Networked Model for Organising ICT Service Development in the University Consortium of Seinäjoki

Lanamäki Arto [1]
Lahti Juho [2]

[1] University of Tampere, Institute of Extension Studies (TYT), Seinäjoki, Finland
[2] University of Vaasa, Seinäjoki, Finland

Kampusranta 9 C, 60320 Seinäjoki, Finland
arto.lanamaki@sjoki.uta.fi, juho.lahti@uwasa.fi +358 6 4274315

Keywords: open source, institutional development, software development, knowledge-creation, SIP, networked model

Abstract

Networked model is based on cross-organisational ICT-service development processes in academic, public or semi-public organisations that are willing to use open source in their ICT service development. The organisations co-operate and share common resources. Networked model consist of 6 phases: 1) monitoring and participating to development, 2) requirements and needs, 3) testing and piloting 4) launching and providing services 5) legitimizing stabilized services and 6) results of legitimization. The model developed in the University Consortium of Seinäjoki (UCS), Finland, but it is transferable to other organisations as well. What combines these organisations is that they have their own development resources, but these resources are quite limited in some way. By co-operation it is possible to achieve better quality, inexpensive, suitable solutions and services. The implementation of the networked model sets new challenges for management.

Our networked model is realised in a SIP-protocol based VoIP telephone system adoption process which took place during 2004-2007.

1. Introduction

In this article we present a networked model that is based on cross-organisational ICT-service development processes in the University Consortium of Seinäjoki (UCS), Finland. Our model is presented by introducing a SIP-protocol based VoIP telephone system project case. This project is chosen because it contains the core factors of the model. There have been other projects as well where this model has been realised, such as the adoption of Access Grid multi-point video conferencing system.

Our approach is the exploitation of open source (Kavanagh 2004) in institutional development and the management of the development work in a networked academic environment.

We both authors of this article are project managers in UCS member organisations. Both of us have also been involved in the development project of SIP in the UCS. Our article creates a theory (Järvinen 2001) of networked development work.

2. UCS - Physical Organisational Context

There has been university activity at Seinäjoki ever since 1981, when the University of Tampere (TYT) arrived on the scene. Then came the University of Helsinki, the Sibelius Academy, the University of Vaasa and Tampere University of Technology. These five universities constitute the University Consortium of Seinäjoki (UCS). The mission of the UCS is to accomplish university research, education and services to society in co-operation with other actors in the region. University consortium is a cooperation innovation enabling universities to fulfil their regional functions. There are six such consortia in Finland, located in Kokkola, Pori, Lahti, Mikkeli and Kajaani in addition to Seinäjoki (www.ucs.fi).

Seinäjoki is a regional centre that does not have a science university of its own. There are total of 110 employees in UCS member organisations. In our article we focus on the topic, how we have organised common ICT-services in UCS using a networked model. We present a case study of how a SIP-protocol based VoIP (Voice over Internet Protocol) service has been adopted into UCS during 2004-2007.

UCS organisations have premises in three different locations in Seinäjoki: Frami Technology Center, Mediwest Health Technology Center and Kampus area. They are connected with a fibre network managed by the Seinäjoki University of Applied Science. The fibre network is connected to the Internet via Finnish University Network (FUNET).

Information and communication technology (ICT) services in UCS consist of various environments: Local Area Networks (LAN), Wireless Local Area Networks (WLAN), IP-telephony, video conferencing facilities and software, Virtual Private Networks (VPN) and Radius-based user authentication.

3. The SIP Process in UCS

3.1. Background

"The Session Initiation Protocol (SIP) is an application-layer control (signalling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences." (cit. RFC 3261)

Wirlab, located in Seinäjoki, was a research consortium that did research in networking real-life application environments. The consortium was founded in the year 2000. Aside from network research, Wirlab supported other research, education and enterprise activities in the networking business.

Wirlab served as a testbed for new technologies like SIP, both hardware and software related. Its testbeds were practical application environments in enterprises and non-profit organisations. Wirlab offered these testbeds to be used in other research done by different research centers and enterprises.

Wirlab had funding from the Finnish Technology Agency (Tekes/ERDF), Cygate Networks Ltd, Alajärven Puhelinosuuskunta, VLP Ltd, Employment and Economic Development Centre for South Ostrobothnia and the city of Seinäjoki.

