Fun and Software Development

Benno Luthiger
ETH Zurich
Zurich, Switzerland
benno.luthiger@id.ethz.ch

Abstract – **The FASD study gathered 1330 answers about fun and software development from open source developers as well as 114 answers from programmers working in commercial software projects. The analysis of these data proves that fun plays an important role when software developers decide to get engaged in an open source project. Moreover, the comparison of the answers gives evidence for the hypothesis that programming in an open source project is significantly more fun compared to the same activity under commercial conditions. The reasons for this fact are that open source projects are able to attract software developers with a credible project vision and that they can offer them an optimal challenge.**

I. INTRODUCTION

The continuing success of open source software has also attracted the interest of the research community. As a result of this academic interest, a number of empirical studies were carried out to account for the open source phenomenon (see for example [1] to [12]). The study about Fun and Software Development (FASD) builds on these existing studies (see [13]).

II. RESEARCH QUESTIONS

The aims of the FASD study are twofold:

First, the study aims to make a quantitative estimation of the importance of fun in order to explain the open source developers' commitment. Second, it aims to verify the hypothesis that open source developers have more fun when programming than developers of commercial software. If this hypothesis can be confirmed, the FASD study aims to identify those elements which are responsible for the fact that open source developers have more fun with their work than commercial developers.

A. Research methods

In order to achieve the aims of this study, a questionnaire for an online survey was developed. There were two versions of this questionnaire, one addressing open source developers, the other addressing programmers working in commercial software firms. The questionnaires consisted of 53 questions. The first part of the questionnaire was identical for both versions. The purpose of this part was to measure the flow software developers experience during their work. I used the flow construct introduced by Csikszentmihalyi (see [14], [15]) to operationalise the fun developers have while programming.

In the following part, I asked the open source developers about their readiness for future activities in open source projects, about how many patches and modules they have developed so far, and how much of their working hours and spare time respectively they spend on developing open source software. With these questions, the criterion variables were established, i.e. they function as a measure of the developers' commitment.

In a further part of the questionnaire, the open source developers were asked about the reasons why they initially joined an open source project or why they started one themselves. In the concluding part of the questionnaire, I gathered demographic data about the respondents and tried to elicit information about the opportunity cost they have when they work for open source projects in their spare time, e.g. by asking them how much spare time and how many hobbies they have.

In the questionnaire for the developers in commercial software firms, I asked them about their willingness to work overtime and about how many checkins they did in the past few days in order to get an impression of their commitment. In a further part of the survey, these developers were asked about their relation to their employer, i.e. how proud they are of their employer and how well they can develop personally and professionally at their workplace. I further asked them how frequently they feel deadlines, about the project visions behind the software projects they work in and about the project managers' formal authority and professional competence. Again, gathering demographic data concluded this questionnaire.

The questionnaire I used was a standardised questionnaire that contained no open questions. About 80% of the questions were formulated as statements; the respondents then indicated their agreement with the statements from "completely unimportant" to "very important" and "is never the case" to "is always the case" respectively on a six point Likert scale.

This study design allows answering the questions formulated in the beginning and to test the hypothesis. I used a simple model which combines the open source developer's commitment as a dependent variable with the explanatory variables consisting of the fun (i.e. flow) and the opportunity cost of time (i.e. the availability of spare time). Using statistical methods like regression and variance analyses, it is possible to test the connection between fun and spare time one the one hand and commitment on the other. The proportion of the variance that can be explained with this model can be used as a measure of the importance of fun for the motivation of open source developers.

By directly comparing the answering behaviour of open source developers and programmers working for commercial firms, it is possible to test the hypothesis that the two groups differ significantly concerning the experience of flow. And by asking the commercial developers about deadlines, project visions and formal authority - all characteristic elements of the commercial software development model which distinguish it from the open source model - I am able to identify the factors responsible for a potential difference between them.

The open source version of the questionnaire was launched on May 3, 2004. I sent a mail to the mailing lists of the various projects hosted by SourceForge, GNU/Sa-

Proceedings of the First International Conference on Open Source Systems
Genova, 11th-15th July 2005
Marco Scotto and Giancarlo Succi (Eds.), p. 273-278

vannah and BerliOS. This questionnaire was open during 53 days until June 25, 2004 and was filled in by 1330 respondents. For the second questionnaire, I found six software companies in Switzerland ready to collaborate with the FASD study. The questionnaire for the developers working for these companies was open from September 20 until November 10, 2004. 114 software developers filled in this version of the questionnaire.

