Free/Libre and Open Source Software: Survey and Study

FLOSS

Deliverable D18: FINAL REPORT

Part 4: Survey of Developers

Rishab A. Ghosh
Ruediger Glott
Bernhard Krieger
Gregorio Robles

International Institute of Infonomics
University of Maastricht, The Netherlands
June 2002

The original version of this document is available at http://www.infonomics.nl/FLOSS/report/

Contents

List of Figures

Executive Summary

Although Open Source and Free Software are no new phenomenon, they have shown a considerable increase of their importance just in recent years. However, many aspects of this domain still appear unknown or even strange. Economic exchange relations, as they occur within the community of OS/FS developers as well as in the traditional parts of capitalist economies, are usually based on the fundamental principles of private property and monetary payments. However, these principles seem not to be applicable to OS/FS, and still this domains functions very well and gains more and more importance in the leading software markets.

Based on an online survey on 2784 Open Source/Free Software developers, this report provides insights in fundamental features of the OS/FS community and its economic principles. It sheds a light on personal features of OS/FS developers, of their work and project organization, their motivations, expectations, and orientations. Finally, it illustrates the fundamental dividing lines that characterise mainly the OS/FS community and cause its outstanding position, which are the distinction between monetary and non-monetary rewards, the distinction between OS/FS and proprietary software, but also the internal distinction between Open Source Software and Free Software.

The results of the study have shown that the OS/FS community is a rather young and predominantly male community with a strong professional background in the IT sector and a high educational level. The developers are mostly singles or only loosely associated with their partners. They feature a high degree of mobility, whereby the European Union appears as attractive only for developers from its member states, but not for developers from the United States of America or other world regions.

Overall, developing OS/FS still resembles rather a hobby than salaried work. Besides (software) engineers and programmers, students play also a significant role in the community, but project performance and leadership is primarily a matter of professionals. Most of the developers feature networks that consist of rather few people. Nevertheless, we found a considerable large group of OS/FS developers that showed regular contacts to more than 50 other developers and that provided undoubtedly the “professional elite” within the community.

Comparing the motives to start with the development of OS/FS and the motives to continue with it, we found an initial motivation for participation in the OS/FS community that rather aims at individual skills and the exchange of information and knowledge with other developers, but over time a maturing of the whole community with regard to both, commercial (material) and political aspects. To learn and to share knowledge have also been the most important issues of OS/FS developers’ expectations from other developers.

Finally, regarding the main dividing lines we found the sample clearly one-sided with respect to the differences between Open Source/Free Software and proprietary software. Positive features are generally associated with OS/FS, and negative features with proprietary software. The difference between monetary and non-monetary rewards does not play a major role within the OS/FS community.

The internal differentiation of the community by self-assignments to either the Open Source or to the Free Software community does not provoke a polarization of the community into two different parties. Rather, we found six distinguishable types of orientations in this respect, ranging from those who clearly assign themselves to one of the two domains and claiming fundamental differences between them to those who do not care to which domain they belong.

1. Introduction

The FLOSS developer survey was started in February 2002 and ended in the beginning of April 2002. It was intended to get insights into the motives of software developers to develop, distribute, and exchange Open Source / Free Software and into the ways in which the OS/FS community is organized. Key issues of the examination in this context were:

- the relationship between non-monetary motives of software developers to provide Open Source / Free Software, like the wish for a good reputation, and monetary motives, like the wish for better paid jobs
- software developers' perception and valuation of the Open Source / Free Software domain compared to that of commercial software
- incidence and distribution patterns of Open Source / Free Software and the role of this kind of software for the economy as a whole
- personal backgrounds of software developers

The developer survey was conducted in form of an online-survey. The questionnaire consisted of closed questions, i.e. every question was associated with a variety of possible answers the developer had to choose from. The questionnaire revealed following topics:

- Work situation and experience
- Personal features (age, sex, status etc.)
- Involvement and activity in the Open Source / Free Software community
- Activity in the commercial software area
- Motives for involvement in the Open Source / Free Software community (especially the role of monetary and non-monetary remuneration)
- Comparisons of experiences in the Open Source / Free Software community and in the field of commercial software
- Remuneration of contributions to the OS/FS scene

The scope of the survey was not limited, neither by the number of interviewees nor by countries or similar criteria. The FLOSS team utilized the well-known phenomenon that questionnaires of the described type are distributed within the OS/FS community by the developers themselves, thus enabling the project team to reach a large and diverse part of the whole group under consideration

Since the questionnaire, once developed, was posted to a few OS/FS communities and then distributed further within the whole scene by OS/FS developers themselves, the survey covered a broad scope of the OS/FS developers' community as a whole. Within the two months the survey was conducted, 2784 OS/FS developers filled in the online questionnaire, a number that provides a good basis for a deep-grounded description and analysis of the realm of OS/FS development. The size of the sample is, thus, smaller than the size of the sample of the WIDI survey (www.widi.berlios.de), but considerably larger than the sample size of the "hacker survey" of the Boston Consulting Group and OSDN (www.bcg.com/opensource).

The WIDI survey asked Free Software developers for their nationality, their residence, and some technological aspects and reached approximately 5600 persons. Compared to this survey, the FLOSS survey concentrated very much on motivations, orientations, and economic aspects of the OS/FS scene, thus providing deeper insights into the functioning of this community. To fill in the FLOSS questionnaire required more time than to fill in the WIDI questionnaire, which explains largely the differences in the sample size.

The hacker survey of BCG and OSDN was concentrated on the leading figures in the Open Source scene and reached thus only 660 persons. Consequently, the personal features of the BCG hackers, like age structure and occupational background of the respondents, deviate considerably from the personal features of the general OS/FS developer as it was revealed by WIDI and FLOSS.

Based on a source code analysis that was conducted in parallel to the developer survey, the FLOSS team was able to identify a sub-sample of approximately 500 OS/FS developers and to crosscheck some of their answers to the survey by their documented contribution to software code. This sub-sample provided a validated group of OS/FS developers, to which the large majority of OS/FS developers who could not be validated in this way could be compared. The aim of this procedure was to check the validity of the results of the FLOSS survey. The result of this crosscheck is attached to this document as appendix A. It shows that the group of validated OS/FS developers consists of slightly more active and "professionally" experienced persons, but their answers do not differ significantly from those of the non-validated OS/FS developers, especially in terms of orientations and motivations.

The whole procedure of the validation kept, of course, to the privacy requirements of the respondents. The first step, identification of the sub-sample, was conducted separately from the analysis of the data gained by the survey. Only the ID-number (a randomly generated serial number for each respondent) and two variables providing personal features were used for the first step. After identifying a sufficiently large group for a validation of the whole sample from the source code analysis, the data of the sub-sample were made anonymous by replacing all personal information by the single information "validated" or "not validated". Only after this transformation, the validation data were re-integrated into the data set of the survey.

The results of the survey are presented in the following chapters. The topics cover the following issues:

- Personal features of OS/FS developers
- Characteristics of work in the OS/FS community
- Motivations, orientations, and expectations of OS/FS developers

Thus, the report is structured in six chapters. After this introduction, chapter 2 deals with personal features of OS/FS developers. Chapter 3 examines main characteristics of work in the OS/FS scene. Chapter four deals with motivations, orientations and expectations of OS/FS developers from their engagement in the OS/FS community. In chapter five we will perform a deeper analysis of these subjects by distinguishing typical groups within the community of OS/FS developers with respect to the three main issues that have motivated the FLOSS project: the role of monetary and non-monetary rewards, the role of the distinction between OS/FS and proprietary software, and the role of the distinction between Open Source and Free Software. Thus, we will come to a better understanding of the internal differentiation of the whole community in ways to think and work. Chapter six provides the conclusions from the FLOSS developers survey.

2. Personal Features of FLOSS Developers

2.1 Gender and Age

The FLOSS survey on OS/FS developers confirms the findings of the WIDI project that women do not play a role in the development of Open Source and Free Software; only 1.1% of the FLOSS sample is female.

We did not ask for the current age of the developers because this is for some people, especially in the United States, a delicate question that may be perceived as a violation of their privacy. Still, we asked for the age of the respondents when they started to develop OS/FS and we asked for the year they started. On this basis we could compute an approximation to the current age of the OS/FS developers in the survey. Figure 1 shows that the age of the respondents ranges from 14 to 73 years, while there is a clear predominance of people between 16 and 36 years. Only 25% are older than 30 years, and only 10% are older than 35. The average age (mean) of the respondents is 27.1. However, the median is 26, indicating that the larger share of the sample is younger than the average age.

The question for the year in which the respondents started to develop OS/FS revealed that the main dynamics of OS/FS development took place in the second half of the nineties (figure 2). Still, some of the respondents claim to have started with OS/FS already in the fifties, and some more others ticked a year in the seventies or eighties. However, until 1990 there were only 8.2% of the sample already active in the OS/FS scene, and just in the following years the development accelerated considerably. Although the dynamics have accelerated again from 1998 onwards, the average starting year (mean) was 1996.7. Nevertheless, the median is 1998, indicating again the skewness of the distribution.

The average starting age of the OS/FS developers was 22.9 (median: 22.0). The distribution of the starting age, as it is indicated by figure 3, shows that only 7% started below an age of 16 years, one third was between 16 and 20 years old, another third between 21 and 25, and a quarter was more than 26 years old when started OS/FS development.

Figure 1: Current Age of OS/FS Developers

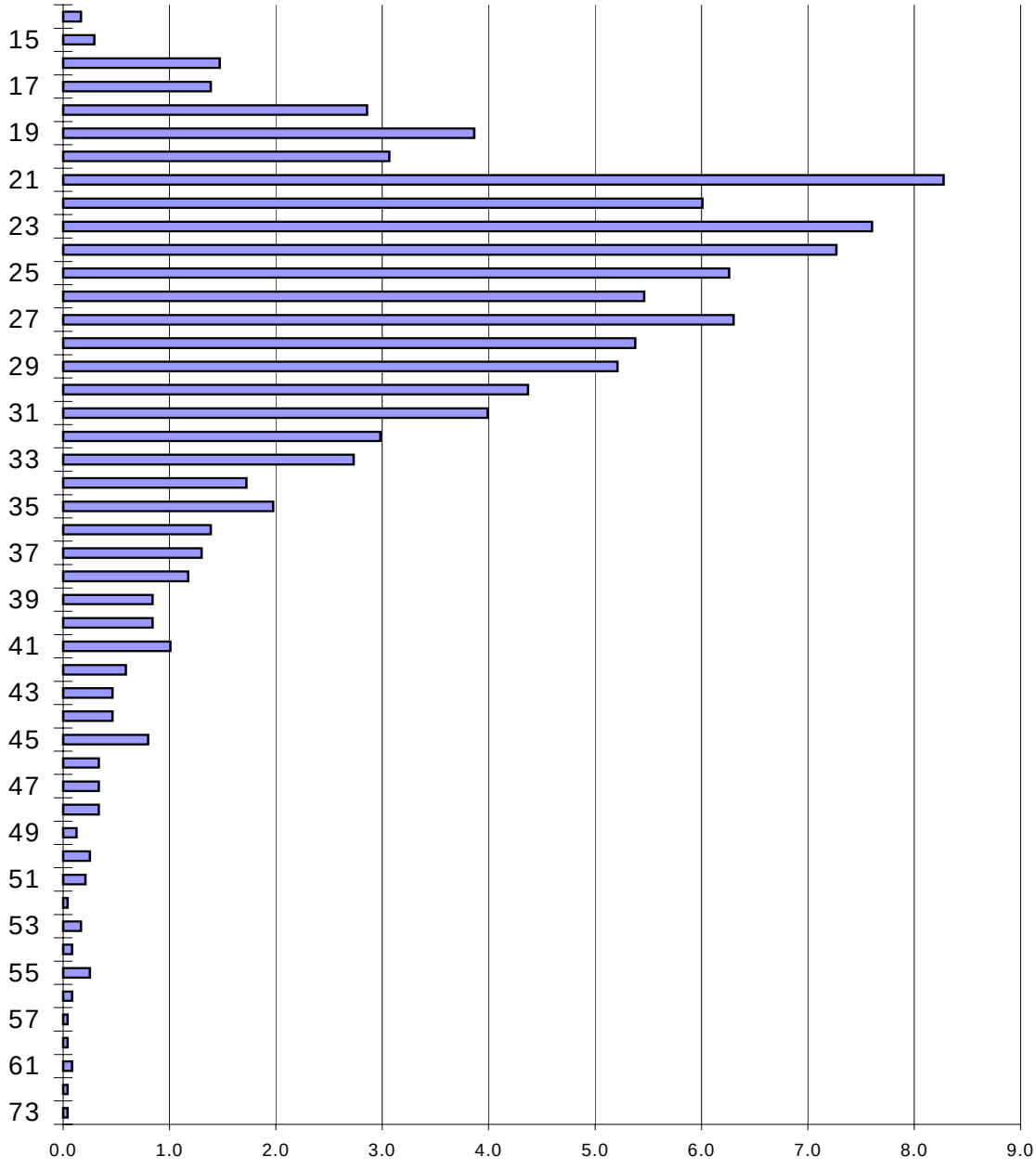


Figure 2: Starting Year in OS/FS Community

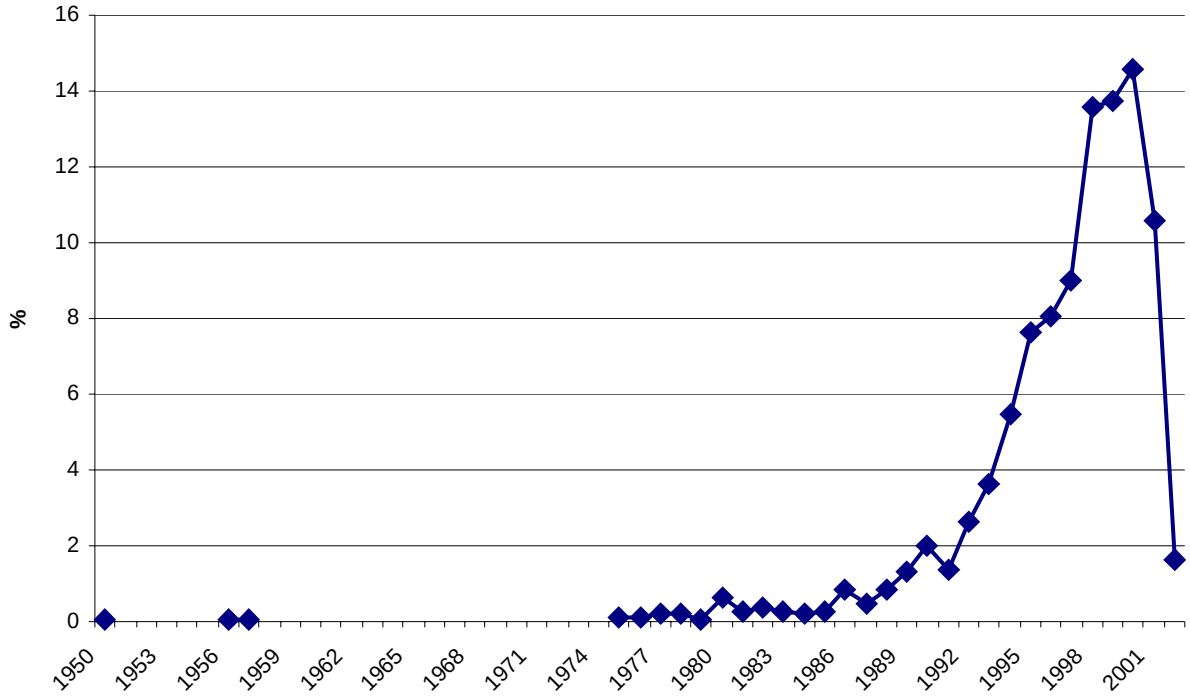
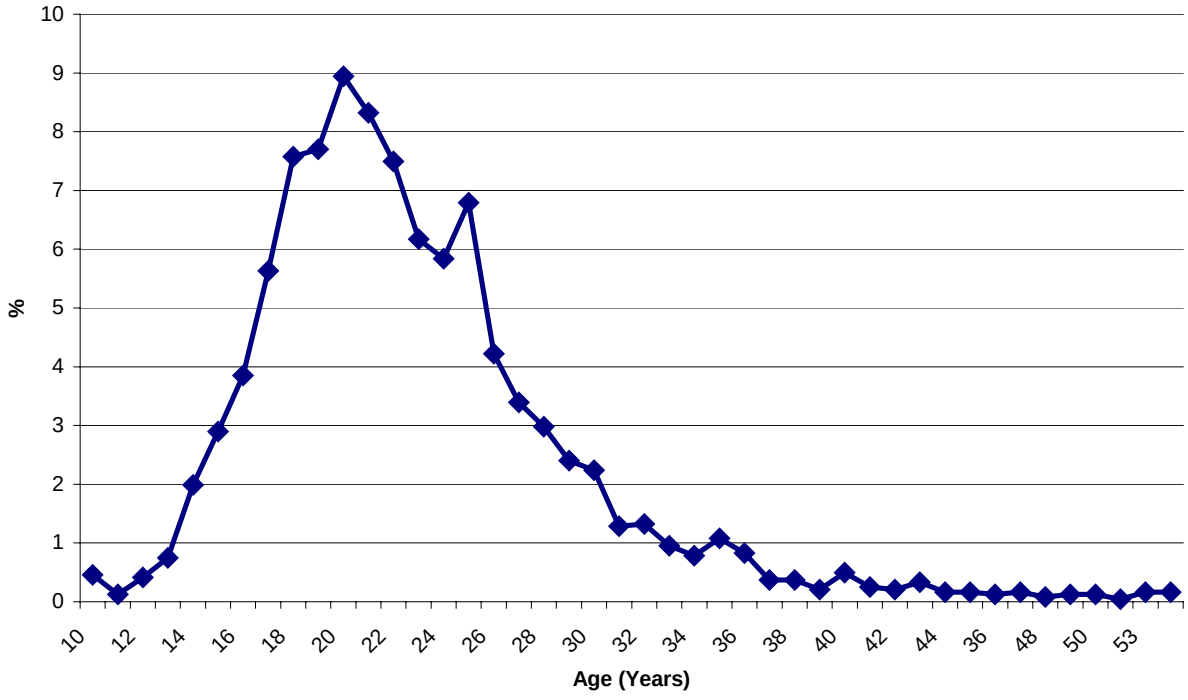


Figure 3: Starting Age in OS/FS Community

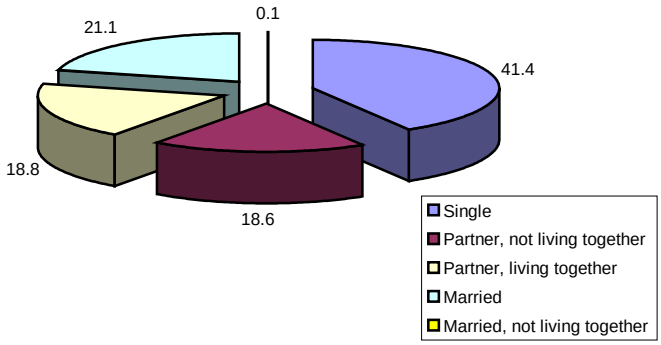


Altogether, the results indicate that developing Open Source / Free Software is rather a matter of the rising generation than one of experienced software developers. However, taking into account that Open Source / Free Software is by no means a phenomenon of only the recent years, the young age of OS/FS developers cannot be explained only by generational effects. Other reasons like changes in the market for software products or in the production of software seem to be the key factors for the increasing importance of Open Source / Free Software, and due to these changes, more and more young people become attracted by this community.

2.2 Partnership / Family Background

Two fifths of the sample are singles, roughly 60% live in a kind of partnership (figure 4). Therefore, the often-mentioned assumption that OS/FS developers are singles that are bored and have no partnership obligations and responsibilities is apparently not true.

Figure 4: „Civil Status“ of OS/FS Developers

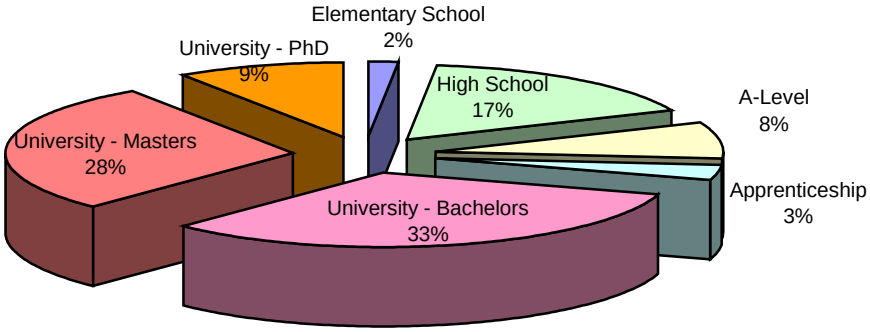


Further examination of the data revealed that only 17% of the OS/FS developers of the FLOSS survey have children. Almost half of the developers with children have only one child, 31% have two children, 14% have three, and 6% have more than three children.

2.3 Educational Level of OS/FS Developers

OS/FS developers feature a rather high educational level. University degrees make up 70% of all degrees, while only 17% of OS/FS developers have a high school degree and only 8% name an A-level as their highest educational degree (figure 5). However, a PhD seems not to be a necessary prerequisite to become an OS/FS developer.

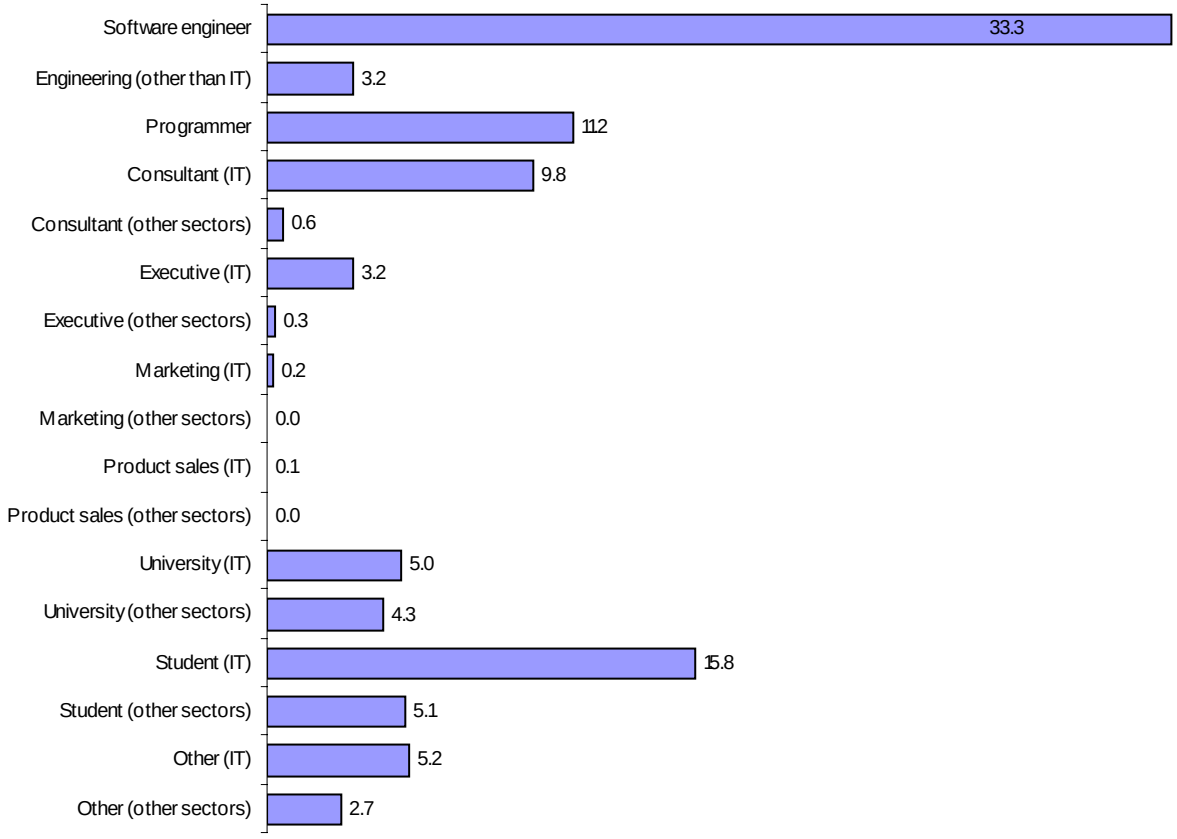
Figure 5: Highest Level of Education of OS/FS Developers



2.4 Professional Background

Not surprising, professions and university courses related to the IT sector play an important role for OS/FS development. 83% of all developers in the sample are employed in the IT sector or deal with similar tasks at universities (figure 6). Software engineers provide one third of the sample and thus the largest single group. With a share of 16%, students are the second largest group, followed by programmers and IT consultants. Executives, marketing and product sales experts do not have a significant impact on the professional structure of the OS/FS community.

