

Puzzling with a top-down blueprint and a bottom-up network

**“An explorative analysis of the Free/Open Source Software world
using ITIL and Social Network Analysis”**



Sanne te Meerman
sanne@opensourceadvies.nl
Master Thesis
Sociology
University of Groningen
Supervisor: Dr. A. Flache
Referee: Dr. R. Popping
February 2003

Preface

This writing will explain some necessary tasks that need to be performed for the construction and maintenance of software. Next, an investigation is conducted how these tasks are performed in the open source world. Hopefully, this thesis is of practical use for those who are working with F/OSS, or are considering the possibilities of F/OSS. For them, there are also a lot of url's included where interesting facts and background information can be found.

I would like to thank the following people for their contribution to this thesis: The family te Meerman, Marieke van der Wal, Andreas Flache, Roel Popping, Wouter Teepe, Tom Roberts for his useful advice, extra information and pictures, and of course Erik Kuipers for his pictures.

Also I would like to thank all the people I have interviewed. Thank you for your time and the interesting conversations.

In order of appearance: Wouter Teepe, Hugo v/d Kooij, Robert Brink, Mendel Mobach, Bas Wijnen, Geert Stappers, Roeland Jansen, Fred Mobach and Erwin Mulder.



Special thanks to the Open Source Adviesgroep: www.opensourceadvies.nl

Contents

| | |
|--|-----------|
| Chapter 1 Introduction | 4 |
| Chapter 2 ITIL | 7 |
| 2.1 Configuration Management | 7 |
| 2.2 Incident Management | 8 |
| 2.3 Problem Management | 9 |
| 2.4 Change Management | 10 |
| 2.5 Software Control & Distribution | 10 |
| 2.6 The Helpdesk | 11 |
| Chapter 3 Social Network Analysis | 12 |
| 3.1 Where ITIL and SNA meet | 12 |
| 3.2 Actors | 12 |
| 3.3 Relations between Actors | 13 |
| 3.4 Resources | 13 |
| Chapter 4 Methodology | 15 |
| 4.1 Roles | 15 |
| 4.2 Genereal & Specific Research Questions | 16 |
| 4.3 Research Techniques | 17 |
| 4.4 The interview questions | 18 |
| 4.5 Analysis of the interviews | 19 |
| 4.6 Snowballing in practice | 20 |
| Chapter 5 Results | 22 |
| 5.1 Configuration Management | 22 |
| 5.2 Incident Management | 26 |
| 5.3 Problem Management | 29 |
| 5.4 Change Management | 32 |
| 5.5 Software Control & Distribution | 35 |
| Chapter 6 Conclusion | 39 |
| 6.1 Research Questions | 39 |
| 6.2 Discussion | 41 |
| 6.3 Further Research | 42 |
| Appendix A Interviewees | 43 |
| Appendix B Interviews | 44 |
| Literature/sources | 47 |

Chapter 1: Introduction

World Wide there are more and more countries that become aware of the possibilities of Free/Open Source software (F/OSS). Because of its open nature it is stated that for example security issues are easier to trace, and it is easier to make adjustments. Also considering the new licensing system Microsoft has adopted for Windows with which they have managed to double their profit in one year (Volkskrant, November 2002), it seems reasonable that governments but also companies increase their awareness of alternative software.

However, there is still substantial doubt about the 'chaotic, bazaar style of development' of F/OSS that apparently lacks any co-ordination or management. How can, for example, technical and functional support be expected? And how can any contractual guaranties be offered, when the construction of such software seems almost exclusively executed by individual volunteers?

So far, research has already made attempts to shine a light on the 'black box' of F/OSS development (i.e. Dafermos 2001, Krishnamurthy 2002, Mockus, Fielding & Herbsleb, 2002). More and more this research seems to suggest that F/OSS development is far from chaotic, and does not seem to be lacking management, or central co-ordination. This thesis falls into that category of research, and will extend upon the knowledge there is about the organisation of the F/OSS world. But maybe more important, it tries to be of practical use for companies or governments that want to use F/OSS to their benefit.

This thesis explores the development of F/OSS with two tools. First, ITIL is used. ITIL stands for Information Technology Infrastructure Library. It describes a collection of 'best practices' that were identified in service oriented IT-organisations (ITSMF 1999). The assumption in this thesis is that the processes that are identified by ITIL, must also be present in some form during the construction and maintenance of F/OSS, to explain its evident success.

Second, this thesis will use Social Network Analysis. ITIL has its limitations, mainly because it focuses on formal organisations. Its approach is very much top-down. It describes how certain processes can be organised in such a formal setting. F/OSS is often created in a much less formal setting. This does not mean F/OSS Projects are unstructured, but the organisation has often evolved from the bottom-up. Social Network Analysis is used to investigate how actors in the F/OSS world relate to each other, what resources they use to communicate, and how they organise the ITIL processes that are assumed to take place, -in some form-, somewhere within the F/OSS world.

Research indicates that formal organisations can benefit from informal networks within their organisation (Lundberg 1975; Krackhardt & Hanson; Meyerson 1992; Albrecht and Ropp 1984). However, the synergy of formal organisations & informal networks that is present in the F/OSS world seems to suggest that companies can also benefit greatly from informal networks outside of them (and vice versa). This study will provide insight into how some important processes in the F/OSS world are executed, and also how formal organisations and informal networks work together to accomplish this. By increasing our understanding of this, the co-operation between the involved actors can be extended upon, for the benefit of the corporate world, the contributors to F/OSS projects, and innovation in general.

Research question

This explorative study aims to answer the following general question:

“How are the necessary activities that must be employed for the construction and maintenance of a computer system fulfilled for the construction and maintenance of F/OSS? Which actors are active, how are they related and how do they communicate?”

Overview of this thesis.

In *chapter 2*, ITIL will be explained in detail. I intend to use the processes described by ITIL, of the construction and maintenance of software and hardware, as a starting point to describe and understand construction and maintenance of F/OSS¹.

In this thesis, ITIL will give an idea of which processes must take place in the F/OSS world to explain its evident success. As mentioned in the introduction however, ITIL has its limitations in this context because it is a top-down and process approach. To analyse how these processes are actually performed in the F/OSS world, a theoretical framework that is appropriate for analysing networks has to be used. Therefore Social Network Analysis is introduced in *chapter 3*.

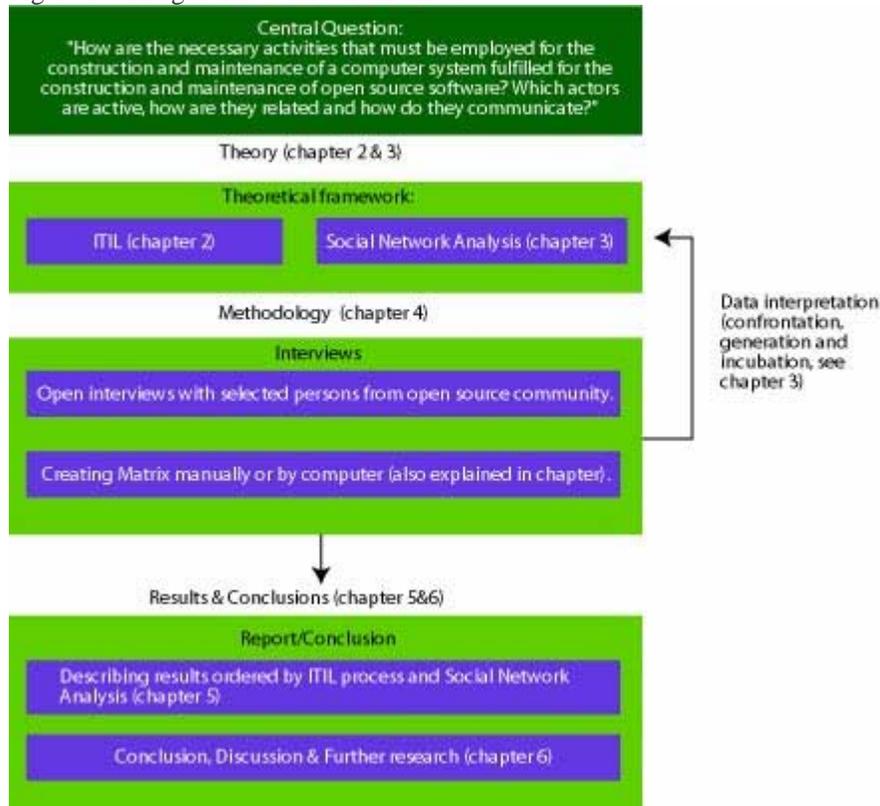
Chapter 4 explains how the general research question result in more specific questions, what research techniques are used, why they are used, and how the combination of ITIL and Social Network Analysis is transformed to interview questions, what respondents are being interviewed and how they were selected. It is also explained in this chapter how the data was analysed.

In *Chapter 5* the results are discussed. It describes how the F/OSS community performs the processes that are described by ITIL, which actors are involved, how they are related, and which resources are being used.

In *chapter 6* the general conclusions are presented. Also, the strengths and weaknesses of this thesis are being discussed. Implications for further research are given in chapter 6 as well.

¹ Throughout this thesis the often used term F/OSS is used. I felt this term respects equally the ‘fundamental’ ideology of Free Software, and the more practical ideology of Open Source Software.

Figure 1.a The general outline of this thesis

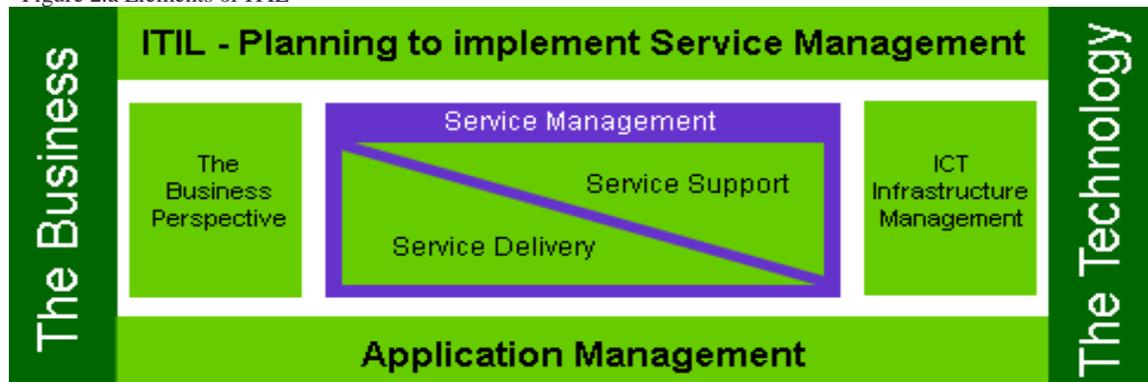


Chapter 2: ITIL

In order to design, construct, innovate and maintain a computer system, there are a number of activities that have to be employed. Until this day, these activities are mostly handled by corporate² organisations. People with the same working hours, and often in the same vicinity of each other, with many means of communication available, are responsible for these activities. Of course, companies differ in the way they organise the necessary activities to make this happen. And naturally, not all of these ways to organise companies are successful. Many theoreticians and practitioners have twisted their brains on how to organise things in the best way. In the United Kingdom for example, the Office of Government Commerce (OGC) developed a set of books called ITIL. ITIL stands for Information Technology Infrastructure Library. These books describe an integrated, process based, best practice framework for managing IT services.³

ITIL describes the processes involved that have to be performed to maintain and exploit the IT infrastructure. A process can best be described as ‘a series of activities that, when performed have the intention of achieving a certain goal’. By giving a detailed description of both operational as well as tactical processes such as (financial) planning, contingency management etc. ITIL gives a very comprehensive blueprint that can be used to structure IT companies. Because these processes are basically too comprehensive to take them all into account in this writing, I will focus on ‘Service Support’ of Service management.

Figure 2.a Elements of ITIL



Source: Koppens; Peters; Vonk 2001: 20

ITIL's Service Support set contains the following processes: Configuration Management, Incident Management, Problem Management, Change Management and Software Control & Distribution, and the Helpdesk. These processes are too comprehensive to be described in detail, and for the purpose of this study, it is actually more important that the 'core' of the processes described are captured somehow. To say this differently, I will focus on the actual 'Mission' of the process, or what the process really does, and why it is there in the first place. For illustration, I will also describe some of the tasks that each of these processes entails.

2.1 Configuration Management

The mission of the first process that will be described, Configuration Management, is to keep track of the means of production. This is important, because with these means of production, the

² Note that 'corporate' is used in the American way throughout this thesis, and refers to private companies and not to government institutions.

³ Definition from: <http://www.itilalumni.com/aboutitil.htm>

products or services that justify the existence of the company are realised. Like other means of production, the IT-infrastructure must also be controlled (ITSMF 1999).

ITIL names the parts of which the infrastructure consists 'configuration items', or CI's. Information about these configuration items is stored in the so-called CMDB, the 'configuration management database'. Examples of such Configuration Items (CI's) are printers, keyboards, Manuals, Documentation, Collections of software, etc.

The activities done by Configuration Management to 'complete its mission' are:

- Identification: describe and administrate the components of the infrastructure and how they relate to each other.
- Management: making sure the contents of the CMDB is up to date.
- Monitoring: monitoring the status of the components.
- Verification: verifying if the information in the CMDB is up to date.
- Reporting: informing the management about trends and developments.