In our model Wirlab represented an open source community. It was also linked to other open source communities, such as OpenSER and SEMS. OpenSER is an open source SIP server implementation. SEMS stands for SIP Express Media Server. It is a server application that enables the creation of different kinds of SIP-based VoIP-services.

Wirlab consortium merged into TYT's ICT-section in the beginning of 2004. Occassionally the Wirlab name is still used, but all their development projects are owned by TYT.

Wirlab's technology adviser Dr. Juha Heinänen introduced SIP for Wirlab consortium in January 2001. Heinänen is long-time network technology key person with a large curriculum including working as a CTO in major network operator companies like Sonera, Telia and Song Networks.

One of the first SIP-related Wirlab projects was KPhone, a Linux-based SIP client software. KPhone was originally developed by a nowadays-Google-developer Billy Biggs until 2000. KPhone development was done in Wirlab from 2001 to 2005. A Windows version was never released, which kept KPhone within quite narrow user base. However, KPhone was seen as a technologically advanced software in its time. It even was reviewed in Linux Journal as one of the best VoIP software products available for Linux. KPhone was the first VoIP-software to use internet Low Bit Rate Codec (iLBC), a codec later used also by many commercial VoIP software, such as Skype and Google Talk.

3.2. SIP adoption in UCS

The University Consortium of Seinäjoki (UCS) was founded in 2003. In the following year it received allowance from the Finnish Ministry of Education to develop educational and research environments. Some of this allowance was decided to be used in developing common information and communication technology infrastructure.

Before 2003, Wirlab projects were targeted to network operator companies due to investors' and supporters' decisions. In the projects funded by network operators open source software was developed to commercial software. The allowance UCS received made it possible also for UCS to benefit from Wirlab's SIP-knowledge.

Wirlab first introduced SIP for UCS in the end of 2003. UCS called its member organisations for proposals of the use of granted allowance. Wirlab consortium was managed by the University of Tampere, Institute of Extension Studies in Seinäjoki (TYT), which gave a proposal of using part of the allowance to develop a SIP-based VoIP-service to all units of UCS. The situation then was that all of the UCS organisations had their own analog telephone exchange and all the employees used analog telephones. Addition to that, most had a GSM mobile telephone. The plan was to create a common SIP-based VoIP-proxy to replace the old analog system. This also

required adoption of SIP-based VoIP-phones to all employees. The proposal was accepted in the summer 2004. The development was started immediately.

The developers of Wirlab/TYT had compared the features of various SIP-phones. The experience was that Sipura was the leading vendor at the time. Its SIP-phone had a sophisticated web-based maintenance interface, versatile call forwarding and management services. Sipura phone was compatible with the SIP server software that was used. Also Sipura provided enough features for users.

The Sipura 841 phone was chosen to be the UCS VoIP phone. An order of one hundred Sipura 841 phone units was made in September 2004. However, the Sipura phone was not tested well enough in practice, before the order decision was done. When the Sipura 841 phones were taken into use, the user experiences were bad. The phone was very weak in audio quality. The audio quality problem was tried to solve by a software update, but it did not help. It was found out that the weak audio was due to a hardware problem. Especially the heavy-users, people whose work routines involve a lot of telephone use, noticed the problems in audio quality. For the employees whom the telephone was not such an essential tool, were not disturbed by the audio troubles enough to complain about it.

For the heavy-users, an alternative was offered. This was a Sipura ATA-box, which contains the same SIP-software as the Sipura 841 telephone. ATA-box is an equipment that provides the same SIP-features, but the audio side was done by using a regular analog telephone receiver. The voice quality was good, but compared to Sipura phone, the phone memory features were not available.

Furthermore, the heavy users demanded that the phone system should provide a phone-call-capturing feature. This means that if any telephone rings in an office, it is possible to answer the call from any other telephone. This was a feature that was possible in the old analog era, but was not anymore supported in SIP. Call-capturing was a strongly adapted procedure in one UCS member organisation, and it made people resistant when this feature was not available anymore.