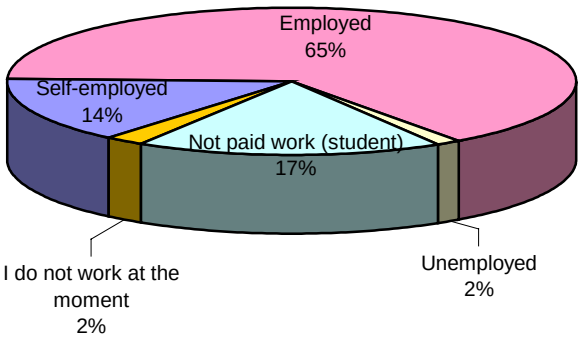
Figure 6: Professional Structure of OS/FS Developers



2.5 Employment Status

Almost two thirds of the OS/FS developers within the sample are employees, but a relatively high share of 14% is self-employed (figure 7). Students and people who claim that they are not working currently sum up to a fifth of the sample. Unemployment does not play a role within the group of OS/FS developers.

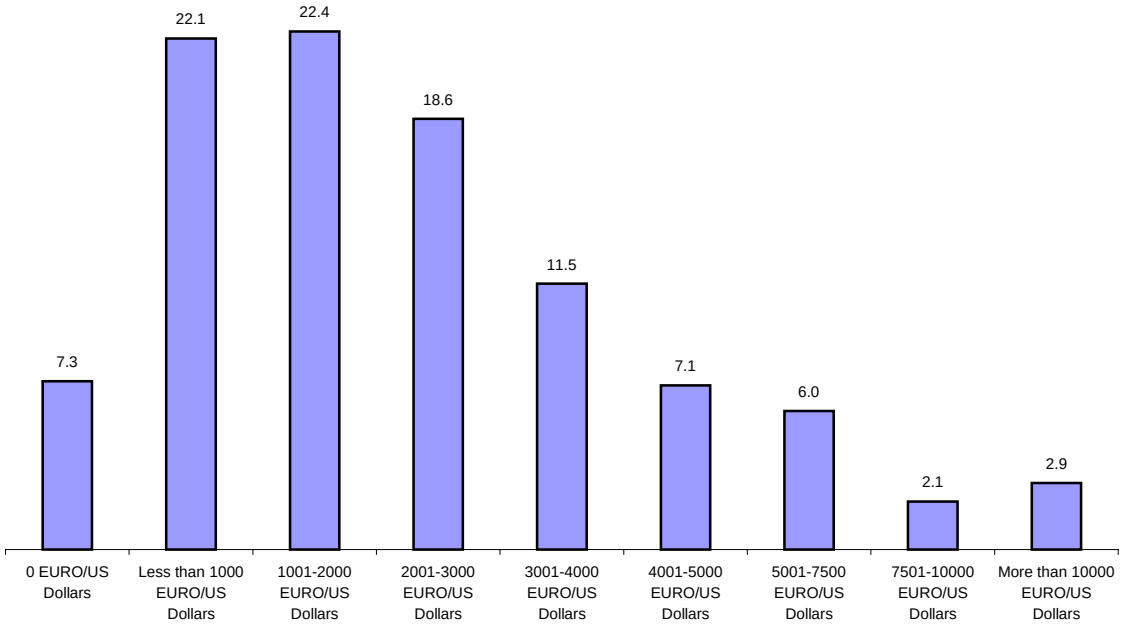
Figure 7: Employment Status of OS/FS Developers



2.6 Income

OS/FS developers are no top earners, which may be due to the rather large share of students. 7% do not earn anything, and another 45% reach no higher gross income than 2000 EURO/US Dollars per month (figure 8). After all, 19% earn between 2000 and 3000 EURO/US Dollars, and 12% reach an income level of 3000 to 4000 EURO/US Dollars. 13% earn between 4000 and 7500 EURO/US Dollars, and only 5% reach an income above 7500 EURO/US Dollars.

Figure 8: Monthly Gross Income of OS/FS Developers



2.7 Nationality, Residence, and Mobility Patterns of OS/FS Developers

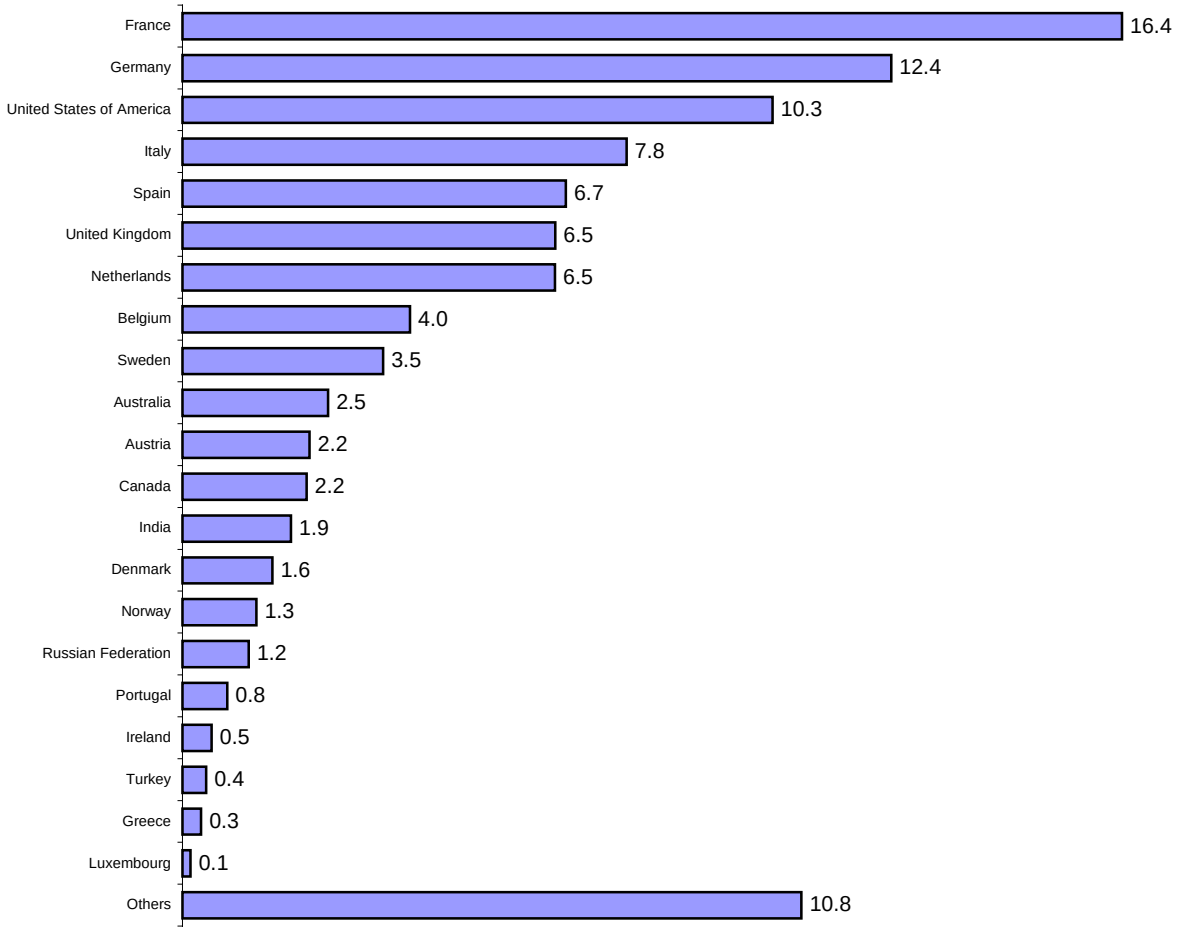
71% of the OS/FS developers in the sample stem from EU-countries, while 13% come from North America (USA and Canada), and another 17% from other countries in Europe and all over the world. A closer look to nationalities allows distinguishing five groups (figure 9). The first group consists of the three countries that provide the lion shares of the sample, led by France (16.5%) and followed by Germany (12.4%) and the United States of America (10.4%). This result may in part be due to different diffusion patterns of the questionnaire in Europe and America, but the FLOSS survey nevertheless clearly indicates that in terms of Open Source and Free Software Europe is no longer behind the USA.

The second group consists of countries that provide shares between 6% and 8% of the sample. This group includes two Mediterranean (Italy and Spain) and two Northern European countries (UK and The Netherlands) and confirms thus the strong position of Europe in the OS/FS scene.

The third group consists of countries that provide shares between 2% and 4% of the sample and is made up by Belgium, Sweden, Australia, Austria, and Canada. Countries with shares between 1%

and 2% of the sample provide the next group, which is led by India and followed by the Scandinavian countries Denmark and Norway, and finally by the Russian Federation. The fifth group is a residual category, consisting of countries, which shares lie below 1% of the sample, like Ireland, Turkey, Greece and Luxembourg.

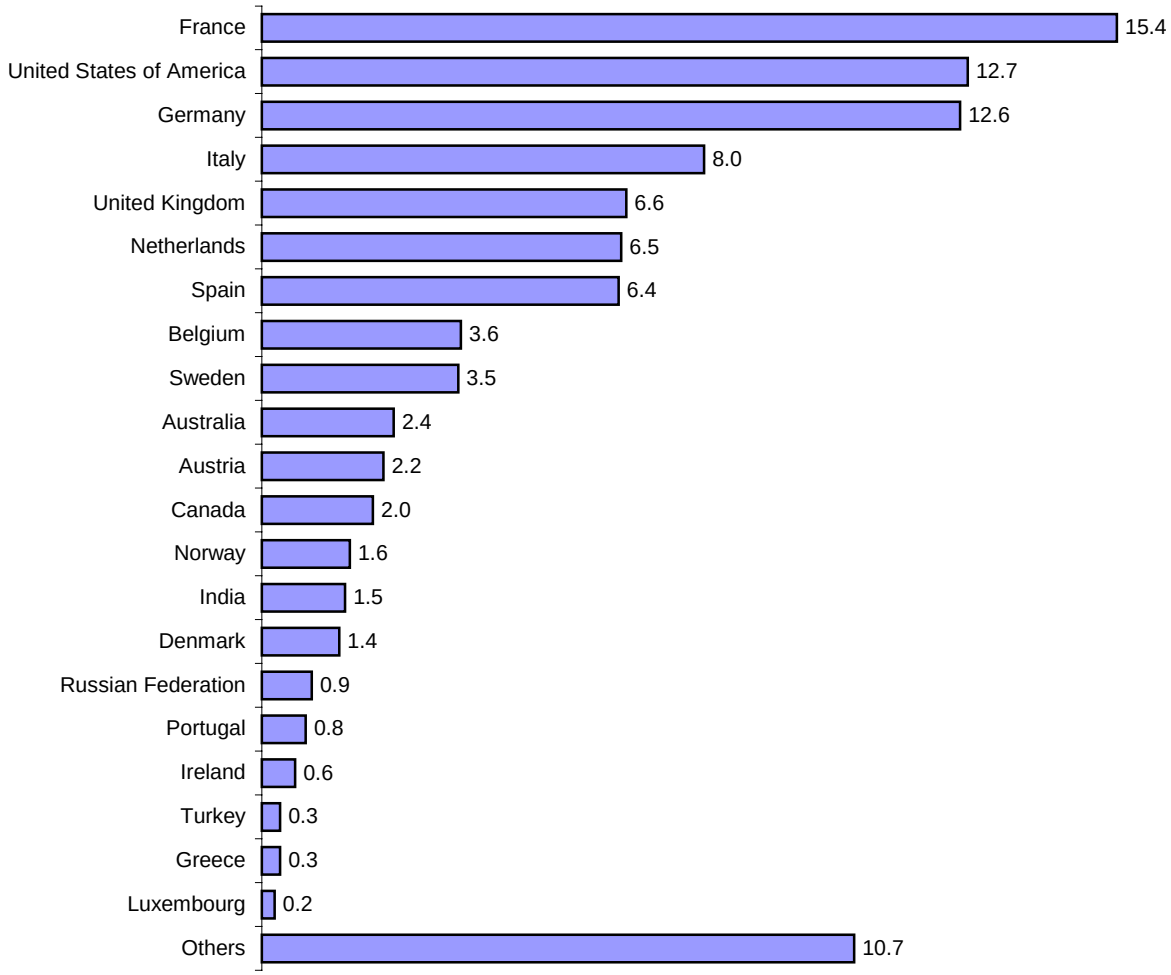
Figure 9: Nationality of OS/FS Developers



However, the question for nationality only provides information about the origins of the OS/FS developers in our sample. It is also important where these people live and work. 70% of the sample live in EU-countries, 14% in North America, and another 16% in other countries. 10% of all OS/FS developers who answered the respective questions lived in another than their home country. Nevertheless, figure 10 shows some, but no great differences between the residences of OS/FS developers and their home countries. The composition of the five groups is almost the same, while the order of the countries within each group is sometimes changed.

For instance, in the first group Germany and the United States of America have changed their position, while in the second group, Spain shows an evident loss of attractiveness for OS/FS developers to work in this country, compared to the other countries in this group. The biggest changes occur in the fourth group, where rank order and number of countries diverges from the pattern shown in figure 1. Russia, in this regard, does not belong to the fourth, but to the fifth group.

Figure 10: Country of Residence or Work of OS/FS Developers

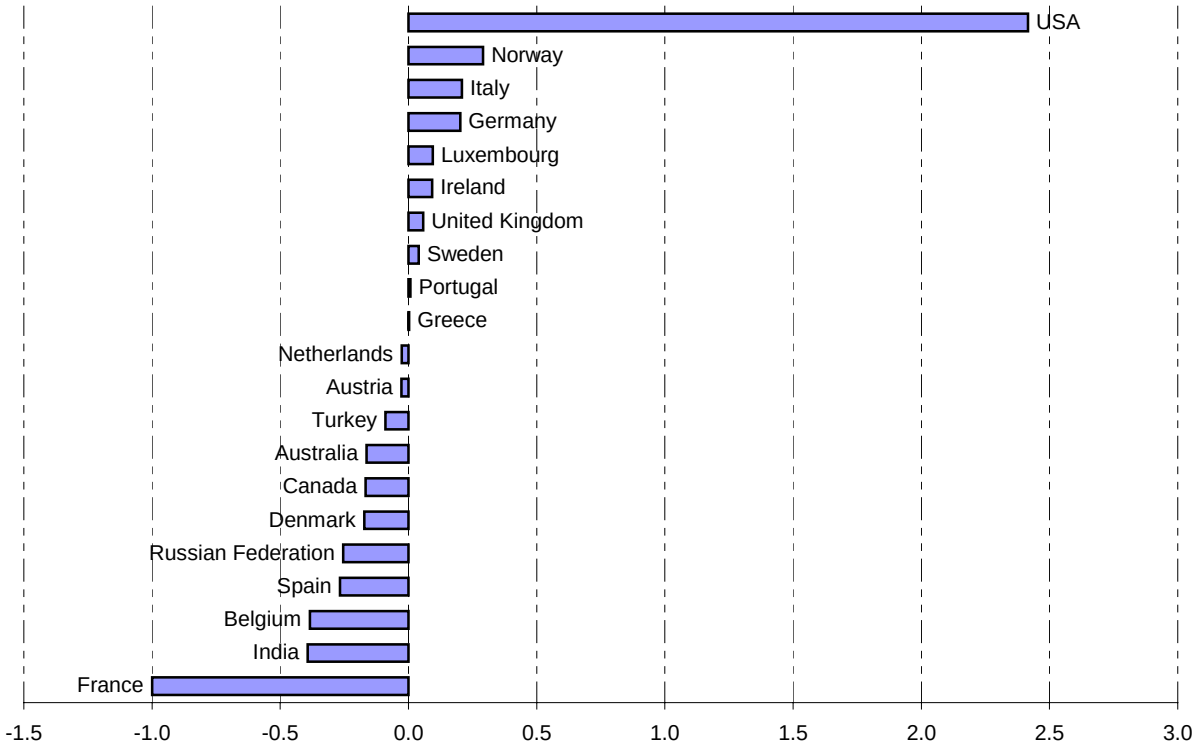


These results lead to question the attractiveness of countries and regions for OS/FS developers and to a further examination of mobility patterns of OS/FS developers. Comparing the nationality of the OS/FS developers and the place where they live and / or work, we can compute country-specific mobility balances and distinguish countries that attract OS/FS developers from countries

that appear as not as attractive to OS/FS developers to live and work. As revealed by figure 11, the United States of America are obviously most attractive for OS/FS developers, while France does apparently not offer attractive opportunities to them.

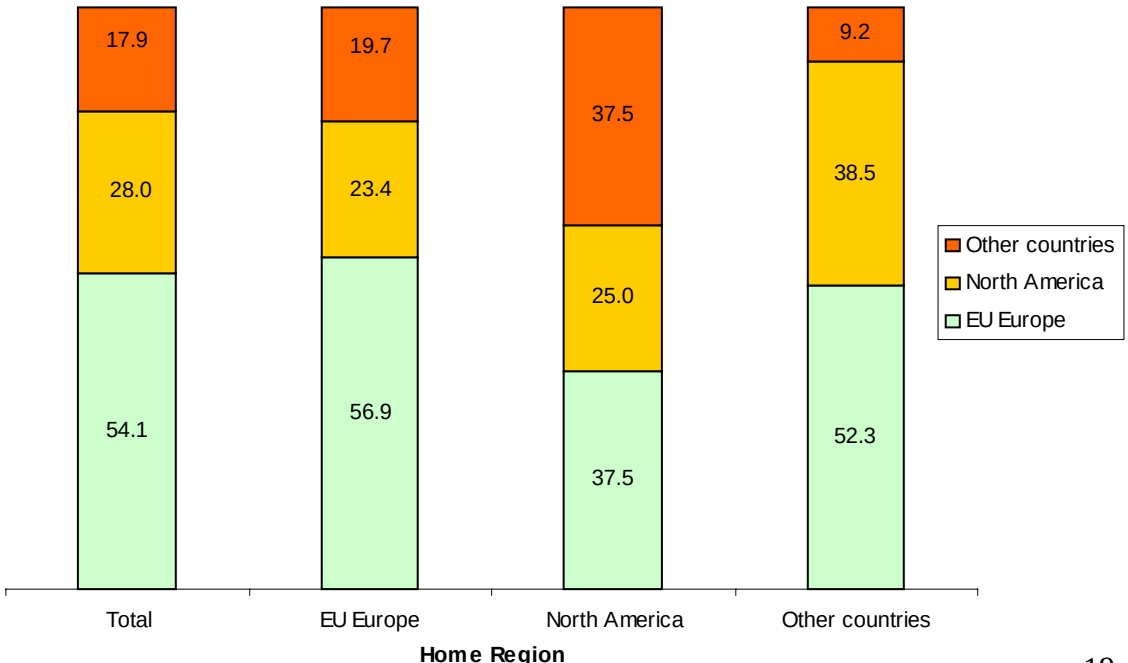
Although there are also European countries with a positive mobility balance of OS/FS developers, like Norway, Italy, and Germany - and, on a very low level, Luxembourg, UK, and The Netherlands -, the attractiveness of these countries is left far behind the attractiveness of the USA. Besides France, also India, Belgium, the Russian Federation, Spain, Denmark, Canada, Australia, Turkey, Austria, and Sweden, show a negative mobility balance. These results limit our former finding of the strong position of Europe within OS/FS development, because they clearly indicate that European countries cannot really compete with the USA in terms of attractiveness as a residence for OS/FS developers. Since we did not ask for reasons why people moved to other countries, we cannot specify directly the factors that would explain the shortcomings of most of the European countries. However, in following steps of the analysis we will check whether factors like income, job satisfaction, and similar features, will have an effect on the choice of residence.

Figure 11: Country-based Mobility Balances



Finally, the data allow identifying the direction of mobility flows of OS/FS developers. Regarding only those developers who have left their home country, we find a striking difference in the mobility patterns of European and North American OS/FS developers, probably mainly explained by the different political structure of the two regions (i.e. the number of states offered in the region to go to), but again shedding a light on EU's strengths and weaknesses in OS/FS development. While almost three fifths (57%) of the European OS/FS developers who leave their home country stay in Europe, only one quarter of the North American OS/FS developers who leave their home country stay in this region. Roughly one quarter of the European OS/FS developers who leave their home country went to North America, and another fifth to other countries in the world. 38% of the North American OS/FS migrants went to EU-Europe, and the same share went to other countries in the world. Thus, the mobility pattern of EU-European OS/FS developers is characterized by an above average migration to other EU-countries and other countries in the world, while migration to North America is below the average. In contrast, the migration pattern of North American OS/FS developers is clearly characterized by an above average migration to other countries in the world, while migration into the EU and other countries in the same region keeps below average. Finally, OS/FS developers from other countries are clearly attracted by the EU and North America. However, although the lion share went to EU member states, this mobility flow kept below average, while the migration of this group into North American states was above average.

Figure 12: Direction of Mobility Flows of OS/FS Developers



To conclude: The European Union is very attractive to OS/FS developers who originate from that region, but it is overcome by the United States of America in attracting developers from other regions than the USA or EU, and it is overcome by other countries to attract OS/FS developers from the United States.

