Failure and success - how to move toward successful software development in Networking

Vilmos Bilicki
University of Szeged / NIIF, Szeged, bilickiv@inf.u-szeged.hu

Ivana Golub
Croatian Academic and Research Network - CARNet, Josipa Marohnica 5, 10000 Zagreb, Croatia, Ivana.Golub@CARNet.hr

Pavle Vuletic
Serbian Research and Education Network - AMRES, Kumanovska 7, 11000 Belgrade, Serbia, pavle.vuletic@rcub.bg.ac.rs

Marcin Wolski
Poznan Supercomputing and Networking Center, ul. Noskowskiego 12/14, 61-704 Poznań, marcin.wolski@man.poznan.pl

Abstract
This paper presents the key factors driving successful software development in NRENs and reasons that may cause defective software production. This article is focused on the characteristic of the software development in a multi-domain environment. Basing on lessons learnt from GÉANTX projects, we present several good practices in software development that lead to valuable software product.

Keywords
software requirements, software development, network management, NREN, multi-domain environment

1. Introduction
In the last decade the main software development activities in the network management domain were typical for the OSS (Operational Support Systems) arena. Network elements were managed through the command line interface or network management systems developed by the equipment vendors, and other components communicated with them through the northbound interfaces. Network managers were rarely involved in the development of the management solutions.

Recent appearance of new open network and service management styles and methods like Software Defined Networking (SDN)¹, Network Function Virtualization (NFV)² or Interface to routing system (i2rs)³ require complete rethinking of the management procedures in the existing networks. Although at this moment it is not realistic to expect complete replacement of legacy networks with those based on SDN, NFV or similar technologies, it is likely that NRENs will have to adopt them at least in some parts and for certain functions.

Such a change brings new light to the software development, and emphasizes the importance for quality of the products and the process itself. It will also affect the way the NRENs operate and maintain the software, taking into account closer collaboration between software development teams and operation teams.

Software development for and in one organization brings its own requirements on the processes and products. Software development aimed at more than one organization, either customized for a set of similar organizations, or aimed at the general market (commercial or open-source), or as multi-domain solutions bring even stronger requirements, starting from the product design phase, over the development phase until the production phase.

The paper presents lessons learned from GÉANTX projects towards successful software development in networking.

2. Software development challenges in multi-domain environment
The world of national research and education networks (NREN) is brought together by GÉANT projects into one big multi-domain community. They all keep individual organizational view on what is useful, needed, desirable etc., and have one additional characteristic: cooperation of individual entities (organizations as well as related users) that share common interest and are open to collaboration and joint development of new solutions that will

¹ https://www.opennetworking.org/index.php
² http://www.etsi.org/technologies-clusters/technologies/nfv
³ http://datatracker.ietf.org/wg/i2rs/charter/
The multi-domain world of NRENs poses additional challenges when organizing and leading the software development (SWD), since NRENs can have different style, experience, expectations and approach to SWD. In addition, all NRENs have demanding and knowledgeable users, maintainers and owners, and highly developed advanced networks and systems. These challenges are often addressed by deploying various types of software that aim to manage a selected part of the network domain (network inventory, network monitoring, dynamic circuit provisioning etc.) [1]. Moreover, the off-the-shelf software is usually mixed with legacy and in-house solutions developed by the NOC teams.

The in-house developed software in NRENs is still a relative minority comparing to the commercial or open-source solutions. For the majority of management functionalities in-house developed solutions are used between in 2% and 23% cases [1] which make NRENs relatively small software producers which usually aim to solve some operational issues with quick development rather than to have a holistic approach to network management automation. Some NRENs even do not have a software department which would develop service supporting solutions, but the development is done by the knowledgeable and enthusiastic network engineers trying to solve their own everyday problems. The diversity of tools, communication models, application-specific data schemas used for network management alone causes many problems with managing, updating and evolving these complex environments, and presents even a bigger challenge for NRENs which are typically not organized to be software producing organizations, and often do not have capacities to procure provider-grade OSS solutions. Apart from that, recent analysis of the current status in the field of networking tools and systems shows some common and also distant views on solutions in network monitoring and management. In addition, dynamic expansion of computer networks and the increasing level of their complexity pose new challenges for network management procedures. More and more network services are being introduced. At the same time a growing need of having a supporting infrastructure to provide efficient data transport capabilities delivered on the users' service requirements is observed.