Call-capturing feature was a requirement which was not well received among the SIP development community. The community claimed that call-capturing was against the rules set in the SIP standard. Or maybe the standard supported this, but the SIP equipment vendors did not follow the standard strictly, so it was not supported in the telephones. However, some UCS-users saw call-capturing as such an essential feature, that they would rather move back into the analog telephone system, if it would not be possible in SIP. And they really did this. This lead the developers thinking about possible ways of solving the call-capturing requirement. It seemed like an impossible task – and it was. However, an alternative solution was invented by an individual developer in TYT in the beginning of 2007.

There had been negotiations with the Sipura company right after when the audio problems were discovered to be of major scale. This process started in March 2005. The developers in UCS complained and claimed that the SIP-phones would have to be changed to working newer models. Sipura Technologies Inc. was acquired by Cisco Systems' Linksys division in April 2005. The UCS developers had a chance to test one new Linksys-branded SIP-phone, which was evaluated to have an excellent audio

quality and even more advanced user interface. Much because of the Sipura-Linksys fusion, the vendor's decision about UCS's reclamation was not handled until spring 2006. Then Linksys agreed that the Sipura 841 phones had a major audio problem, and they replaced the old Sipura phones with newer Linksys SPA 941 SIP-phones. So the process of getting usable SIP-phones took one whole year. Finally during the summer 2006 all the old Sipura phones had been replaced by the new Linksys SIP-phones.

The failed adoption of Sipura SIP-phones had generated a lot of pressure for the management of TYT. A lot of criticism was heard, which lead the management to search for alternative solutions in co-operation with a local network operator. The company provided some SIP-products for testing purposes, but these were quickly seen insufficient and unfitting for UCS's needs. Also the confidence grew that the highest know-how of SIP was already in UCS and network operators or vendors could not provide any better knowledge. It also created some tension between the management and the developers in UCS, when the management intervened into a problem the developers knew that could not be solved any easy way.

After the adoption of Linksys SPA941 SIP-phones, the criticism stopped and the penetration grew significantly. Concerning SIP and other technologies implemented by the same allowance, the UCS organisations agreed about a common technology adoption and training strategy. This includes processes such as providing accounts to new users, delivering an up-to-date UCS phone book, organisation of user support and division of costs. Regardless of UCS-level co-operation, it is up to each organisation to choose when to close down their analog telephone systems.

Only a small percentage of the users is technologically oriented. SIP server software provide a lot of features and possibilities that have not, at least yet, been found useful in UCS work environment. For humanist-science-oriented majority of employees the SIP-phone is just a phone and they do not care about its advanced features. Some IT-minded employees were interested enough to deeply familiarise themselves to all available SIP-features. This exposed some software bugs. These bugs were soon fixed after those were recognised, and the required resources were available.

After the development phase, the SIP-system has come to a stable maintenance phase. TYT is the main responsible of the SIP maintenance. Its responsibilities include maintenance of SIP-servers, account management, provision of new accounts and telephone configuration. Users are supported by IT support personnel in each UCS member organisations. TYT takes care of network operator co-operation, including PRI (Primary Rate Interface) cable to provide PSTN and GSM network connections.

4. Discussion

We have presented a real life case of how a SIP-based VoIP system was adopted to the University Consortium of Seinäjoki. Our example of SIP development represents a networked model to organise the development and maintenance of ICT services.

According to the SIP adoption example presented before, we have created a six-phase model of networked development.