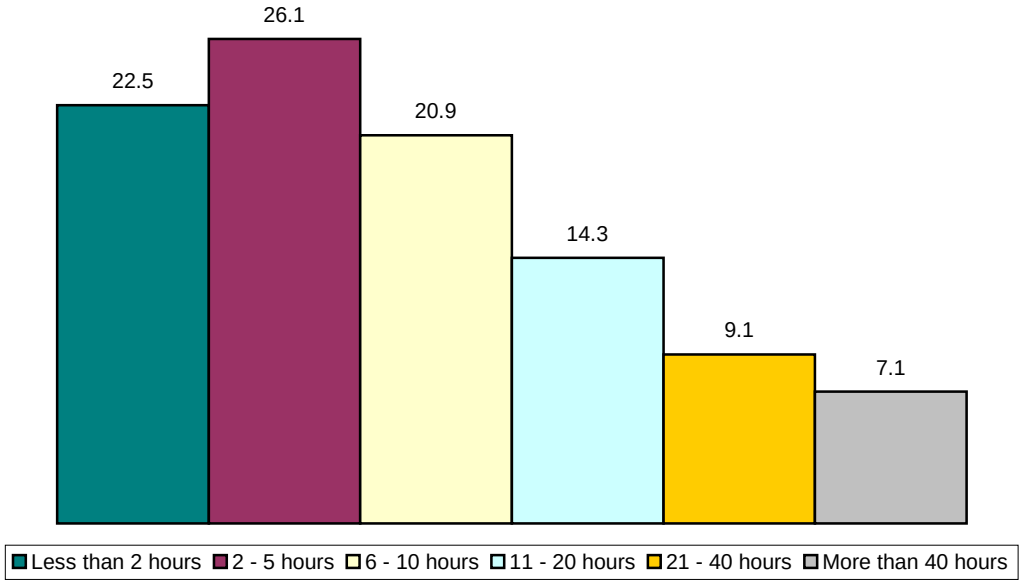
3. Organisational and Work Characteristics

In this chapter, the main organisational and work characteristics of developing Open Source or Free Software are scrutinised. The analysis comprises patterns of time spending for OS/FS development, work areas, preferred working tools, degrees of involvement in OS/FS projects, experiences in project leading, contacts to other OS/FS developers within the community, and finally aspects of income from software development.

3.1 Patterns of Time Spending for OS/FS Development

Although there is evidence of a strong professional background, for most of the developers in the sample developing Open Source / Free Software is rather a hobby than a profession. Almost 70% do not spend more than 10 hours per week for developing OS/FS (see figure 13). Roughly 23% spend only 2 hours per week for this, 26% expend 2-5 hours per week, and 21% spend 6-10 hours per week for developing Open Source / Free Software. 14% spend an amount of time for OS/FS that could be compared to professional part time work (11-20 hours per week), and 9% spend 20 up to 40 hours per week for the development of OS/FS. Finally, for 7% of the sample the time used for developing OS/FS exceeds 40 hours per week.

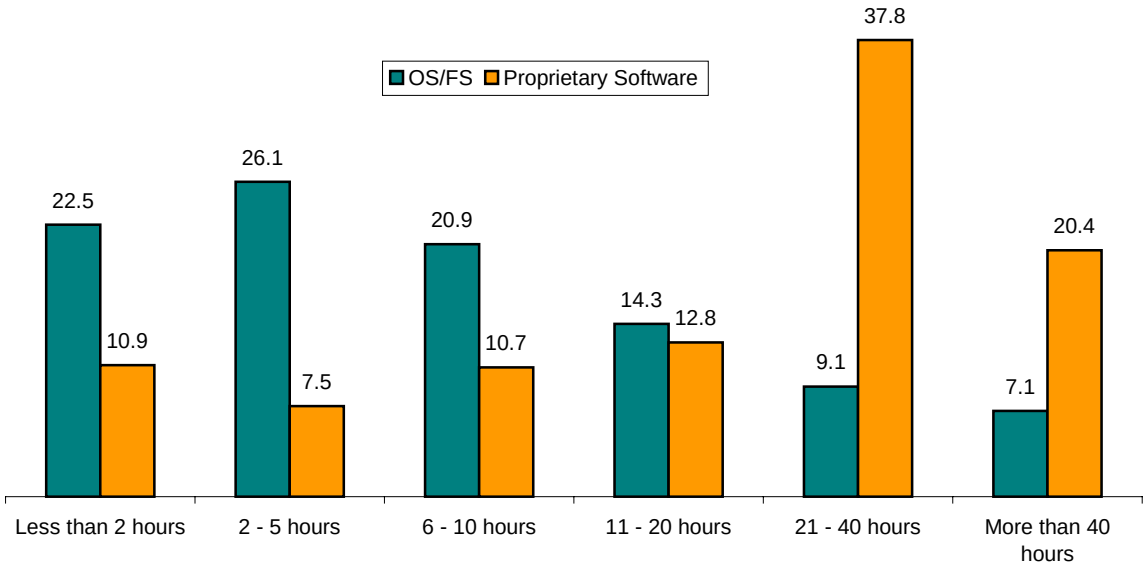
Figure 13: Time Spent for the Development of Open Source / Free Software



The development of Open Source/Free Software is not at all a matter of leisure "work" at home. 95% of the sample claim that they use OS/FS at work, school, or university. Thus, the professional background seems to be a very motivating factor for developing OS/FS.

Half of the developers of the FLOSS sample (52%) do not only develop Open Source / Free Software, but also proprietary software. Figure 14 shows the amounts of time the developers spend for developing proprietary software in comparison to the pattern of time spent for developing OS/FS.

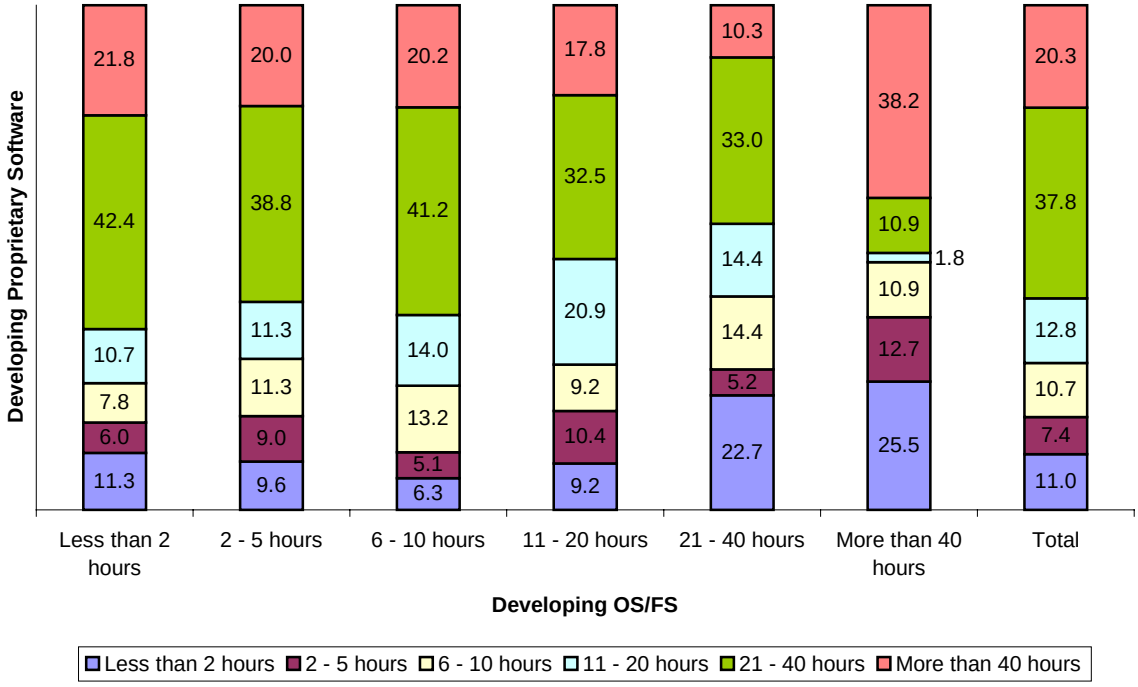
Figure 14: Comparison of Patterns of Time Spending for the Development of OS/FS and of Proprietary Software



The patterns of time spending for the development of OS/FS and of proprietary software are converse, which seems to be explained by the simple fact that those who spend much time for OS/FS have less time to perform proprietary software and vice versa. However, a deeper insight into the data shows that this assumption does not reveal fully the reality of OS/FS developers. Instead of two poles, one provided by those who spend much time for OS/FS and only few time for proprietary software, and the other provided by those who spend much time for proprietary software and only few for OS/FS, we found that those who are very active in developing OS/FS - i.e. those who spend more than 40 hours per week for this task - are also very active in developing proprietary software (figure 15). It is not surprising that those who invest more than 40 hours per week in the development of OS/FS feature the highest shares of developers who spend only less than two hours per week (25.5%, while the average is 11%) and, respectively, two to five hours per week (12.7%, average is 7.4%) for proprietary software. Yet it is astonishing to find that 38.2% of this group do spend more than 40 hours per week for developing proprietary software (average: 20.3%). However, regarding the whole sample, this extraordinarily active group amounts to only 0.8%.

There is also some evidence that those who do not spend much time in developing OS/FS do spend more time in developing proprietary software, as indicated by 42% of those who do not spend more than two hours per week for developing OS/FS who spend 21 to 40 hours per week for developing proprietary software.

Figure 15: Time Spent for Developing OS/FS and Proprietary Software

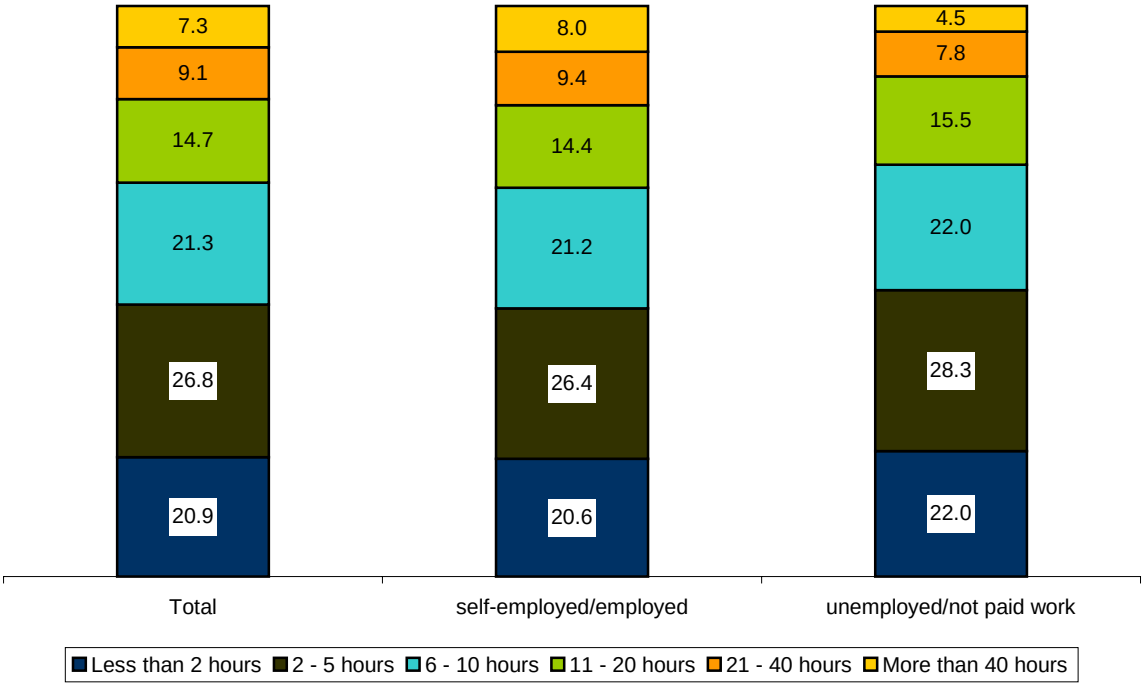


Those FLOSS developers who spend six to ten or 11 to 20 hours per week, respectively, for developing OS/FS provide a third interesting group. Both groups, but especially the latter one, spend six to ten respectively 11 to 20 hours per week for proprietary software, either. Thus, they seem to follow a "double part-time concept", spending half of their working week for developing OS/FS and the other half for the development of proprietary software.

One could reasonably assume a correlation between developers' activity within the OS/FS community and the employment status in the way that the latter will have an impact on the time that is available to people to develop OS/FS, either in the sense that unemployed people can spend more time in developing software then employed people, or in the sense that employed developers spend more time in developing OS/FS because they do it on their job and have to count their working time. However, figure 16 shows that none of these correlations really exist,

although the share of those who work more than 40 hours per week is slightly higher among the employed than among the unemployed. More striking than this single and rather small difference is the analogy of the time patterns between the unemployed and the employed in all other categories. Thus, other factors must explain the differences in time used for developing Open Source / Free Software.

Figure 16: Hours per Week Spent for OS/FS by Employment Status



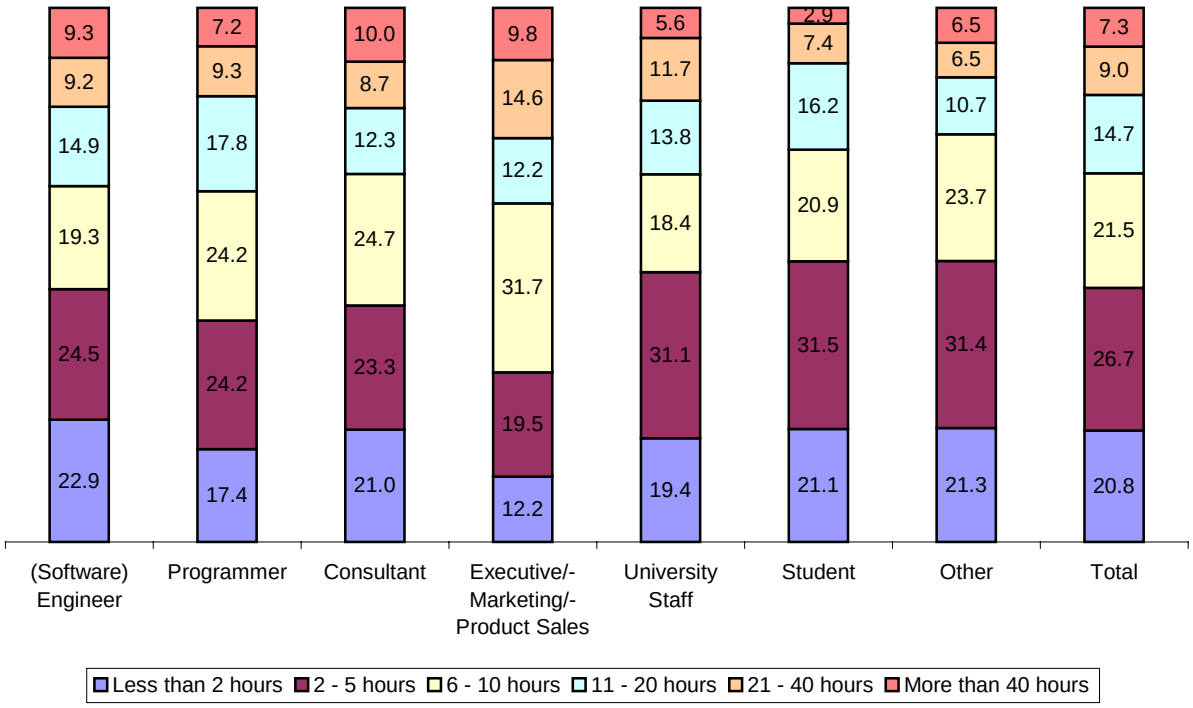
Further examination of personal features indicated a slight tendency of older developers (more than 30 years old) to spend more than 40 hours per week for developing OS/FS, but altogether the results were too disparate to provide a good reason for the differences in time spending patterns of the OS/FS developers, too. However, it becomes evident that developing OS/FS is not a subject that is primarily performed by the youngsters. Rather, the result indicates a growing stability of the individuals' interests in developing Open Source/Free Software over time.

Similar conclusions apply to the impact of the living situation (living alone, either as a single or as part of a partnership, or living together with the partner) on the time spent for developing OS/FS. Assumed that people who live together with their partner have to spend a certain share of their time for their partnership, it is reasonable to presume these people spending less time for

developing software than people who live alone. However, we found that developers who live together with their partner show a slightly higher tendency to spend more than 40 hours per week for OS/FS development, but again the data are too weak to explain the overall differences. Here, the disparateness of the findings is probably explained by the fact that 83% of those OS/FS developers who live together with their partner and who spend more than 40 hours per week for developing OS/FS feature occupations that are closely linked with OS/FS and usually associated with long working times: They are software engineers (58%), programmers (12%), and IT consultants (13%).

This leads to the examination of the role of the professional background on the patterns of time spending for the development of Open Source /Free Software. Figure 17 illustrates the time spending patterns regarding the development of Open Source / Free Software for different occupational groups.[1]

Figure 17: Time Spent for Developing OS/FS by Occupational Background of the Developers



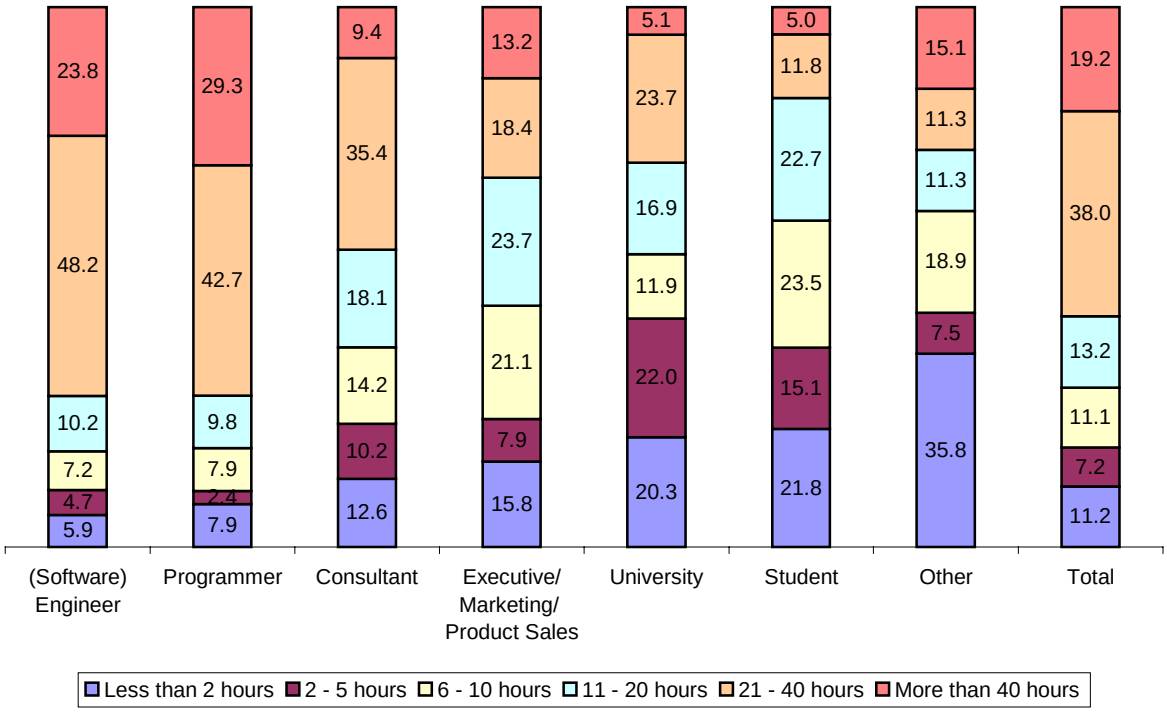
[1] Students, unemployed and people performing not paid work are simplified considered as occupational groups, too. It is however clear that these groups do not really feature an occupation.

Software engineers provide the predominating group within the sample and thus determine the average to a large extent. Still, they show slightly above average shares in the two extreme categories, working less than 2 and working more than 40 hours a week, while they feature below average shares in the lower middle categories, i.e. 2-5 and 6-10 hours a week. *Programmers* distinguish clearly from this pattern, as they do not work overtime to an extent that exceeds the average, and as they show also an below average share in the minimum category, less than 2 hours a week. *Consultants* and *executives, marketing and product sales professionals* provide the two groups that most typically spend more then 40 hours a week for the development of OS/FS, whereby the latter group features extraordinarily high shares in the categories of 21-40 and 6-10 hours per week. However, this group consists of rather small and heterogeneous occupational sub-groups, which impedes the interpretation of this time spending pattern. *University staff* features an above average engagement in developing OS/FS only in two categories, 2-5 and 21-40 hours per week. *Students* differ from the general pattern as the highest share of this group invests two to five hours a week in developing OS/FS. They scarcely occur within the category of more than 40 hours a week, and they are also underrepresented in the category of 20 to 40 hours per week. This clearly indicates that the often drawn picture of the PC-sticking student as providing a large part of the OS/FS community is not true. Rather, students show a certain interest into OS/FS and are willing to invest more than just 2 hours a week, but apparently they are not able to engage fully in this scene like software engineers, consultants and executives tend to do. Finally, representatives of *other occupations* show shares above the average only in the lower middle categories, while they are underrepresented in the higher categories.

For those OS/FS developers in the FLOSS sample who are also active in the field of proprietary software, figure 18 illustrates the pattern of time spending for the development of this kind of software. As a consequence of its greater proximity to salaried work, the workload in this respect is much higher than the workload related to OS/FS. 19% of all developers of proprietary software in the FLOSS sample spend more than 40 hours per week for this task (compared to only 7% in OS/FS development), and another 38% spend 21 to 40 hours for proprietary software (OS/FS: 9%). Consequently, the shares of those who spend only few hours for developing proprietary software are smaller than with regard to Open Source / Free Software.

This overall pattern is clearly determined by (software) engineers and programmers, while students and university staff provide the strongest deviation from this overall pattern.

Figure 18: Time Spent for Developing proprietary software by Occupational Background of the Developers

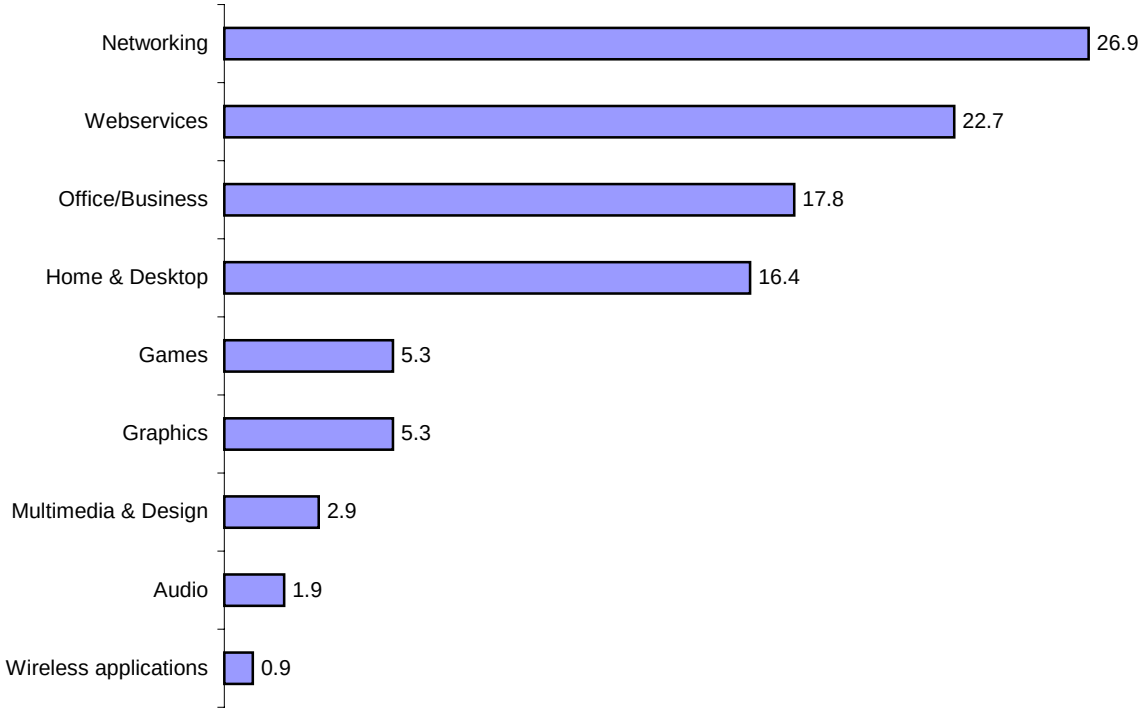


As a conclusion, it is evident that developing OS/FS is aligned with clear IT competence, usually acquired through the occupational context of the developers and often embedded in a professional / salaried environment, which is especially illustrated by the strong inter-relation of OS/FS development and the development of proprietary software. There is a broad variety of degrees of activity among the OS/FS developers, but overall we find that the large majority takes OS/FS very serious and invests more than just 2 hours a week for this interest. A strong interest into IT issues seems to be the most important push factor for OS/FS, as it became revealed by the professional background of the developers' time spending patterns.