Of course, in the F/OSS world, things are 'slightly' different than in a corporate IT company. According to 'De Automatiseringsgids', ITIL is seen as too theoretical, too demanding, and too dry (Automatiseringsgids, October 2002). Apparently, it is already hard for traditional companies, with their formal organisation structures, to implement ITIL. It seems obvious then, that it is unlikely that within the F/OSS community that thrives on informal connections of people, processes described by ITIL will be found in their ideal theoretical way. Still, the 'mission' of Configuration Management seems as crucial for the F/OSS community as for traditional companies. The question still remains how this mission is completed. It can be expected that in the F/OSS world Configuration Management is handled very differently. The main argument for this is a demographic one: the people who actually build F/OSS are living all over the world. This means these people are responsible for their own means of production. However, there are also sites on the Internet where a lot of F/OSS development takes place, for example Sourceforge⁴. These sites and the tools that are present there must also be controlled. It seems that corporate actors are involved here, but it will become clear that other actors play a role here as well.

2.2 Incident Management

Nowadays organisations depend more and more on Information Technology. When something fails in the IT infrastructure, this can have a huge impact on many crucial processes in a company. Therefore, it is very important that any failure is dealt with as fast as possible. The next process Service Support describes deals with this and is called 'Incident Management'. The Mission of Incident management is described as: restoring the services that an IT company offers, by minimising the effects of failures so the end-user can continue his work (ITSMF 2000:55)

To complete its mission, the following activities are part of incident management:

- detection and registration;
- classification and assigning of the incident;
- investigation and diagnosis;
- problem solving and repair;
- closing of the problem;
- monitoring and communication;

⁴ <http://sourceforge.net/> On this site there are approximately 53.000 projects hosted, created by approx. 500000 people. Sourceforge is provided by VA software and is in fact a 'downstripped' version of a unified development environment for managing code, content and process, that is also sold commercially in a more complete form by VA software.

Incident Management, when it is implemented according to ITIL, contains some sub-processes like the administration of the impact and urgency of an incident. Also, ways to ‘escalate⁵’ an incident are described, which means the incident is transferred to a different part of the organisation that will deal with it. In the case of F/OSS, things will –again- be slightly different. As F/OSS is not being manufactured in an organisation, and there is no helpdesk in the physical sense, there probably will be no such thing as a pre-defined process to deal with incidents. However, with F/OSS, as with any other piece of software, incidents and failures will occur, and there will be a process, more or less visible that ‘Manages’ these incidents and failures.

Again, ITIL may prove helpful in reconstructing the process as it is performed by the F/OSS community. Because in the F/OSS world, as well as in any formal setting where software is created, incidents that are stumbled upon by users must be investigated, diagnosed, repaired and monitored to see if the problem is actually solved. ITIL’s Incident Management can give direction to investigate how the F/OSS Community shapes their efforts to manage incidents.

2.3 Problem Management

Some incidents that occur will be no more than incidents, and after a solution or work around is offered, they will not reappear. In some cases however, there will be incidents that are caused by serious malfunctions of i.e. the infrastructure. In these cases, it is imperative that a more thorough investigation is conducted and that a structural solution is found. The process that is responsible for the investigation of the malfunction and the fixing of the problem is called ‘Problem Management’.

Like the other processes, Problem Management is a comprehensive process. Also the way it interacts, and sometimes overlaps with incident management is quite intricate. I have therefore not been able to fully take its complexity into account. But the core of the process can be described as follows: In essence, an effort has to be made to trace the causes of incidents, and offer a solution. (Koppens: 50) This effort is called problem management. Of course, we need a little more than this to grasp the complexity of Problem Management as it might occur in the F/OSS community. Here are some of the sub processes Problem Management executes in order to investigate problems and offer solutions:

- Problem control: defining the problem and making a diagnosis.
- Error control: as soon as a problem is identified and has become a so called ‘known error’, error control is the monitoring of the problem, making the ‘Request for Change’ and monitoring the change and evaluating after the change is done.
- Proactive support: prevention of problems before they appear.
- Reporting to the management.

Some of the aspects of problem management will no doubt also be performed somewhere within the F/OSS community. The power of F/OSS is exactly that a lot of users world-wide detect flaws of the system, and either report these to others, or make repairs themselves (Given enough eyeballs, all bugs are shallow⁶). How these changes are actually being monitored is not so obvious, and less visible. It is also unsure whether or not there is some form of proactive support and proactive scanning for problems.

⁵ From the French word ‘Escalier’ (staircase). Escalation can be vertical or horizontal.

⁶ Linus’ Law. This has become a well-known quote from E. Raymond’s book (and article) ‘The cathedral & the bazaar’.

2.4 Change Management

If a bug causes a malfunction, and this bug is discovered, it has become a 'known error', according to ITIL a 'request for change' (RfC⁷) has to be made, and the change has to be executed. When a lot of people are using a certain application that has this bug, in all of these applications the bug has to be fixed. Software is never free from bugs, so a software package that is being used will often lead to a lot of incidents, resulting in a lot of request for change. There are of course also other sources of requests for change, like customers who want a certain functionality to a program, or a new law, that demands changes to increase security.

The bottom line is: a lot of changes are being made. Change Management is the process that is responsible for these changes to be made in an orderly fashion. Put more formally: Change Management is responsible for assuring that standardised methods and techniques are used so changes can be made in a direct and efficient manner. In reality Change Management is of course a complicated and intricate process that consists of a lot of sub-processes.

Again, these are too numerous to take them all into account, but the following are the most important activities that are executed by Change Management.

- Filtering the RfC's and accepting them for 'further treatment'.
- Classification of the RfC's (according to size and priority)
- Planning of the execution of the changes, and determining the resources needed.
- Co-ordination of the constructing, testing and implementing.
- Evaluation to see if the change was a success.

(Koppens 2001, ITSMF 1999)

Again, it seems rather obvious that F/OSS, in order to be improved, to get problems solved, and to keep up with changes that are made in the surrounding technical environment, has to 'manage changes', in the broader sense of the word. How these changes are managed exactly, is not quite so obvious. If there is something as well defined and structured as change management in the case of F/OSS, it is done by a group of geographically dispersed people that often communicate only by means of their PC. How this is done exactly is not known, but ITIL might prove to be helpful starting point to gain insight in this.

2.5 Software Control & Distribution

With the description of Configuration Management in section 2.1 the need for controlling the means of production of an organisation was discussed. Configuration Management is mostly concerned with Configuration Items that are physically present in an organisation. Software, that is not physically present, has a special place among these means of production. Therefore, there is a separate process that is concerned with software only. This process is called 'Software Control & Distribution'.

With the increasing amount of software that is being used nowadays, and considering the vulnerable nature of software, the need for such a process should be clear. Also taking into account the incidents that will occur over time, resulting in many changes of the software, many different (altered) versions can be found scattered across the organisation, if not somehow managed in a systematic fashion. Periodically, or in urgent situations, the latest or most stable versions of these programs must be released or distributed.

The activities that Software control & distribution does to manage and distribute the different software versions are the following:

⁷ Not to be confused with the RfC that is part of the vocabulary of the Open Source World. For them it means: 'Request for Comment'.

- Managing the ‘Definite Software Library’. This means the storing of the physical software in a database.
- Release Build. This means that a selection is made from the software in the ‘Definite Software Library’ or DSL, and is being ‘released’. This can be a selection of all the latest versions of the programs and application consists of, or a smaller selection of only the programs that have been altered since the latest release.
- Distribution. The release is copied to the test environment, and later to the production environment (often a customer has both environments so a new release can be tested on the test environment before actually using it for production.)
- Implementation. This means that the release is installed on the right location, and goes ‘live’. The release is now installed and being used.

The need to manage different software versions is of course as important for F/OSS as it is for Closed Source software. As was also the case with configuration management, Incident Management Problem Management and Change Management, having looked at how this process is shaped according to ITIL, it will be more clear what to look for in the F/OSS world.

2.6 The Helpdesk

IT companies nowadays often have a physical place, a separate organisational unit with which customers can have contact to address their problems. It is not so much a separate process, as much as it is a physical place they can visit, or contact by phone, email, fax, etc. Basically the processes described so far are often centred around the helpdesk. Usually customers who experience problems will contact the helpdesk. The helpdesk will often do the registration of the problem, and assign the problem. Incident management is one of the processes described by ITIL that is mostly done by the helpdesk. Because the helpdesk is not a completely different process like incident management, or problem management, it falls outside the focus of this thesis.

Summary

This chapter described five processes of ITIL’s Service Support set. First, Configuration Management was explained. This process is basically responsible for managing the means of production of an IT organization. Second, Incident Management was explained. This process explains what must happen when incidents occur. These incidents must be administrated and solved, so the end-user can continue his work. Next, Problem Management was described. When an incident is caused by a serious malfunction, the incident is no longer ‘just an incident’ but a structural problem that needs to be dealt with efficiently. Problem Management gives a framework of how this can be done. After this, Change management was introduced. It is necessary that changes are managed, because after several problems have been solved, and many changes need to be made, it is easy to lose track of these changes. Change Management tries to coordinate these changes by using standardized procedures and techniques. Finally, Software Control & Distribution was explained. Software has a special place among the means of production of a company, because it is not physically present, can be damaged easily, and there are often different versions and different programs around. Software Control & Distribution is responsible for managing these different versions, and distributing it. After the description of these ITIL processes, it is argued that these processes must also be present in some form in the Open Source World. Finally the helpdesk is mentioned, but this process falls outside the focus of this thesis.

Chapter 3: Social Network Analysis

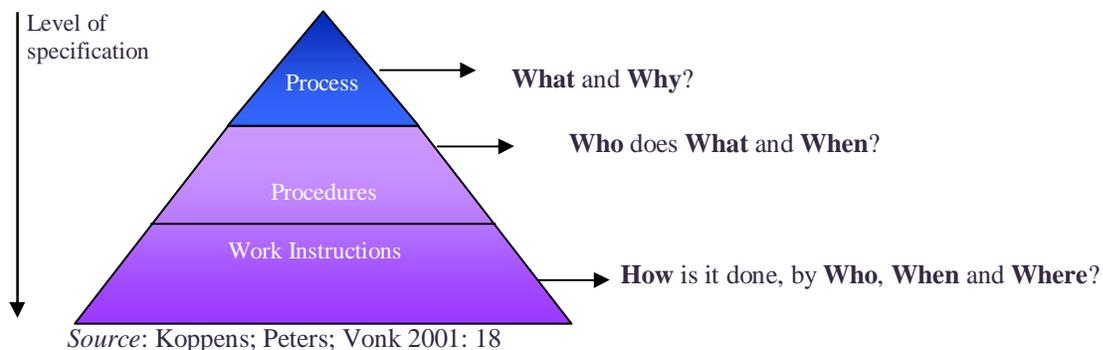
In this chapter it will become clear that the title ‘puzzling with a top-down blueprint and a bottom-up network’ is really what this thesis is all about. In the last chapter, five ITIL processes have been outlined to explain some crucial processes that need to take place in some form to create and maintain software. The ‘puzzling’ means that it is basically a matter of trying to reconstruct how these processes are performed in the F/OSS world. Of course, a serious effort is made to make this ‘puzzling’ as scientific and well-structured as possible. However, Social Network Analysis (SNA), that is introduced in this chapter, will not be used to its full capacity to investigate and analyse networks of people with different types of quantitative measures.

Rather, for the explorative nature of this thesis, some useful concepts and ideas that can be helpful to get a better understanding of how networks of people can be investigated are used of SNA.

3.1 Where ITIL and SNA meet.

The following picture, that explains what ITIL does and does not intend to do, also illustrates where SNA is used in this thesis.

Figure 3.a Process, procedures, work instructions



Although there is debate about the exact purpose of ITIL, Koppens & Vonk emphasise that ITIL intends mostly to describe processes: what is done, and why is it done. According to them, ITIL does not want to get involved with the procedures and work instructions too much. In this thesis, ITIL is used accordingly: it is used to get a broad idea of some important processes that need to be executed in order to create and maintain software. To find out who does what, when it is done, how it is done and where is it done to perform this process in the F/OSS world, SNA is used.

3.2 Actors

“Social network analysis is concerned with understanding the linkages among social entities and the implications of these linkages. The social entities are referred to as actors. Actors are discrete individual, corporate, or collective social units (Wasserman & Galaskiewicz, 1994).

By looking at this definition of SNA more thoroughly, it will become clear that SNA is a very useful framework for the study of the F/OSS world. The concept of ‘actor’ is the first concept of SNA that is used. As shown above, actors can be ‘individual, corporate or collective social units’. Such a comprehensive definition is very useful for this thesis, because although there seem to be many volunteers active, who are working from their computer at home, in other cases it seems

that groups are fulfilling certain tasks, and that even corporate actors (companies like Suse for example) play a significant part in the construction and maintenance of Open Source Software.

3.3 Relations between Actors

“Social network analysis is concerned with understanding the linkages among social entities and the implications of these linkages. The social entities are referred to as actors. Actors are discrete individual, corporate, or collective social units (Wasserman & Galaskiewicz, 1994).

Traditionally, when describing networks of people, Social Network Analysis did not focus so much on these linkages itself, but mainly on attributes of the actors that were subject of study. Later on, these linkages or relations between the actors came more into focus. It is also exactly this relational data which distinguishes SNA from other viewpoints. SNA has evolved so far that there are instruments available for the quantification of this relational data. As stated before, I will not come to this in this study, but the concepts and variables that are used for studying the relations between actors, give an interesting and useful perspective. Here are some examples of how relations can be characterized

:

- Transfer of material resources (business transactions, lending, borrowing)
- Transfer of non-material resources (communications, sending/receiving information)
- Formal Relations
- Physical connection (a road, river, etc.)
- Evaluation of one person by another (for example: expressed friendship, liking or respect)

(Wasserman & Galaskiewicz, 1994)

The following table will give a brief overview of differences between the corporate world and the F/OSS world, concerning these relational ties:

Table 3.b Difference relational ties between Corporate World & F/OSS world

| | <i>Corporate world</i> | <i>F/OSS world</i> |
|--|--|---|
| <i>Transfer of material resources</i> | Many possibilities (gifts, money, transactions etc) | Low (mostly exchange of information) |
| <i>Transfer of non-material resources</i> | More possibilities (meetings, face to face contact, verbal/non verbal cues etc.) | Less possibilities (dependant on Computer Mediated Communication) |
| <i>Formal Relationships</i> | Yes (corporate world depends on formal relations and contracts) | No (F/OSS world depends mostly on volunteers) |
| <i>Physical connection</i> | Possibility available (employees working together, physical intra & inter organisational contact). | Most contact using CMC (computer mediated communication) |
| <i>Evaluation of one person by another</i> | Many means available (verbal/non verbal cues, money etc.) | Mostly by CMC (computer mediated communication) |

3.4 Resources

Although SNA has some interesting concepts that can be used to study relations between people, SNA is not specifically designed to study relations in the relatively new, and very technological environment of the internet. Because resources that people use to communicate with each other create boundaries and opportunities for the relations between people, these resources can give a lot of information about the nature of the relationships between people. For example, if people communicate by telephone, it is not possible to transfer material resources. Also when

communicating by phone, there is a limitation of how many people can contribute to the conversation.