4.1. Networked Model

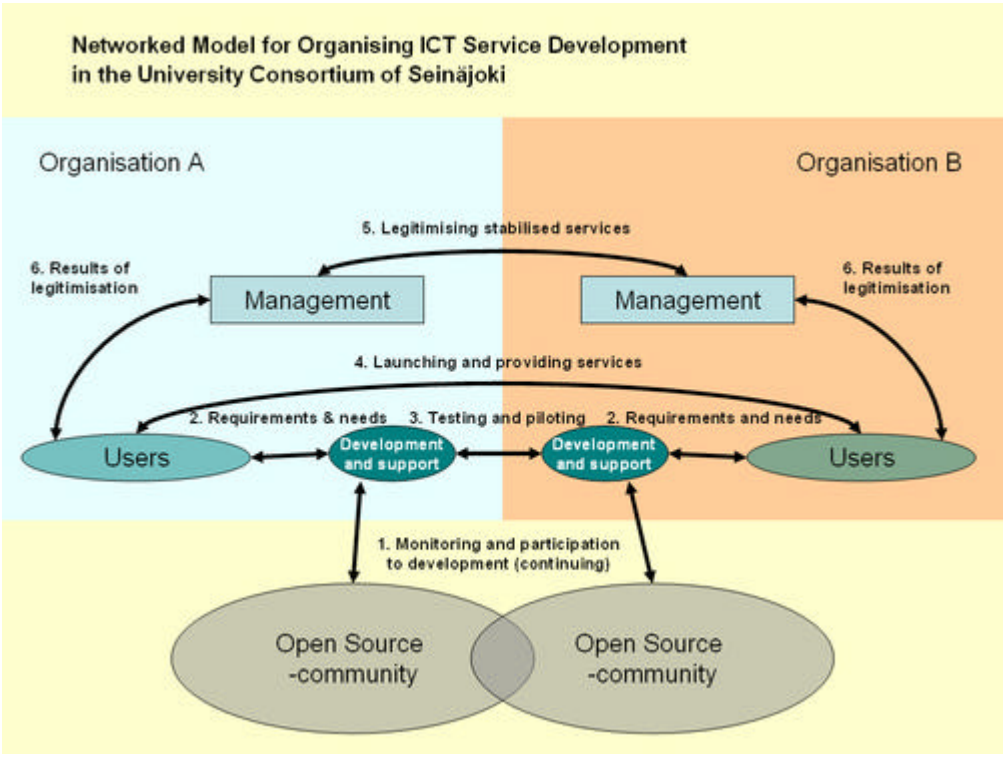

Chart 1 - Networked Model for Organising ICT-Services in University Consortium of Seinäjoki

The networked model for organising ICT-services in University Consortium of Seinäjoki consists of six phases.

1) Monitoring and participating to development

   UCS developers are continuously monitoring the activities of various open source communities. Some UCS developers are active participants in certain open source projects. This is a continuing process where the freedom of a individual developer at its highest.

   In our SIP case this phase lasted for three years from 2001 to 2003. It happened simultaneously by doing different kinds of SIP projects, but also as a undocumented process where the developers familiarized themselves with the SIP technology and many open source communities that were involved with this.

2) Requirements and needs

   This phase requires an input that turns the first phase into organisation-specific development. In our case the input was the granted allowance that made the development financially possible.

   Software developers and support personnel discussed with UCS employees and estimate the applicability of developed services in the use of particular

organisation. At the same time the developers listen to requirements given by users.

3) Testing and piloting

   Developers test and pilot the system, according to information gathered in the phase 2. Advanced ICT-oriented users often participate to this phase.

   This phase can be a quick one, because a significant amount of the development is done already before this phase. But it is also possible that this phase takes a lot of effort.

   In this phase, all feedback is given by individuals and no organisation-opinion exists.

4) Launching and providing services

   When there is enough evidence of suitability of services, it is spread out into wider use.

   If testing has not been done thoroughly, it is probable that some kind of crises will happen in this phase, as it happened in our SIP case. In the management point of view, this phase is the most challenging because a large amount of end-users must adopt the technology. It must be noticed that here the developer freedom decreases because of increasing institutional demands.

   In this phase a remarkable amount of feedback is gathered. As employees have had time to use the technology in real life and discuss about it, these opinions have grown into strong institutional opinions. Institutional opinions are always stronger than the opinions of their individual members.

5) Legitimizing stabilized services

   When the service has been used for a while and the users agree of its suitability and functionality, the managers of UCS organisations agree about the service and required resources.

   Legitimization aims to minimize the amount invoicing transactions. Instead of them it is preferred to use common resources over the institutional boundaries to achieve co-operation that benefits all stakeholders.

6) Results of legitimization

   After the cross-organisational management-decisions are made, the services turn into established routines and enable development of educational and scientific procedures.

While the model is definitely not problem-free, it has several advantages that could not be achieved in any other way.

It is notable that depending on the case, the amount of network-participating organisations can vary heavily. There can be two, three or even five different academic organisations that collaborate to the development. Also the organisational roles vary. Some of the organisations participate only after a long period of previous development. In our SIP case, one organisation was the responsible partner in development, some arrived in the testing phase and some after the service was fully launched.