3.2 Preferences of Work Areas and Working Tools

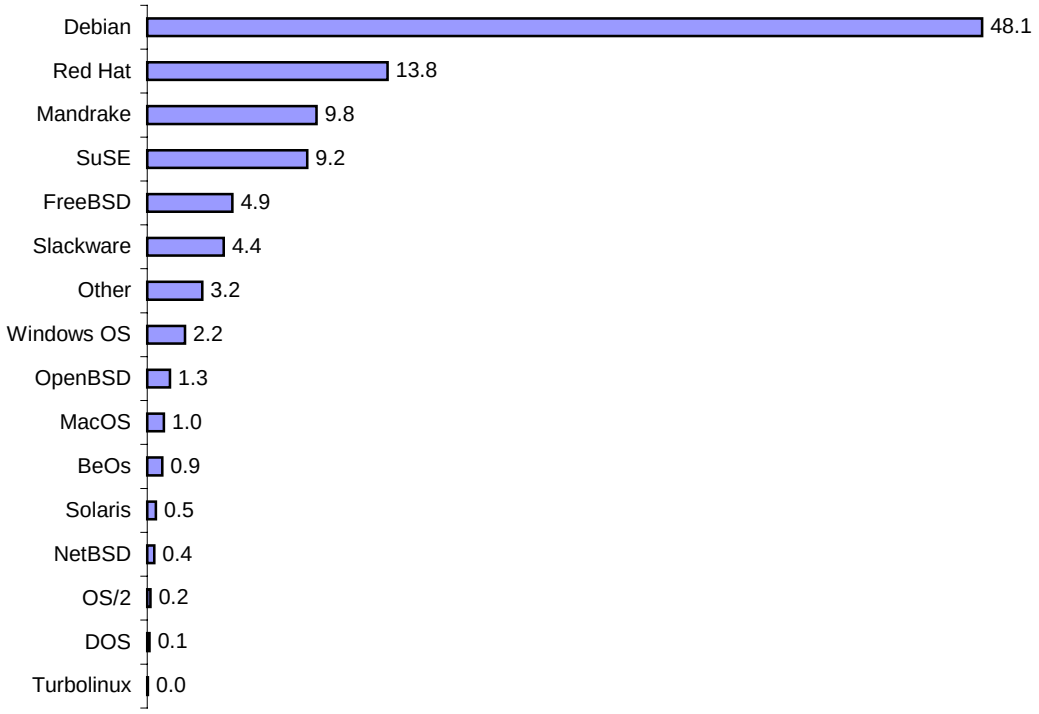
Networking and web services, together with office/business and home & desktop applications provide the areas for which the respondents develop primarily Open Source / Free Software (figure 19).

Figure 19: Areas for which OS/FS is Primarily Developed



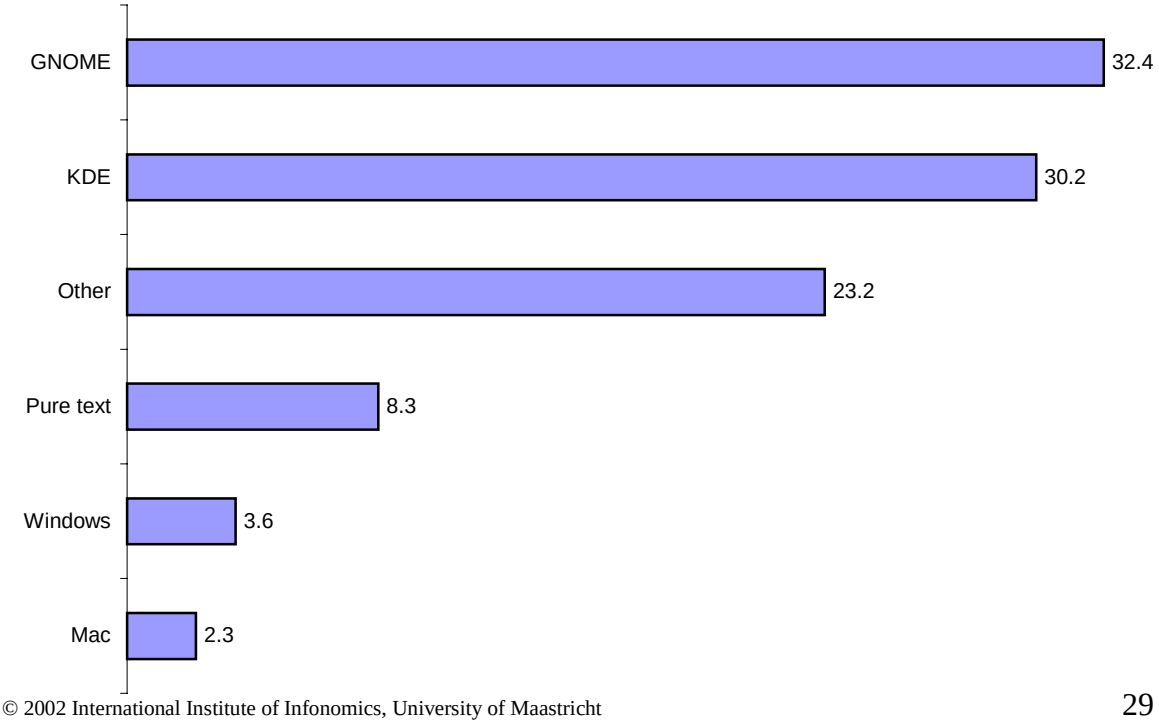
We also asked the developers for their preferred working tools, like distribution / operating systems and editors. Figure 20 shows the preferences of the OS/FS developers with respect to distribution / operating systems. The results correspond clearly to the expectations, which are based on visible communications of OS/FS developers in the Internet and the market position of the main distribution systems.

Figure 20: Favoured Distribution System



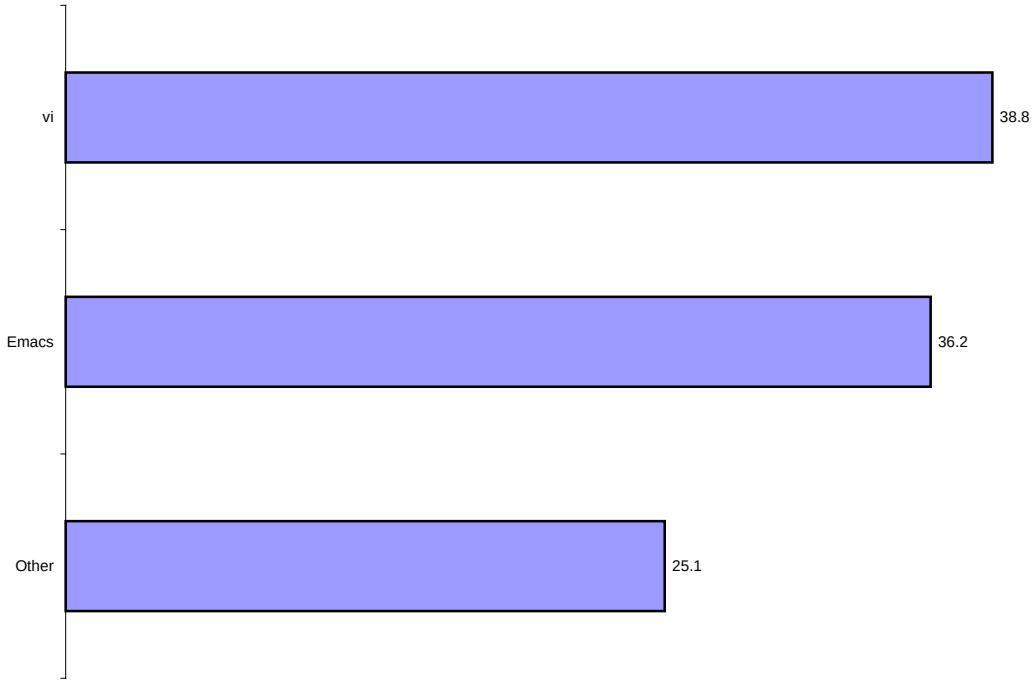
The same applies to the order of the favoured desktop, as it is illustrated by figure 21.

Figure 21: Favoured Desktop



Finally, figure 22 shows that there is a very balanced valuation of the main editors within the OS/FS community.

Figure 22: Favoured Editor



3.3 Project Involvement

After we considered the level of working time patterns, the nature of work in the OS/FS community will be further examined on the project level. Questions under scrutiny are:

- In how many projects do OS/FS developers currently work, how much project experience do they feature?
- How is the OS/FS community organized on the level of contacts between OS/FS developers?
- How is leadership in OS/FS projects organized?

Figure 23 illustrates that a predominating share of the OS-FS developers (72%) got experience from a rather small number of projects, while roughly one fifth of the sample still features experience from six to ten projects. A third group worth mentioning is provided by developers

who participated in 11 to 20 projects, which amounts to 6% of the sample. However, there is only a very small group of OS/FS developers (3%) who participated in more than 20 projects since they began to participate actively in the OS/FS community. 0.5% provide obviously the most experienced elite within the community of OS/FS developers, claiming to have performed more than 100 projects since they started to develop Open Source / Free Software. Three quarters of this group are over 30 years old, but there are also members that are just 22 and 23.[2]

Figure 23: Number of Performed OS/FS Projects So Far

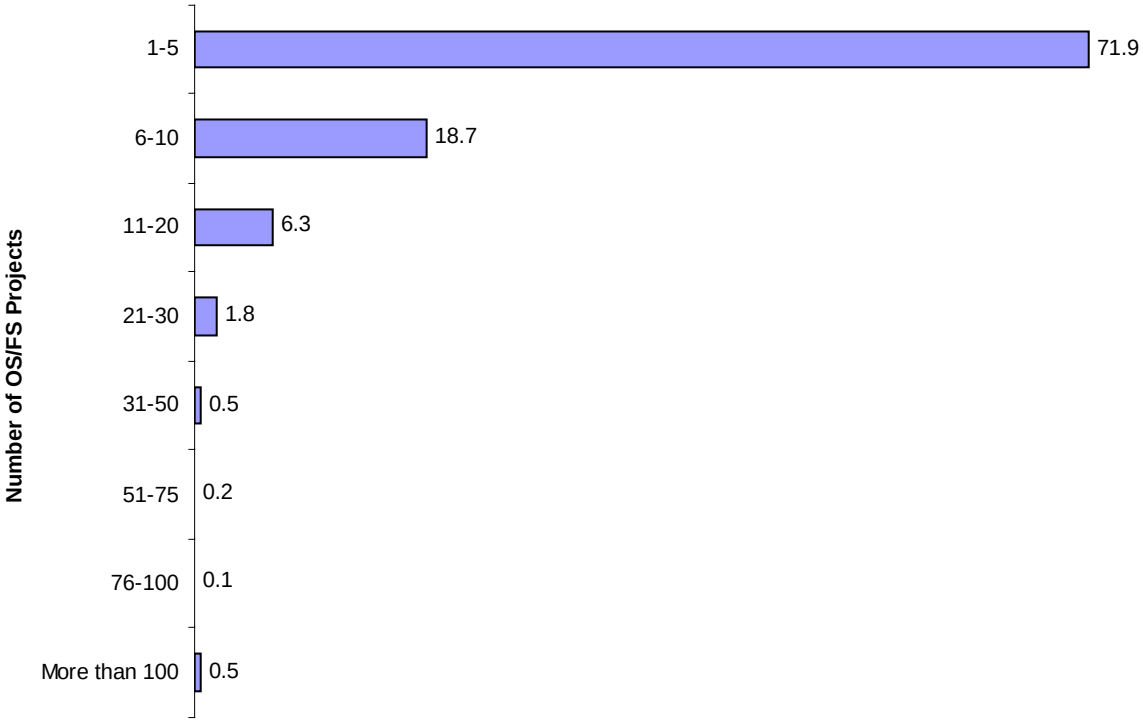
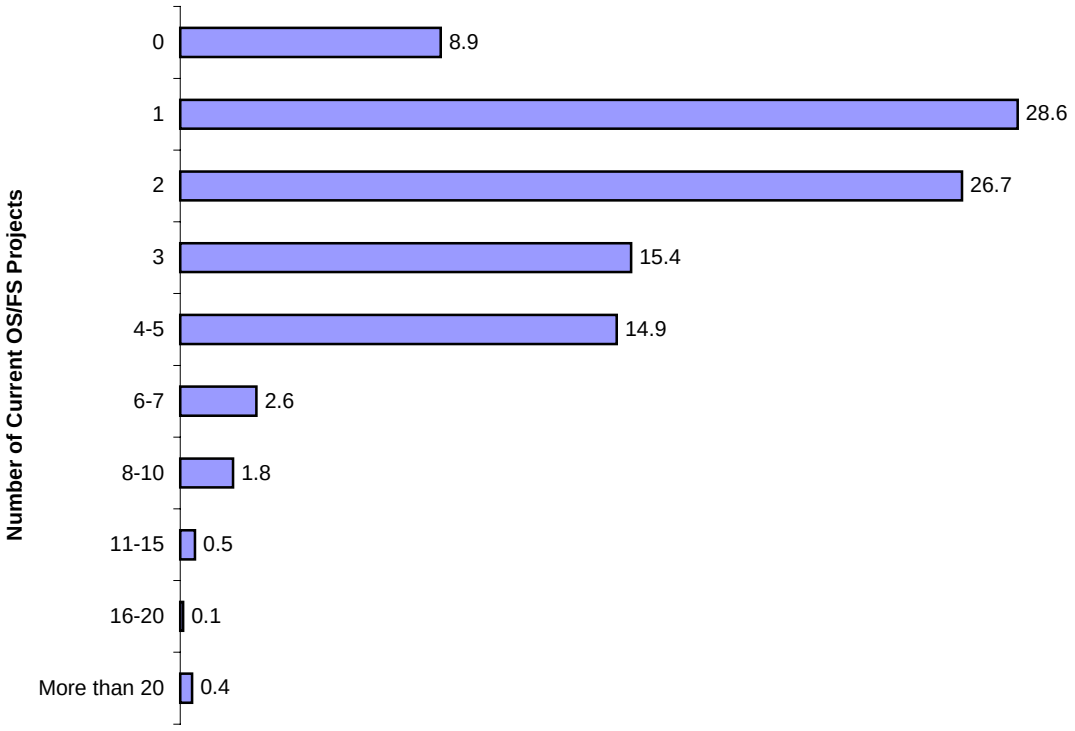


Figure 24 illustrates the number of OS/FS projects in which OS/FS developers are currently involved. The fact that only 9% have not been involved in a project while the survey was conducted reveals the high degree of activity in the community. The large majority of OS/FS developers limit their activity to one or two projects (29% and 27%, respectively). A second considerable group participates in three OS/FS projects (15%), and another 15% are currently involved in four to five projects. Only 5% are busy with six or even more projects at the same time.

[2] However, this group consists of only 12 people.

Figure 24: Number of OS/FS Projects Involved In at Current



In the following sections, the major concern is given to the overall experience in OS/FS projects of developers, because this figure indicates the role of the developers over a longer period, while current project involvement may be influenced by accidental or casual circumstances. Therefore, we concentrate on the overall number of projects performed and on the duration of membership in the OS/FS community.

The number of projects a developer has participated in so far is, by nature, determined by the age of the developer, as indicated by the small share of the youngest and the large share of the oldest in the group of those with experience in more than ten projects. In turn, we find a relatively large share of the youngest among those with experience in only one to five projects (figure 25). However, besides these findings the overall pattern does not show a strong influence of the age of the developers on their project experience.

Figure 25: Number of Performed OS/FS Projects So Far by Age

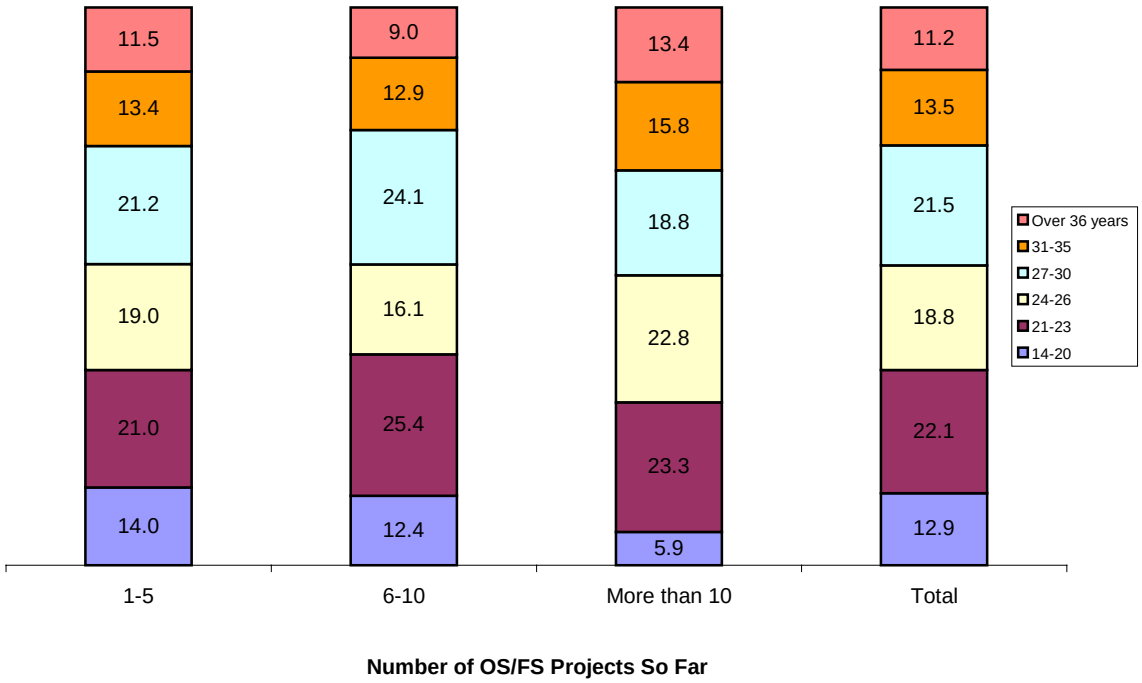
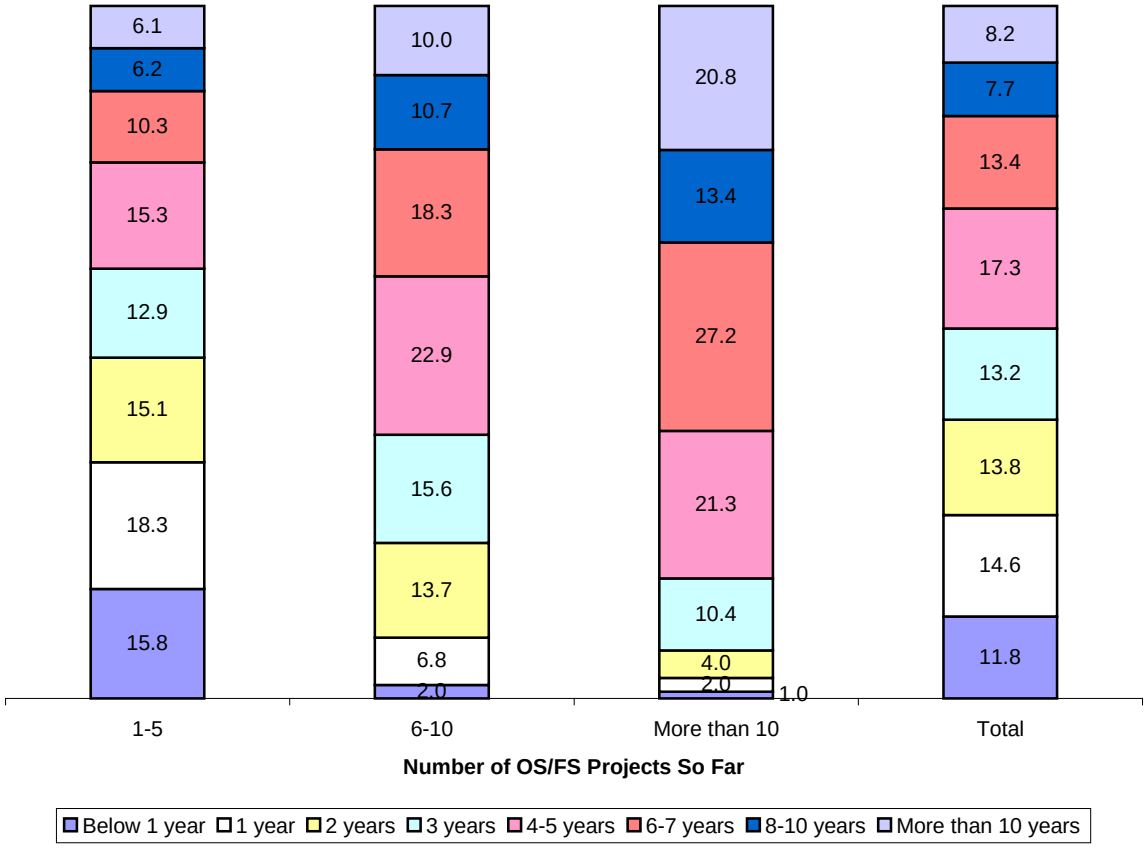


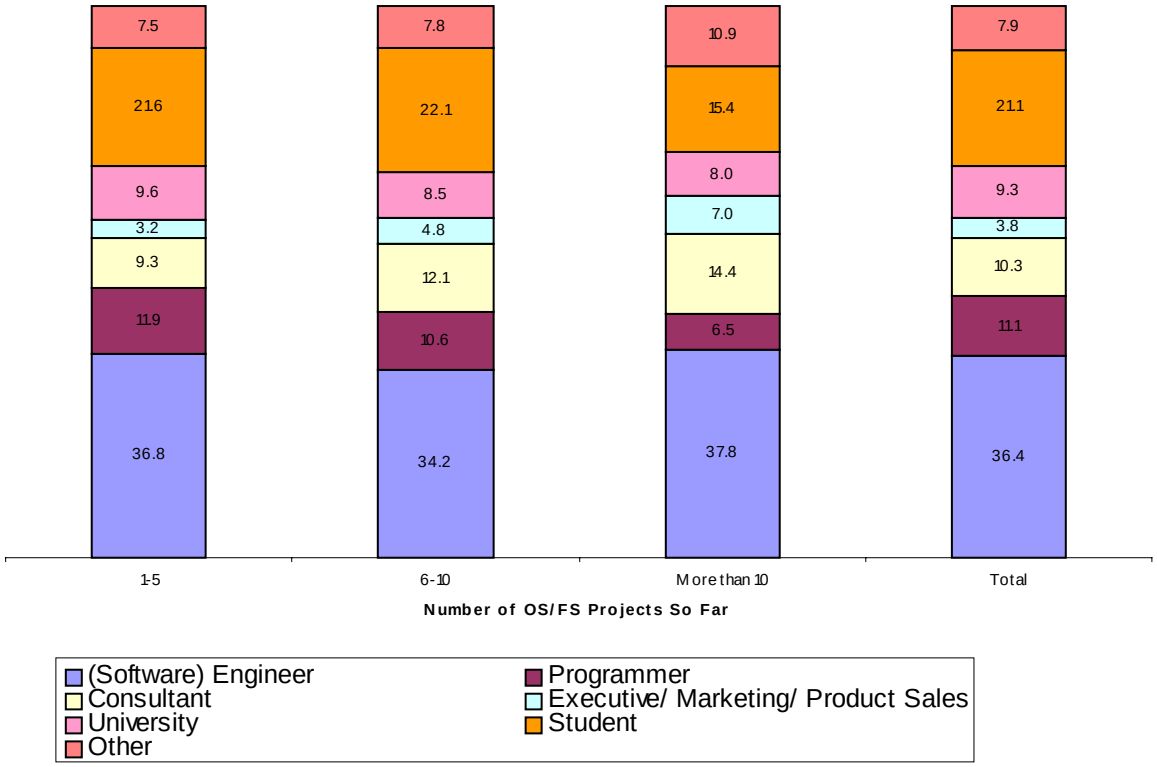
Figure 26 illustrates the correlation between project performance and duration of membership in the OS/FS community. The degrees of activity within the community are obviously very disparate. On the one hand, we find developers who are members of the community for more than 10 years, but who did not participate in more than five projects during this time. On the other hand, we also find developers who stay in the community for not more than two years, but who claim to have already participated in more than 10 projects. Of course, the overall pattern is that the longer a developer stay in the community, the more project experience he gets.