III. RESULTS

A. General results

The study participants come from 74 countries, 19.1% thereof come from the USA, 18.9% from Germany, 5.1% from France and 4.6% from Great Britain. The remaining contributors come from other European countries, from South America, Australia and a few African and Asian countries. Only 1.9% of the contributors are female and only 22.3% have children. Nearly 60% of the respondents are full time employed, only 3.3% declared to be out of work and 28.1% are students. More than half of the contributors are between 20 and 29 years old, almost 28% are between 30 and 39 years old, the youngest is 12 and the oldest is 65 years old. Most of the contributors have only been actively engaged in developing open source software (OSS) for a relatively short period of time. The median is 3 years, the mean 4.8 years. Individual contributors have been active for over 40 years, so that the OSS experience varies to a great extent. The majority of the developers have reached the role of project leader as their highest position (64.6%). This can be explained by the fact that numerous of the existing OSS projects worldwide are one-person projects; therefore, this person must inevitably be the project leader. In contrast to this figure, the proportion of developers (27.2%), bug fixers (3.3%) and persons only registered in mailing lists (3.9%) is relatively small.

B. Use of time for OSS in general

The contributors spend an average of 12.15 hours per week on OSS activities. 7.32 hours thereof are spent during spare time and 4.82 hours during working hours. This statistical analysis corroborates the impression that the commitment for open source projects mainly happens in the spare time. Yet, the share of development of open source projects during working time amounts to considerable 41%.

C. Use of time in dependence on the project role

As expected, the time invested in OSS varies according to the contributor's role within the open source project. Project leaders spend a total of 14.13 hours per week on average on OSS (8.51 hours in spare time), developers devote 11.10 hours (5.87 hours in spare time), bug fixers 5.6 hours (3.6 hours in spare time) and the remaining users about five hours per week (ca. 2.5 hours in spare time). The amount of time committed to OSS therefore increases with the importance of the contributor's role. An F-test confirms a significant difference of the means (significance level $\alpha = 1\%$), though this is only true for the first three categories.

D. Use of time in dependence on time constraints

When analysing the commitment of time for OSS in dependence on the developer's number of hobbies, it turns out that maximal commitment occurs with programmers who have one additional hobby besides OSS. However, the difference between a person having two and a person having three pastimes (including programming) is only weakly significant (significance level $\alpha = 10\%$). When the free time invested in OSS is analysed in dependence on the programmer's degree of employment, an expected tendency is confirmed: The lower the degree of a developer's employment, the more he programs in his free time.

The difference between a fully employed person, who spends 6.5 hours per week on OSS, and a student spending a total of 8.2 hours per week, is statistically significant ($\alpha = 5\%$), as well as the difference between a student and a jobless person, who spends 17.7 hours per week on OSS ($\alpha = 1\%$). However, the difference between people who are employed 100% and people who are employed 70% (9.56 hours per week) is not significant. The latter group only consisted of 15 people, though.

IV. FLOW COMPONENTS

By means of a factor analysis, I was able to investigate the decisive factors which underlie the 28 questions concerning flow experience. Interestingly, this analysis resulted in two different sets of factors for the two versions of the questionnaire. The factor analysis with the answers of open source developers led to the following six factors: *Concentration* (7 items), *clearness of the task* (6 items), *flow/fun* (5 items), *immersion* (4 items), *attention* (2 items) and *reversed wording* (2 items) (see Table I).

The factor analysis of the questionnaires filled in by commercial software developers resulted in the following five factors: *Flow/fun* (6 items), *concentration* (6 items), *immersion* (5 items), *clearness of the task* (4 items) and *challenge* (3 items). The results of the two factor analyses correspond in four of five relevant factors. But whereas challenge is of great importance to commercial developers, it seems that this factor is less important to open source developers.

The reduction of the 28 variables concerning the experience of flow to a few basic factors sets up the basis for further evaluations.