The concepts of insourcing and outsourcing are quite interesting in our model. TYT sees that they have insourced the telephone system maintanance to themselves. Awhile the other UCS organisations see that they have changed the outsourcing partner from a local network operator to TYT. Some of personnel risks and responsibilities have moved from the network operator to TYT. Nevertheless the amount of personnel and financial resources needed for telephone system maintenance has significantly decreased.

The networked model is transferable to other organisations as well. Such organisations include universities, academic consortia, educational establishments and other public or semi-public organisations. What combines these organisations is that they have their own development resources, but these resources are quite limited in some way. By co-operation it is possible to achieve better quality, inexpensive, suitable solutions and services.

4.3. Conditions

Certain conditions must be realised before this model can be applied. The units must have the will and the ability to work in co-operation with other similar institutions. Also the will and the ability must exist in co-operating with development communities such as open source. If such a co-operation does not exist yet, it takes time and resources to build such a social network.

The organisation should have the ability and the power to make independent decisions considering its infrastructure and services. The UCS organisations have a very independent status towards their parent universities. The UCS organisations have their own network infrastructure, such as their own IP ranges, physical cables, adapters and other network equipment.

The organisations should have some tolerance for risks and uncertainty. The UCS organisations are used to project-based funding, which has grown their tolerance to uncertainty. It has trained them to results-determined thinking, which is a very different mindset compared to unlimited academic freedom.

The UCS allowance started the SIP development process. However in our general model this kind of funding is not essential. This is because the co-operative networked model releases unutilized resources that can be used efficiently in a shared work environment.

The necessary knowledge-potential existed in the UCS organisations before the development of SIP started. By knowledge-potential we mean that there was no need for additional recruitment of employees. The actors in the social network were

confident that they could create successful results even when no one had done a real life SIP hardware implementation before. As there was experience with KPhone SIP software client, enough knowledge existed to start the UCS SIP development project.

Knowledge includes that the actors in the network are able to communicate with the open source community. They must understand to ways of working and the use of specified language. It is a specialized skill to be able to build working software from open source drafts.

The problems that occured during the development process did not stop the development because the relations between the users and the developers was close enough. It is essential considering the networked model that the relationship and communication works between different counterparts. Without this a project might stop completely in a conflict.

The networked development process has facilitated different kinds of spin-off results. One example of this is OpenXG Inc. company (http://www.openxg.com/), which develops a SIP user manager software application for network operators. The UCS environment has provided a development platform that could not have happened in a commercial environment. Spin-off businesses have the possibility to financially strengthen the development processes for the future.

4.4. Management Challenges

One interesting phenomenon was the management's interference into the development process when the audio problems occurred in Sipura telephones. There was a lot of pressure from the users to get the problem fixed. Management had to interfere, because it was not aware if there was enough effort put into trying to solve the problem.

Developers must keep aware of which features are essential and important for the organisation and focus on those instead of creating solutions providing just extra glitter. Also the management must keep aware that the developers are doing the right things.

Development work in distributed networks is certainly a challenging task. The ways of working and thus the requirements for systems developed varies from organisation to another. The management knows its own institution, but they are not able to use the technology-related knowledge as a management principle, especially when considering a network-based development work.

One interesting point is the relation of the networked model and the concept of academic freedom. According to Wikipedia, academic freedom is "the freedom of teachers, students, and academic institutions to pursue knowledge wherever it may lead, without undue or unreasonable interference." (Columbia University 2005)

However, in order to achieve applications that support organisational goals, some level of institutional interference must occur. This interference must not be based on hierarchical and authorial need, but is a consequence of practical goals of problem-solving.

It is necessary to find the right balance between academic freedom and management demands. This question of the right balance has been recognised already in Aulin's (1979) the law of requisite hierarchy. The goal is to preserve freedom and creativity, while being able to quickly solve technical problems and considering user needs. Co-operation with the open source community supports these goals, because it is possible to make your own additions to the source code according to local needs. The open source community also provides more resources and general models to solve local problems.

Time by time institutional pressure occurs. In our case institutional pressure happened in the form of user complains after poor audio quality in Sipura units was noticed. Organisations applying the networked model should have pre-planned methods in processing the pressure.

In our model the institutional pressure must be seen as a positive factor. As noted in our SIP development case, the system developed a lot better because of institutional needs. The need for many important features, such as in our case call-capturing, would have not been noticed without the fact that the system was used by five different organisations. In general it can be said that crises are the most useful phases in a development process, because they make the developers concentrate on the core needs.