In the multi-domain world, solutions for the aforementioned challenges can be sought individually by an NREN itself, or jointly through cooperation of more NRENs. They take part in multi-site projects when a number of NRENs work together to achieve a specific goal. The most recognized project is GÉANT comprising all NRENs in Europe.

Although participation in a multi-domain community gives significant benefits for every NREN, such an environment has more complex operational procedures than single NRENs due to participation of the domains with various policies and procedures. Multi-domain service environments similar to multi-NREN environments are not common today, so there are no clear recommendations or standard specifications and there is not even common understanding on how to organize various activities for handling multi-domain service operations, which makes the development of the supporting tools even more challenging. The complexity of the environment and service operations often pushed NRENs to work from the first day of the multi-domain service design towards the development of the supporting services. These efforts resulted in development of the tools like: AMPS, AutoBAHN, perfSONAR, cNIS[2] and similar.

3. Software development– GÉANT case study

Participation in previous GÉANTX projects brings several points that, if considered in everyday work, can help towards successful software development. These are the following:

1. thorough software and business requirements gathering and management,
2. clear user definition and constant users' participation during the product lifecycle,
3. appropriately chosen development methodology.

Those three aspects will be covered in more detail in the next sections.

3.1 Software and business requirements management

The business requirements phase is intended to place software into a wider perspective determining the value which the software introduces to end users. Business requirements management means defining a consistent framework for the proper requirements definition, benchmarking them and applying into the SWD process. The framework, which shaped the way of creating and managing the dependency between the business requirements and software requirements. The need for such a framework is not always visible from the beginning of the software project. Usually it becomes crucial when a software product is matured and introduced to end users. In practice, the business requirements analysis is often reduced or even omitted. The decision about kicking off the software development effort can then be driven by the technology innovation, instead of the actual business need. Research activities tend to place a technology as leading criteria. In those cases, finding a proper balance between a technology innovation and business goals can be a challenge.

---

For example, GÉANT GEMBus is a standardised SOA platform built upon an existing Enterprise Service Bus (ESB) to address the communication and integration of heterogeneous services [3]. GEMBus is a result of the GN3 Research Activity. The industry specifications identify ESB as a central part of the service-oriented application ecosystem. There is a potential risk that the overhead and side-effects of using the BUS will surpass its benefits. On the other hand, it can be seen as a long-term investment and benefits might be seen with time. Consequently, the ongoing pilot aims to evaluate GEMBus usage towards the goal to ease the integration of heterogeneous services.

When a solution is designed for more than one organization, it might be challenging to consider all relevant processes and business segments individually for each targeted organization, especially if all potential organizations are not known in the design phase. The world of standards helps to answer this challenge. TM Forum Business Process Framework⁵ can be taken as a relevant baseline in the domain of business processes. The software development can be aligned with standardised business processes definitions, as well as the business processes of any individual organization. Thus the potential risk of omitting an important aspect of some internal procedure can be minimized.

To illustrate the above scenario GÉANT ISHARE [6] case can be presented. ISHARE aimed to enable seamless delivery of multi-domain E2E link services and provision of a consistent operational support system across domains. The ISHARE users and stakeholders were active during the design phase which resulted in the delivery of the first version of ISHARE. Valuable feedbacks from the user community were collected during the pilot phase which comprises several NRENs. However, it turned out that some of these requests require re-thinking of the design. As a result, ISHARE business goals were evaluated and ISHARE was mapped as an integral part of the selected standardised network management process (eTOM).

All in all, to handle the business requirements management appropriately one can support it by a cohesive set of best practices inherited from existing frameworks. The example of such a framework is ITIL⁶. Typically, only a subset of practices is chosen and accommodated carefully to the specific conditions – Geant Product Management Lifecycle⁷.

### 3.2 Users definition and support

Another significant aspect to the SWD is end users. They play a target audience for the software product. Proper attention should be put on the users’ definition and benefits they gain from the software. It has to be clearly identified how the software solves and which problems users have in a more efficient way than their current practice.