Introducing the resources used in the F/OSS world as a subject of study, alongside the actors and relations within the F/OSS world, will facilitate the reconstruction of how certain processes are performed in the F/OSS world. The definition of 'resources' that is used will be very broad. Resources will be mentioned in this research:

- If they are used for communication of the involved actors
- If they are helpful in any other way to increase our understanding of how actors are related, and organise certain processes.

Summary

While ITIL serves as a top-down starting point to get an idea of the processes needed to create and maintain software, In this chapter, Social Network Analysis (SNA) is introduced. SNA is not used in its full capacity to analyse networks using quantitative measures. Rather, it is used to get a broad understanding of some useful concepts to analyze the bottom-up organized networks in the Open Source World.

First, the concept of 'Actor' as it is used by SNA is introduced. This concept is useful because SNA gives a broad definition of an actor. According to SNA, an actor can be individual, corporate, or a collective social unit.

Next, SNA's focus on relations between actors is discussed. This is one of the things that makes SNA different from other theoretical frameworks. SNA describes several distinctive 'relational ties', like whether information is transferred, or material goods.

Finally it is argued that to understand how these actors are related, it is necessary to look at the resources these actors use to communicate.

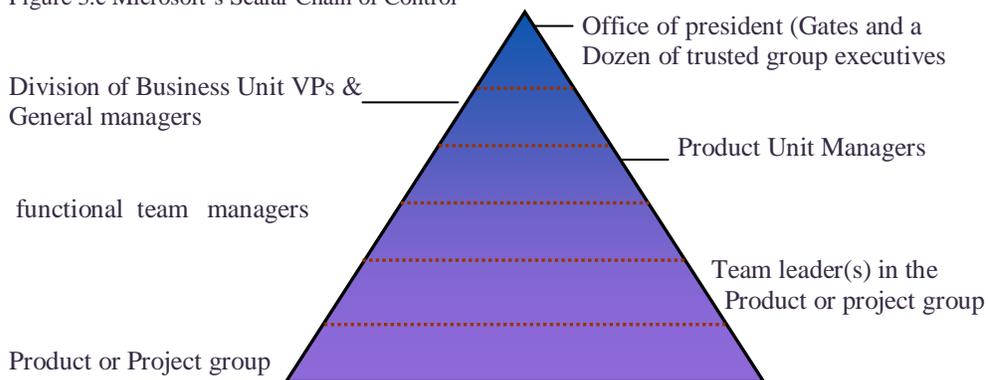
Chapter 4: Methodology

This chapter will get back to the general research question, and will explain in more detail how this general question results in more specific research questions and how ITIL and Social Network Analysis are used to gather information and analyse this information. Furthermore, some specific techniques besides ITIL and SNA that are used to gather information and analyse it are explained. Finally, the first interview questions that result from this are presented.

4.1 Roles

Before I go into this, yet another concept of Social Network Analysis is introduced, namely the concept of 'role'. This basically means that SNA takes into account that actors can have different roles in different circumstances. The reason it was not introduced in the last chapter is because it is in fact a methodological issue; roles are not so much the subject of study. The importance of the concept of roles for this research will become clear, when considering how software is built by a more traditional organisation like Microsoft:

Figure 3.c Microsoft's Scalar Chain of Control

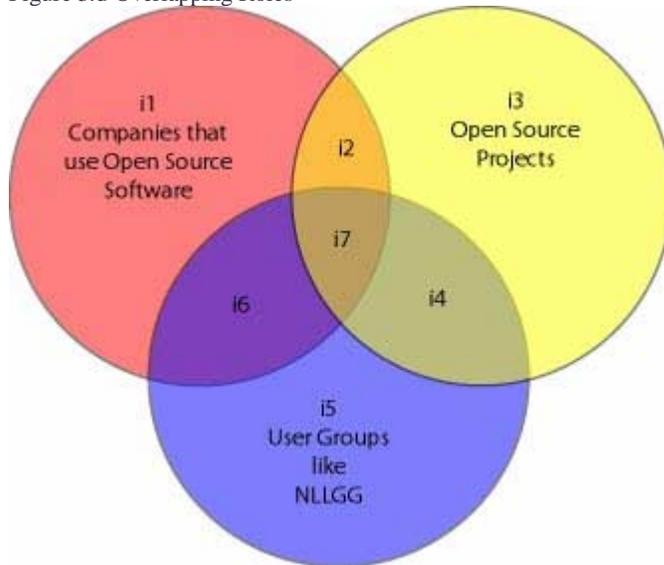


Source: Dafermos 2001: 73

It seems that Microsoft has a clear structure where separated departments create different products (Dafermos, 2001). This suggests that for the construction people have a clearly defined position or 'role' in a well-defined hierarchical structure.

Although I have not found research about this, it seems that for the construction of F/OSS, the people involved have a more diverse set of roles in an organisational structure that seems less hierarchical and less well-defined. Consider for example the following picture, which illustrates how individuals can possibly be involved with the construction and maintenance of F/OSS in different ways at the same time.

Figure 3.d Overlapping Roles



Source: A. Flache 2002
(Personal Communication)

Explanation

In this example, there are three groups active with F/OSS, so an individual can participate in F/OSS in many ways. For example individual **i1** may use F/OSS in a professional setting, and occasionally report problems back to the community. Another individual, like **i4** is a member of the NLLGG and participates in an F/OSS project. Even another is active with F/OSS professionally, as a participant of an F/OSS project, and as a member of the NLLGG (the Dutch Linux User Group, details will follow in chapter 4). Things can get rather complicated because in every group, this particular individual can have specific relations with other important actors. To make things more complicated, within one of such groups people can also have multiple roles. T.R. Madanmohan and S. Navelkar (Madanmohan & Navelkar 2002) have identified several roles concerning the knowledge management of Online Technology Communities. By introducing the concept of 'Role' also used in Social Network Analysis, this problem can be tackled. When interviewing an individual about his or her position in the network it is crucial to determine the different ways in which someone is involved with F/OSS. How this is dealt with when interviewing key persons of the F/OSS world, is explained later in this chapter.

4.2 General & Specific Research Questions

“How are the necessary activities that must be employed for the construction and maintenance of a computer system fulfilled for the construction and maintenance of F/OSS? Which actors are active, how are they related and how do they communicate?”

Looking at this general research question again, it should now become clear how ITIL and SNA can contribute to answering this question:

ITIL

“... the necessary activities that must be employed for the construction and maintenance of a computer system...”

This part of the general question is in fact what ITIL focuses on in this thesis. The specific question that ITIL tries to answer originates from this:

Q1: What are the necessary activities that must be employed for the construction and maintenance of a computer system?

Social Network Analysis

“...which actors are active...”

This is where Social Network Analysis becomes helpful. It provides a broad definition of these possible actors that considers both individual as corporate actors. The specific question that stems from this is:

Q2: Which actors are performing the ‘necessary activities’ that ITIL describes?

“...how are they related...”

Again Social Network Analysis is the suitable framework. It focuses on relations between actors, and has identified many relational ties. For example, if material or non-material goods are transferred between the actors, if they are connected physically etc. This results in the following question:

Q3: How can the relations between the actors be described, in terms of the relational ties between them?

“... and how do they communicate?”

This seems similar to the question of the relational tie between the actors, but this question aims to focus on the technical side of the communication. With the emergence of the Internet, a lot of Computer Mediated Communication came along with it. The specific question that results from this is:

Q4: What resources are being used by the involved actors to communicate, and co-ordinate their efforts?

4.3 Research Techniques

In chapter 2, the different processes that ITIL’s service support consists of are described. With this, the answer to *question 1* is already answered. (What are the necessary activities that must be employed for the construction and maintenance of a computer system?). Answers to the other specific questions remain to be answered:

- *question 2* (Which actors are performing the ‘necessary activities’ that ITIL describes?),
- *question 3* (How can the relation between the actors be described, in terms of the relational ties between the actors?) and
- *question 4* (What resources are being used by the involved actors to communicate, and co-ordinate their efforts?).

In this paragraph the research techniques that are used to answer these questions are introduced.

Literature

Literature about qualitative research, about ITIL, and about Social Network Analysis is used. Especially worth mentioning is the vastly growing amount of on-line literature⁸ about the F/OSS phenomenon. Most of this research is only recently written and contains very up-to-date and relevant information.

Observation

Communication within the F/OSS community is for a great part done on the Internet. There are many mailing lists where communication about F/OSS takes place. By subscribing to these mailing lists, and observing the communication that takes place several things can be determined.

By looking at the contents, it is possible to see how certain problems are discussed, where people find their solutions, and what the nature is of the things that are discussed. A first step in

⁸ <http://opensource.mit.edu/> a lot of research about the open source phenomenon can be found here online, it is published by many scientists worldwide. There is also a discussion list.

investigating how, for example, incidents are handled can be accomplished like this. Also, which people are the people who pose questions, and which people answer them, can be determined by observation.

Most importantly, by observing the contents, but also the frequency of communication that occurs on the mailing lists, key persons can be identified. For example, people who often answer questions of others, are more likely to be experts.

Snowballing

Snowballing is described as the practice by which recruited respondents may be asked if they know another person that is willing and able to contribute to the investigation. Especially when it is difficult to find suitable respondents this technique can be very useful. In this case, it is indeed difficult to find subjects who have the required knowledge about the subject, so this technique is a helpful addition to selecting respondents using observation.

Semi-structured interviews

When important key persons are identified, it is now important to gather information from these key persons. Because of the explorative nature of this thesis, semi-structured interviews are the best choice. The structure of these interviews will be provided by ITIL and SNA. Again, the title of this thesis, -‘puzzling with a top-down blueprint and a bottom-up network’- is very illustrative. Basically, the interviewees will be presented with the processes that are described by ITIL. The purpose of these processes will be explained briefly. Next, using concepts of Social Network Analysis, it is reconstructed how these processes are performed in the F/OSS world. The next paragraph will provide more details about the interview questions, and how the interviews are structured using ITIL and SNA.

4.4 The interview questions

After a brief introduction about the purpose of the interview, the theoretical framework that is being used, and general information about the background of the interview, the interviewee will be asked about his personal involvement with F/OSS and the F/OSS community, or the different ‘roles’ the interviewee fulfils in the F/OSS world. This will put the interviewee at ease and give him a chance to consider the activities of the F/OSS community without already analysing it in the terms and theories used by the study. But most important, this will provide a general structure for the interview. In each of the questions that follow next, the different roles that are mentioned by the interviewee can be considered separately.

The possibility must also be considered that the interviewee is not personally involved with this particular process, but is aware how this particular process is performed in the community. Therefore I will ask these questions more in general after the interviewee has answered the questions about his personal involvement with the process.

I will continue the interview by asking about the ITIL processes. In the beginning of each question, I will first briefly describe the process as it is defined by ITIL. Next, the research questions can then lead to concrete questions:

Question 2 (Which actors are performing the ‘necessary activities’ that ITIL describes?) will lead to asking the respondent:

- If and how he is involved with the process, in the different roles he plays in the community.
- If he can identify other actors involved in this process.

question 3 (How can the relation between the actors be described, in terms of the relational ties between the actors?) leads to asking:

- How these actors are related to each other, are related to him, and how he relates to these actors.

question 4 (What resources are being used by the involved actors to communicate, and coordinate their efforts?) results in asking about:

- What means of communication can the involved actors use to communicate with each other?

The general outline of the Interviews will be identical to the Service Support set of ITIL. I will treat the different processes (Configuration Management, Incident Management, Problem Management, Change Management and Software Control & Distribution) subsequently, in the order just presented.

4.5 Analysis of the interviews

To see if the ITIL processes are a good starting point for this thesis, and to see if the concepts of SNA are useful for this research, the first results from the interviews will be compared with the theoretical framework that is at the heart of the research. This type of analysis is called 'kaderanalyse'⁹, and consists of two phases:

A *confrontation* phase, which means that the theoretical framework will be confronted with the empirical data, and a *generation* phase, which means that new concepts and hypotheses are generated. In this explorative study, instead of a hypotheses, temporary models are created after the first interviews. These models will be a first attempt to reconstruct how such processes as Incident management etc. are performed. In later interviews, these temporary models can be shown to the interviewees, so they can be extended upon, or altered.

