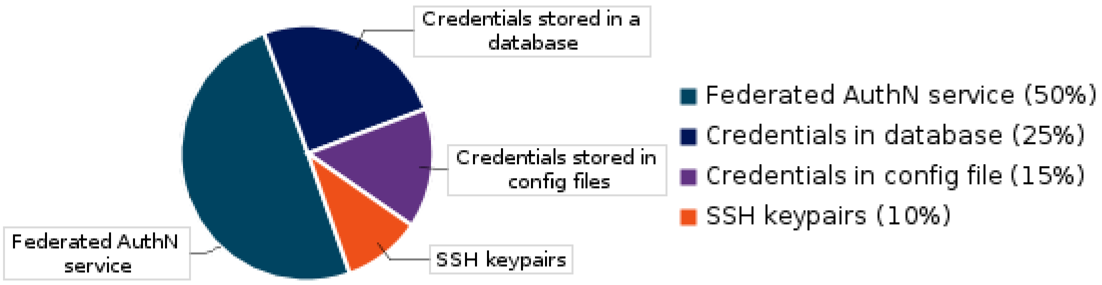

Figure 4.1 Types of authentication applied in software projects

GN4-2 development teams show exemplary practices in securing the applications produced. Very high levels of adoption of federated authentication services in software projects, especially eduGAIN as an authentication mechanism, clearly indicate that GÉANT teams are providing good users experiences, while at the same time lowering development and production costs by using proven and reliable authentication mechanisms.

Further progress in the adoption of authN/authZ mechanisms can be monitored through security audits, which are the part of the transition to production process performed by SA2 T1. Such audits are meant to validate software security levels, identify any existing vulnerabilities and propose concrete recommendations for improvements for software products, including fixes and remedies for any detected bugs and failures.

5 Conclusions and Future Work

Organising and leading software development in GN4-2 can be challenging as developers from different NRENs that might have varying work methodologies in terms of style, experience, expectations and approach. In addition, all NRENs have highly developed advanced networks and systems and must be responsive to demanding and knowledgeable stakeholders.

Since GÉANT is a large multi-domain and international collaboration, with many individuals contributing from different organisations, it has been essential to establish a common approach to software development, and which has been in practice since the 2010 (GN3 project). This work started with the adoption of best practices, followed by the deployment of software development tools and infrastructure, and finally culminated in facilitating training events targeting the software community – "Software School 4 developers" and "Secure code trainings".

Awareness of the importance of software quality as well as responsibility for the software maintained in production has increased, and has resulted in the establishment of a service validation and testing process for SA2 production services in the GN4-2 project.

The work carried out in GÉANT involves collaborations of peers from various countries around Europe and even the world. This means that the quality of products and services that GÉANT offers to its customers not only depends on the development methodology, the practices in use and available skills, but also on the quality of the tools supporting software development, the level of integration of these tools and the help and assistance that developers are able to receive from GÉANT. In a multi-domain environment, the challenges involved in the development of services and tools are similar to those encountered for service transition [MDServiceTransitionChallenges].

The majority of development projects in GN4-2 are in the pre-productions or production stage and present a high level of maturity in terms of documentation management, SWD tools usage and implementation of various types of authentication. Future work opportunities are identified in the harmonisation of software development practices across GN4-2 activities, in the pursuit of even greater efficiency and overall quality. In addition, areas for future development are noted in IPR management, the creation of best practices and guidance documents and in providing expertise to the community.

A possible starting point in the area of harmonisation could be to establish a common software information repository for different stakeholders, including team leaders, product managers, training coordinators, IPR coordinators and others. This would be implemented horizontally across GN4-2 (in contrast to the vertical structure of activities/tasks) and could bring together information on software developers, their products and supporting resources under one hood.

Particularly, the analysis carried out found that having such a common location for software information may act as:

- A coherent and up-to-date repository of software products (ID-cards) that holds the most important information for each product such as contacts, licenses and IPR, links to documentation, development methodologies used etc.

- A consistent and lightweight point of reference to kick-off a new software project, which among other things could create a software ID card and help submit requests for any required GÉANT software tools.

This repository should be easily accessible to any GÉANT project participant and should also include information on past and retired projects. It should contain information on all software being developed, no matter how small the allocated effort. A proposal on how information could be presented is given in Appendix A, which provides an example of the desired functions that should be further elaborated before implementation.