A. Commitment and experience of flow

To what extent does the joy of programming correlate with the commitment for OSS?

In the analysis of the open source developers' commitment in dependence on their experience of flow, some significant correlations can be observed. The amount of time an open source developer is prepared to spend for open source projects correlates highly significant with the experience of flow/fun, immersion, the concentration the developer can devote to the activity, as well as the clearness of the task and the control he can exert on it. The more fun the developer has while programming and the more he is wrapped up in the activity, the more time he spends on open source projects. Table II clearly shows that the experience of flow has its strongest effect on the time spent on OSS during spare time.

Proceedings of the First International Conference on Open Source Systems
Genova, 11th-15th July 2005
Marco Scotto and Giancarlo Succi (Eds.), p. 273-278

TABLE I:
FLOW FACTORS

No	Factor	Loading
Concentration		
5	It's easy for me to concentrate.	59,80%
6	I'm all wrapped up in the action.	57,10%
7	I am absolutely focused on what I'm programming.	76.10%
18	I'm completely focused.	72.90%
20	I am extremely concentrated.	75.80%
27	I completely concentrate on my programming work.	77.10%
28	I am easily distracted by other things.[a]	61.20%
Clearness of the task		
8	The requirements of my work are clear to me.	76.10%
10	I know exactly what is required of me.	80.00%
12	I feel that I can cope well with the demands of the situation.	55.30%
14	I always know exactly what I have to do.	75.80%
17	I know how to set about it.	56.70%
19	I feel able to handle the problem.	56.10%
Flow/Fun		
3	I am in a state of flow when I'm working.	44.90%
21	I'm looking forward to my programming work.	66.80%
22	I enjoy my work.	64.80%
24	Things just seem to fall into place.	45.00%
26	I accomplish my work for its own sake.	58.40%
Immersion		
1	I lose my sense of time.	80.80%
2	I cannot say how long I've been with programming.	79.80%
4	I forget all my worries when I'm working.	41.10%
25	I forget everything around me.	52.00%
Attention		
9	I hardly think of the past or the future.	59.60%
16	I don't have to muse over other things.	77.60%
Reversed wording		
11	There are many things I would prefer doing.[a]	61.70%
15	I'm very absent-minded.[a]	63.70%

[a]Item values reversed for the evaluation.

The effect of the experience of flow is even stronger on the readiness for future activities for open source (see Table III[1]). The correlation of this readiness is highly significant with all flow components. The more fun a programmer has in his activity, the greater is his readiness for future commitment in open source projects.

We can observe a similar pattern with commercial soft-

TABLE II:
TIME SPENT FOR OPENS SOURCE: CORRELATIONS

	Total of time spent	Spare time	Working time
Concentration	0.097**	0.133**	
Clearness of the task	0.143**	0.139**	0.079*
Flow/Fun	0.097**	0.134**	
Immersion	0.127**	0.140**	0.062*

[1]***: significant on 1% level, **: significant on 5% level, *: significant on 10% level

TABLE III:
READINESS FOR FUTURE EFFORT: CORRELATIONS

	Readiness for future effort
Concentration	0.248***
Clearness of the task	0.192***
Flow/Fun	0.467***
Immersion	0.309***
Attention	0.112***

ware developers, too.

V. THE IMPORTANCE OF FUN

To model a connection between the commitment for OSS and the joy experienced while programming, I fell back on the ideas provided by the theory of compensating wage differentials [16]. This theory explains the existence of wage differences between activities that are formally equal or comparable with the existence of additional factors that determine the work place. For example, the theory gives reasons for the wage differences between risky and less dangerous jobs, explaining that employees "pay" for more safety at the work place with lower wages, or, in the reverse case, have to be compensated for their higher risk with adequately higher wages.

Based on such interactions, I developed a model in which I explain the voluntary commitment for open source as a dependent variable by the two independent variables "fun" and "spare time" (1).

According to my model, the engagement for open source increases the more fun the developer has while programming and the more spare time he can make available for this activity. However, the increase is not linear. Both of the independent variables have a quadratic term with a negative sign. Thus, this model is in accordance with the standard economic assumption of decreasing marginal utility. An additional unit of spare time does not lead to the same increase of commitment as the first available hour of spare time does.