There is one more phase that in fact proceeds the confrontation and generation phase: the *incubation* phase. In this phase it will be analysed if the interview does indeed cover the context, the atmosphere of the situation. Possible ideas that come to mind during the interviews will be written down. (Maso & Smaling 1998)

To analyse the interviews in an efficient way and to report the results in an orderly fashion, a 'topic list' will be used. First, the interviews are transcribed and organised by subject. Next, in every row it will be investigated where the respondents correspond and where they differed in their view. (Maso & Smaling 1998)

Table 4.a Matrix used to analyse data

| | Respondent 1 | Respondent 2 | Respondent 3 | Etc. |
|---------|--------------|--------------|--------------|------|
| Topic 1 | | | | |
| Topic 2 | | | | |
| Topic 3 | | | | |
| Topic 4 | | | | |
| Topic 5 | | | | |
| Etc. | | | | |

Source: Maso & Smaling 1998

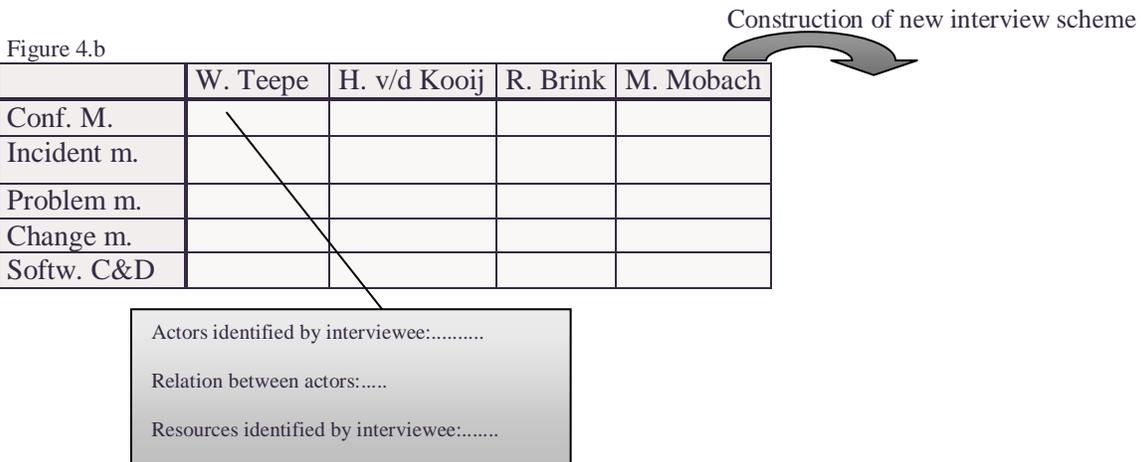
⁹ I have not found an English equivalent for 'kaderanalyse', but it should be something like 'framework analysis', I reckon.

4.6 Snowballing in practice

The methodology described so far has been helpful to provide structure to the process of gathering and analysing the information that was gathered. However, because the respondents were selected using the technique of snowballing, and it was difficult to determine in advance what information would be provided during which interview, changes were made along the way. This paragraph will describe which respondents were interviewed, and how the interview schemes were adapted accordingly.

First, I have interviewed W. Teepe, who is one of the key persons of the SquirrelMail¹⁰ project. In this interview my first assumptions were tested, and some important actors were identified. After this I have ‘surfed the web’ to gather background information. Then, I have observed the Dutch Linux User Group, the NLLGG¹¹. This group had about 750 members at that time. I have subscribed to their mailing list, and have identified important figures as far as supporting fellow members. Several of them were approached and they were interviewed on a conference. After the first round of interviews, these interviews were written out fully, and using the matrix, in every cell the actors, relations and resources that were mentioned were written down in keywords. Using this matrix, it was determined what information was still lacking or where more detailed information was needed. The theoretical framework was reconsidered, new ideas were generated and ideas that came to mind during the interviews were written down so they could be verified in the following interviews (confrontation, generation and incubation phase).

Next, background information was searched on the Internet, and information provided by the interviewees was verified and investigated. Relevant organisations and websites that were mentioned during the interviews were searched, and more possible actors and resources were identified. Also, a first attempt was made to create a model of each of the ITIL processes to verify, and extend upon. Using this model, and the extra information gathered on the Internet, a more in-depth interview scheme was constructed. Figure 4.b illustrates this.



In the next phase, I have directly altered the interview scheme after each interview. Also, after each interview the model of each of the five ITIL processes was tested and extended. First, I have interviewed W. Teepe again, second, I have interviewed B. Wijnen, and next I have revisited the NLLGG and interviewed F. Mobach, R. Jansen, and G. Stappers. Finally, I have interviewed E.

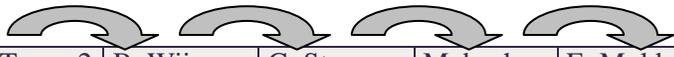
¹⁰ <http://www.squirrelmail.org/> SquirrelMail is a widely used standards-based web-mail package. It is translated in many languages and has thousands of users worldwide. It is hosted by Sourceforge.

¹¹ <http://www.nllgg.nl/> the Dutch Linux user group.

Mulder. The next figure will clarify this phase. More can be read about the respondents in appendix A.

New interview schemes were constructed after each interview

Figure 4.c



| | W. Teepe 2 | B. Wijnen | G. Stappers | Mobach & Jansen | E. Mulder |
|-------------|------------|-----------|-------------|-----------------|-----------|
| Conf. M. | | | | | |
| Incident m. | | | | | |
| Problem m. | | | | | |
| Change m. | | | | | |
| Softw. C&D | | | | | |

Summary

This chapter starts with the explanation of how the concept of ‘Roles’ is used in Social Network Analysis, and how this concept helps to structure this research. Next, the general research question is divided in four, more specific questions. Then it is explained which research techniques are used to gather information, namely: reading (on-line) literature, observation, snowballing, and semi-structured interviews with experts.

Out of these four questions, interview questions are formulated. In short, the interviewee is asked which roles he plays in the Open Source World. Then he is asked how the involved actors are related, and how they communicate, and which resources they use to communicate, in order to execute the Open Source equivalent of the processes described by ITIL.

Next, the way the obtained information is analysed, and how the interview scheme was altered to gain additional information is described.

Chapter 5: Results

In this chapter, the results will be presented. Subsequently, Configuration Management, Incident Management, Problem Management, Change Management and Software Control & Distribution will be dealt with. For each of the top-down ITIL processes, a description will be given of how the F/OSS world manages this particular process from the bottom-up. In each case will be described which actors are relevant, how they relate to each other, and which resources they use to perform the particular process (research question 2,3, and 4). For each process this also results in an illustration to clarify visually how the process is performed.

5.1 Configuration Management

In section 2.1 the mission of Configuration Management was described as ‘to control the means of production’. The process intends to identify and keep track of the ‘configuration items’ (CI’s) of which the infrastructure consists. Such configuration items can be pieces of hardware, but also collections of software etc. In the F/OSS world, both individual and collective actors are handling the means of production. This paragraph will explain how this is accomplished.

5.1.1 Actors

Individual actors

Although I do not have quantitative data about this, a very important part of the necessary work that is performed in the construction of F/OSS seems to be done by a group of people who do this from their computer at home. G. Stappers points out, what is also described by Eric Raymond as one of the main motivations for getting involved with open F/OSS: People do so because they want to ‘scratch a personal itch’. It seems obvious that these people will take care of keeping their computer up to date and making sure that they have the right hardware and software. They themselves install the necessary operating system, applications and tools.

Corporate Actors

Corporate Actors also perform activities that are similar to what Configuration Management intends to do. This is done in a variety of ways:

1. An interesting fact is that companies that distribute Linux and F/OSS on CD-ROM provide their buyers with the necessary developer tools, like compilers, editors, graphic programs, and complete development suites. When seeing the contents of distributions of for example Suse, Red Hat and Caldera, they all have tools like Gcc, Vi and Kdevelop present in some of their distributions¹². This basically means that these companies are involved with managing the means of production of the F/OSS world. The tools just mentioned are all tools that are used to develop F/OSS. In a traditional, top-down organisation, management would organise in some way that the employees have the necessary means of production. In this case, the distributors supply these means, namely, the software development tools necessary to prolong the existence of the F/OSS world, and thus also their own existence.
2. Distributors like Red Hat also supply the means of production for their employees to work on F/OSS. Alan Cox is a good example to mention in this context. He is employed by Red Hat, but contributes a lot to the creation of the Linux Kernel. But also other companies like Digital¹³. According to H. v/d Kooij:

¹² <http://www.caldera.com>, <http://www.debian.org/>, <http://www.mandrakelinux.com/en/>, <http://www.redhat.com/>, http://www.suse.com/index_us.html, all of these distributors, and probably more, all include tools like Gcc, G++, Vi, Emacs, and lots more in their package.

¹³ Digital is now merged with Compaq.

“Digital has contributed a lot to make sure Linux could operate on any random piece of hardware. As far as I know, they had four developers working on this at the time.”

XS4all¹⁴, an Internet provider who uses SquirrelMail is also a good example of this category of companies. According to Mr. Teepe:

“It has happened more than once that XS4All has ‘opened a can of programmers’ to solve the problem that they had with SquirrelMail. They then e-mailed the solution to us”

This means that these companies take care of the means of production of these particular individuals who are contributing to the construction of F/OSS.

3. There are also companies who offer an environment for software developers. Sourceforge is probably the most important, but there are also others like Savannah¹⁵, and DevChannel¹⁶. It is possible to start a project on these sites, and there you can get all the necessary tools for development, like Gcc, Vi, Emacs or other editors. But also there are means of communication and administration like CVS, mailing lists and bugtrackers.
4. The Linux Documentation project (LDP)¹⁷ can also be seen as a project that is dealing with some kind of configuration Management. The LDP is a site that keeps track of the documentation that is about Linux. Managing documentation is also one of the things that Configuration Management is responsible for. Mr R. Jansen and G. Stappers point out that a lot of documentation can be found here, although according to Mr. F. Mobach, this is mostly for people who are interested in the technical details, and not so much the end-users of F/OSS products.

According to W. Teepe, the companies that support F/OSS or are involved with F/OSS can roughly be divided into three categories:

1. The first category consist of companies that have to do with software development themselves: companies like Red Hat, Suse, Mandrake, and Caldera try to get the F/OSS that is available onto a distribution in its most elegant form.
2. The second category consists of companies that profit from the success of a certain Open Source Project. For example, companies like HP, IBM, Sun and VA software, which try to boost the use of their hardware by having free drivers available. (H. v/d Kooij). Also Zend¹ a company who creates software that improves the performance of web-sites. They found out that people were using their product to improve the performance of SquirrelMail.
3. The third category consists of companies that use the products. This category includes ISP’s like XS4all, but also universities and the like.

This categorisation has proven to be solid in later interviews, and will be used throughout this chapter.

¹⁴ <http://www.xs4all.nl/uk/> XS4all also uses the FreeBSD operating system. On their website it is also possible to find the reasons for their use of F/OSS. These are of an ideological reasons as well as reasons of superior quality, according to XS4all.

¹⁵ <http://savannah.gnu.org/> Similar to Sourceforge, except that this site hosts about 1200 projects, and has approx. 12.000 users. Also there is a difference in ideology, as Savannah is rooted in the FSF and not in the open source philosophy.

¹⁶ <http://hpc.devchannel.org/> Also Similar to Sourceforge, but more suitable for larger projects, like Linux.

¹⁷ <http://www.tldp.org/> On this site a lot of documentation can be found. The documentation is written in several languages, and has a lot of mirrors, located al over the world.

5.1.2 Relations

- Companies of category 1, like Suse and Red Hat relate to other actors because they build distributions that contain many tools that are used by people who develop F/OSS. These collections of the Linux operating system, together with many F/OSS applications are 'physical', meaning they can be purchased in shops.
- Companies of category 2, like VA Software, provide hardware. Basically, these companies also provide material resources, although the hardware and software they provide is not moved or transferred in the physical sense.
- Companies of category 3 also provide material resources, in the sense that their employees use the means of production that are made available by the company. People who work for these companies, use many of the companies means of production (like PC, connection to the Internet etc.) to create F/OSS.

Of course, for providing the necessary means of production, people are not only related by 'transferring material resources'. For example:

- F/OSS projects provide each other with many kinds of tools. These tools, that are often also present in many of the distributions, can be seen as non-material resources because they are sent over the Internet. So in this case the relational tie can be classified as the transfer of non-material resources.

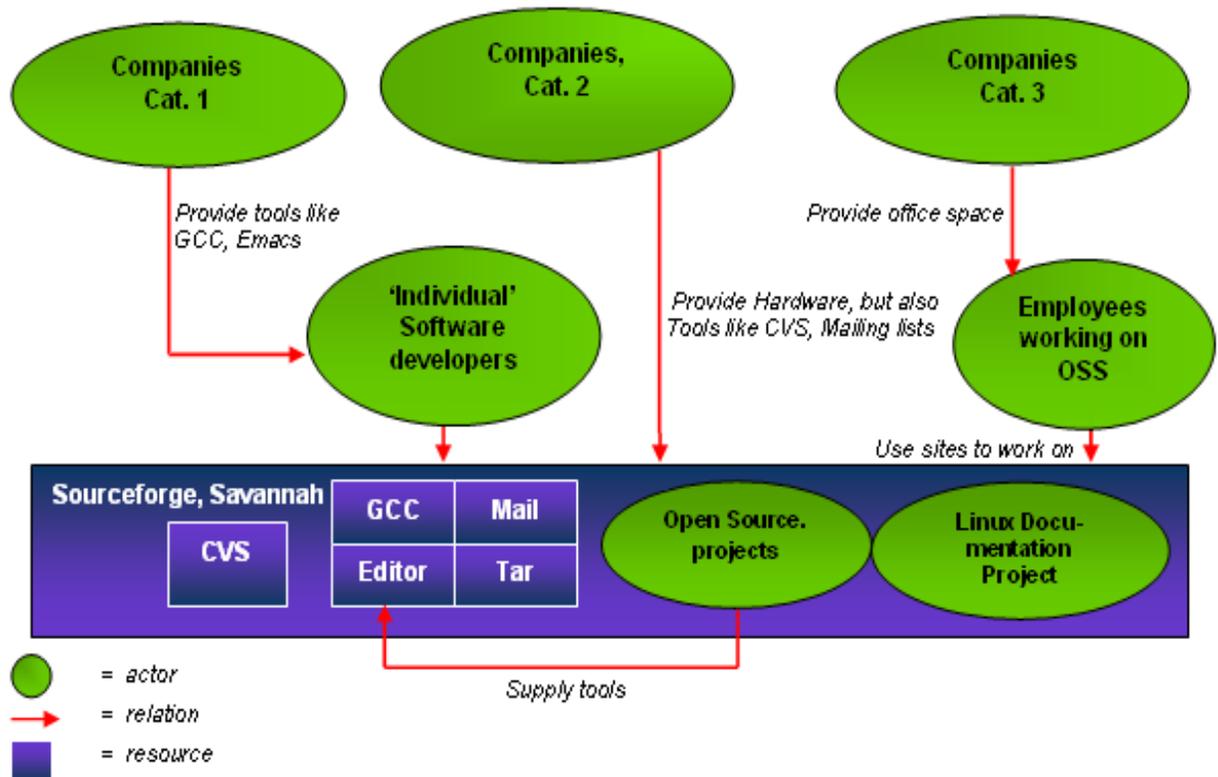
5.1.3 Resources

The following resources are mentioned here because they either give information about how the actors perform the F/OSS world's equivalent of configuration management, or because they can be considered as Configuration Items (CI's).