A second area of work towards the harmonisation of software development practices in GN4-2 is the application of a consistent approach for using software development tools. The survey analysis showed that teams are currently using either GÉANT tools or public services such as GitHub for software development. With regard to the harmonisation of those practices, it is necessary to:

- Consider best practices and recommendations for using software development tools and, if appropriate, establish a process that promotes preferential use of GÉANT tools over public tools (for the sake of coherent security policy and customisation options).
- Define guidelines, manuals and recommendations for the usage of GÉANT software development tools.
- Propose a roadmap for the consolidation and harmonisation of current GÉANT software development tools, including proposals for new tools, alternatives to existing ones and retirement of rarely used tools.

An ongoing effort must be made to present and promote tool adoption strategies with reference to the different categories applying to software products (e.g. maturity, team size, target environment). A roadmap for GN4-2 software development tools should be defined and adopted based on available effort.

Regarding the IPR management aspect, as the corresponding processes are already defined and in place, the next step might be to better integrate these into software development and management, and enhance the competencies of development teams in IPR management.

Finally, another area that would benefit from harmonisation of software practices is that of software development methodologies. The analysis carried out identified a need to increase awareness among development teams of software development methodologies, and of how tools support and can be an integral part of the software development lifecycle. An example of such actions would be the showcases presenting the benefits of software development tools, i.e. what development teams could achieve by using the products, how the tools make it easier to complete routine tasks, etc.

Some of the areas identified above fall under the domain of SA2 T1, while others should be anchored in other activities. For the selected software management aspects, such as the periodic analysis of SWD tools usage, joint actions with SA3 might be undertaken in the areas of tools set harmonisation, IPR management, adoption of software ID cards, etc.

Furthermore, the SA2 T1 team could offer the GN4-2 software development community its guidance and expertise through consultancy services complementing the training that is already provided for software developers. Possible consultancy services might cover areas such as the introduction of best practices in the software development lifecycle, advice on usage of appropriate tools and guidance on software quality and operational and transitional management.

Appendix A Software ID Cards

Software "ID cards" should help collect and maintain key information on software projects in GÉANT and help further analysis of various aspects of software development. What follows are examples of the proposed Software ID Cards for all GÉANT software projects.

A.1 Administrative ID Card

The SWM team proposes the adoption of an "Administrative ID card" containing only basic information, such as project name, purpose, activity and task in GN4-2, and key persons. The SWM team used almost identical "ID cards" such as that described to conducting the survey in preparation for this document. This means that almost all of the data has already been collected, albeit in a slightly different form adjusted for carrying out the analysis presented in this document and currently located in internal repositories of the SWM team. The ID Card includes the type of information that is considered most important and most easily collected. By way of example, the ID Card below shows the information gathered for the psUI project.

Name	Value	Description
Software/Team Name	psUI - perfSONAR UI	Usually the name of the software
Description	Web application that uses perfSONAR toolkit to make network performance measurements in a multi-domain environment.	What is the purpose of the software, usually one marketing oriented sentence
Task	SA2 T3	
Activity Leader	Marina Adomeit, AMRES, marina@amres.ac.rs	Name, NREN (if applicable) and email
Task Leader	Ivana Golub, CARNet, Ivana.Golub@CARNet.hr	Name, NREN (if applicable) and email
Manager	Antoine Delvaux, PSNC, antoine.delvaux@man.poznan.pl	Software or product manager's name and email
Team/Tech Lead	Rade Martinović, AMRES/UoB, rade@rcub.bg.ac.rs	Team or tech lead name and email
Start Date / Expected Completion Date	in 2009 / ongoing project	Important dates can be added
Team size	3	Number of people producing the software, including managers
License Type	GÉANT Standard Open Source Software Outward Licence	The adopted license of the project

Latest release/version	13. July 2016, 1.7.1	
Additional comments		Anything of note that should be mentioned, but doesn’t fall under any of the categories above

Table A.1: Example of Software Administrative ID card

A.2 Technical ID Card

The proposed "Technical ID Card" contains more technical information, such are source code location, the location of different types of documentation, programming languages used, libraries and similar. By way of example, the Technical ID Card below shows the information gathered for the psUI project.