Figure 26: Number of Performed OS/FS Projects So Far by Duration of Membership in OS/FS Community



Occupational issues do have an impact on project involvement, too. As illustrated in figure 27, while software developers play a strong role in all the differently experienced groups, programmers do not show such an impact. Moreover, like with university staff and students, the role of programmers decreases with growing project performance. Astonishing is the role of consultants, executives, marketing and sales professionals, and those who perform occupations that are not specified here. Their shares increase considerably with growing project experience. Thus, these groups may not play a major role in terms of quantity, but they are obviously very important for the community in terms of project organization and performance.

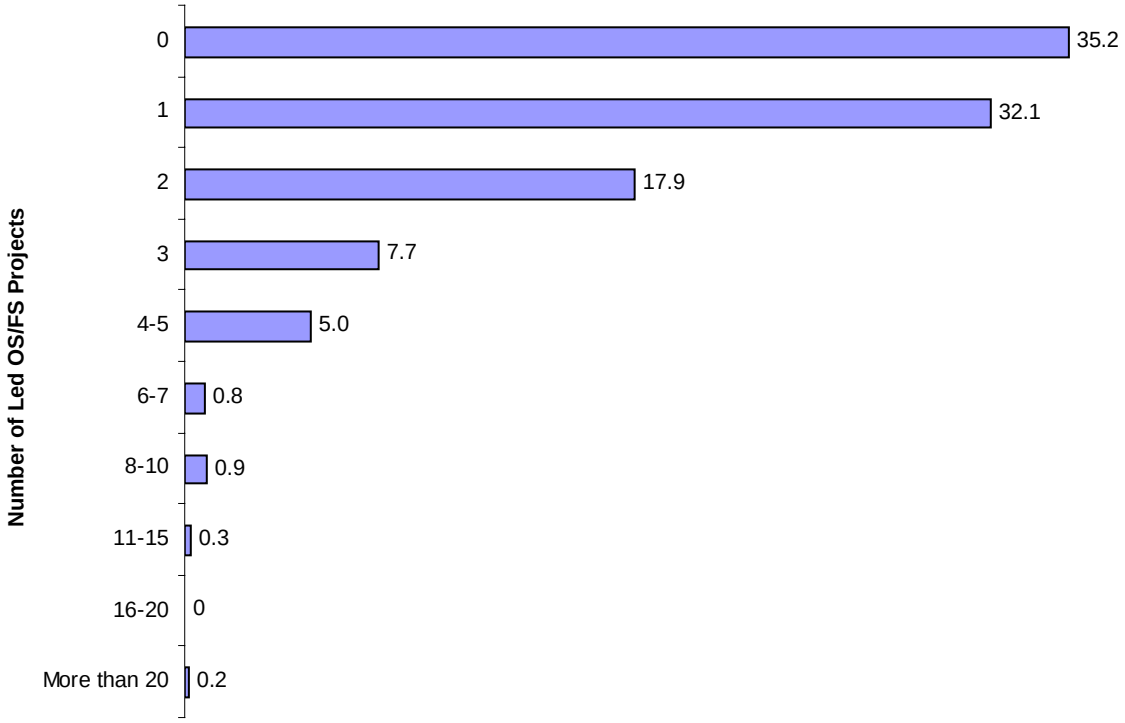
Figure 27: Number of Performed OS/FS Projects So Far by Occupational Background



3.4 Project Leadership

Project leadership provides an important indicator for the experience of a software developer and for the organisational patterns of OS/FS projects. Figure 28 illustrates the leadership experience of the respondents of the FLOSS survey. More than one third of the developers did not lead an OS/FS project since they entered the community, and roughly another third disposes of experience as a leader, administrator or coordinator in one OS/FS project. 18% of the respondents have led two OS/FS projects so far, roughly 8% led three projects, and 5% provide leadership experience from four to five projects. Thus, only 2% of the developers are experienced in project leadership from more than five projects.

Figure 28: Number of OS/FS Projects Involved in as a Leader, Administrator, or Coordinator



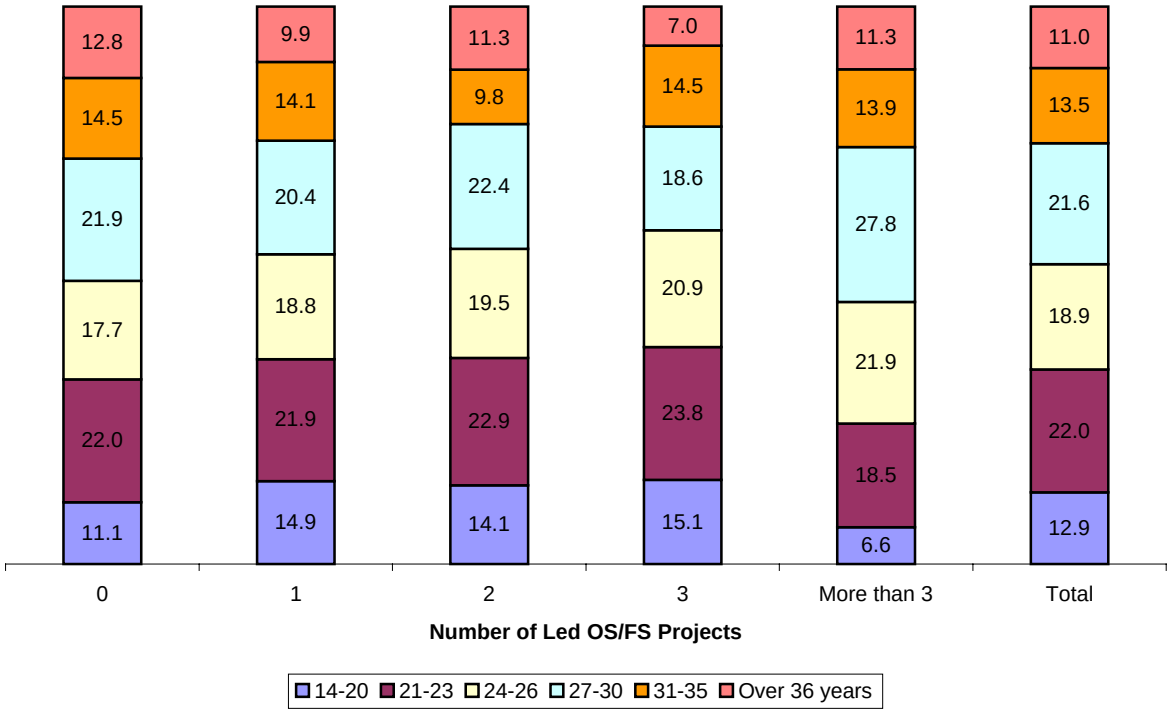
Reasonably assumed, there should be a clear impact of age on leading projects in the way that the older a FLOSS developer is, the more he is experienced in project leadership. Actually, we find the two youngest groups underrepresented in the group of those who have led more than three OS/FS projects so far, while they show above average shares among those who have led one project (figure 29). Accordingly, the shares of the older groups lie above the average in the group with leadership experience in more than three projects, and the oldest group is underrepresented among those who only led one project so far.

However, besides these results we find two considerable exceptions from this overall pattern, so that by and large our findings do not reveal the assumed clear correlation between leadership and age. The first exception is provided by the age structure of those who never have led an OS/FS project. While the three oldest groups are represented here above average, the youngest group appears underrepresented. Those who have led three projects so far provide the second and probably most important exception. Here, the oldest group and those between 27 and 30 are clearly underrepresented, while the three youngest groups show shares that lie noticeably above

the average.

Thus, we conclude that there is a certain impact of age on leadership experience within the community of OS/FS developers, which follows the expected direction. But the community provides also good opportunities for younger members to become experienced as a project leader, administrator or coordinator. We assume that the strong growth of the community in recent years has was aligned with (or maybe even caused by) an increase of the number of OS/FS projects, which exceeded the capacity of the established and experienced heads of the community and which required young and rather inexperienced people to fill the gap. However, this is just an assumption, which becomes more support from the following examination of the impact of the professional background on leadership organisation, but which remains to be examined by further research projects.

Figure 29: Number of OS/FS Projects Involved in as a Leader, Administrator, or Coordinator by Age



Like in the context of project performance, we find a strong influence of the membership duration in the community on leadership experience (figure 30).

Figure 30: Number of OS/FS Projects Involved in as a Leader, Administrator, or Coordinator by Duration of Membership in OS/FS Community

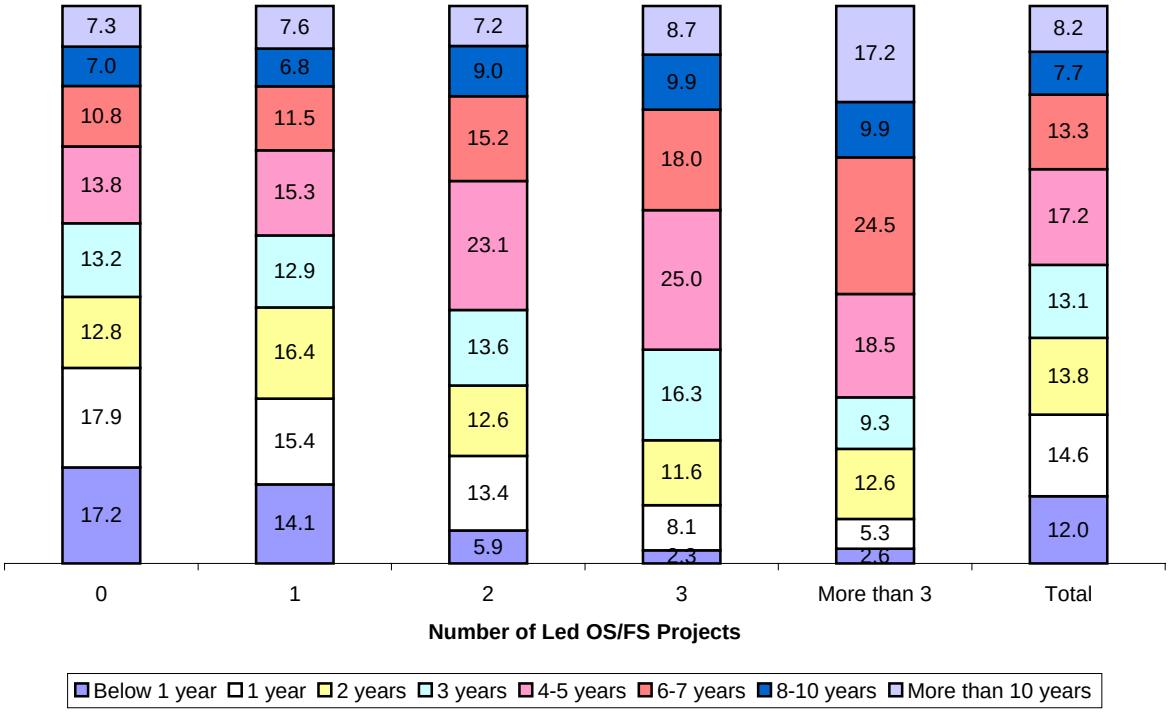
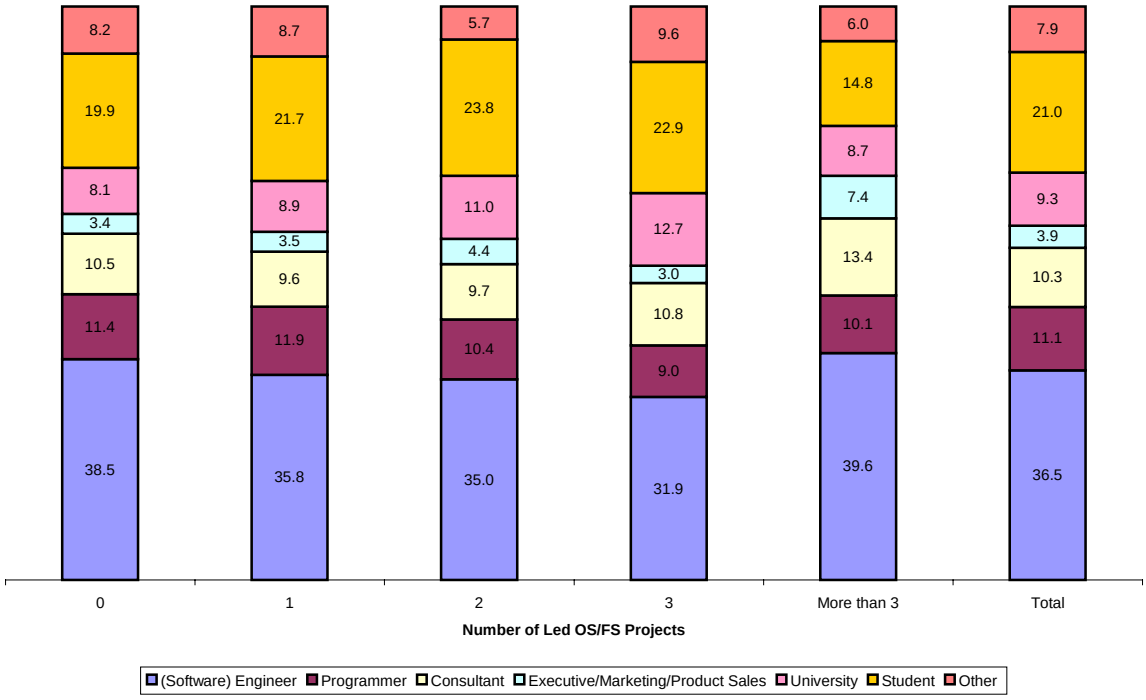


Figure 31 shows the relation between the number of led projects and the occupational background of the FLOSS developers. Again, we find ambivalent trends. For instance, (software) engineers play an important role among those who provide leadership experience from more than three OS/FS projects, but as well among those who never have led a project. In this context, our previous findings about the specific role of consultants, executives, marketing and product sales professionals for the organization and performance of OS/FS projects become more evident, as they feature a very strong tendency towards project leadership. University staff members are also involved in projects as leaders, administrators, or coordinators, whereby they show above average shares within the two middle categories, i.e. the groups of those who have led two or three projects. Recalling the students' pattern of time spending for developing OS/FS, their involvement in projects as leaders, administrators, or coordinators appears rather astonishingly. They are underrepresented among those who never have led a project, but show above average

shares among those who have led one, two, or three projects. However, their involvement decreases considerably within the group of those who have led more than three projects. Probably, this engagement of students will explain largely the previous findings about the role of young people in project leader positions.

Figure 31: Number of OS/FS Projects Involved in as a Leader, Administrator, or Coordinator by Occupational Background

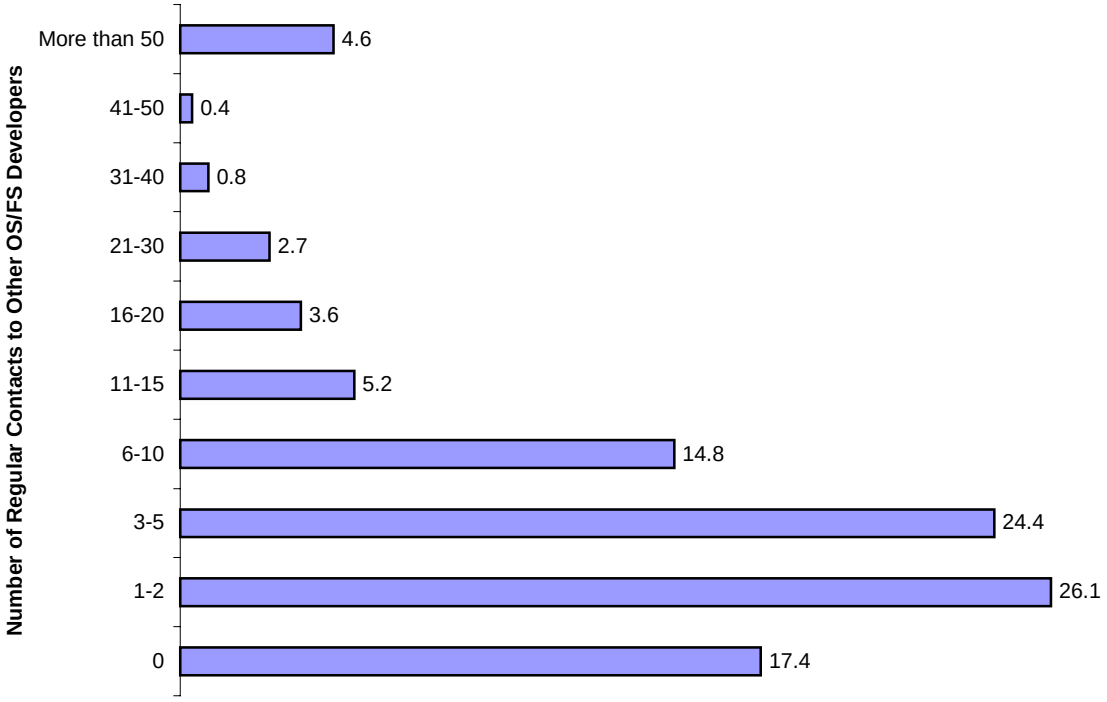


3.5 Contacts and Central Players within the OS/FS Community

As a specific feature of the development of Open Source / Free Software, everybody in this community is free to take up existing software code, to refine it, and then to distribute it again. Thus, OS/FS projects are usually aligned with a high degree of collaboration and communication between numerous people. Therefore, members of the OS/FS community often stress the socialising effects of collaboration according to the principles of this community. However, the crucial question is how the individual developer perceives this collaboration and communication. Due to different life-styles, different degrees of engagement in the community, and different social capacities, it is to assume that there are strong differences in the socialising behaviour of OS/FS developers and in their perception of their OS/FS-related environment. For instance, a

developer who takes up software code of other developers may consider himself as part of the OS/FS community as a whole, not caring for the names of these other developers; or he may consider himself as part of a team, regardless of its size and whether he knows the other team members personally or not; or he may consider himself as a performer of a one-man project, leaving thoroughly aside the contributions of others.[3] Of course, this problem cannot be clarified by the means of an online survey. However, the FLOSS survey allows a first approach to this problem by assessing the number of contacts to other OS/FS developers within the community and the consciousness of the respondents of OS/FS developers that are considered as prominent representatives and "heads" of the community.

Figure 32: Number of Regular Contacts of OS/FS Developers to Other Members of the Community



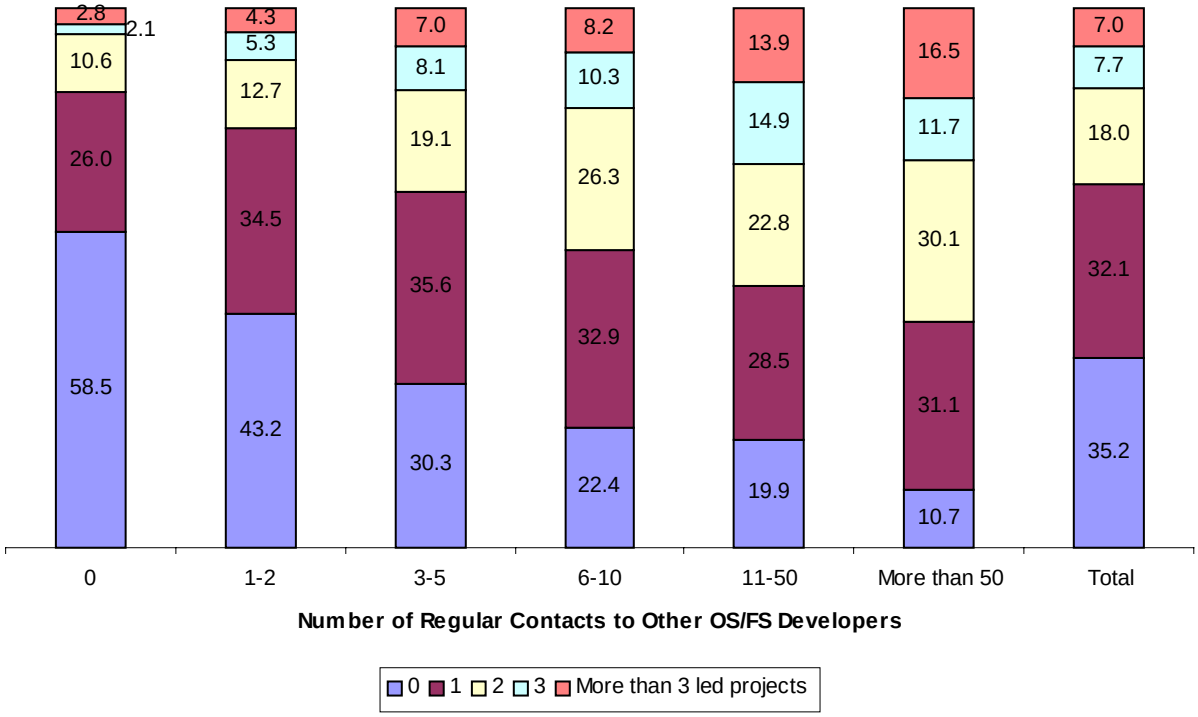
As revealed by figure 32, 17% of the OS/FS developers do not maintain any regular contacts to other members of the community. Approximately one fourth, respectively, features regular

[3] This is the reason why we did not offer a pre-formulated definition of „project" in our online questionnaire and to leave it to the respondent what he understands as a project. A project can thus either be ascribed to a specific team or consortium, to the specific product or to oneself. Thus, it may be possible, however not very likely, that *leading a project* could mean every own effort to contribute to the production of a specific product without direct collaboration with other OS/FS developers.

contacts to one to two or to three to five other OS/FS developers. Thus, two thirds of the sample keeps regular contacts to no other community member or to a very limited number of other OS/FS developers. 15% keep contacts to six to ten developers, and 5% of the sample claim to have regular contacts to 11 to 15 other developers. 4%, respectively, maintain contacts to 16-20, to 21-50, and to more than 50 other developers.

