

Analysing the process of joining in KDE project

Copyright (cc) 2008 Pablo J. Sangiao Roca. Some rights reserved

This document is distributed under Creative Commons license Attribution-Share Alike 3.0 available in:

<http://creativecommons.org/licenses/by-sa/3.0/>.

COLLABORATORS

	<i>TITLE :</i> Analysing the process of joining in KDE project	<i>REFERENCE :</i>	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Pablo J. Sangiao Roca	January 2008	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Introduction	5
2	Related Research	5
3	Methodology	6
4	Results	6
5	Threats to validity	9
6	Conclusions and Further Work	10
7	References	10

List of Figures

1	Data gathered for group1 of developers	7
2	Mean calculated for group1 of developers	7
3	Data gathered for group2 of developers	7
4	Mean calculated for group2 of developers	8
5	Data gathered for group3 of developers	8
6	Mean calculated for group3 of developers	8
7	Data gathered for volunteers	9
8	Data gathered for professionals	9

Abstract

In *The cathedral and the Bazaar*, *Eric Raymond* told how the development in libre software¹ projects is similar to a Bazaar where a group of different people try to do their business and all together bring up a stable system. Nevertheless in the libre software projects exists really one kind of organization or process in order to become a member of the community. The Onion model try to identify this process establishing the phases that developers go through in order to become a members of the core group. In this model is assumed that developers start their activity as project users. Then they send some kind of messages to the mail lists of the project and maybe they can report some bug to Bug Trucker system. From here on they begin an active work, they usually start fixing bugs until get into development core group. In this paper it will be trying to analyse if the Onion model goes for KDE project and if it does, to find out how much time it is necessary to become a member of KDE core group.

1 Introduction

As it was said in the abstract, this study try to find out if the Onion model can be applied to KDE project. The study will be done for two different people: volunteer and professionals. Apparently, volunteers should follow the Onion model, because they work as a developers in their free time, but professionals should not follow it instead because they need to get into the core team as soon as they can. This last point has very importance because KDE project is based on a library called *Qt* property of *Trolltech*. In fact, a lot of KDE developers work for this company, include his creator *Mattias Ettrich*. Besides there are a group of companies involved in KDE under the name of *KDE League*. Most of them are *GNU/Linux* distributions that they are interested in collaborating with the development of KDE cause their business models. For that reason they hire developers to contribute with the development of KDE. From the other hand, the study will be done also taking into account the evolution of the libre software projects in relation with the time. It will be tried to find out if the Onion model is satisfied or not since very beginning of the project or if it is necessary to reach certain state of maturity in order to satisfy the onion model. The study will consist in monitor the public activity of the developers through the public resources available, that is, *Subversion*, *mailing list* and *Bug tracker system*. It will be try to get this information for the core developers and to measure the time from they sent their first message to some mailing list to they did their first commit.

2 Related Research

This paper is based in another one [1] which it was elaborated by Libresoft group. In that paper the authors try to measure the average time that developers need in order to get into the core group of GNOME project. That study has been elaborated taking into account the two different roles for developers: volunteer and professional. The Onion model is used also by the authors as a theoretical approximation about how the joining process in a libre software projects is. This model do not have to be always true. Libre software projects pass through some different phases during their life-cycle. In this phases the number of developers is changing. According to *The Cathedral and the Bazaar* is necessary that the number of developers grows up so that libre software projects are successful. This has been studied by Capiluppi and Michlmayr. In their paper [2] they have studied two

¹Through out this paper we will use the term “libre software” to refer to any code that conforms either to the definition of “free software” (according to the Free Software Foundation) or “open source software” (according to the Open Source Initiative).

libre software projects, Wine and Arla, in order to get information about their life-cycle, development process and communities. The conclusions of this paper are that "the Cathedral and the Bazaar phases are not mutually exclusive". According to this paper many libre software projects start as a Cathedral, with a small number of developers and then the community grow up until to reach the Bazaar phase. This point is directly related with the easiness to join in a libre software project when it is starting, this is, when it is in the Cathedral phase. As the time passes if the project get more developers and it changes to Bazaar phase probably it will be a bit more complicated to get into the project.