1. A site that was considered important by many of the interviewees is Sourceforge (H. V/d Kooij, W. Teepe, B. Wijnen.) In the same category are Savannah and Conglomerate. These sites are hosted by companies like VA software. This means that these companies provide and maintain the hardware and software like databases, tools, server software etc. that is necessary to make these sites operational.
2. An important tool that is used a lot by the F/OSS community is CVS. This tool allows people to work on software simultaneously, and integrates the changes made by the different programmers. Tools that are used are for example Gcc, Vi- editor, Emacs. Etc. These tools are mostly created by other F/OSS projects, and either downloaded from their site, or installed from a distribution.

The following figure will clarify how the actors are related, and which resources are being used.

Figure 5.a Configuration Management in the F/OSS world



Explanation of figure 5.a

This picture demonstrates in a simplified way how the F/OSS world manages their means of production. Obviously, some interconnections between actors are left out so the picture should be easier to understand. In reality, for example, it is also companies like Red Hat, and Suse (cat 1.) who in fact have people working (sometimes fulltime) on F/OSS projects.

Various interesting facts are shown: note that companies of category 1 (page 22) provide buyers of their distributions with various means of producing their own (F/OSS) software: compilers, editors, etc.

Companies of the second category benefit from the success of F/OSS because this increases the use of their hardware (if there is software that is compatible with, or controls their hardware, this will also increase the sale of this hardware). Also, contribution to F/OSS projects is good publicity. VA software, a company that seems to contribute for both these reasons, seems a good example of this. VA software is the company that makes Sourceforge available, and also maintains the server, and offers support for developers who contribute to projects on Sourceforge.

Companies of category 3 provide 'Office Space'. These companies contribute to F/OSS because they use it. When necessary these companies have some of their developers contribute to the F/OSS projects. Because these developers do so in "the boss' time", their employer is the actor that provides the necessary hardware, like a computer, tools and connection to the Internet.

What also happens is that F/OSS projects download these tools from other F/OSS projects. This is also mentioned in section 5.1.2. According to W. Teepe:

"Concerning editors; everybody has it's own taste. You can fight holy wars over this, but we're not going to, so everybody just uses what he likes best. Because it's all F/OSS, and open standards are being used, it works very well"

5.2 Incident Management

In research published on the F/OSS Research Community¹⁸, K.R. Lakhani and E. Von Hippel explore how, and why the ‘mundane but necessary task of field support’ is executed in the case of Apache¹⁹. In their definition; ‘Field support involves provision of assistance to users having difficulties with a product (..) because of defects in the product itself or because of the state of the user’s own understanding.’ (Lakhani & von Hippel, 2002). This user-to-user field support is in fact one of the F/OSS world’s alternatives for Incident Management as it described by ITIL. One of the differences is that in the latter case, Incident Management is expected to be performed by a service desk²⁰. This paragraph will show that although there is not a ‘helpdesk that can be contacted’, the F/OSS world does have very interesting ways to accomplish the same as what ITIL’s Incident Management sets out to do.

5.2.1 Actors

Individual actors

Lakhani and von Hippel (2002) stated that 50 % of the answers were provided by 2% of all ‘providers’, and that in general frequent information providers appeared to have more expertise than others. W. Teepe added to this that especially experienced users are important in this process. He explained that on the mailing lists of SquirrelMail, there are a lot of people who do not write software, but have installed the package so often that they know a lot about which settings to adjust, where possible errors may occur, and where difficulties may appear.

As all interviewees pointed out, it is also a custom to look for someone’s email address in the source code. It must be pointed out that this is not always appreciated by the developer in question. In general, it appears that this option is appreciated if the person who emails has already been searching for the cause of the problem, or a possible solution. If someone emails the programmer with ‘dumb’ questions, he might be asked to post his question on the mailing list, or risk the response: ‘RTFM!’²¹. (W. Teepe, R. Jansen, B. Wijnen)

Corporate actors

1. Companies who create distributions²² (category 1, see page 22) are important actors here as well. This is mentioned by several of the interviewees. A lot of the user to user assistance occurs on their mailing lists.
2. Distributors also offer support by means of telephone or email. This is part of a generally limited time warranty that is made available to purchasers of the distributions.
3. User groups, like the NLLGG also have a lot of mailing lists. Some of them generate a lot of mail. World-wide, there are about 500 of these user groups, in 80 different countries²³
4. IRC, Internet Relay Chat is mentioned by B. Wijnen to get questions answered.
5. Usenet is also mentioned as important for Apache Support²⁴ (Lakhani & von Hippel 2002)

¹⁸ <http://opensource.mit.edu/>

¹⁹ <http://www.apache.org/> Apache is F/OSS and currently the most popular web server software on the internet, used by more than 60% of the 8 million World Wide Web sites. (Prettejohn, 2001 in: Lakhani and von Hippel, 2002)

²⁰ The research by Lakhani & von Hippel indicates that users appreciate this Apache support as ‘somewhat better than that of it’s major commercial rivals in the server software field.’ (Ibid, page 16).

²¹ ‘Read The F-word-of-choice Manual! (Fine, or, F\$@#% ^!)’

²² <http://www.suse.de/en/private/support/maillinglists/index.html> is a good example. There is a warning on the website, some of the mailing list can create up to 200 messages a day.

²³ <http://lugww.counter.li.org/> this is the Linux User Groups worldwide homepage. Links to these user groups can be found here.

²⁴ <http://groups.google.com/> Apache is one of many open source projects you can find support for here.

5.2.2 Relations

Although a lot of incidents are discussed on-line, using a variety of resources that will be discussed shortly, there are also meetings of people who are interested in Linux. The NLLGG has approximately one meeting every two months. During these meeting users also help each other with problems they are experiencing. The NLLGG also organises ‘install parties’. This means that they visit a company, or hire a location where they install Linux on computers that visitors can bring along. Assistance is provided as well on such occasions.

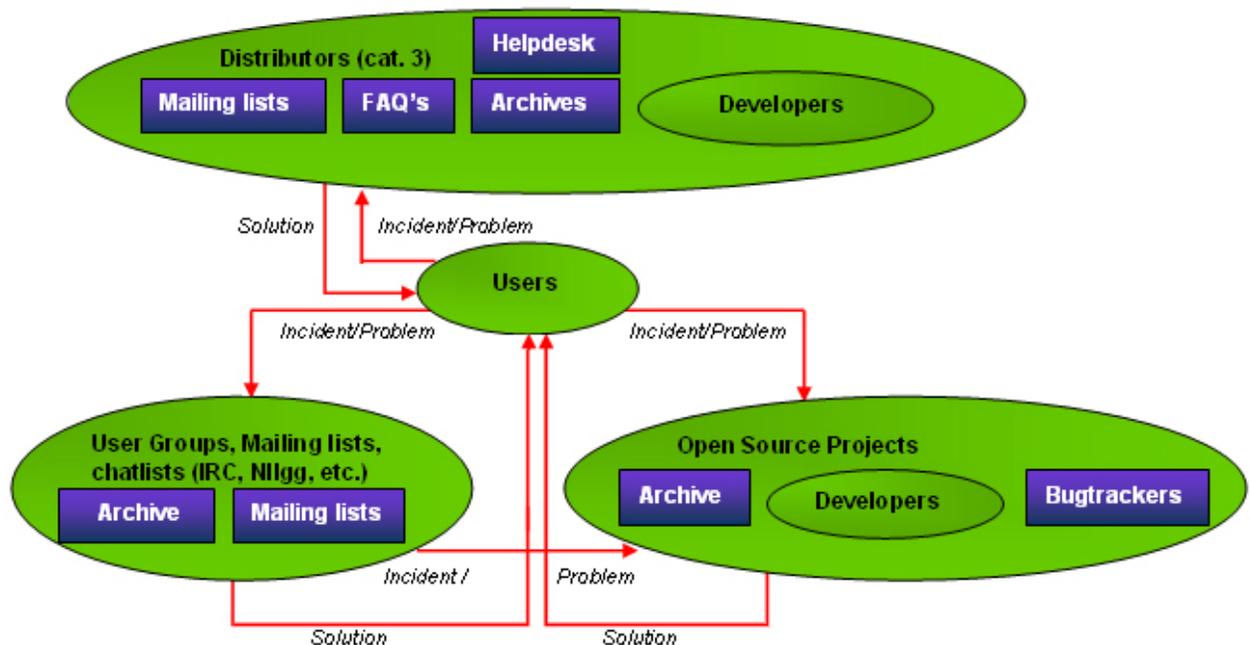
5.2.3 Resources

In earlier research, it has already been pointed out that there are several resources available for managing incidents.

1. There are often lists of FAQ's, Frequently Asked Questions present. In these FAQ lists, particular questions that resurface often, are listed.
2. The mailing lists are also an important tool. On these mailing lists people can post questions so everyone who is subscribed to these mailing lists can read the questions, answer these questions, or read answer provided by others.
3. Bugtrackers. These are a type of database application that make it possible to record a bug in detail, describe how it is processed, where a fix can be found, etc.
4. Archives are also present for a lot of mailing list. By using archives, if a problem has been posted and answered on a public site, it can always be traced by search engines like Google, Altavista, etc.

The following figure clarifies how incidents are managed in the F/OSS world:

Figure 5.b Incident Management in the F/OSS world



Explanation of figure 5.b

This picture clarifies what options a user has to get support in case of emergency. Again, the picture gives a simplified view of reality, to keep it understandable and easy to read. I.e. some user groups also have FAQ's, and not all F/OSS projects will be part of a distribution. But for the

more popular F/OSS projects, which are also added to distributions like Red Hat and Suse, there seems to be an overwhelming amount of communication and support.

W. Teepe stresses that not all support is of the same quality. There are often a lot of ‘mirrors’ of the mailing list and also sites like bugtrack²⁵, where security issues can be addressed. But not all of these sites are equally good in updating their information about the status of the bugs. The best place to go to remains the site of the project itself.

NB The fact that in the illustration the ‘user’ seems the only person who contacts the different mailing lists and FAQ’s suggests that only individual users can use F/OSS, and that it is not interesting for companies. It may also suggest that if companies do use F/OSS products and incidents occur, the user must contact these mailing lists himself. This is definitely not the case. As a matter of fact, support when using F/OSS is no different than support from companies who sell proprietary software according to F. Mobach.

According to him, if F/OSS is used by a company, the users can just contact the helpdesk of their own company, as they normally do with closed source software. The people working on the helpdesk and the BackOffice, are the people that have installed, tested and used the application, before it was actually taken into production. The people from the helpdesk will know how to handle most incidents. This finding opposes to the often heard objection that F/OSS can only be used by people ‘who are very good with computers’.

R. Jansen adds to this that even if the helpdesk is not able to solve the problem, the helpdesk has far more possibilities to solve the problem than they have with closed source software. According to him, with closed software you can only fall back on the company who has made the software, and often you do not even get in touch with the person who has actually written the software. In the case of F/OSS, he argues, you can get in touch with the person who has actually written the software. But even if this fails, there are still numerous other ways to get the problem fixed. Because it’s open, anyone with coding skills can look for the problem. If a company does not have these people available, it can be someone on the Internet, or a contract can be made with companies like Suse, or just a local computer expert.

²⁵ <http://www.bugtrack.com/>

5.3 Problem Management

Linus' law states that: 'given enough eyeballs, all bugs are shallow'. It seems indeed that because of its open character, problem management seems to be well taken care of with the creation of F/OSS. The distinction between an incident and a problem is not always clear from the beginning. This is so because before a bug is identified, sometimes extensive communication is required (W. Teepe). So during communication about what starts as an incident, the incident can turn out to be caused by a bug. For example, this discussion can take place on the mailing list of a user group, or at the site of the distributor. If it appears to be a bug, the problem can be 'escalated' (see footnote page 9) to the developer in question, or a bug report can be made at the site of the project. Because some of the interviewees were closely involved with this process, I have been able to not only ask about the essence of Problem Management, but also get into the details a bit more.

5.3.1. Actors

Individual actors

1. As stated, when an incident has been discussed, and it turns out to be caused by a bug, someone can address this at the relevant F/OSS project. H. v/d Kooij remembers that this has occurred several times on the mailing list. In this case, an individual within an organisation like the NLLGG performs the 'escalation'.
2. If someone is capable of doing so, he can also create the solution himself. If he brings the solution to the F/OSS project in question, they will make the change in CVS, or in some cases they will grant him access to CVS, so he can make the change himself (W. Teepe, R. Brink, H. v/d Kooij, F. Mobach, R. Jansen).
3. The developers of the F/OSS projects are also very important for finding and solving problems. Often they are the people who use the application with the most extreme settings, and because of this they discover difficulties themselves (W. Teepe).