It appears noteworthy that the shares of developers decrease considerably between 16 and 50 contacts, but than increase again with those who claim to have more than 50 regular contacts in the scene. A deeper analysis of the data has revealed that it is by no means only the group of the oldest (over 36 years old) that makes up this highly connected group. To a larger extent, this group consists of developers aged 21 to 26. However, this group provides clearly the most active community members in terms of project leadership (figure 33).

Figure 33: Leadership Experience of OS/FS Developers by Regular Contacts to Other Members of the Community

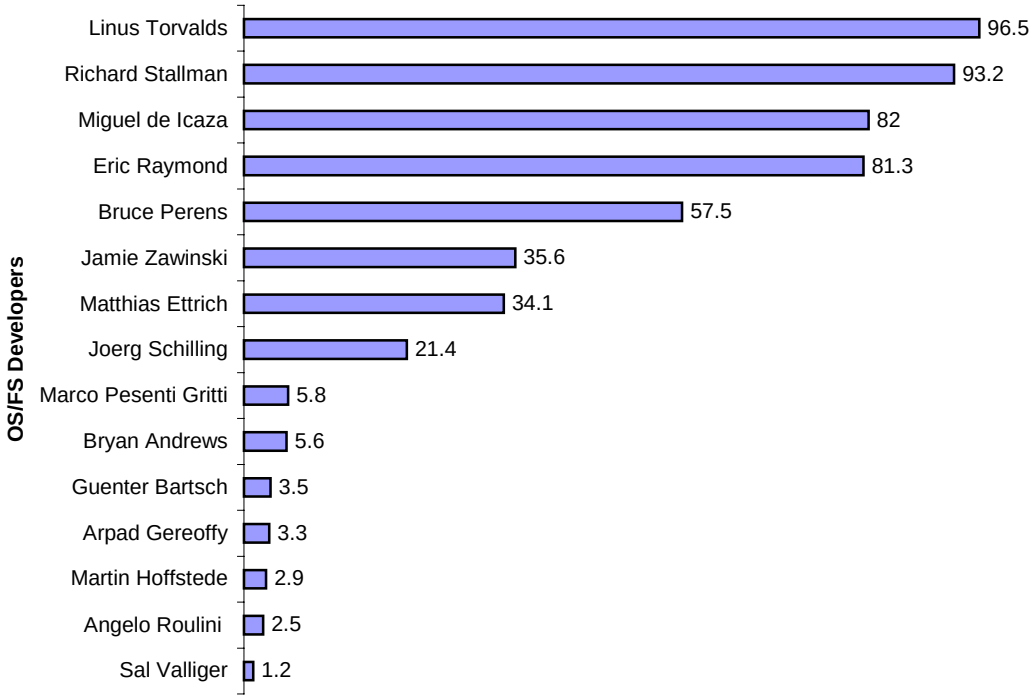


As shown in figure 33, there is a very strong correlation between leadership experience in OS/FS projects and contacts to other OS/FS developers: The more contacts an OS/FS developer has to

other members of his community, the more experienced he is in leading OS/FS projects – and vice versa.

Finally, we offered a list of names of publicly known OS/FS developers and asked the participants of the survey to tick all names they know. The result of this multiple response question is shown in figure 34. The main purpose of the list was not to find out the most famous OS/FS developers, but to identify people who are not very well oriented about the key players of the OS/FS scene and the perception of this scene by the public. For this purpose – and in order to get a (weak) indication of people who do not belong to the OS/FS scene, but filled in the questionnaire - we added the names of three non-existing persons to the list. These names are Martin Hoffstede, Angelo Roulini, and Sal Valliger. As figure 34 shows, the respondents were very well oriented about the key players of the community. The shares of the three names lie between 1% and 3%, i.e. the amount of "wrong" answers is insignificant.[4]

Figure 34: Known Central Players of the OS/FS Scene



[4] In the whole sample appear only four respondents who claim to know all of these three persons. Further examination revealed that these developers are strongly engaged in the OS/FS scene in terms of project participation and leadership. Thus, their answers rather indicate that they have recognised that these names are wrong by purpose than that they have ticked these names without knowing these persons do not exist.

4. Motivations, Expectations, and Orientations of OS/FS Developers

4.1 The Social and Political Dimensions of the Open Source/Free Software Community

The OS/FS community is often considered - by its members as well as by outsiders - as not only a community of people who merely develop software, but as a rather political community. There are several different reasons that may have caused this perception, of which the following three may be the most important. Firstly, the kind of struggle comes in mind that members of the OS/FS community fight against proprietary software and large companies like Microsoft. To limit the power of single economic actors is an objective of every market economy, thus the fight against monopoles and trusts fits very well into the capitalistic concept of free markets and fair competition. However, that is the second point, the OS/FS community seems to have a certain wish to escape from the fundamental laws of capitalist economies, as apparently expressed in the rejection of private property rights on their products and the way in which their software code is exchanged and refined - free of any charge. The fact that these "anti-capitalistic" procedures are realized practically, in real life and basically right within the environment of capitalism, may trigger many people who criticize the rigidity of markets, but could not find a solution to escape their grasp. Secondly, the internal discussions about the application of either the term "Open Source Software" or "Free Software" has attained the character of a fundamental philosophical discourse that reaches far beyond the realm of developing software. Implicitly, it reflects the choice between two different fundamental self-perceptions, aligned with different life-styles and political conceptions of the world.

Nevertheless, it is evident that the world of Open Source/Free Software is not strictly separated from capitalistic principles, and that a lot of money can be earned by the development or application of OS/FS, like it is illustrated by the example of LINUX. Therefore, the individual wish to contribute to the OS/FS community can be caused by a variety of reasons. Preferences for technical aspects of software; political convictions like the wish to change the way how society and economy deal with software; social aspects of sharing information and interests with others; the wish for self-realization; the wish to make profit; and other reasons, can be mixed-up and, thus, cause essential differences within the OS/FS community as a whole.

Therefore, the project has to elicit the motives of software developers to develop, distribute, and

exchange Open Source / Free Software.

4.2 Motivations for developing Open Source / Free Software

We have been interested into the motives of people to join the OS/FS community from two different perspectives. Firstly, we wanted to know which motives have been causal to join the community, and secondly, we were interested in the motives that keep the developers staying in this community. Figure 35 illustrates the answers two these two questions, whereby the respondents had the choice to tick a maximum of four answers.

Most of the respondents ticked reasons that resided on the individual skills level, but there is also evidence of a social aspect. Almost eight out of ten software developers started with OS/FS because they wanted to learn and develop new skills, and half of the sample claimed that they wanted to share their knowledge and skills with other software developers. Not surprising, with regard to the reasons to stay in the community we observe that the first reason has lost some of its importance, while the second reason has increased.

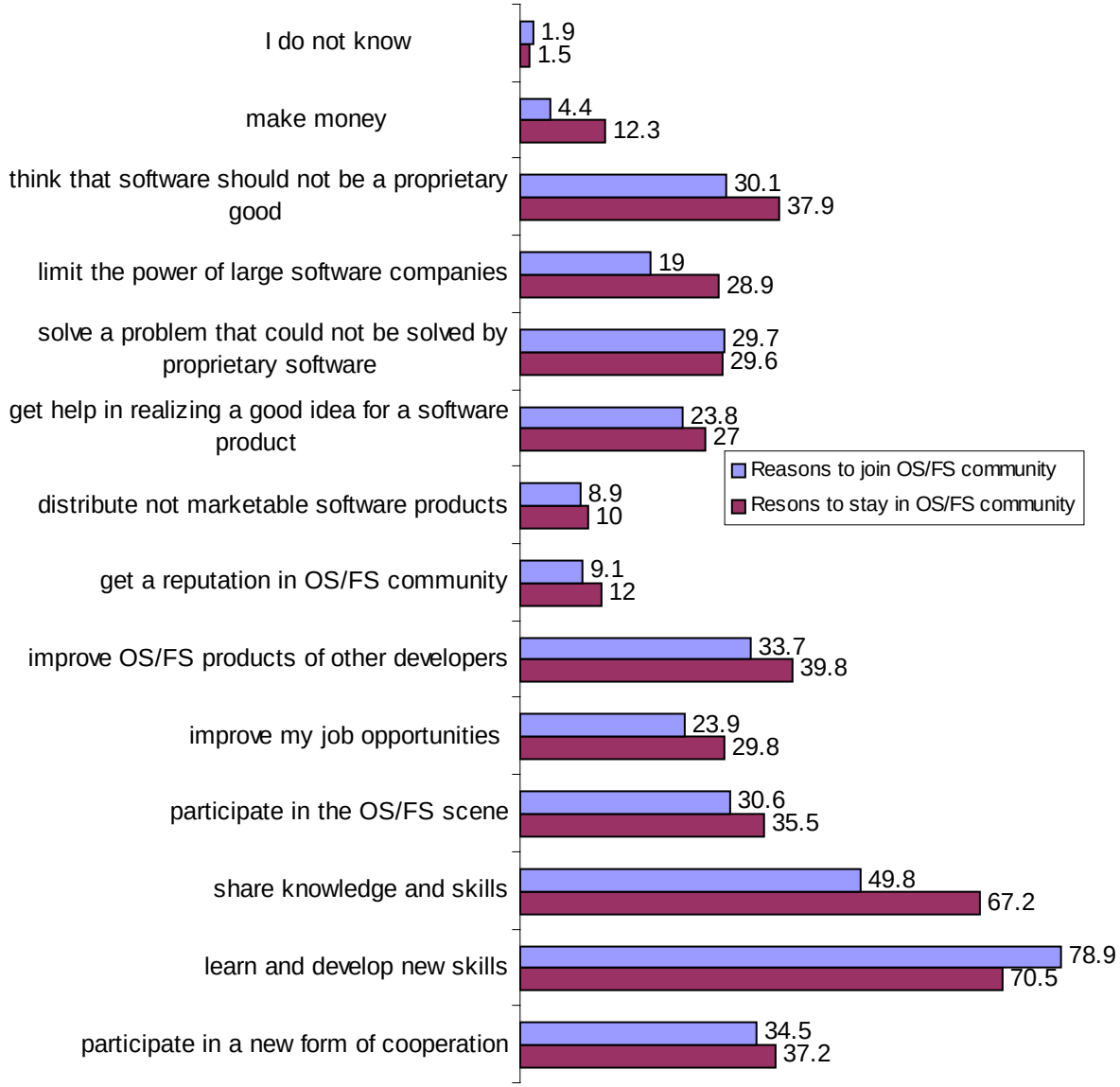
The third important group of reasons is provided by motives that reach shares between 30% and 35%. 35% of the sample emphasized their wish to participate in new forms of cooperation that are associated with OS/FS development, and another 34% emphasized aspects of the goods produced in OS/FS by stating that they wished to improve software products of other developers. 31% state that they wanted to participate in the OS/FS scene, and 30% were convinced that software should not be a proprietary product. Thus, after skill-related aspects, a set of rather heterogeneous reasons motivate people to join the OS/FS community, ranging from social and work- as well as product-related issues to a rather political opinion. It is noteworthy that all these reasons gain importance after the developer has joined the community and got some experience, which particularly applies to the product-related and the political item.

The next important group of motives, featuring shares between 20% and 30%, again comprises product-related issues (“solve a problem that could not be solved by proprietary software” and “get help in realising a good idea for a software product”, but also a material motive (“improve my job opportunities”). While the motive to get help in realizing an idea for a software product shows no change in its importance, the other two items, especially the motive to improve job opportunities by contributing to OS/FS, gain considerably importance.

Finally, another important motive to start and to go on with OS/FS resides in the wish to limit the power of large software companies, which is ticked by one fifth of the sample. This item shows a very strong increase with growing experience of the OS/FS developers within their community.

All the other motives that were offered to the respondents reached only shares below 10% and will be neglected here. However, we have to except the motive to make money from this rule, because this items gains a lot of importance as a reason to continue with OS/FS, growing from 4% to 12%.

Figure 35: Reasons to Join and to Stay in OS/FS Community



As a conclusion from these observations, we witness an initial motivation for participation in the OS/FS community that rather aims at individual skills and the exchange of information and knowledge with other developers, but over time a maturing of the whole community with regard to both, commercial (material) and political aspects.

4.3 Expectations Related to the OS/FS Community

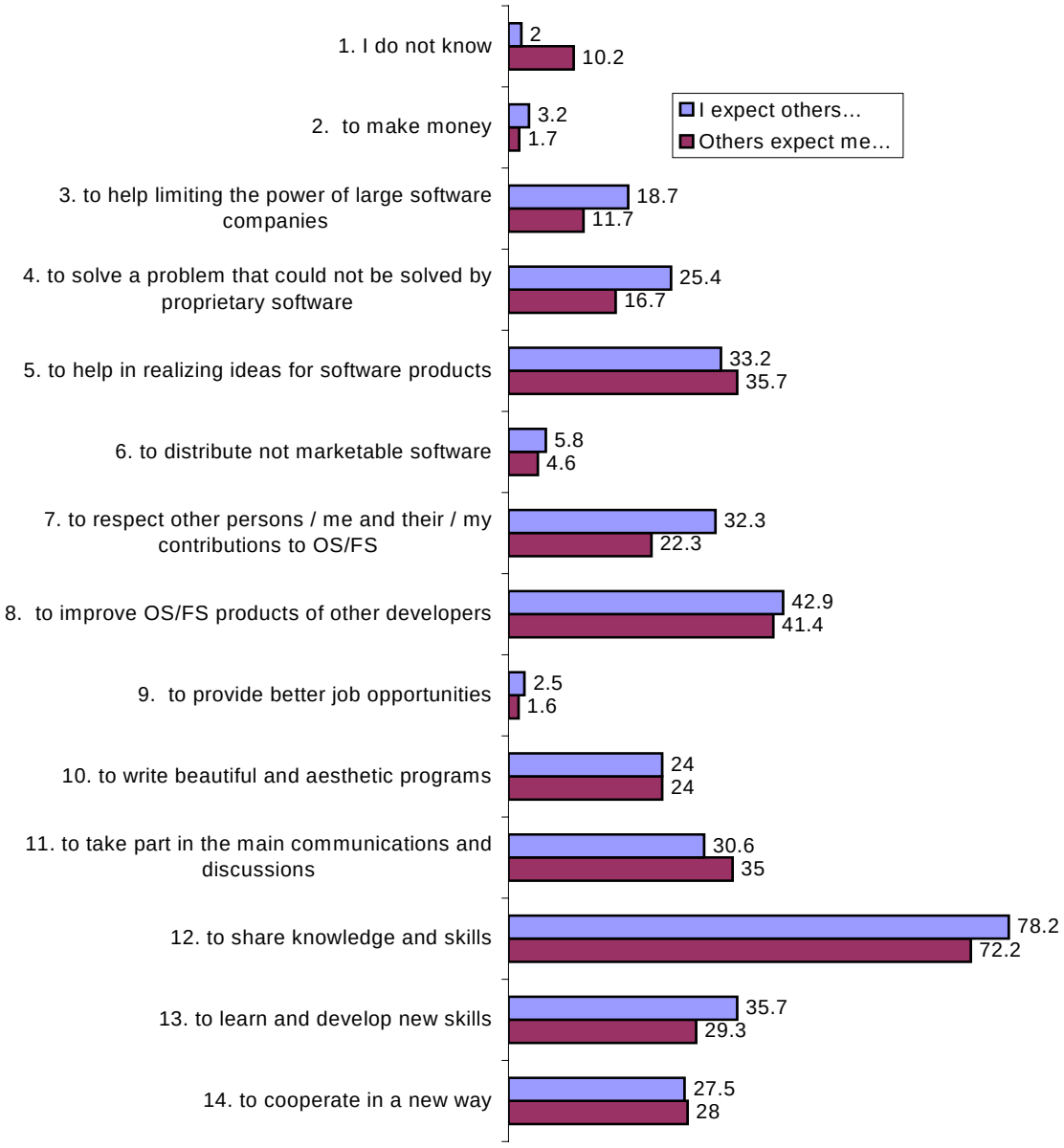
We examined the aspects of motivation also in a different context, as we asked the respondents about their expectations from other OS/FS developers and about their assumptions about other OS/FS developers' expectations from them. The purpose of our return to these aspects is to crosscheck their motives in another perspective, which reflects rather the perceived characteristics and general rules and principles of the community than insights in personal (individual) motivations and interests.

If we compare the own expectations of OS/FS developers from other OS/FS developers' expectations with the assumed expectations of other OS/FS developers from the respondent, we find a considerable degree of coherence, but also some remarkable differences (figure 36). Apparently, OS/FS developers are to a certain degree convinced that they can learn more from other OS/FS developers than others can learn from them, as indicated by items 12 and 13. Besides these skill-related expectations, product-related orientations play an important role in the perception of the requirements of the OS/FS community by its members (items 5 and 8).

There is a considerable difference in the "political" dimension of OS/FS development, as 19% of the developers expect others to help limiting the power of large software companies, but only 12% assume that they are expected to do so by other software developers. Since this aim is valued higher on the individual level than it is assumed as a general expectation or goal of the community, we would like to conclude that there is at least uncertainty about the importance of this issue within the OS/FS community.

Finally, in contrast to the motivations expressed in figure 35, material aspects do not play a role in the expectations from other community members, and they either play no role in the assumed expectations from other developers from the respondents.

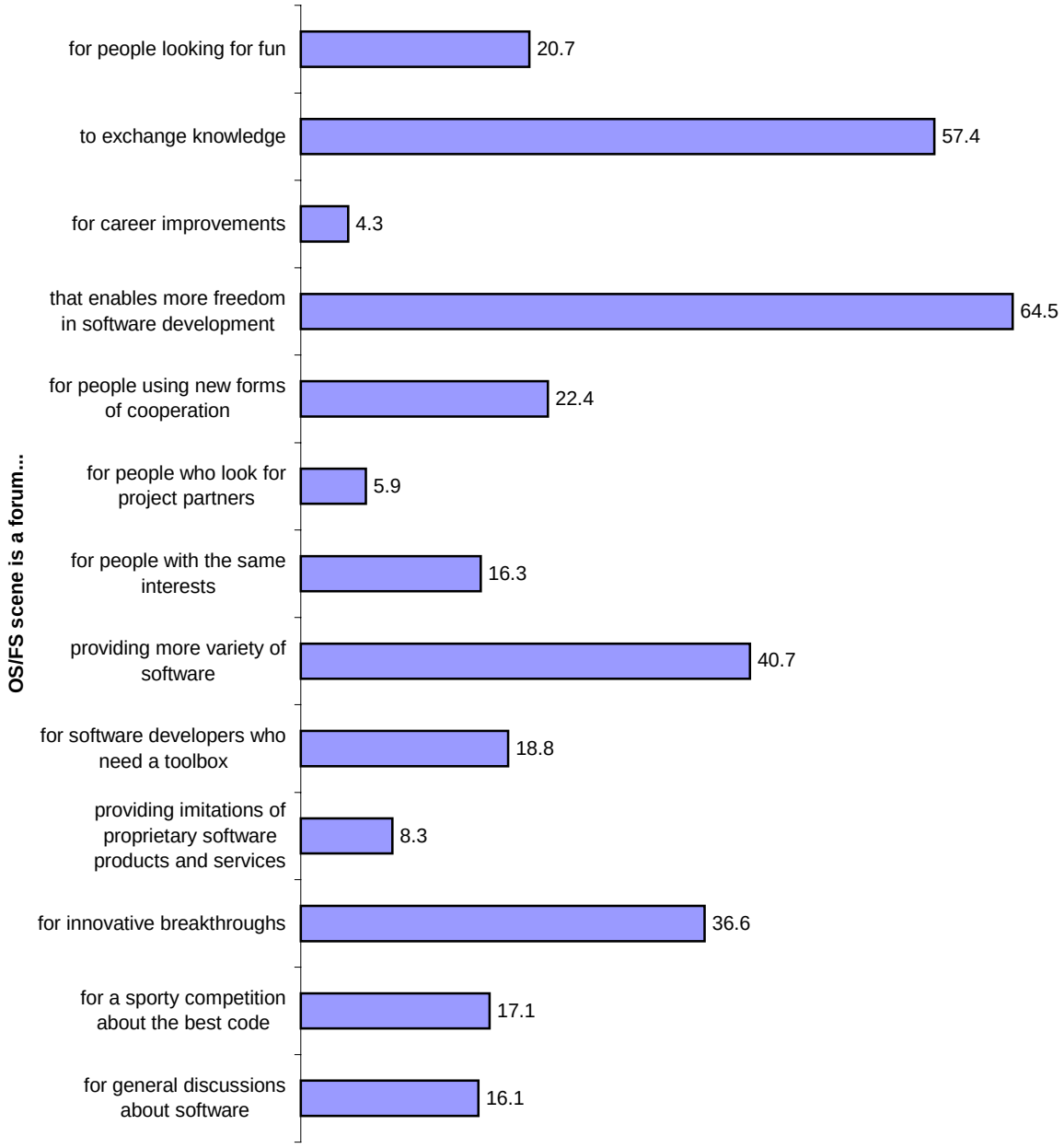
Figure 36: Expectations of OS/FS Developers from other Members of the Community and Assumed Expectations of other Community Members from the Respondents



As a last attempt to approach the general orientations and attitudes of the FLOSS developers from different perspectives, we tried to find out how they perceive the community of OS/FS developers as a whole, i.e. what purposes this community serves according to their opinion. Therefore, we draw a picture of the community as a forum where information can be changed, contacts can be established and maintained, and a variety of interests can be satisfied. Figure 37 shows how the

developers valued the different items. Thus, the societal and economic utility of the community comes under scrutiny.