It is important to emphasize that the service user is not always the same as the user of the OSS component supporting that service. Taking the GÉANT multi-domain connectivity service as an example [4]. Even though the service is intended to serve end user researchers or members of the higher education community, actual users of network connectivity services are usually perceived to be network managers, mainly at the level of campuses. In practice, however, a user of this service is anyone who requests it (e.g. network operators at other levels, such as regional network managers) from the service area. The above assumptions led to the definition of the service access hierarchy which then implied the definition of users for GÉANT BoD service and the supporting service tool – AutoBAHN⁸.

Furthermore, users have to be present and actively involved throughout the whole SWD project lifecycle. They should take active part in the SWD evaluating the incremental results, deliver continuously new requirements and provide details for existing ones. Otherwise the project results might not fit the users expectations. It is significantly important in a diverse environment which evolves fast incorporating innovative technologies. Software projects which are managed according to the waterfall model paradigms [8] concentrates the users activity on the selected phases only. Yet user requirements usually evolve in time and it is highly important to engage them in a more systematic way.

To explore this issue further GÉANT cNIS case can be presented. cNIS is a unified repository of all relevant network information about a single administrative domain[5]. After finishing the technological pilot, cNIS evolved into a powerful network topology engine with the ability to manage SDH, ETH, IP layers via configurable drivers. However, delivering the tool with a wealth of functionalities to NRENs and getting initial feedback resulted in a growing set of requirements for new features and improvements. We found proper management of these requirements more and more problematic, as the requirements sometimes represented contradictory interests. Eventually the development of cNIS tool was divided into two parallel tracks – the production one shaped to the GÉANT BoD service and the advancement one as a standard-based Resource

---

⁵ [http://www.tmforum.org](http://www.tmforum.org)
⁷ [http://plm.geant.net/](http://plm.geant.net/)
⁸ [http://services.GÉANT.net/bod/about_BoD/Pages/Who_benefits_from_BoD.aspx](http://services.GÉANT.net/bod/about_BoD/Pages/Who_benefits_from_BoD.aspx)
Inventory Management tool [7]. To sum it up, in a dynamically changed environment, especially when it is led by innovation, there should be a periodical review of the main goals of the SWD project. The initial assumptions and requirements for a long-running project should be periodically checked, evaluated and changed if necessary; otherwise it is likely they become obsolete.

3.3 Software development methodology

The common and long established approach for SWD is based on procedures, detailed structuring and modelling as well as documenting. It is called the process-oriented method. NRENs have adopted some of these principles when it comes to software validation and testing, change management, release management and others. The process-oriented approach can be seen as opposite to the agile (informal) development which is more people-oriented [10]. Agile development relies on informal interactions, lightweight processes and cohesive, well-motivated and well-skilled teams. The agile approach is considered as the most promising one for achieving the results in a timely manner with the ability to adapt to constantly changed business and software requirements.

Although the NRENs community is well prepared for that type of working and the teams have proven their ability to finish successfully the multi-site projects, there are still some areas for improvements. These are, for example, the distributed requirements management (propagating and changing the requirements), communication issues and application knowledge management [9]. Moreover, making the software teams fully agile means in most cases changing the developers’ routines, behaviours, focus and roles. Usually it indicates forgetting about the waterfall model (specifications, analysis, coding, testing, etc.) and concentrating on user stories, talks, discussions, needs and incremental release cycle. The companies and teams that had passed this process underlined how difficult and long the process might be, and how important it is from the beginning to have proper support and coaching [10].

The next framework that one can see as promising for NRENs is the OSS (Open source) practices. These practices offer broad and mature knowledge of building and sustaining the communities working on software in a collaborative manner. The open source development is similar to the IETF word of networking where people believe in the running code. It rarely follows the traditional waterfall software development methodology. It focuses on iterative software development trying to capture the attention of the community. As Abramson [10] describes it: “if the introduced prototype gathers enough attention, it will gradually start to attract more and more developers”. Beside the iterative model most of the opens source software development project follow the “Release Early, Release Often” approach defining short development cycles. With this approach they can decrease the burden of the integration and building which is a pain point of the waterfall approach. The short documentation is also a key feature of the early releases. The NREN community applying a similar methodology can:

- win early feedback,
- build confidence,
- show the project activity,
- facilitate manageable upgrades.