Name	Value	Description
Software/Team Name	psUI - perfSONAR UI	Usually the name of the software
Programming languages used	Java, HTML, CSS	Programming languages
Frameworks/libraries/technologies used (list the most important ones)	Google Widget Toolkit, OpenJPA, Maven, Derby Database, RPM package manager, DEB package manager	Tools and libraries used in the project
Source code	svn.geant.net/fisheye/browse/SA2T3-ps-visualisation-tools	
Installation documentation	http://docs.perfsonar.net/install_psui.html	
User documentation	http://docs.perfsonar.net/using_psui.html	
Architecture guide	/	

Table A.2: Example of Software Technical ID card

Appendix B Software Tools and Management Surveys

In order for the team to perform the analysis of software management requirements, it was essential to gather information about the usage of software development tools and existing software management practices of GÉANT software development teams. For this purpose, a two-part survey was developed. The outlines of these two surveys are presented in this appendix below.

Outline of the first part of the survey “Software Development Tools” (“Tools” survey):

1. Basic Contact Information
2. Basic Software Information
3. Source Code Version Control and Repository
4. Issue/bug tracking
5. Licensing, license tools and licensing practices
6. Build tools
7. Continuous integration
8. Documentation repository
9. Binary repo
10. Collaboration tools
11. Project management tools
12. Text editors and integrated development environments
13. Authentication and Federated AuthN Services
14. Virtual machines and other tools

Outline of the second part of the survey “Software Process and People Management” (” Management” survey):

1. Software development methodology
2. Issue/bug tracking practices
3. Testing practices
4. Source code management practices
5. Continuous integration practices
6. Release management
7. Support management
8. Solving non-development issues
9. Documentation

10. On-boarding procedures
11. Team Roles
12. Investment in training and development of human resource

Appendix C Software Development Infrastructure

A list of the tools that constitute the Software Development Infrastructure provided by SA3 T4 is given below:

- Tools for Issue Tracking / Project Management:
  - GÉANT Jira Development Service https://issues.geant.net/JIRA
- Source Code Management repositories:
  - GÉANT Subversion Development Service http://svn.geant.net (ssh keypair authentication)
  - GÉANT Bitbucket Development Service - Git repository https://code.geant.net (GNAD authentication)
- Tools for Quality Assurance:
  - GÉANT SonarQube Development Service https://ci.geant.net/sonar/
  - GÉANT FishEye&Crucible Development Service https://svn.geant.net/fisheye/
- Binary Repositories provided:
  - GÉANT Artifactory Development Service https://artifactory.geant.net/artifactory
  - GÉANT Downloads Development Service http://downloads.geant.net/
- Continuous Integration services:
  - GÉANT Bamboo Development Service https://ci.geant.net/bamboo (production)
  - GÉANT Jenkins Development Service https://ci-archive.geant.net/jenkins (archive)
- Identity management services:
  - GÉANT Crowd Development Service – https://crowd.geant.net/ proxy to GÉANT Active Directory
- Wiki pages:
  - GÉANT Confluence Development Service https://confluence.geant.net
  - GÉANT Forge Development Service https://forge.geant.net (archive projects pages since GN3)
- Virtual Machines Provisioning Infrastructure:
  - GÉANT QA Testbed Service (managed in JIRA: https://issues.geant.net/jira/projects/QATB based on VMWare Sphere solution)

References

[Bitbucket] https://www.atlassian.com/software/bitbucket

[D8.1] M. Wolski (PSNC), I. Golub (CARNet), G. Frankowski (PSNC), A. Radulovic (MREN), P. Berus (PSNC), S. Medard (RENATER), S. Kupinski (PSNC), T. Appel (DFN/LRZ),T. Nowak (PSNC), S. Visconti (GARR), I. Smud (CARNet), B. Mazar (CARNet), B. Marovic (AMRES), P. Prominski (PSNC) ”Deliverable D8.1 Service Validation and Testing Process”

[D8.1-support] Marcin Wolski (PSNC), Bartosz Walter (PSNC), Szymon Kupiński (PSNC), Patryk Promińsk (PSNC), Ivana Golub (CARNet) : GN4-1 White Paper: Supporting the Service Validation and Testing Process in the GÉANT Project, Doc ID: GN4-1-16-106bc68