Figure 37: Perception of the Purposes of the OS/FS Community by OS/FS Developers

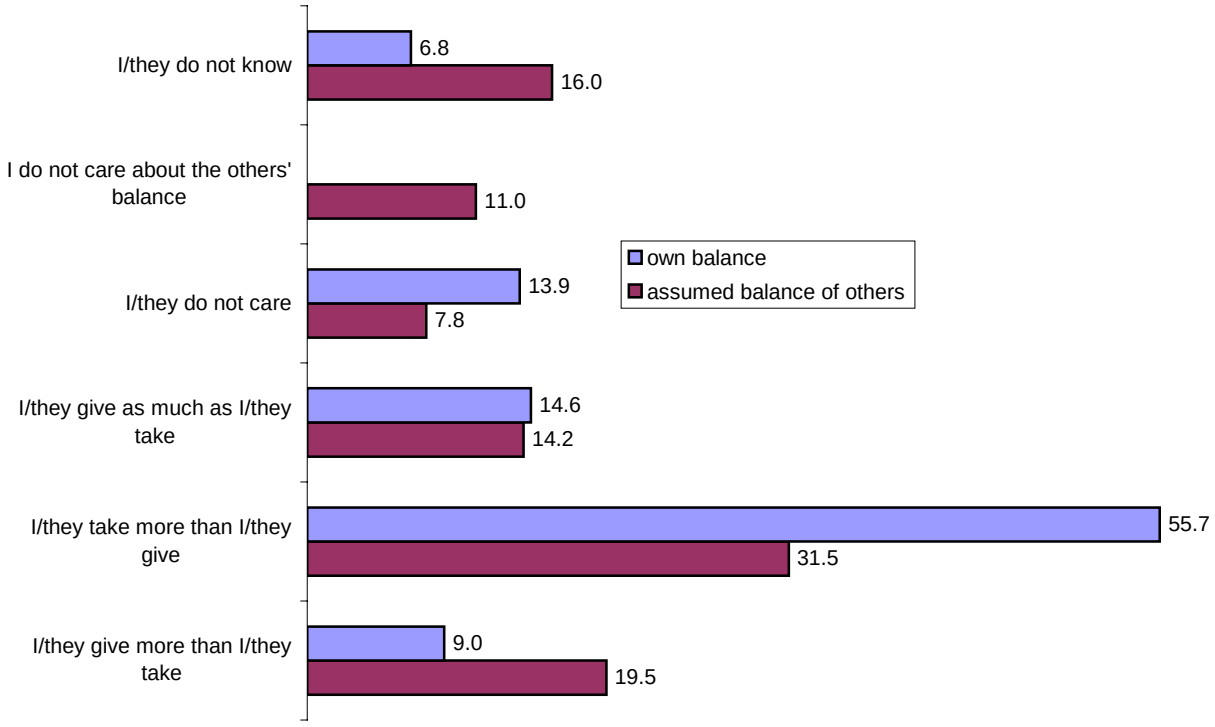


To provide more freedom in software development and to serve as an institution where knowledge can be exchanged are the two most pronounced purposes of the OS/FS community, followed by the provision of more variety of software and by support for innovations, which are very similar items. Thus, we witness a clear coherence with the otherwise mentioned motives and orientations, the strong interest in skill improvements and a strong wish for a large variety of software products, which apparently is considered as a value itself. It is noteworthy that all material or hedonistic interests are clearly outplayed by these items.

Finally, we asked the respondents to evaluate their participation in the OS/FS community in kind of a balance. Again, they were asked to give their personal balance as well as an assumption about the balance of the other OS/FS developers in order to get more insight in the perceptions of the community and of the own role compared to the role of other members.

Figure 38 shows that the own balance differs considerably from the assumed balance of the other members of the community. Regarding themselves, the developers are apparently convinced that they get more out of the community than they give in. In principle, the same applies to the assumed balance of other developers. However, while the shares of those who state that they give more than they take is only 9%, the assumed share of other developers to which this statement seems applicable is much higher and reaches almost a fifth. In turn, the share of those who claim that they take more than they give is extremely high (56%), while this share decreases to only 32% when the balance of other developers is esteemed.

Figure 38: Balancing Give and Take



5. Dividing Lines – Free Software versus Open Source, OS/FS versus Proprietary Software, and Monetary versus Non-Monetary Rewards

5.1 Free Software versus Open Source Software

To distinguish the fundamental orientations of Open Source/Free Software Developers has to start with the individual self-perception of these developers as part of the "Free Software" community or as part of the "Open Source Software" community. The term "free software" is the older one of the two terms, and the term "open source software" was invented in 1998. Although members of both communities collaborate intensively on practical projects, they claim that, on the level of the underlying ideas and philosophies, both communities have to be considered as entirely separate movements.[5] According to members of the free software community, the intended meaning of

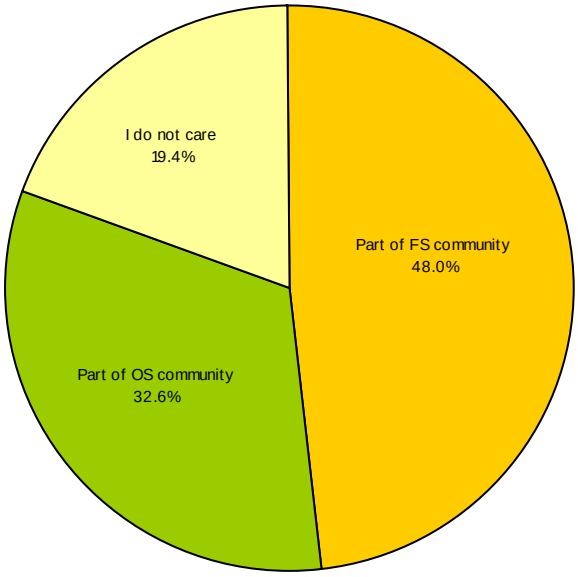
[5] See for example: Open Source - GNU Project - Free Software Foundation (FSF): Why "Free Software" is better than

"Free Software" is "software that gives the user certain freedoms", but the term also invites to the unintended interpretation as "software you can get for zero prize" (cf. www.gnu.org/philosophy/free-software-for-freedom.html). These freedoms contain ethical issues, aspects of responsibilities and of convenience. Members of the open source software community define "Open Source Software" as software that allows everybody to have a look at its source code and stress the practical benefits of such software, while aspects of freedom are rather neglected in the definition. "Open Source Software" contains a broader variety of software than it is allowed by the term "Free Software", it comprises free software as well as semi-free software and even certain proprietary programs. This may be the reason why the term "Open Source Software" is also ambiguous. While members of the respective community stress the closeness of their concept to the concept of "Free Software", it is often merely understood as a strategy to improve the opportunities to "sell" (www.gnu.org/philosophy/free-software-for-freedom.html) the software more effectively to users, especially to business companies.

According to this ongoing discussion, one would expect a sharp polarization of the whole community of developers of non-proprietary software into two very different parties, one of Open Source developers and one of Free Software developers. However, figure 39 shows that, although there is clear evidence of these parties, still a share of almost one fifth of the whole sample does not care anyway if they belong to the one or to the other party. As indicated in figure 41, "Free Software" still plays the most significant role in the community of developers of non-proprietary software.

"Open Source"; http://www.gnu.org/philosophy/free-software-for-freedom.html; last revisited: March 23, 2002.

Figure 39: Self-Assignment of OS/FS Developers to Open Source or Free Software Community

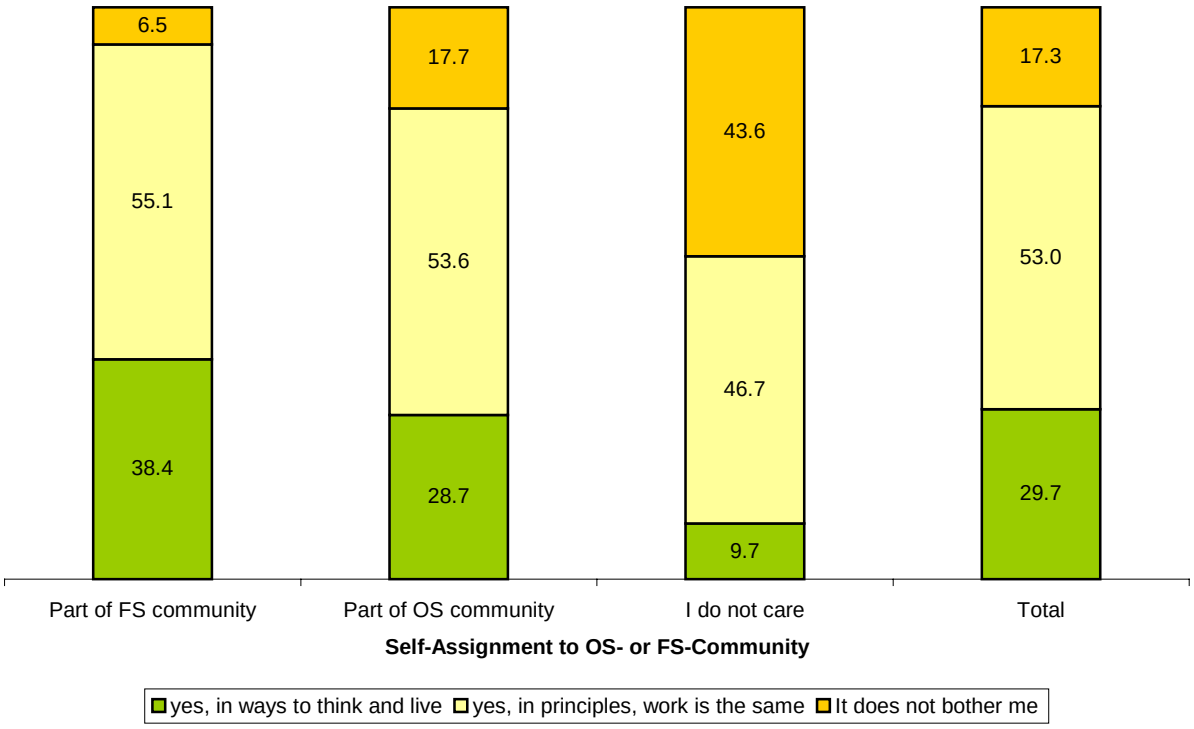


Nevertheless, the self-perception and self-assignment of the developers do not tell much about the significance they ascribe to the distinction between Open Source Software and Free Software. Therefore, we also asked the respondents how they understand the differences between the Free Software and the Open Source Software community. More than half of the sample answered that the difference between the two communities exists only on principle, while work in both communities is considered the same. On the other hand, we can clearly identify a share of approximately 30% that can be seen as the ideological hardcore movement of the two communities. This group claims that the difference between the two communities is not only one in principle, but a clear distinction between two completely different ways to think and work. Finally, a remarkable share of 17% does not care at all about the differences between the two communities (see figure 40).

Figure 40: Evaluation of Differences between Open Source and Free Software

Figure 41 shows that especially those who assign themselves to the Free Software community are striving for a sharp distinction between their community and the Open Source Software community, while the members of the Open Source Software community do largely correspond with the average distribution of answers. Not surprising, those who do not care to which of the two communities they belong, claim significantly less than the other two groups that there are clear distinctions between the two communities, and the lion share of them does not care about any difference between the Open Source Software and the Free Software community.

Figure 41: Evaluation of Differences between Open Source and Free Software by Self-Assignment to OS- or FS-Community

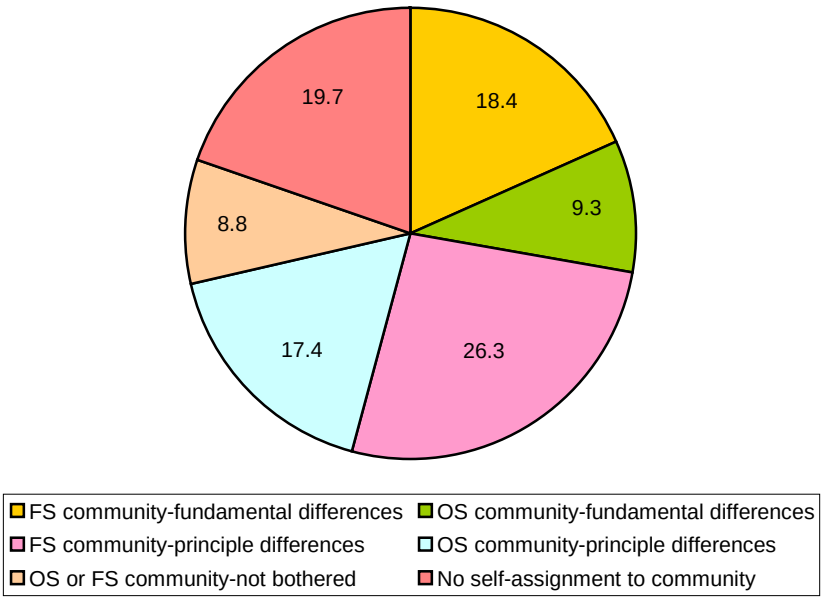


Based on this self-assignment and the respective evaluation of the perceived differences between Open Source and Free Software, we can distinguish six types of OS/FS developers, as they are illustrated by figure 42:

- The first type consists of those developers who assign themselves to the Free Software community and who see fundamental differences between the two communities (18%).
- The second type consists of those developers who consider themselves as part of the Open Source community and who perceive fundamental differences between the two communities (9%).
- The third type is made up by those developers who assign themselves to the Free Software community and who perceive only principle differences between the two communities, but consider work in the two communities the same (26%).
- Accordingly, those developers who assign themselves to the Open Source community and see principle, but no fundamental differences between the two communities provide the fourth type (17%).

- The fifth type consists of developers who assign themselves to either the Free Software or the Open Source Software community, but are not bothered by differences between the two communities (9%).
- Finally, those developers who do not care to which community they belong provide the sixth type (20%).

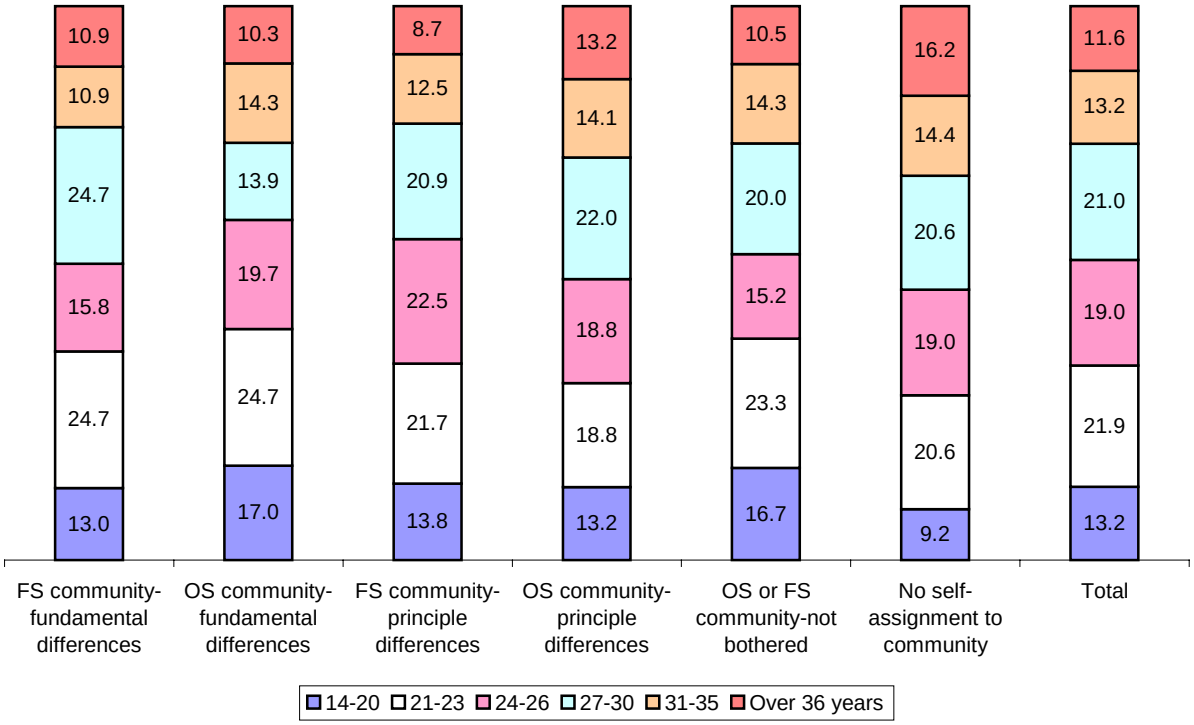
Figure 42: Perceived Differences between Open Source and Free Software - Typology of OS/FS Developers



The age of the developer has a certain, but no strong impact on this typology. As shown in figure 43, type 1 shows above average shares of those who are between 21-23 and between 27-30 years old, while type 2 shows above average rates of the two youngest groups. The two middle types (only seeing principle differences) do not differ noteworthy from the average, with the exception of type 4, which features a high share of members older than 36 years. Those who belong to either the one or the other community, but are not bothered by differences between them (type 5) show relatively high shares of the two youngest groups, but do not differ considerably from the average, either. Finally, those who do not assign themselves to a community are clearly characterized by above average shares of the oldest and the second oldest group. Thus, we see a

rather disparate picture of the age structure of our typology, with a small tendency of the younger generations to the first two types, while for the older generations the differences between the two communities seem to lose their importance.

Figure 43: Perceived Differences between OS- and FS-Communities (Typology) by Age

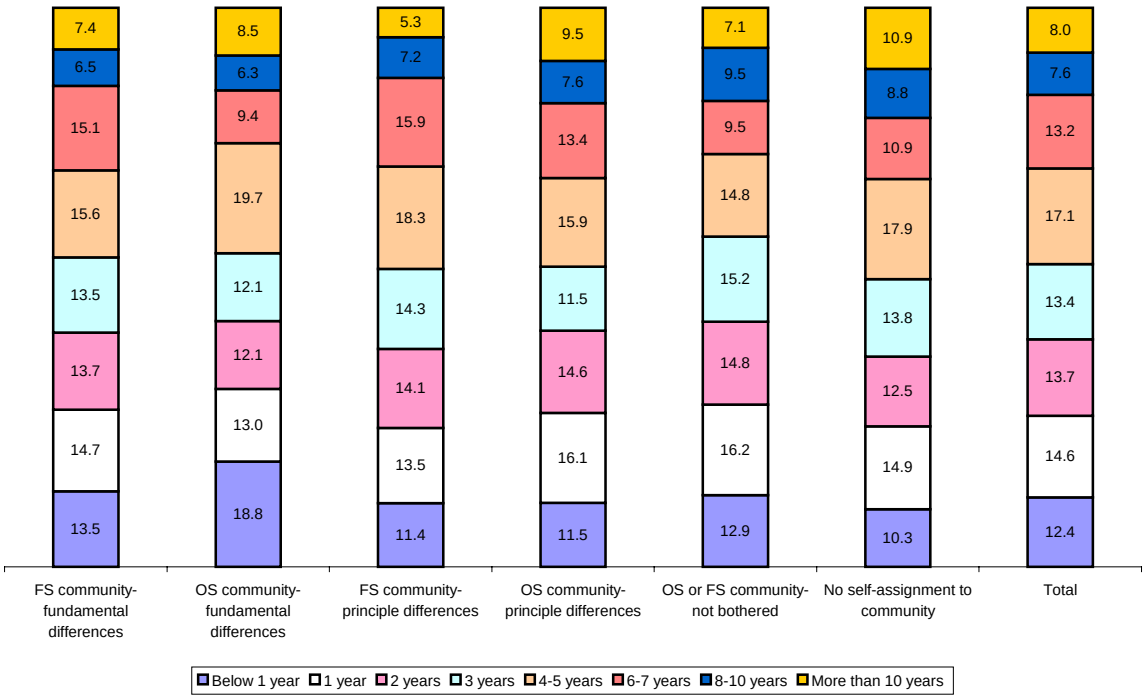


Although the duration of membership in the OS/FS community is to a certain extent a function of the age of the developer, we can identify some specific features from this characteristic (figure 44). For instance, with the exception of those who stay in the community for 6-7 years, the first type (belonging to FS-community and seeing fundamental differences to OS-community) shows no remarkable deviation from the average. Thus, belonging to this ideologically clearly defined group is obviously not a matter of maturing in the scene, although we already found that political issues gain importance with growing experience in the OS/FS community. In contrast, the counterpart to this type (type 2) is characterised by a clearly above average share of the youngest group, and a slightly above average share of those staying in the community since 4-5 years, while those staying there since 6-7 years are clearly underrepresented. The third type seems to be in between type one and two, as it features above average shares of those who stay in the community for 4-5 as well as 6-7 years. The fourth type is characterised by an ambiguity, as it

shows relatively high shares of rather new members of the community (up to one year membership) as well as of developers who belong to the community for more than 10 years. Those who belong to the one or the other community, but are not bothered by differences (type 5), are qualified by relatively large shares of members who belong to the scene from one up to three years, but also members who belong to it for 8-10 years. Like with the age structure, the sixth type is characterised by only slight differences to the average structure, whereby those who belong to the scene for more than ten years are represented clearly above the average.

Conclusively, neither age nor membership duration in the OS/FS scene have a strong impact on the ideological orientation of the developers.

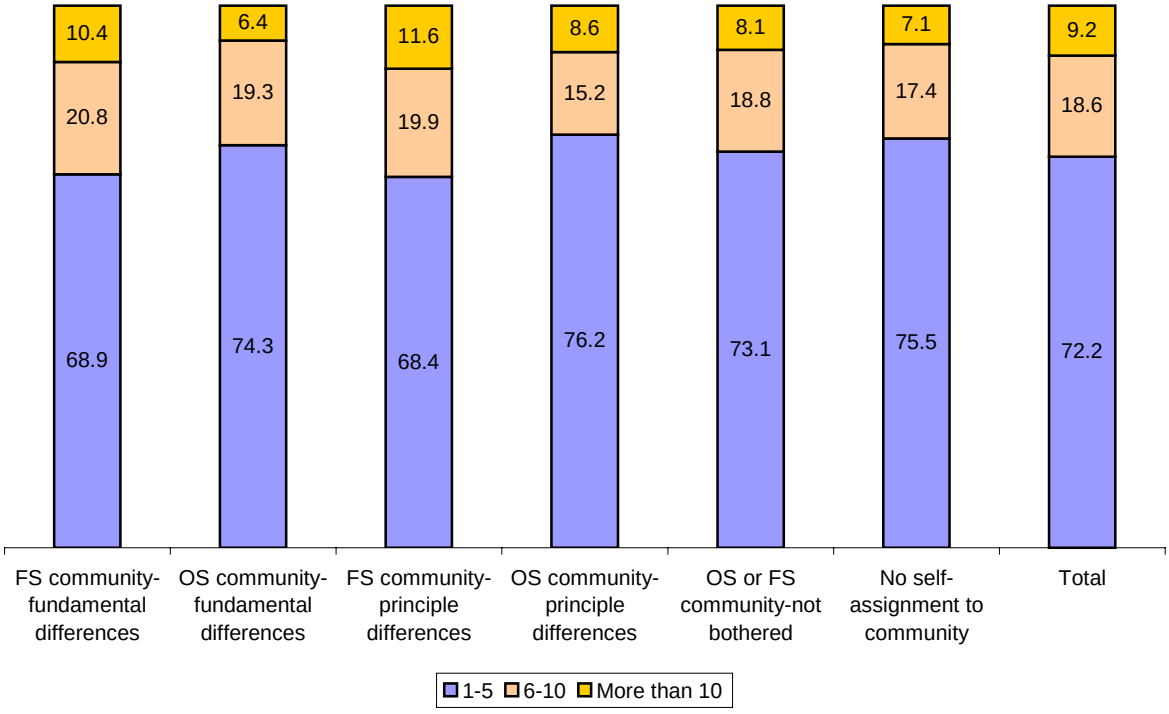
Figure 44: Perceived Differences between OS- and FS-Communities (Typology) by Duration of Membership in the OS/FS Scene



Still, the number of performed projects has a slightly stronger impact on the perception of differences between the two communities than age or membership duration (figure 45). Type 1 is observably built by developers who have participated in at least 6 or even more than 10 projects, while type two is characterised by developers that have participated in 1-5 or 6-10 projects. Thus, one may conclude that a growing project experience is aligned with a tendency to become a

"hardcore" Free Software developer. However, our previous results give reason to be careful with such a conclusion. Another reason to doubt such a conclusion is provided by type three, as it features the highest share of the most experienced developers of all types. The contrast to its counterpart is rather strong, as type four is characterised by a relatively high share of developers who provide experience from not more than five projects. While type five resembles very much the overall structure, type six again shows a relatively large share of rather inexperienced developers.