Corporate actors

1. Companies of 'category 1' will also play a role here. As was already described in section 5.1, these companies often have people working on F/OSS projects. According to H. v/d Kooij, there are often extensive contacts between people of these companies and F/OSS projects.
2. Companies of category 2 also do some programming to F/OSS projects. XS4All is a good example of this (W. Teepe), but there are lots more.
3. Most problems are fixed by the people who are involved with the F/OSS projects. Within these projects, there is often a certain 'division of labour' on how the problems are solved (W. Teepe).

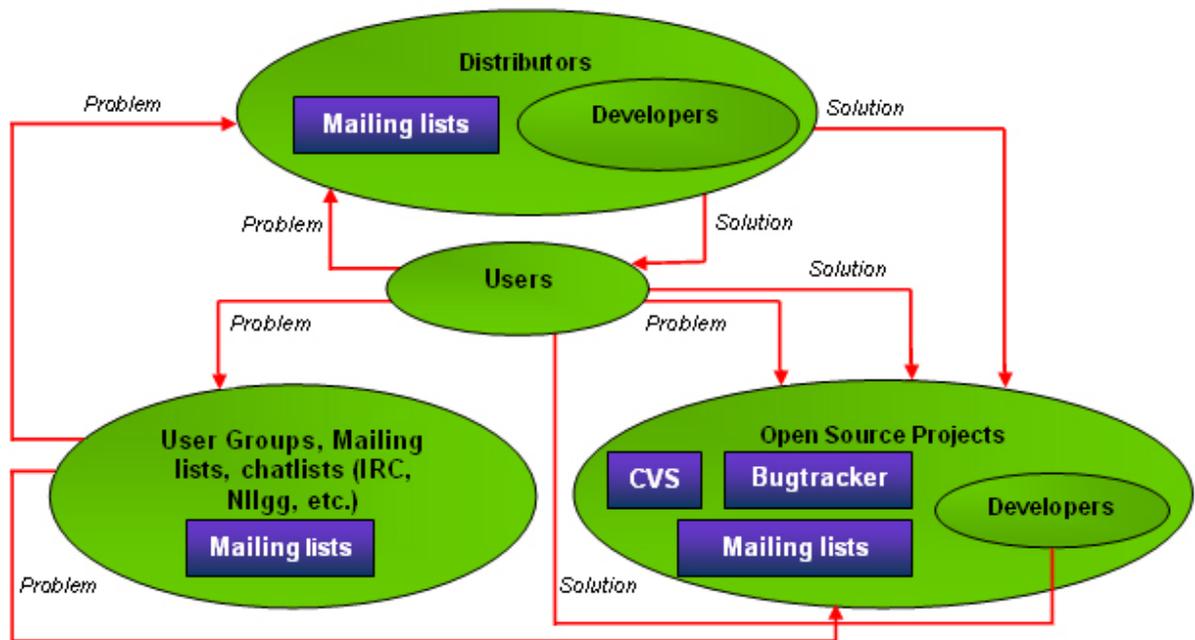
5.3.2 Relations

With Problem Management, the same actors are involved as with Incident Management. A lot of information is exchanged using resources on the internet. For example, users address problems at the mailing list of a user group. When discussing the problem, it can turn out to be a bug. Someone can then contact the F/OSS project that has built the application that contains the bug. Of course, there are many other possibilities. Users sometimes directly contact the project or they provide a solution themselves. Distributors can also provide solutions to users, or to the F/OSS project.

5.3.3 Resources

1. Bugtrackers are used. For example, problems can be appointed automatically or manually to people with these bugtrackers (B. Wijnen). Bugzilla²⁶ is a good example of such a device (F. Mobach).
2. Mailing lists. There are several types of mailing lists and discussion groups. The NLLGG alone has about ten. The developer-mailing list that is present at most F/OSS projects seems to be most relevant for problem management. Here, the programmers themselves have discussions about problems, and divide tasks amongst themselves (W. Teepe) (B. Wijnen).

Figure 5.c Problem Management
In the F/OSS world



Explanation of 5.c

This picture explains where problems are addressed, and who provides the solutions, and what resources are used in the process. Notice that this picture is again a simplification. For example, companies of the second category are also involved (companies that profit from the success of a certain Open Source project, page 23).

P.S. W. Teepe has received dozens of mails where people complain about SquirrelMail, and want the problem to be solved within fifteen minutes, or else... W. Teepe acknowledges that this can be very frustrating²⁷. Most people contributing to SquirrelMail do so for fun, and do so in their spare time, and when people become demanding, this can be annoying. This example does underline the issue that because F/OSS projects are based on voluntary work, it can be difficult to hold anyone accountable, and have a warranty that the problem will get fixed. This is in fact an often-heard objection to the use of F/OSS, that was also briefly mentioned in the introduction of this thesis. Several of the interviewees have in fact

²⁶ <http://www.bugzilla.org/> according to themselves, Bugzilla 'is now the de-facto standard defect-tracking system against which all others are measured'.

²⁷ W. Teepe sometimes replies that if they are not happy, they can always get their money back!

countered this objection with several arguments. According to W. Teepe, R. Jansen, F. Mobach and R. Brink:

- Security issues are dealt with very swiftly. Although F/OSS projects can not be held accountable for problems with the software, apparently there is a code of honour surrounding security issues.
- Because the software is open, anyone with coding skills can solve the problem. This can be someone from within the organisation, or someone recruited on the Internet.
- The previous arguments are not really a ‘warranty’ in the sense that someone can be held accountable, but merely that most problems are solved quickly or could be solved quickly if necessary. Nevertheless, it is even possible to make a deal with an outside organisation. Companies like Suse and Red Hat offer support contracts, but also a local computer expert can be contracted for this²⁸.

N.B. Updating the relevant documentation after incidents or problems have occurred is also an activity that should be done by Problem Management (ITSMF 1999). According to W. Teepe people at SquirrelMail try to do this, but this is a difficult process:

‘I like to tell people RTFM, but I would like to be more secure that the solution is really there!’

²⁸ <http://www2.mobach.com/> is F. Mobach’s company (one of the interviewees).

5.4 Change Management

The methods and techniques that are used to manage changes in the F/OSS world may not be as standardised and well defined as Change Management according to ITIL, but to say changes are made in a chaotic fashion seems untrue. At least in the case of SquirrelMail, changes are made in a very structured, and professional way. For example, there are certain procedures, like 'Feature Freeze, Code Freeze, and String freeze, in use. This means that the involved actors have an agreement that during a certain period, no changes can be made to features of the application, or to the source code, or to the strings that are part of the code. Furthermore, CVS is used, which is a technical means to structure the changes that are made. The contributors to the F/OSS project divide the work amongst themselves. And, as with many other F/OSS projects, the parallel release structure is used. I will not go into the details of this, but it is basically a means to balance innovation with stability²⁹. By having a stable and an instable release simultaneously, the instable release can be used to try new innovations, while the stable release contains only features that have already proven to be reliable. All these procedures and tools that are used are in fact ways of 'Managing Changes'. The involved actors, the relations between these actors, and the resources that are used will now be described so a light is shed on how changes are managed in the F/OSS world.

5.4.1 Actors

Individual actors

1. According to W. Teepe, the developers who are involved with the actual project are probably the most important actors for Change Management.

"Typically, a programmer makes a change, just because he wants to, and suddenly it's there. And if it's a nice feature, why not keep it? Luckily, we have a plug-in system, to keep things under control a little bit"

2. The people who use SquirrelMail, en these include also the developers, often use it in completely different ways. According to W. Teepe:

"Personally, I have about 800 mail folders, in which I keep about 5000 mail. That is a peculiar way of using SquirrelMail where certain limits are being tested" .

According to W. Teepe, other people will use the program with other extremes. If these individuals just use SquirrelMail for themselves, it is already being tested quite extensively. (Note that testing is one of the things ITIL's Change Management aims to do (see section 2.4).

3. A lot of the requests for change are being made by individual users. To meet their wishes, there is a wishlist on the SquirrelMail homepage. With these wishlists, people can request what they want. However, these wishlists are 'everlasting long' according to W. Teepe.

Corporate actors

1. The F/OSS projects seem to be most important for changes that are made. This means that people involved with these projects do execute a lot of the changes, and also do a lot of the testing individually. But sometimes the people involved also co-ordinate their efforts. In the case of SquirrelMail, there are about 7 taskforces. W. Teepe is one of the so-called 'HeadNuts' of one of these teams. These headnuts assign who will fix which bug. Although this does not mean that this is forceful in any way. It's more a matter of matching the right bug with the most suitable person to fix it (W. Teepe).

²⁹ For more information about the parallel release structure of many open source projects, read: Dafermos, Management & virtual decentralized networks: The Linux Project.

2. F/OSS projects also seem to be organising test activities. At least in the case of SquirrelMail procedures like Code-freeze and Feature Freeze are used during these activities. These procedures that are explained briefly in the introduction of this section were indeed recognised by the interviewees after W. Teepe mentioned them in his second interview. In these final stages before a new release, the SquirrelMail project announces that there is a new version to be tested, that people can download from the site. With the first half an hour there are sometimes 600 downloads. These people are then urged to test it to the extremes, and report errors back.
3. In some cases, changes have to be made that involve many developers simultaneously. In these cases there is often an extensive discussion by the developers in question. (H. v/d Kooij).
4. There are also other constructions possible. If people want a certain functionality for a program, they can strike a deal with a distributor or a local computer expert to simply build this. According to F. Mobach, & R. Jansen, there are many possibilities here. Distributors can also play a role here³⁰, but it is also possible to recruit someone on the Internet.

5.4.2 Relations

Change Management, as it is executed by the F/OSS community, is executed completely on the Internet. This occurs in several ways:

- New releases are announced on Mailing lists
- F/OSS is downloaded from the Internet to be tested.
- Requests for Change are posted on the internet using wishlists
- Changes are co-ordinated using Bugtrackers, CVS and Mailing lists.
- Users report problems back to the F/OSS projects.

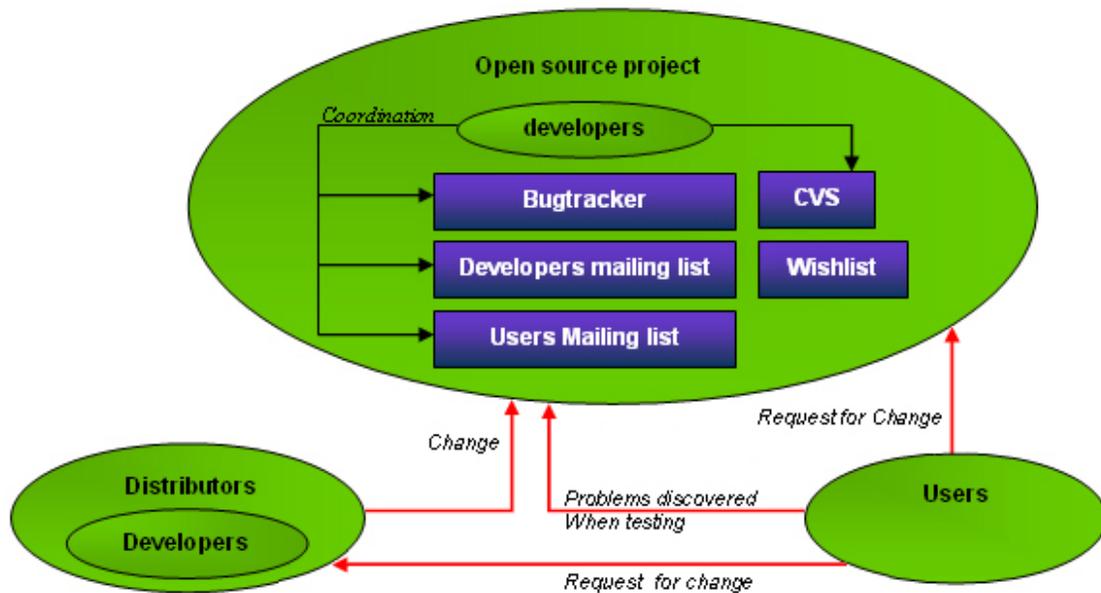
5.4.3 resources

1. Wish Lists on F/OSS projects are used to record wishes from users.
2. CVS is used as a co-ordination mechanism. According to W. Teepe, projects are very different in the amount of people they grant access to CVS. In the case of SquirrelMail, there are about 60 people who have access. In other projects there are often less people with access. CVS is very important for Change Management it seems. CVS is in fact a tool that helps to co-ordinate the efforts made by the different developers. Also the 'Fall-back plan' as ITIL describes as being very important for Change Management, is apparently very easy when using CVS (W. Teepe).
3. The Mailing lists are also very important. Especially the developers list, where the developers discuss changes etc. But also the announce mailing list that is used to tell people about new releases to be tested, etc.
4. Bugtrackers, also used for Problem Management, can also be seen as a tool to co-ordinate changes. Bugtrackers are applications built on database technology. Bugs can be administrated, and can be given a certain status like: 'assigned to...', or 'solved' (B. Wijnen).

³⁰ <http://www.suse.com/us/business/services/support/index.html> For example, Suse is advertising on their web-site with a variety of support they offer.

The following figure illustrates how changes are managed in the F/OSS world:

Figure 5.d Change Management in the F/OSS world



Explanation of figure 5.d

Again, the illustration is a simplification of reality. In the picture it seems for example that individual users are the only people who make requests for change. In reality, people working in companies can also do these request for changes. Also a lot of changes are made because the developers of the F/OSS project desire this. How the changes are co-ordinated varies between the different projects, according to F. Mobach. The smaller projects will do this mostly by using their developers mailing list. The larger projects may need more sophisticated tools like CVS and bugtrackers.