3 Methodology

The first step was getting the group of committers who were going to be under study. In order to do this, the strategy chosen was analysing the current *trunk* in the *Subversion* of KDE. Making use of CVSanaly² were obtained the top twenty committers more active in the last six months.

When the group of committers was chosen, the next step consisted of analysing their activity by means of the public means that libre software provide, these are, *mailing list*, *Subversion* and *Bug tracker systems*. In addition to use CVSanaly in order to analyse the *Subversion*, it was utilized Mailing Lists Stats³ to analyse the *mailing list*. The *Bug tracker system* used by KDE is Bugzilla⁴ which provide a web interface in order to do basic queries. This web interface was used to get the information about the first bug reported by the group of the *committers* under study.

In order to can check the Onion model it is necessary to know for each *committer* the date of their first *commit* and the date of their first contact with the project, normally through *mailing list* or *Bug tracker system*. With these data it can be calculated the mean in time that the developers need in order to get into the *core team*.

For each of the *committers* were measured the time that they needed to do their first *commit* since they have sent their first message to some of the *mailing list*. With these times expressed in months was calculated the mean also in months that a developer need to become in member of the core developers team.

In order to try to classify the developers according to their role, this is, if they are professional developers hired by some company to work developing the project or volunteers that collaborate with the project in their free time, the methodology was as follow:

KDE project by means of its web page⁵ publish some interviews⁶ that they make to the most important developers of the project. One of the questions is if they have been paid for their contributions to the KDE project. Making use of this resource it is possible to find out if a developer is professional or volunteer. For each member of the group under study was looked for the existence of this interview and they were classified according to their answer. Developers for whom it was impossible to get this interview were ruled out in this classification.

4 Results

Among the top twenty developers more active was ruled out one of them called "scripty" because it is a bot.

When the dates of the first *commit* was found out, it was seen clearly that exists three different groups differentiated regards to the time. A first group with the developers that made their first commit before 2001. KDE project started in 1997 and the version 2.0 was released in 2002. Taking into account this information in this group would be the developers who started the project. A second group would be composed of developers that made their first commit between 2001 and May 2005. This date, May 2005, is the creation date of the current trunk in which it is being developed the version 4.0. Besides, KDE 3.0 was released in 2002 so this group is composed of developers that made the version 3.0 and the next ones until version 3.5, the last one before starting the development of the version 4.0. The last group is composed of developers that made their first commit after May 2005, this is, in the current trunk so they only took part in the development of the version 4.0.

The results for each group are as follow:

²<http://tools.libresoft.es/cvsanaly>

³http://tools.libresoft.es/mailling_list_stats

⁴<http://www.bugzilla.org>

⁵<http://www.kde.org>

⁶<http://behindkde.org>

Committer_id	#Commits	#Messages	#Bugs	1st Commit	1st Message	1st Bug Reporter
21	24262	347	77	1998-03-10	1997-12-12	1999-02-06
34	23297	200	40	1999-08-24	2000-06-05	2000-10-03
41	12475	2856	23	1998-02-17	1999-03-01	2000-07-03
158	2265	16	15	2000-10-08	2000-08-01	2001-07-04
36	3310	43	38	2000-07-17	2002-08-21	2001-05-18
40	1749	14	3	2000-12-06	2004-08-06	2005-12-30

Figure 1: Data gathered for group1 of developers

Mean = -14.505
Standard Deviation = 18.12336
Variance = 328.4561
Max = 2.93
Min = -44.63

Figure 2: Mean calculated for group1 of developers

In this first group, *committers* who did their first *commit* before 2001, only two out of six committers sent their first message to some mail lists before doing their first *commit*. For that reason the mean is negative so it seems that the Onion model is not valid for this first group.