In the NREN community knowledge management is a critical point. The question is how to share the knowledge of the system administrator with the programmer. A formal method for this was described in the previous section but real interaction should be motivated by the “Release Often, Release Early” approach.

The human aspect of the open source development (as described in article [12]) should be carefully studied as the NREN community is a fixed one and there is a project-based hierarchy of participants. For example, the Meriocratic principle of the OSS development where contributors are judged transparently and this is done on public mail lists could be adapted only with the help of project-specific management disciplines. As a summary, the NREN community should adapt to the hybrid version of the OSS development: the technical part of software development principles are good examples which should be considered, while the project management part of the OSS software development should be strictly tailored to the actual project/community.

To conclude this section, properly chosen and implemented software development methodology is an important factor, but the key to deliver high quality software are people who develop it, form well-motivated teams, have appropriate basic skills, and know what to do and how. Procedures, detailed structuring and modelling, and documenting constitute valuable assets but cannot be placed over people.

Although not deeply covered in this document, software governance that comprises several activities related with the software development effort such as delivering software development infrastructure, quality source code audit and software design best practices has to be considered and carried out in relevant phases in the SWD process [13].

4. Conclusion

The paper presents the authors’ view on good practices in software development based on the experience in
multi-domain software development projects under the GÉANT umbrella. Solid requirements definition, constant users’ involvement, as well as proper development mechanisms are seen as very important aspects that can help towards successful software development. We found the blended approach for software development, still relying on the basic processes and trying to find proper balance between people- and process-oriented control, together with hybrid OSS practices that can be used as guidelines for NREN-based development in the future.

Having in mind the broad diversity and variety in the multi-domain world of European NRENs, the role of standards has been recognized as very important in order to be able to systematically cover and take into consideration different options of the internal organizations and business processes. Hence, a new tool, product or system can fit best and be a candidate for a long-lasting, useful and used solution.

References


Biographies
Dr. Vilmos Bilicki is an assistant professor at the Department of Software Engineering at the University of Szeged. His main research topics are distributed storages, compression of the SIP protocol, discovery of the distributed hash table based botnets, and the scalability issues of the distributed hash table based small degree P2P networks. He is the lecturer of different programming and networking related subjects. He was the technological leader of more than 20 R&D projects among them different FP6 and FP7 projects. He is a founding member of the Intelligent Medical Systems section of the Hungarian Scientific Association for
Infocommunications.

Ivana Golub is a Deputy CEO at CARNet, in charge of the Network Infrastructure Department which is responsible for designing, building and maintaining the CARNet network, network and multimedia services. She received her B.Sc. and M.Sc. degrees in Electrical Engineering at FESB, the University of Split. For over a decade Ivana has worked in networking, in areas ranging from LAN to WAN, and in positions ranging from architect to manager. In the GÉANT3plus project Ivana leads the Network Support Services activity (SA4).

Pavle Vuletić is an assistant professor at the University of Belgrade, School of Electrical Engineering, teaching Advanced Networking course, and a deputy director of AMRES - Serbian NREN. He used to work at the University of Belgrade, Computer Centre as a senior network engineer and head of the networking team when UoB was responsible for the development of AMRES. He was involved in several EC projects in the area of research and education networking. From 2009 till 2013 he was a leader of a research task exploring multi-domain control and management principles in GN3 project. Currently he is leading the extension of the same task in the GN3+ project.

Marcin Wolski is the head of the Advanced Data Management System Team in Poznan Supercomputing and Networking Center. He also holds a position of Network Services Specialist at PSNC. His main interests include database systems, software development and enterprise application frameworks. He has participated in several projects funded by Polish Ministry of Science and Higher Education. During the GN3 project Marcin Wolski led the development of multi-domain workflow tools. Currently he is participating in GN3+ SA4 T3 which aims to enhance the design and use of existing monitoring tools and systems by aligning them with business processes.