The management problems in UCS SIP process might have been solved using a management model that considers the institutional challenges and technological character of distributed development work. One possible solution might be adoption of Agile-based work methods, such as Scrum. Scrum is known as a model where information is distributed and exchanged efficiently. Our requirement is that the management will not turn into bureaucracy and over-documentation. Consideration has to be done to avoid panic reactions and authorial decision-making.

4.5. Benefits

It is notable that there are two kinds of benefit types in our model. There are *general benefits* that are due to the networked model itself. *Project-specific benefits* are such that rise from a certain project and its results. The projects are made possible using the networked model.

In our SIP-case there are many project-specific benefits. Contactability of employees has grown as the SIP system facilitates advanced phone call distribution, forwarding, and other easily manageable web-based services. As an economic benefit, all SIP to SIP calls are free. This is a major cost-saving factor. Additionally there is not need to operate analog telephone exchange any more, which has facilitated savings in work time and salaries.

During the SIP development process from 2001 to 2007 the number of office staff in TYT has decreased from 3 to 1. The same kind of process has happened in other UCS member organisations as well. While SIP is not the only reason for this, it certainly has been one strengthening factor in this process.

The development of SIP within UCS has also created regional technological expertise. Various network operators have applied this expertise in their own VoIP service development.

Concerning our networked model we were able to find several general level benefits rising from our SIP-case. However, the networked model probably has a lot of other benefits as well, that just did not appear in our SIP case.

Employees' technological knowledge has grown. This has facilitated more outside finance for new projects.

Open source community has benefited. Local development and use has brought forward the kinds of software bugs and needs for new features, that would not had happened otherwise.

Organisations have themselves learned to identify their core processes. Identification of these processes is a key factor in developing these processes more dynamic.

The benefits of open source are remarkable. By using open source software there are no software licence fees. Participation to open source community is an efficient way to build organisational knowledge-creation. As UCS consists of multiple universities,

knowledge-creation is not limited to just one organisation. It is a commonly shared activity between all participants.

The networked model is only possible by using open source software. Traditional closed source software would not facilitate the same amount of testing and making additions to the source code according to organisations' needs. In closed source the written agreement is required in the initial phase. This often leads to non-suitable and financially unsustainable results. Open source makes it possible to legitimize the software after it is found suitable.

The results of the UCS development model can be seen in different ways by different types of employees. For office administration this provides new ICT-services for better communication in everyday work. Researchers are able to make use of ICT-services in their personal and research use. The services provide many new research topics, such as combining use of these services in their own research fields. For software developers this is a great opportunity to participate into top-level open source development. This is a good way to develop personal know-how. Open source community participation provides two-way motivation for developers. The communities are based on voluntary activity. Also when the management notices the benefits of this activity, they can use it as a management asset to motivate developers to continue their work and to motivate the whole organisation to adopt more efficient work methods.

4.6. Conclusions

We have created a new theory of networked development. Our article offers possibilities for further research to test this theory in practice. Further research projects may have many different approaches, including open source development, management and dynamic organisational models.

It is notable that the networked development model does not create new organisational units or structures. All development is done using existing resources. New forming structures, mostly support processes such as maintenance, are functional, modifiable and reshapable according to actual needs. In the development process it is tried to ensure that the new technologies and supporting processes will be suitable for organisations' changing needs also in the future. This can be seen as flexible institutional development.

Refences

Järvinen, P. (2004). On research methods. Opinpajan kirja, Tampere, Finland
Aulin A.Y. (1979): "The Law of Requisite Hierarchy", Kybernetes 8, p. 259-266.
Kavanagh, P. (2004).: Open source software : implementation and management ISBN: 1-55558-320-2 Boston, Mass. Elsevier Digital Press.
Columbia University (2005). "First Global Colloquium of University Presidents, "Statement on Academic Freedom"
http://tools.ietf.org/html/rfc3261 RFC3261 18.5.2007
http://www.wirlab.net Wirlab 8.5.2007
http://www.ucs.fi/In_english.html University Consortium of Seinäjoki 8.5.2007
http://en.wikipedia.org/wiki/Academic_freedom 8.5.2007
http://www.openxg.com OpenXg Inc. 11.5.2007