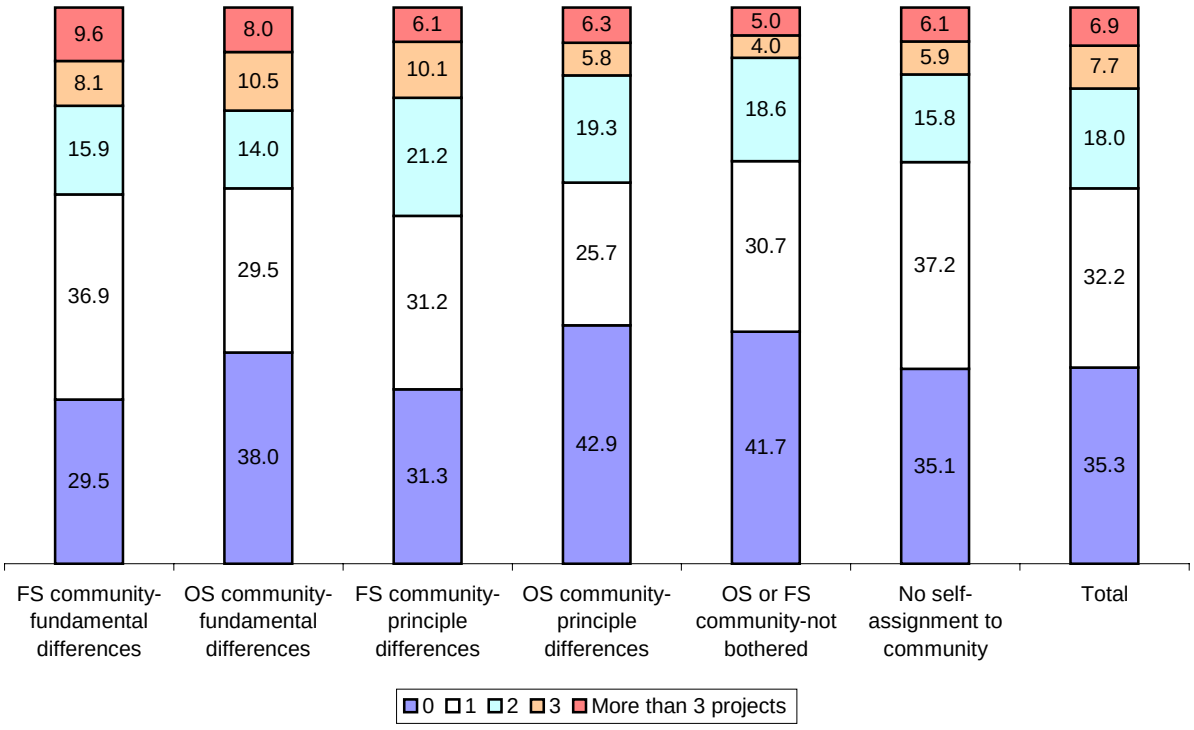
Figure 45: Perceived Differences between OS- and FS-Communities (Typology) by Number of Performed Projects



Finally, we find the strongest impact on the ideological orientation of OS/FS developers provided by the leadership experience of the developers (figure 46). While the most experienced developers show above average shares in both, type 1 and type 2, those who never have led a project are clearly underrepresented within type 1, but play a significant part in type 2. On the other hand, we find an above average share of those who have led only one project among the developers of type 1, but they are underrepresented in type 2. In contrast, type three is mainly characterised by higher shares of developers who have led two or three projects, while type four

is again characterised by inexperienced developers. The same applies to type five. Finally, those who do not care to which community they belong (type 6) are dominated by developers who have led no more than one project.

Figure 46: Perceived Differences between OS- and FS-Communities (Typology) by Number of Led Projects

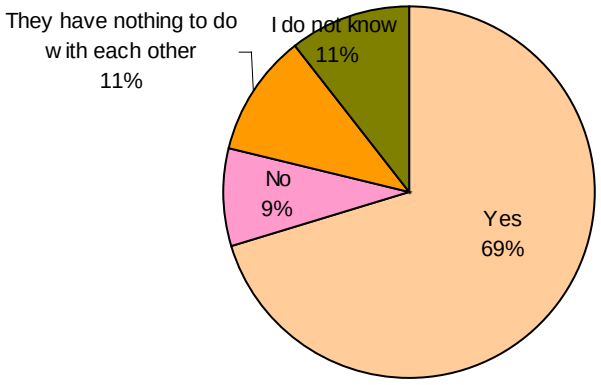


5.2 Open Source / Free Software versus Proprietary Software

After the analysis of differences between Free Software developers and Open Source Software developers, we will continue by considering the two groups as opposites to the development of proprietary software. As we already showed in chapter 2, 52% of the developers in our sample are developing proprietary software as well as Open Source or Free Software. Therefore, it appears reasonable to assume that the dividing line between the two software domains is not very selective. However, the results of the FLOSS developer survey show a clearly polarized perception of the two domains by OS/FS developers, where OS/FS provides undoubtedly "the best of both worlds".

A first closer look on the perception of proprietary software within the community of Open Source / Free Software is provided by the question whether the respondents find that OS/FS satisfies today's requirements for software better than proprietary software. As to be expected in an environment of OS/FS developers, a large majority of 70% endorsed this question, while 9% refused this item. 11% stated that OS/FS and proprietary software have nothing to do with each other, and another 11% did not know how to answer this question (figure 47).

Figure 47: Comparison of OS/FS and Proprietary Software

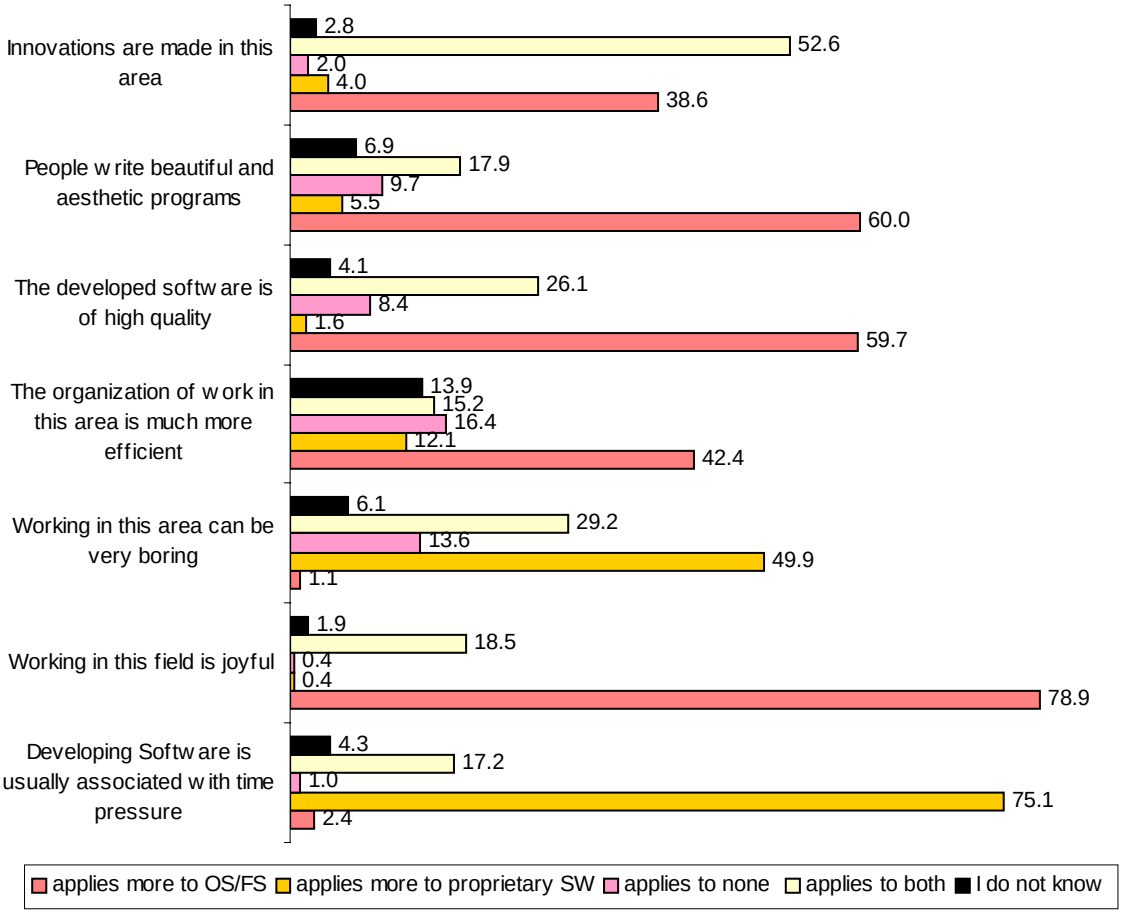


The perception of OS/FS compared to proprietary software has been further examined by asking the respondents to adjoin items providing typical features of software to one of the two domains, to none, or to both of them. Figure 48 illustrates how predominantly OS/FS developers associate positive features with OS/FS and negative features with proprietary software. While proprietary software is associated with time pressure and boring work, OS/FS is associated with joyful work, beautiful and aesthetic programs, high quality software products, and a better work organization.

However, one important exception is provided by the feature of innovativeness, which is clearly adjoined to both of the two domains. Although the picture that is drawn by these results is again very biased, this exception and the fact that some items have relatively often been adjoined to none or to both of the two domains indicate that the responses are rather the result of a weigh-out

process than of pure prejudices. One should bear in mind that more than half of the sample is able to compare OS/FS and proprietary software development directly on a practical level.

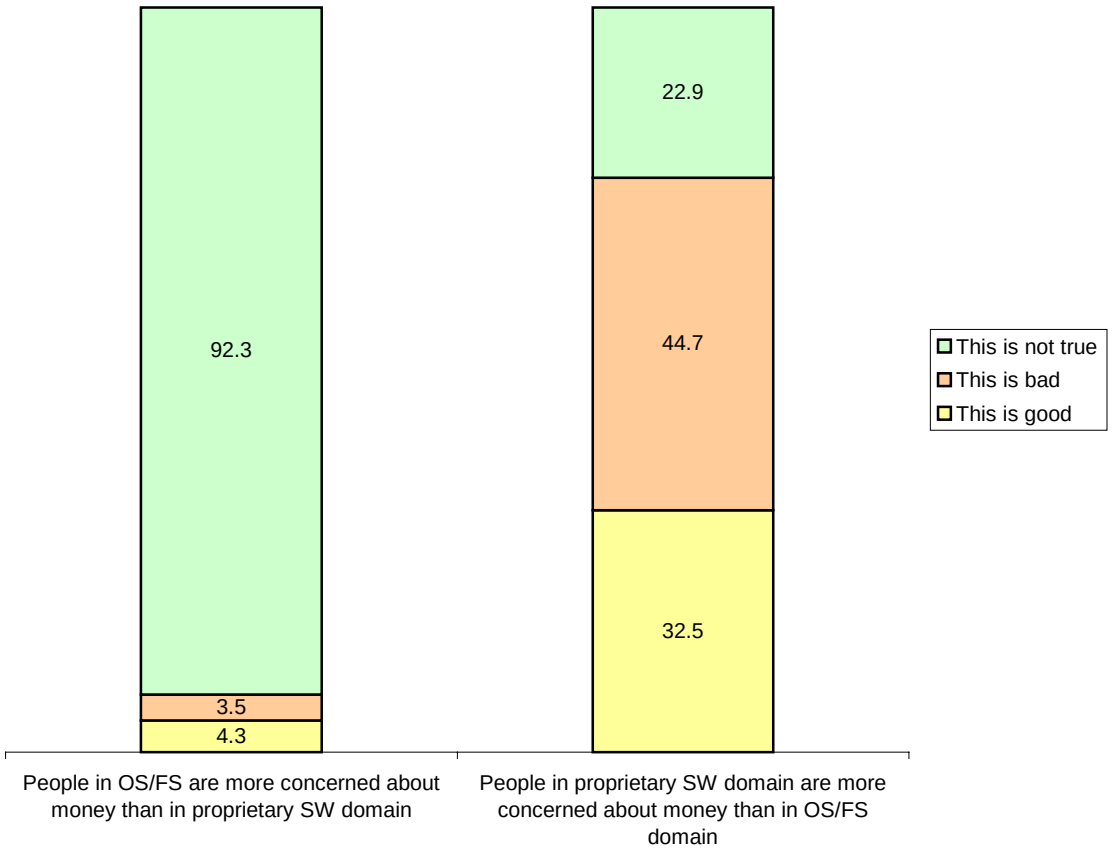
Figure 48: Positive and Negative Features of OS/FS and Proprietary Software



In the public perception, money concerns are usually rather associated with proprietary software than with Open Source / Free Software.[6] Nevertheless, we confronted the software developers with to contradictive statements about money concerns, one saying that people in the domain of proprietary software are more concerned about money than people in the domain of OS/FS, the other stating just the opposite. The respondents were asked to decide whether they find the statements good or bad or whether they hold it for untrue. Again, the result was noticeably biased (figure 49).

[6] This point will be further examined in the next section.

Figure 49: Assessment of Money Concerns in the Domain of Proprietary and in the Domain of Open Source/Free Software



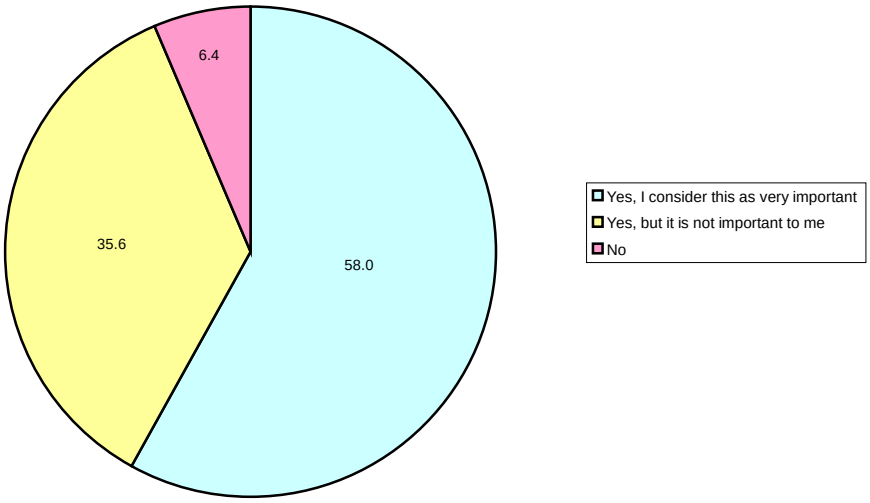
While 92% of the respondents are of the opinion that it is not true that people in the domain of OS/FS are more concerned about money than people in the domain of proprietary software, only 23% state that the opposite statement is true. Moreover, one third of the sample declares that it is good that people in the domain of proprietary software are more concerned about money than people in the realm of OS/FS. Still, 45% say that this is bad.

Money, as a conclusion, plays surely a much greater role in the sphere of proprietary software than in the domain of OS/FS.

Finally, we had to scrutinize the meaning of personal property, which is valued very high in the domain of proprietary software, in the domain of OS/FS. For this purpose, we asked the developers whether they mark their contributions to OS/FS (in form of source code) as theirs or not. The result is clear-cut: 94% of the OS/FS developers mark their contribution to software

projects as theirs (figure 50). Almost three fifths even declare that they consider this as very important. Thus, although differently defined than in the domain of proprietary software, property as a possibility to prove and to claim own efforts does definitely play a fundamental role within the domain of Open Source/Free Software.

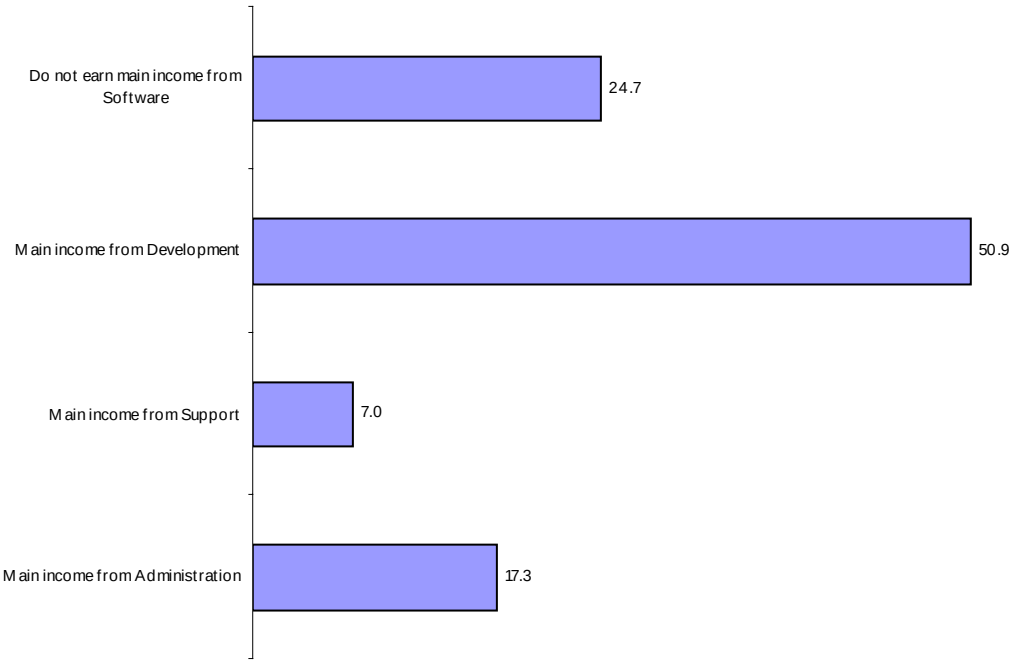
Figure 50: Marking of Source Code in the Domain of OS/FS



5.3 Monetary versus Non-Monetary Rewards

Due to the fact that a large share of OS/FS developers deals with software on a professional basis, it is clear that many OS/FS developers earn their main income from administrating, supporting or developing software. Figure 51 shows that half of the sample earns the main income by software development, while a quarter of the sample does not earn the main income from handling software.

Figure 51: Software as Income Source



However, this result does not tell anything about the role of Open Source and Free Software as a source of income, because there is no distinction between OS/FS and proprietary software. Figure 52 provides an overview about the extent to which the developers of our sample receive direct and indirect payments and non-monetary rewards from developing Open Source / Free Software. Due to the fact that one can achieve different kinds of rewards for contributions to OS/FS, we allowed the respondents to tick more than one answer. Therefore, the percentages in figure 52 add up to a value that is higher than 100.

Almost half of the sample (46%) does not earn money from OS/FS, neither directly nor indirectly. In turn, this means that the majority of the OS/FS developers receives some kind of reward for contributions to OS/FS. Comparing the amount of monetary and non-monetary rewards with regard to the respective shares of developers in the different items, both kinds of rewards seem to have the same importance for the community. Within the scope of directly earned money from OS/FS, administrating plays a more important role than developing OS/FS. Within indirect earnings, to get a job because of expertise in OS/FS issues is observably the most important factor, followed by the development of OS/FS at work.[7]

[7] „Develop OS/FS at work“ is ticked by those who get paid by their employer for developing OS/FS during their usual

Figure 52: Monetary and Non-Monetary Rewards from OS/FS

working time. In contrast, „job description does not include OS/FS development“ means a more indirect and vague way of being paid for OS/FS development at work, i.e. the boss does not know or care or require that the developer deals with OS/FS.

6. Conclusions

Although OS/FS is no new phenomenon, the dynamics of the community have only in recent years accelerated. The OS/FS community is a rather young and predominantly male community with a strong professional background in the IT sector and a high educational level. Despite this professional background, the average OS/FS developer does not reach top incomes.

Most of the developers are singles or only loosely associated with their partners. They feature a high degree of mobility, indicated by the fact that 10% of the sample live and work in a country that does not correspond to their nationality. In this context, the European Union appears as attractive for developers from its member states, but not for developers from the United States of America or other world regions.

The patterns of time spending for developing OS/FS show that this activity still resembles rather a hobby than salaried work. The employment status and the family background of the developers do not have a noticeable impact on the time spending patterns for developing OS/FS.

Besides (software) engineers and programmers, students play also a significant role in the community. However, the often drawn picture of the computer-sticking student, spending hours and hours for developing OS/FS and participating in the community's communications, holds not true. Project performance and leadership is primarily a matter of professionals, although the strong increase of the OS/FS community in recent years has apparently led to the fact that newcomers have been charged with these tasks, too.

In terms of project involvement and leadership, the community appears on the one hand as very active, as less than 10% of the sample have not been involved in a project during the two months the FLOSS survey was conducted. On the other hand, the lion share of the OS/FS developers provides project experience and leadership experience in only few projects. However, we have to take into account that many members of the community have only entered the OS/FS scene after 1997.

Regarding the number of regular contacts to other OS/FS developers, we found that most of the developers feature networks that consist of rather few people. Nevertheless, we found a considerable large group of OS/FS developers that showed regular contacts to more than 50 other developers and that provided undoubtedly the "professional elite" within the community, as

measured by their experience and involvement in project leadership. Overall, the OS/FS developers seem to be well oriented about the leading heads of their community.

Comparing the motives to start with the development of OS/FS and the motives to continue with it, we found an initial motivation for participation in the OS/FS community that rather aims at individual skills and the exchange of information and knowledge with other developers, but over time a maturing of the whole community with regard to both, commercial (material) and political aspects. To learn and to share knowledge have also been the most important issues of OS/FS developers' expectations from other developers.

The own balance of the OS/FS developers of their contributions to and rewards from OS/FS differs considerably from the assumed balance of the other members of the community. They are convinced that they get more out of the community than they give in. In principle, the same applies to the assumed balance of other developers, but they are assessed to invest more and to get less back.

Finally, regarding the main dividing lines we found the sample clearly one-sided with respect to the differences between Open Source/Free Software and proprietary software. With the exception of innovativeness, which is associated with both kinds of software, positive features are generally associated with OS/FS, and negative features with proprietary software. This is in so far astonishing, as more than half of the sample does not only develop OS/FS, but also proprietary software.

The difference between monetary and non-monetary rewards does not play a major role within the OS/FS community. While 45% of the sample do not receive any reward, the remainder receives to a comparable extent both, monetary as well as non-monetary rewards.

The internal differentiation of the community by self-assignments to either the Open Source or to the Free Software community does not provoke a polarization of the community into two different parties. Rather, we found six distinguishable types of orientations in this respect, ranging from those who clearly assign themselves to one of the two domains and claiming fundamental differences between them to those who do not care to which domain they belong.

Overall, we found some ambivalent results while checking the impact of age or of the

professional background on the organisation and functioning of the community, which may be caused by the homogeneity of the community with regard to the young age of the OS/FS developers and their heterogeneity with regard to other features.