[FSIPR] Fact Sheet - IPR management in software development, European IPR Helpdesk,https://intranet.geant.org/gn4/2/Activities/NA1/T2/IPR/PublishingImages/SitePages/Home/Fact-Sheet-IPR-Management-in-Software-Development.pdf

[FTESDSKAIP] Five Things Every Software Developer Should Know About Intellectual Property, http://saperlaw.com/2007/09/27/five-things-every-software-developer-should-know/

[GIT-FLOW] https://danielkummer.github.io/git-flow-cheatsheet/

[GN-COmanage] GÉANT Membership management tool

[GN-Forge] https://forge.geant.nethttps://forge.geant.net/

[GN-IPR] Dai Davies (DANTE), Jana Zmatlikova (CESNET), Anand Patil (DANTE), "GÉANT IPR Policy v1.2,", Document Code: GN3-10-325

[GSOSSOL] GÉANT Standard Open Source Software Outward Licence, https://intranet.geant.org/gn4/2/Activities/NA1/T4/IPR/Shared%20Documents/GEANT_Standard_Open_Source_Software_Outward_Licence.txt

[IFFCSDM] Influence Factors For The Choice Of A Software Development Methodology, ftp://ftp.repec.org/opt/ReDIF/RePEc/ami/articles/10_4_3.pdf

[IPRG] IPR Glossary, https://intranet.geant.org/gn4/2/Activities/NA1/T4/IPR/Pages/IPR-Glossary.aspx

[ITIL] ITIL Wiki, http://wiki.en.it-processmaps.com/index.php/Main_Page

[MDServiceTransitionChallenges] I. Golub, B. Radojević, "Challenges of a service transition in multi domain environment", Mipro, Opatija, 2016.

[PMMCA] Project Management Methodologies: A Comparative Analysis, http://cibw117.com/journal/index.php/performance-info-and-value/article/viewFile/75/73

[QABestPractices] GN3 Quality Assurance Best Practice Guide 4.0 http://geant3.archive.geant.net/Media_Centre/Media_Library/Media%20Library/GN3-09-184_Quality_Assurance_Best_Practice_Guide_4%200.pdf

[SADA] Selecting A Development Approach, https://www.cms.gov/research-statistics-data-and-systems/cms-information-technology/xlc/downloads/selectingdevelopmentapproach.pdf

[SDBestPracticesGuide] GN3 Software Documentation Best Practice Guide, http://geant3.archive.geant.net/Media_Centre/Media_Library/Media%20Library/GN3-12 330_Software_Documentation_Best_Practice_Guide_2.0.pdf

[SSDMOC] Selecting a Software Development Methodology based on Organizational Characteristics, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.614.9644&rep=rep1&type=pdf

[SwASBestPractices] GN3 Software Architecture Strategy Best Practice Guide - http://geant3.archive.geant.net/Media_Centre/Media_Library/Media%20Library/GN3-09-185_Software_Architecture_Strategy_Best_Practice_Guide_4%200.pdf

[SwDevBestPractices] GN3 Software Developer Best Practice Guide - http://geant3.archive.geant.net/Media_Centre/Media_Library/Media%20Library/GN3-09-186_Software_Developer_Best_Practices_Guide_4%200_with_knowledge_share.pdf

[YGTIP] Your Guide to IP in Horizon 2020, European IPR Helpdesk, https://intranet.geant.org/gn4/2/Activities/NA1/T2/IPR/PublishingImages/SitePages/Home/EU_IPR_IP-Guide.pdf

Glossary

GN4-2	GÉANT project iteration GN4-2
NREN	National Research and Education Network
SA	Service Activity in GÉANT project
SA2	GÉANT project activity that provides operations and delivery of Trust and Identity and Multi-Domain services
SWM	Software Management
SWD	Software Development
SWDI	Software Development Infrastructure
SCM	Source Code Management
SVN	Apache Subversion, a software versioning and revision control system
Git	Git version control system
IP	Intellectual Property
IPR	Intellectual Property Rights
ONOS	Open Network Operating System - project providing the control plane for a software-defined network, managing network components, such as switches and links, and running software programs or modules to provide *communication services* to end hosts and neighbouring networks.
S4D	School For Developers, annual training for software developers organized by SA2 with topics covering wide areas of software development
SCT	Secure Code Training, annual training for software developers organized by SA2 covering security aspects of programming