Committer_id	#Commits	#Messages	#Bugs	1st Commit	1st Message	1st Bug Reporter
28	1826	205	20	2002-07-25	2004-04-04	2003-01-27
48	1372	315	20	2002-01-18	2001-11-07	2002-03-30
23	3308	487	9	2002-01-08	1999-08-03	2001-12-17
91	2903	187	24	2003-10-25	1997-09-03	2003-07-09
12	4589	120	32	2004-09-08	2005-02-21	2004-12-11

Figure 3: Data gathered for group2 of developers

For the second group, composed of five *committers* only two of them have negative result, this is, they did their first *commit* before having sent a message to some of the mail lists.

Mean = 16.128
Standard Deviation = 37.51622
Variance = 1407.466
Max = 74.77
Min = -20.63

Figure 4: Mean calculated for group2 of developers

In this case the mean is positive although as it was said before, there are two negative results to distort the mean. Besides the standard deviation is elevated what it means that the population is not homogeneous. For these reasons it is not possible to say that this group follow completely the Onion model.

For the last group, the most recently taking into account the enter date into the core team, everyone have positive result, this is, all of them did their first *commit* after sending a message to some mail list. In this case, the Onion model seems to be valid.

Committer_id	#Commits	#Messages	#Bugs	1st Commit	1st Message	1st Bug Reporter
31	467	298	20	2007-01-26	2005-04-21	2005-04-21
50	909	99	1	2007-03-05	2007-02-04	2007-12-05
62	570	387	0	2007-04-20	2006-05-10	
63	1323	303	2	2005-12-11	2005-10-15	2004-09-24
65	676	208	2	2006-11-21	1999-07-01	2007-03-08
51	1173	203	74	2006-11-17	2003-06-20	2004-12-16

Figure 5: Data gathered for group3 of developers

The mean obtained in this case are twenty eight months. This piece of information must be taking with caution because if we look the standard deviation, we can see as in the last case that it is big enough what it means the values of our sample are not homogeneous.

Mean = 27.9
Standard Deviation = 33.92147
Variance = 1150.666
Max = 90
Min = 0.97

Figure 6: Mean calculated for group3 of developers

Next are detailed the results making the classification according to the role of each *committer* this is, volunteers or professionals. Among volunteers seem that the Onion model is satisfied. In this group have been identified six people and four of them have sent a message to some of the *mailing list* before have done their first commit.

Committer_id	#Commits	#Messages	#Bugs	1st Commit	1st Message	1st Bug Reporter
48	1372	315	20	2002-01-18	2001-11-07	2002-03-30
23	3308	487	9	2002-01-08	1999-08-03	2001-12-17
91	2903	187	24	2003-10-25	1997-09-03	2003-07-09
12	4589	120	32	2004-09-08	2005-02-21	2004-12-11
41	12475	2856	23	1998-02-17	1999-03-01	2000-07-03
158	2265	16	15	2000-10-08	2000-08-01	2001-07-04

Figure 7: Data gathered for volunteers

The mean obtained in this case is over fifteen months.

In the group of the professionals it has been able to be only three people and none of them satisfied the Onion model what it was expected because the professionals need to get into the core team as soon as possible.

Committer_id	#Commits	#Messages	#Bugs	1st Commit	1st Message	1st Bug Reporter
21	24262	347	77	1998-03-10	1997-12-12	1999-02-06
34	23297	200	40	1999-08-24	2000-06-05	2000-10-03
36	3310	43	38	2000-07-17	2002-08-21	2001-05-18

Figure 8: Data gathered for professionals

5 Threats to validity

It is necessary to make some comments about the process and the tools used in order to get the data. Due to the problems with the tools which were used to get the information from the different sources like *Svn* or *mailing list* and the lack of some tool to analyse the *Bug tracker system*, the process had to be carried out in manual way sometimes, therefore the possibility of making mistakes has been increased.