5.5 Software Control & Distribution

Software Control & Distribution is a process that seems to be handled particularly well in the F/OSS world. In Eric Raymond's book *the cathedral & the bazaar*, the reasons for this are discussed. The strategy 'release early, release often', as it is called by Eric Raymond seems to be used often by F/OSS projects. At least, this is the case with SquirrelMail. But apart from this, one could argue that Software Control & distribution is handled so well in the F/OSS world, because of its open nature, and the ever improving information technology, that allows people to download new releases instantly. Also, the fact that not only the volunteers of F/OSS projects are involved here but that the commercial world also contributes a great deal to this process seems relevant. This way, new releases can be downloaded from the Internet, as well as purchased in a store, along with other F/OSS.

Anyway, the actual reasons for the success of this process fall outside the focus of this thesis. For the last process of ITIL that is being discussed in this thesis, again ITIL seemed a useful tool to gather knowledge about the ins and outs of how the F/OSS world controls its software and creates distributions.

5.5.1 Actors

Individual actors

It seems that this process is quite structured, and involves mainly the F/OSS project as a whole, or other corporate actors as a whole.

One of the interviewees was occasionally sent a distribution by Suse, so he could test if the semi-automatic installation was functioning well. He was then asked to report errors back to Suse. In exchange, he got the distribution for free, and apparently received another small gift. (M. Mobach). R. Jansen & F. Mobach acknowledged that similar things happen quite often in the F/OSS world. More on this in 'Homesteading the Noosphere' by E. Raymond³¹.

Corporate actors

1. The F/OSS projects themselves seem to be the most important actor for building new releases (see section 2.5) of the software that is created by the project. A lot of the F/OSS projects are using the parallel release structure briefly explained in section 5.4. This means that they also distribute the software themselves by announcing on their mailing lists that new releases or patches can be downloaded from their site. The F/OSS projects are also responsible for other activities that Software Control and distribution describes: managing the definite software Library (DSL). In this case this means controlling CVS, that can in fact be seen as a DSL, although CVS has more features than a DSL.
2. Distributors like Suse, Red Hat, Caldera, Slackware seem to play a very important role here. What they typically do is scout the web, looking for interesting software. (W. Teepe) Although their role is different in every F/OSS project, in some of the projects, they actually are also involved with the development (see Problem Management). In the case of Red Hat, they are involved with the Kernel³². In the case of Suse, they are very involved with KDE³³. According to H. v/d Kooij, there are sometimes close connections between the people of the distributors and some of the F/OSS projects. In this case it is relevant that they gather interesting software on the Internet that are packed together on a CD-ROM and sold for commercial purposes. This is basically distribution, a sub-process of Software Control & Distribution.

³¹ Part of E. Raymond's book 'the cathedral & the bazaar, which contains 5 articles, including 'Homesteading the Noosphere, also available on the internet:

http://www.firstmonday.dk/issues/issue3_10/rammond/#d2

³² <http://www.kernel.org/> This is the primary site for the Linux Kernel, the heart of Linux.

³³ <http://www.kde.org/> KDE is an Open Source graphical desktop environment for Unix or Linux workstations.

3. Possibly, some of the category 2 companies can also play an interesting role. W. Teepe told of a construction they had with XS4all, where XS4all basically carried out a sub-process of software control & distribution. XS4all had created a test environment that downloaded the latest version of SquirrelMail every fifteen minutes. This way, developers from SquirrelMail could make changes, and fifteen minutes later immediately see and test the result in the test environment of XS4all. The distribution of new releases and changes that are made were handled automatically by XS4all.

5.5.2 Relations

The relational ties between the involved actors are partly the transfer of non-material resources:

- New releases are announced on mailing lists and newsgroups.
- The co-ordination that is necessary for making new releases and distributing them, is done on the developers mailing list.
- Problems discovered when testing are reported back to the F/OSS projects, by email or mailing list.
- New releases are downloaded from the site of the F/OSS project.

There are also relational ties present of a more material nature:

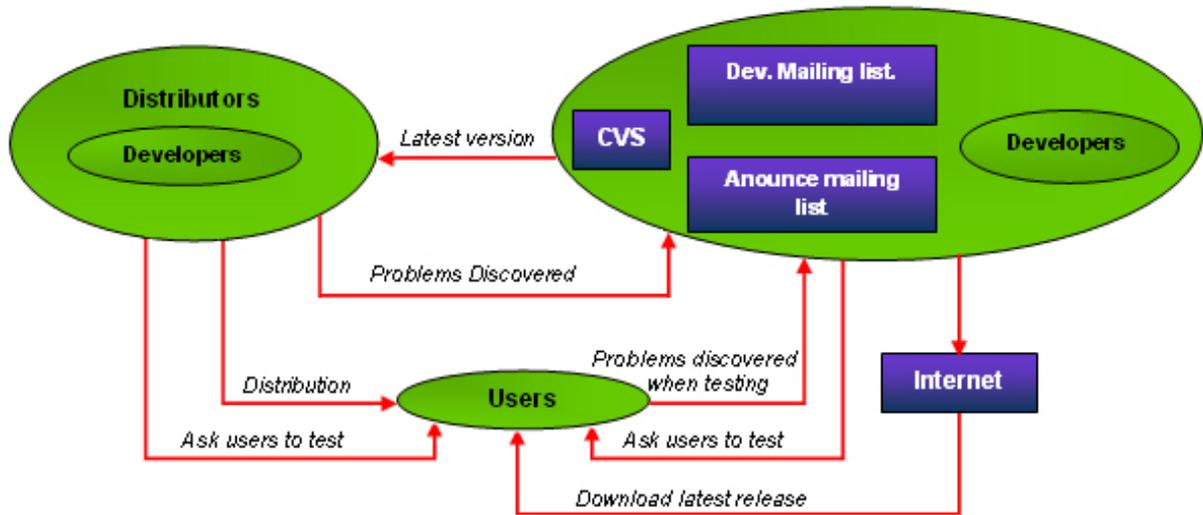
- Real, tangible distributions are constructed by Distributors.
- Distributions are sent to volunteers for testing.

5.5.3 Resources

1. CVS serves as a Definite Software Library. This is according to ITIL a place where the latest software is stored.
2. Announce mailing list. When a new release is due, it can be announced here, so people can be gathered for testing.
3. Newsgroups. New releases can also be announced here.
4. The developers mailing list is here also used as a co-ordination mechanism.

The following figure clarifies this:

Figure 5.e Software Control & distribution in the F/OSS world



Explanation of figure 5.e

It is important to note the different releases, and distributions there are. When an F/OSS project has a new release available, this is solely their own application. When a distributor has a new distribution, this means they have a collection of the latest stable releases of all the software they distribute. When users are testing for a distributor, they are looking at the installation process, and not so much at the functioning of the applications. The testing of an application however, is organised by the F/OSS project.

It is also important to emphasise the difference between a new distribution by a distributor, and the latest release from for example SquirrelMail. The distribution is physical, and can be purchased at a bookstore or computer shop. (except Debian). When SquirrelMail has a new release, this means that it is available for downloading from their site.

Notice also the difference between testing a release, and testing a change. Both change management and Software control & distribution do testing. In the case of change management, the actual change itself is being tested. In the case of Software control & distribution, the whole application, or the installation of the distribution is being tested³⁴.

Summary

In this chapter, the results are described. For every process that ITIL's service support describes, a description is given of how this process is performed in the Open Source World. For every process the involved individual and corporate actors are mentioned, how they are related, and which resources they use to interact.

Apart from individual actors who create or and/or use F/OSS, there are also corporate actors involved. These include user groups, but also open source projects on Sourceforge or Savannah for example. Furthermore, companies are involved with F/OSS in many ways. They provide an on-line development environment like Sourceforge, create distributions of F/OSS, contribute to open source projects and more.

Roughly, there are about three categories of organisations that contribute to F/OSS: The first category consists of companies like Red Hat and Suse, that distribute F/OSS. The second category contains companies like HP, IBM, Sun and VA software that profit from the success of F/OSS, because it can potentially increase the sales of their own products. The third category consists of organisations that use the open source products themselves. For example, Internet service providers like XS4all, and some universities.

The involved actors communicate and coordinate their efforts mostly by CMC (computer mediated communication). They use mailing lists, chat rooms, wishlists, bugtrackers, and tools like CVS (concurrent version system). The Internet is also used in a variety of other ways, like downloading tools, bugfixes etc. and making announcements.

Chapter 6: Conclusion

In this chapter, the questions that were asked in chapter 4 will be answered. Also, this chapter will discuss the usefulness of ITIL and Social Network Analysis for the investigation of the F/OSS world. Furthermore, problems that occurred during the investigation will be discussed and implications for further research will be offered. If anything, I hope the results presented in this thesis and the answers to the research questions will provide an insightful, and practical outline of the F/OSS world. Using a theoretical framework (SNA) combined with a more practical one (ITIL) I hope this thesis appeals to theoreticians but maybe even more to people who are working with software in a professional setting. Whether or not these people are familiar with ITIL and SNA, I hope this paper will improve their understanding of how the F/OSS community is performing the necessary processes to create and maintain reliable software. Ultimately, I hope to strengthen their believe that the F/OSS world is in fact capable of doing so.

6.1 Research Questions

Q1: What are the necessary activities that must be employed for the construction and maintenance of a computer system?

This question is answered in chapter 2. With the description of the service support set of ITIL, processes that are necessary for the construction and maintenance of a computer system are identified. In this chapter I also argued that these processes can also be distinguished in the world. The results show that indeed many of the processes that ITIL describes are performed somewhere in the F/OSS world.

Some of the processes were more difficult to grasp than others, especially Configuration Management. This process is different from Incident Management, Problem Management, and Change Management, because it is responsible for managing mostly physical, tangible resources. Therefore information about this process was more difficult to obtain, because it could not be traced using the internet, and none of the interviewees were closely related to companies like VA Software (in control of Sourceforge, where a lot of software development takes place).

Incident Management, Problem Management, and Change Management are all processes that are performed completely on the Internet, by means of computer mediated communication. Or at least, they could be, theoretically. This means that information about these processes can be found on the Internet. In the case of Incident Management, this is probably the easiest. Mailing lists where users help each other on the Internet are numerous, and it is easy to participate in them. Problem Management and Change Management are somewhat more difficult to trace on the Internet, because communication also occurs between two persons, or on mailing lists that are more private, and tools like CVS that are not always open to everybody.

Software Control & Distribution was somewhat different than the others because relational ties are both of a non-material and a material nature. New software can be downloaded from the Internet, but can also be purchased in a store. Also, companies like Suse apparently give these distributions away for free, in exchange for test-activities.

Q2: Which actors are performing the 'necessary activities' that ITIL describes?

Individual actors are very important for the F/OSS world. Whether or not they have multiple roles, and also contribute to F/OSS projects for example, individual actors play a very important part. They use the software, report problems back to the people who have created it, solve problems, make suggestions on wishlists, test the software, make changes, and support other users who are experiencing problems.

These individual actors can of course also participate in organisations and projects that are considered as corporate actors in this thesis. For example, the Dutch User Group NLLGG is such

an actor. On the meetings they organise, a lot of interesting information is exchanged, distributions are sold, training is provided, etc. The NLLGG also has many mailing lists where a lot of user-to-user assistance occurs, and problems are forwarded to F/OSS projects.

Of course the many F/OSS projects that can be found on the Internet are extremely important. These projects build a substantial amount of the F/OSS that is available. To do this properly, many things have to be organised. Some of these things occur outside of these projects, like Incident Management, that can happen also on mailing lists of amongst others, the NLLGG, just previously mentioned. But these things often also occur on the sites of the F/OSS projects, and much more. For example, changes that have to be made are co-ordinated and executed here, innovations are experimented with, bugs are fixed, releases are built, testing is organised by these F/OSS projects, etc. etc.

Of the corporate actors that are involved with F/OSS, their appear to be roughly three separate categories of companies that contribute to the F/OSS world. This categorisation is given by W. Teepe in section 5.1 (page 22), and seems very useful and is used in the other sections as well. In this first category are companies that contribute to F/OSS by making distributions, they benefit from this by selling these distributions. These companies also contribute in other ways. They often have people working on F/OSS projects, they provide mailing lists, and support.

The second category consists of companies that contribute to the F/OSS world, because they profit from the success of certain F/OSS projects. When for example drivers are created for certain hardware, this can boost the sales of this hardware. VA Software, Digital and IBM are all examples of this.

The third category consists of companies that use F/OSS themselves. These companies often contribute by having some of their employees working on the software, or by giving them an on-line test environment for example. XS4all is a good example of this.

Q3: How can the relations between the actors be described, in terms of the relational ties between them?

Perhaps not surprisingly with a subject as Information Technology, the relational ties between the involved actors can be described as the transfer of non-material resources, in this case, information. Whether it is user-to-user assistance, reporting problems, making a request for change on a wish list, discussing a problem, solving a problem, changing software, or downloading releases, it all has to do with altering information, and transferring it to other actors.

Mostly, the transfer of non-material resources is enough to keep the F/OSS world going. As mentioned before, Incident-, Problem-, and Change Management can all be executed on the Internet using Computer Mediated Communication. In some cases however, it is necessary that Material Resources are transferred, or at least, that actors have the possibility to make physical changes. If someone is experiencing problems on Sourceforge, and this is caused by for example a malfunctioning hard disk, this hard disk can only be replaced by someone who has physical access to that disk. For this reason, actors like VA software have a different position than other actors in the network, because they are the only actors who can make physical changes to the resources. Also, companies in category 1, like Suse and Red Hat, transfer material resources to other actors, like individual actors, by selling distributions.

Q4: What resources are being used by the involved actors to communicate, and co-ordinate their efforts?

The involved actors have many resources available for communication, and co-ordination.