With this simple model, I am able to deal with the question about the importance of fun to explain the commitment for open source. With a regression analysis, I can determine how much of the variance of the commitment can be explained by the model. The percentage of the explained variance informs us on the model's validity and, thus, on the importance of fun.

$$E = c + a_1 * F - a_2 * F^2 + b_1 * T - b_2 * T^2 + \varepsilon \quad (1)$$

where
E: voluntary, unpaid engagement
F: fun
T: spare time
$a_1, a_2, b_1, b_2 > 0$
ε: error term

For the linear regression, I use the flow components that I determined by means of the factor analysis as well as the available spare time in hours per week. As a proxy for the commitment, I use the number of hours per week that the developer spends on open source projects during his spare time. I calculated a second regression with the readiness for future engagement as the proxy for OSS commitment.

Dependent variable	Hours per week in spare time for open source
Indepentent variables	**Estimated coefficients**
Concentration	0.624*** (0.071)
Clearness of the task	0.522** (0.064)
Flow/Fun	0.837*** (0.094)
Immersion	0.943*** (0.109)
spare time	0.353*** (1.020)
spare time^2	-0.002** (-0.586)
R^2	0,3

Remark: Standardized beta coefficients are displayed in parenthesis.

The result of the regression calculated with the hours per week spent on open source in the spare time as a dependent variable is displayed in Table IV. In this evaluation, all flow factors except "attention" emerge with linear terms. Only the variable "spare time" shows a quadratic term. This regression explains 30% of the variance of the dependent variable, i.e. the commitment for open source in the spare time.

Table V displays the results of the regression calculated with the readiness for future engagement as a dependent variable. It is remarkable that spare time has no significant contribution in this model. Only the flow factors "flow/fun" and "clearness of the task", the latter with quadratic term, are significant. This model is able to explain 17% of the variance of the dependent variable, i.e. the readiness for future effort.

I get an even better result if, instead of the flow factors, I use the single log transformed questionnaire items as independent variables for the regression analysis. It turns out that the combination able to account for the greatest share of the variance consists of five (log transformed) questionnaire items belonging to the factor "flow/fun", of two items belonging to the factor "concentration" as well as the age of the software developer. Age and one of the concentration factors correlate negatively with the developer's readiness for future commitment. This regression is able to ex-

TABLE V:
COMMITMENT AS FUNCTION OF FUN AND TIME

Dependent variable	Readiness for future effort
Indepentent variables	**Estimated coefficients**
Clearness of the task	2.599*** (0.896)
Clearness of the task^2	-0.202** (-0.824)
Flow/Fun	1.134*** (0.393)
R^2	0,17

plain 27% of the variance of the dependent variable, i.e. the readiness for future engagement.

This analysis allows three conclusions: 1) Fun matters. A simple model containing fun in a broad sense can explain between 27% and 30% of the open source developer's commitment. 2) The availability of spare time does not matter for the open source developer's readiness for future commitment, whereas this fact is of great importance when we examine the developer's actual engagement, i.e. the amount of time spent on open source. 3) The joy of programming does not wear off: each additional unit of fun is transferred linearly into additional commitment.

VI. DEVELOPMENT MODELS IN COMPARISON

Because the questionnaires for both the open source and the commercial developers were identical in the part concerning the experience of flow, it is possible to compare the answers of the open source programmers with those of the software developers in a commercial context.

This comparison (see Fig. 1) indicates significant differences in the components "flow/fun" (significant at $\alpha = 1\%$), "attention" ($\alpha = 5\%$) and "challenge" ($\alpha = 1\%$). The comparison of the mean values of the factor "flow/fun" for open source and commercial developers confirms my hypothesis about fun and software development. Software developers in open source projects have much more fun than their colleagues in commercial contexts. For OSS developers, the mean value of this factor lies more than 10% above the mean value for commercial programmers.

After having asserted this difference, the question arises why the same activity is less fun when it is practised under commercial conditions. Which characteristic of the open source development model is responsible for the fact that software developers in open source projects have more fun? Table VI shows an overview of the relevant characteristics of both software development models.