CVSAnalY

With this tool was expected to analyse the *Svn* of KDE project. It was not possible to use it in order to analyse the complete log because of a bug that the CVSAnalY version used in that moment had. It could be only analysing the log of the current *trunk*, which was created in May 2005, in order to get the list of the more active *committers* in the last six months.

In order to find out the date of their first commits, the log file was parsed by the next regular expression:

```
grep | commiter_username file.log
```

Mailing List Stats

At the time of making this paper the *mboxes* of the mail lists⁷ of KDE project were not available. This impeded to use the tool

⁷<http://www.kde.org/maillinglists/>

Mailing List Stats which lets analyse the *mboxes* of the one mailing list creating a database with all data.

The solution found in order to get the dates of the first message sent by the *committers* to some of the *mailing lists* of KDE was using the web interface that the *mailing lists* provide. This web interface let us do simple queries by subject or author.

Bug Tracker System

In this case there is not any tool that let analyse these kind of systems. KDE project provide a web interface⁸ to manage the *Bug tracker system*.

In order to get the dates of the first *bug report* that the *committers* did, this interface was used to make simple queries.

6 Conclusions and Further Work

As first conclusion that it is derived directly from the results we can see that the two parameters chosen in order to group the *committers* have directly influence on the results.

The first parameter, the role of the *committer*, this is, if they are volunteers or professionals, as it was expected, it is very related with the results. Volunteers need more time than professionals in order to get into the core team of the KDE project. They work in the project like a hobby and in their free time. Probably they start to collaborate with the project because they have used it before as normal users, and then, they decide to improve it for their own benefit and for the benefit of the community. They have not a final objective and they follow step by step the Onion model.

On the other hand, professionals have a final objective because of their work. Therefore they need to get into the core team as soon as they can in order to start the development. Probably they have not used the program before as normal users. Because of this reasons is normal that they not follow the Onion model and they make the first commit before to send a message to some of the *mailing lists*.

The second parameter used, the time, looking at the results obtained, it seems that have also influence on the results.

In the first group, when the project was starting the Onion model does not serve. At these moments the project is more close to the *cathedral model* than the *bazaar model* in this stage. There are few developers and probably they are more worried about to work in the source than to organize *mailing lists*, the web page or a *Bug tracker system*.

In the second stage, the project has already certain magnitude, the number of developers, most of them volunteers, has increased considerably what implicate that it is necessary to paid attention to the organization. The project is now more close to the *bazaar model*.

Finally, in the third group analysed, the project is totally in the *bazaar model*. The Onion model is served by all volunteers developers that get into the core team. Most of them start as normal users of the project, in this case KDE, and little by little they acquire certain implication in the project what implicate that they want to improve it.

Summarizing, it seems that the two hypothesis have been validate with the results of the study. Volunteers follow the Onion model but professionals do not as it was expected. Besides it seems that it is necessary that the project reach certain grade of maturity in order to the developers follow the Onion model. The project at the beginning is more close to the *cathedral model* and only when it reaches the maturity change to the *bazaar model* as Capiluppi and Michlmayr said on their paper.

Nevertheless it is necessary to extend the study in order to validate these results. It would be necessary to increase the number of *committers* and more important than this, it would be to do more homogeneous the groups of the developers to obtained in a more strict way the mean of the time necessary in order to get into the core team.

At the same time it would be great making use of the same data sources and with the appropriate tools, try to make a study about the generational change-over in the KDE project.

7 References

- [1] Israel Herraiz, Gregorio Robles, Juan Jose Amor, Teofilo Romera, and Jesus M. Gonzalez-Barahona, *The Process of Joinning in Global Distributed Software Projects*

8

<http://bugs.kde.org>

- [2] Andrea Capiluppi and Martin Michlmayr, *From the Cathedral to the Bazaar: An Empirical Study of the Lifecycle of Volunteer Community Projects*
-