- Widely used are the many different flavours of mailing lists. Developers use these for co-ordination, and discussion of problems. New releases are announced on announce mailing lists. Users use them for user-to-user assistance.
- Bugtrackers are used to report problems, keep track of the action that is undertaken to solve the problems, and assign them to people who can fix them.
- E-mail is used for more personal communication, for example, directly contacting the developer of a certain F/OSS application.
- Wishlists are used to archive requests for change.
- CVS, functions as a software library, as well as a means of co-ordination.

6.2 Discussion

Both ITIL and Social Network Theory have proven to be very useful tools. Although the processes of ITIL served as the main structure for the interviews, and basically as the starting point of this thesis, Social Network Analysis has proven to be very helpful in identifying which actors actually performed the ITIL processes, and how they were related. The possibility to see both individuals and groups as actors, was very important. In some cases, the outcome of individual actions was most important, but in others the outcome of more collective action was more relevant. For example, in the process of configuration management, the F/OSS projects as a whole are important, because they deliver certain tools. But in other processes, like change management, the focus is more on the actual individuals within those projects that perform the necessary activities.

Another concept that was useful was that of 'Roles'. During the interviews this provided structure, but it also gave the possibility to get more information out of the same individual. Often the interviewees were indeed involved with F/OSS in different ways. Because the interview scheme was prepared for this, the maximum amount of information was acquired.

There were also some difficulties encountered:

1. It was difficult to find people who were active in the F/OSS world, and also available for a face to face interview. Although it turned out to be a clever decision to become a member of the NLLGG, because I encountered several interesting people here who have been very helpful, it was still difficult to meet them for a face to face interview. Many NLLGG members were living far away from my hometown Groningen, and central meetings were only organised every two months. Unfortunately, of the people who were indeed active on-line, in one of the ten forums that the NLLGG uses, not all of them went to the meetings. By careful selection of the interviewees however, the people that were interviewed eventually, did indeed have a lot of expertise, fortunately.
2. It was hard to determine in advance, though, on what specific areas the interviewee's were most knowledgeable. Some of the interviewees were already familiar with ITIL. In effect, this was not always an advantage. In some cases, it turned out to be more difficult to explain how I intended to use ITIL combined with the F/OSS world, because they knew ITIL so clearly from its use in traditional organisations.
3. Because the results are based of interviews with only 9 people, it is of course difficult to generalise the findings. I did try to mention consistently how many people confirmed a certain finding. This does not have much power of expression however, because the total amount of interviewees was not very high. Also, a certain fact could surface in one of the later interviews, in which case it was difficult to determine whether or not the interviewees that were interviewed earlier confirmed these findings.

6.3 Further research

Despite of the limitations mentioned above, the study has tentatively used Social Network Analysis for analysing the F/OSS world, and SNA has proven to be a promising framework to do so. For example, the distinction that SNA makes with regard to Material and Non-material relational ties, has revealed the specific nature of Incident Management, Problem Management & Change Management. These processes seem to be executed completely using only Computer Mediated Communication and are entirely based on relational ties of a Non-material nature. Contrary to this, Configuration Management and Software Control & Distribution do have relational ties that involve exchange of material goods.

By using ITIL's service support it has become clear that processes ITIL identifies in the work-practice of formal, top-down organised organisations could also be distinguished in the F/OSS world. Service Support has proven to be a very useful blueprint for learning to understand the processes that take place in the F/OSS world. Considering that this thesis only 'scratched the surface' of Service Support, more detailed use of this ITIL module might be fruitful. Not to mention the other modules ITIL entails, like Service Delivery, or Application Management, that could also help to increase our understanding and use of the very promising phenomenon that F/OSS is.

Appendix A interviewees

(In order of appearance)

Wouter Teepe is a PhD of Artificial Intelligence at the RUG, the University of Groningen. He investigates security, data structures, and workflow-related subjects. He has created an online voting guide, (<http://www.stemadvies.com>) and is one of the ‘headnuts’ of SquirrelMail. He is also a teacher at the RUG.

Hugo van der Kooij is a Support Engineer and is specialised in Firewalls. The tools they use run on a Linux platform. He is one of the ‘founding fathers’ of the Dutch Linux User Group, the NLLGG.

Robert Brink works as a software developer and system administrator. Is an experienced user of F/OSS, but has not done any coding for projects. Is one of the early members of the NLLGG

Mendel Mobach apparently works with computers since the tender age of 2. Has been working with Linux since 1995 and has contributed to several F/OSS projects. Works for the centre of Mathematics and computer Science, and is a member of the NLLGG.

Bas Wijnen is a student and mostly politically involved with F/OSS. He is a frequent user of F/OSS. He also contributes to an F/OSS project that creates an MSX simulator.

Geert Stappers works as a system administrator with Linux amongst others, and is a Database Administrator. Is also one of the early members of the NLLGG.

Roeland Jansen is also one of the earlier NLLGG members, and does occasional testing for F/OSS projects, and system administration on different machines.

Fred Mobach is the founder of Systemhouse Mobach BV. and is working with Linux and F/OSS professionally for 6 years. Offers complete F/OSS solutions after first explaining to companies what the differences are between open-source and closed source software, and what this can mean for their business.

Erwin Mulder is also a teacher at Artificial Intelligence at the University of Groningen. Has been a user of Linux and F/OSS for 8 years. Has contributed to different F/OSS project, like the popular Multi User Dungeon games.

Appendix B interviews

Presented here is the first interview scheme that was constructed for the first four interviews, that served as a starting point for the following interviews.

Introduction

I am writing a thesis for the R.U.G, the Rijksuniversiteit roningen. The subject of my thesis is the description of the organisation of the Linux community. To grasp the processes that are going on in the Linux community, I am using concepts of 'ITIL'. ITIL stands for IT infrastructure Library, and it is a collection of 'best practices' that are used in the IT to structure the processes that are needed for any IT organisation to be successful. Of course, ITIL is only one of many models for structuring an IT organisation. It describes processes that have to be performed in an IT organisation to make it an efficient and effective organisation. Possibly, these processes do not always fit exactly on the Linux community, and sometimes it might also be difficult to understand immediately just what the processes actually describe. Of course, if something is not clear let me know, so I can clarify it.

Question 1: Personal Involvement

- First, can you describe to me how you are personally involved with Linux? When did you first hear from Linux, and when did you become a member of the NLLGG?
- Since then, have you done any programming for the Linux community, did you write any documentation? Did you contribute to mailing lists?

Question 2: Configuration management.

ITIL describes a process called Configuration Management. This process is defined as the process that is responsible for gaining control and maintaining control over all parts that constitute the IT infrastructure and also the control over the documentation that comes with these parts.

When this process is performed by an IT company, the company will be responsible for the administration of the parts that constitute the company's infrastructure. If the company builds certain applications, it will need to provide their employees with the necessary hardware, and also the software so the employees can perform their task.

In the case of Linux, things will be different. Of course, as far as your own PC is concerned, you will be responsible for your own 'mean of production'. But a lot of the maintenance and production also has to take place on the internet. Sourceforge is a good example, but also the NLLGG provides its members with a website and a place where documentation is stored.

- Are you personally involved with this process? For example, are you an active member of the NLLGG, and therefore involved with for example the maintenance of the web-site etc.?
- Do you have an idea what parts constitute the infrastructure that is being used by Linux? Which important Internet sites are active? Which mailing lists that are related to Linux? Do you know of other organisations like the NLLGG? Also, do you know something about the involvement of corporate actors, like Suse, Red Hat etc? What kind of virtual space do they offer on the Internet? What ways of interaction do they have?

Question 3: Incident Management

ITIL identifies a process called Incident Management. This process is responsible for dealing with all sorts of incidents as they occur with a certain application or in the infrastructure of an organisation. The nature of these incidents can be very versatile. For example, it can be a user who has a blank screen, or who is unable to start his computer, but also someone who wants to know how to use a certain feature of an application. Again, within an IT organisation, there is often a clearly visible and structured process that deals with these incidents. In the case of Linux, things are not quite so obvious.

- What are your personal experiences with this process? For example, can you recall having experienced malfunctions with Linux? How did you address this problem within the Linux community?
- Do you also recall having helped others who were experiencing problems? How did you help them? How did they get in touch with you, were did they address the problem?
- Do you know other important sites/people or perhaps organisations that you can go to when you are experiencing problems?

Question 4: Problem Management

In some cases, if there is an incident, this will mean that a user doesn't know how to use Linux, and needs advice on how to use Linux properly. The problem that the user is experiencing is in fact caused by a lack of knowledge from the user. There will also be cases when someone has stumbled on a bug. According to ITIL, when an incident is in fact caused by an actual bug, a process called "Problem Management" will become active. This means that the cause of the problem needs to be traced, and that the problem needs to be solved. If the problem is caused by a bug in the software, the bug needs to be fixed.

- Have you stumbled on bugs when using Linux? How did you address these bugs, and where? Have you ever discovered the cause of a malfunction and fixed the problem yourself? What happened with your solution of the problem? Did you send it somewhere by email, or was there another way to address the problem?
- Did you ever provide others with solutions to certain problems?
- Do you know how bugs are fixed in general? Do you know where problems can be addressed, and who will actually investigate the problem? How do the involved actors communicate with one another?

Question 5: Change Management

If an incident is caused by a bug, and if a solution is found, it is necessary to alter the software. Because the faulty program is probably used in a lot of different places, they all need to be changed.

Change Management is the process that is responsible for these changes to be made in an orderly fashion. Put more formally: Change Management is responsible for assuring that standardised methods and techniques are used so changes can be made in a direct and efficient manner.

- Are you personally involved with this process? Did you ever discover and change a bug, and did you actually make sure that other users became aware that this bug was there?
- Maybe you are not personally involved with this process, but do you know how this process is organised in general? Which actors are involved with this process? How do they communicate?

Question 6: Software Control & Distribution

If a program, or application has been used for a while, and several bugs have been discovered, and a lot of changes have been made to it, there will be different versions of the program that are being used. Somehow IT organisations must keep track of all these versions. ITIL has identified a process called Software Control & Distribution. Also Linux has a lot of different Distributions, and versions of programs that are used. Somewhere within the Linux community, a similar process like the one described by ITIL must be active.

- Do you know how Linux performs this process? Which actors actually play a role here? I'm familiar with a couple of corporate actors that seem to be important for making distributions of the available Linux versions and software for it. I'm talking about companies Like Suse, Red Hat etc.
- Do you know if these actors are indeed active with this process? Where do Suse and Red Hat actually get their software?

Literature/Sources

- Burt, R. & Minor, M. (1983) Applied network analysis London: Sage Publications.
- Dafermos, George N. (2001) Management & virtual decentralised networks: The Linux Project.
<http://opensource.mit.edu/home.html>
- Emirbayer, M. (1997) Manifesto for a Relational Sociology, American Journal of Sociology
- Faraj, S. (2001) The web of knowledge: An investigation of Knowledge Exchange in Networks of Practice.
<http://opensource.mit.edu/home.html>
- Flap, H., Bulder B., Volker, B. (1998) Intra organisational Networks and Performance: A review.
Kluwer academic publishers: the Netherlands
- Flick, U. (2002) An introduction to qualitative research. London: Sage Publications,
- Hahn, R. W. (2002) Government Policy toward F/OSS: An Overview
<http://opensource.mit.edu/home.html>
- Homans, G. C. (1951) The Human Group. London: Routledge & Kegan Paul LTD.
- ITSMF (IT Service Management Forum) (1999) IT service management, een introductie
(Introduction to IT service management) Zeewolde: ITSMF
- Koppens, S. Peters, L. Vonk, J. (2001) Operationeel beheer van informatiesystemen (Operational Management of information systems) Den Haag: ten Hagen & Stam.
- Krishnamurthy, S. (2002) Cave or Community? An empirical examination of 100 mature open source projects
<http://opensource.mit.edu/home.html>
- Lakhani, K.R., v. Hippel (2002) How F/OSS works: "Free" user-to-user assistance
<http://opensource.mit.edu/home.html>
- Madanmohan, T.R., Navelkar, S. (2002) Roles and knowledge Management in Online Technology Communities: An ethnography Study
<http://opensource.mit.edu/home.html>
- Matzat, M. (2001) The management of online communities Groningen: Rijksuniversiteit Groningen
- Microsoft & ITIL: <http://www.microsoft.com/traincert/syllabi/1737Afinal.asp>
<http://www.internetnewsbureau.com/archives/2002/june02/microsoft.htm>
- Mockus, A. Fielding, R.T. Herbsleb, J. (2002) A case study of F/OSS Development: The Apache Server
<http://opensource.mit.edu/home.html>
- Pedersen, S.T. (2002) Open source and the network society
<http://opensource.mit.edu/home.html>

- Raymond, E. (2000) The Cathedral & the Bazaar O'Reilly & Associates
- Seidman, I. E. (1998) Interviewing as qualitative research: a guide for researchers in education and the social sciences. New York: Teachers College Press.
- Stephenson, N. (1999) In the beginning...was the command-line New York: Morrow, William & Co.
- Tuomi, I. (2000) Internet, Innovation, and Open Source: Actors in the Network
<http://opensource.mit.edu/home.html>
- Wasserman, S. & Galaskiewicz, J. (1994) Advances in social network analysis London: Sage Publications.
- Wester, F., Smaling, A. Mulder, L. (2000) Praktijkgericht kwalitatief onderzoek Bussum: Coutinho b.v.
- Maso I., Smaling, A. (1998) Kwalitatief onderzoek: praktijk en theorie Amsterdam: Boom
- Wendel de Joode, R. van (2001) Providing goods on the internet, a research to collective action and co-ordination in communities on the Internet.
<http://opensource.mit.edu/home.html>
- Zhang, W. Storck, J. (2001) Peripheral Members in Online Communities
<http://opensource.mit.edu/home.html>