Open source projects are built on a strong project vision. The project leader knows why he launched the project and why he released it under an open source licence. He has clear ideas about the goals of his project. Moreover, he has good reasons to convey the project goals to the actual and the potential programmers. A convincing project vision is the appropriate means to persuade potential contributors to
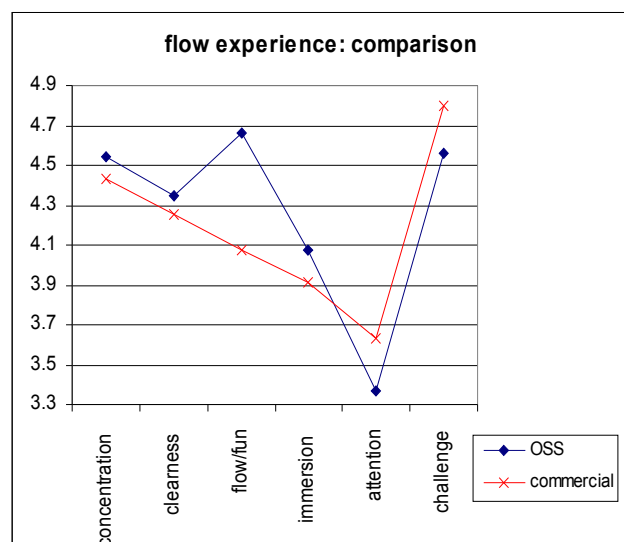
Fig. 1: Comparison of the flow factors

Proceedings of the First International Conference on Open Source Systems
Genova, 11th-15th July 2005
Marco Scotto and Giancarlo Succi (Eds.), p. 273-278

TABLE VI:

CHARACTERISTICAL DIFFERENCES OF SOFTWARE
DEVELOPMENT MODELS

Characteristic	OSS	commercial
project vision	+	?
formal authority	-	+
monetary incentives	-	+
deadlines	-	+
optimal challenge	+	?

get involved with a project, as well as to coordinate the contributions of the current developers and to align them with a common goal. Project visions matter in commercial projects, too. However, it is the formal authority of the hierarchically superior person that is in the first place responsible for the fact that a programmer joins this or that project, and that decides which tasks he actually has to fulfil. Thus, the project vision does not have the same coordination role as in open source projects. In fact, the corresponding data of the FASD study show that the project vision in open source projects has higher statistical significance.

The hierarchical organisation of a firm provides a further distinguishing feature. Hierarchies establish formal authority. The owner of formal authority can ask for a certain kind of behaviour from his employees, something which is impossible for project leaders in open source projects. The latter must try to bring about a behavioural change by means of professional competence and by objective arguments if they are not satisfied with the contributions of the developers participating in their project.

The background of hierarchies and formal authority is formed by the monetary incentives that a firm can establish: Every contribution is paid. This possibility is not available to an open source project that is based on the participants' voluntary cooperation.

Usually, commercial software projects are embedded in a commercial production and exploitation process. The software has to be delivered on a specified date and with a defined functionality. Deadlines are an inevitable part of the commercial production process. On the other hand, each participant in an open source project can avoid a deadline prompted by the project owner without difficulties. Like every other aspect of voluntary cooperation, deadlines are established by agreement. Consequently, the evaluation of the FASD data shows that deadlines play a significantly more important role in commercial software projects.

A final distinguishing feature between the two software development models concerns the challenge for the soft-

TABLE VII:

CORRELATIONS WITH EXPERIENCE OF FLOW

Effect	Correlation coefficient
deadlines	0.256***
project vision: actual	0.397***
project vision: desired	0.189**
formal authority	0.175*
optimal challenge	0.406***

ware developer. In an open source project, the programmer can determine the challenge by the free choice of the project he wants to participate in. Therefore, an optimal challenge can be taken as given, whereas in a commercial context, participation in a specific project is due to the company's circumstances. The programmer's potential regarding his technical knowledge and his experiences as well as his wishes concerning professional and personal development can only be taken into consideration to a limited degree.

In order to find out which of the influencing factors is responsible for the difference in the experience of fun, I correlated the experience of flow of commercial developers with the various factors. A factor that occurs in the commercial development model should correlate negatively with fun and must be significant in order to explain the difference.

Indeed, Table VII shows various significant correlations with the feeling of fun. However, "deadlines" and "formal authority" correlate with the wrong sign. The more deadlines are felt during project work, the more likely the software developers experience flow. On the other hand, the significant correlations with project vision and optimal challenge occur with the correct sign.

Therefore, this analysis permits the following conclusions: Software development is significantly more fun when it takes place in the context of an open source project than when it is carried out under commercial conditions. However, the reasons for this difference are not the deadlines, but the differences concerning the project vision and the optimal challenge.

It is comprehensible that the project situation of a commercial project offers fewer possibilities to software developers to contribute their potential in an optimal way to the project. It is also understandable if the responsible persons in commercial software firms do without project visions that can be understood by the programmers. However, this is not imperative. It is at most expensive to formulate a project vision and to find an optimal challenge for the software developers participating in the project, but it is not impossible. For this reason, one can conclude from my analysis, it is not impossible either that a software project in a commercial context can be as much fun as a project occurring under open source conditions.

VII. REFERENCES

[1] A. Bonaccorsi and C. Rossi, "Altruistic individuals, selfish firms? The structure of motivation in Open Source software", *http://opensource.mit.edu/papers/bnaccorsirossimotivationshort.pdf,* 2003

[2] B. Dempsey, D. Weiss, P. Jones and J. Greenberg, "Who is an Open Source Software Developer?" *Communications of the ACM,* vol. 45, no. 2, 2002, pp. 67-73.

[3] R.A. Ghosh, R. Glott, B. Krieger and G. Robles, "FLOSS: Free/Libre and Open Source Software: Survey and Study", *http://www.infonomics.nl/FLOSS/report/,* 2002

[4] A. Hars and S. Ou, "Working for Free? - Motivations of Participating in Open Source Projects." in *34th Annual Hawaii International Conference on System Sciences*, 2001

Proceedings of the First International Conference on Open Source Systems
Genova, 11th-15th July 2005
Marco Scotto and Giancarlo Succi (Eds.), p. 273-278

[5] K. Healy and A. Schussman, "The Ecology of Open Source Software Development", *http://opensource.mit.edu/papers/healyschussman.pdf* ,2003

[6] A. Hemetsberger, "When Consumers Produce on the Internet:. The Relationship between Cognitive-affective, Socially based, and Behavioral Involvement of Prosumers" *http://opensource.mit.edu/papers/hemetsberger1.pdf*, 2003

[7] G. Hertel, S. Niedner and S. Herrmann, "Motivation of Software Developers in Open Source Projects", *Research Policy,* vol. 32, no. 7, 2003, pp. 1159-1177.

[8] N. Jorgensen, "Putting it All in the Trunk", *Information Systems Journal,* vol. 11, no. 4, 2001.

[9] S. Krishnamurthy, "Cave or Community? An Empirical Examination of 100 Mature Open Source Projects", *http://www.firstmonday.org/issues/issue7_6/krishnam urthy/index.html*, 2002.

[10] K. Lakhani and R. Wolf, "Why Hackers Do What They Do: Understanding Motivation Effort in Free/Open Source Software Projects", *http://opensource.mit.edu/papers/lakhaniwolf.pdf*, 2003

[11] G. Robles, H. Scheider, I. Tretkowski and N. Weber, "Who is doing it? A research on Libre Software developers", *http://widi.berlios.de/paper/study.html*, 2001.

[12] S. Shah, "Understanding the Nature of Participation & Coordination in Open and Gated Source Software Development Communities", *http://opensource.mit.edu/papers/shah3.pdf*, 2003

[13] B. Luthiger, "Alles aus Spass?", in *Open Source Jahrbuch 2004*. Eds. R. A. Gehring and B. Lutterbeck, Berlin: 2004.

[14] M. Csikszentmihalyi, *Beyond boredom and anxiety*, San Francisco: 1975.

[15] M. Csikszentmihalyi, *Flow. The Psychology of Optimal Experience*, New York: 1990.

[16] U. Backes-Gellner, E. P. Lazear and B. Wolff, *Personalökonomik. Fortgeschrittene Anwendungen für das Management*, Stuttgart: 2001.

Proceedings of the First International Conference on Open Source Systems
Genova, 11th-15th July 2005
Marco Scotto and Giancarlo Succi (Eds.), p. 273-278