License Risks from Ad-Hoc Reuse of Code from the Internet: An Empirical Investigation

April 2011

Manuel Sojer[1] and Joachim Henkel[1,2]

[1]Technische Universität München, Schöller Chair in Technology and Innovation Management, Arcisstr. 21, D-80333 Munich, Germany. sojer|henkel@wi.tum.de

[2]Center for Economic Policy Research (CEPR), London

Abstract

Reusing code that is downloadable from the Internet—particularly open source software (OSS) code—in commercial software development is attractive for both firms and their software developers. However, to avoid serious economic and legal consequences for firms, the license obligations of the reused code have to be met. While this risk seems to be manageable in systematic reuse, colloquial evidence suggests that when reusing Internet code in ad-hoc fashion, individual professional software developers sometimes do not treat license obligations properly. Quantitatively investigating this issue, we explore the ad-hoc Internet code reuse of professional software developers with a particular focus on license issues by analyzing a unique global dataset of 869 professional software developers. We find that ad-hoc Internet code reuse has become prevalent in commercial software development. Despite this, when reusing Internet code in ad-hoc fashion, professional software developers appear not to fully account for license issues potentially resulting from their behavior. Moreover, our results point out that professional software developers receive little effective training and information on the topic of Internet code reuse from official channels. Furthermore, professional software developers are on average not fully aware of many common Internet code license obligations, and tend to overestimate their own knowledge. Most firms also do not provide close guardrails to their software developers regarding Internet code reuse through policies. Consequently, a considerable share of professional software developers has violated Internet code license obligations in the past. Based on our findings we discuss practical implications for firms developing software and suggest levers to reduce the economic and legal risks from license violations through professional software developers' ad-hoc reuse of Internet code.

We are grateful to many people who shared their insights or commented on earlier drafts of the paper. Special thanks go to Oliver Alexy, Martin Bichler, Timo Fischer, and Dirk Riehle.

Introduction

Reusing existing software artifacts when developing new software is an attractive lever to reduce development costs, shorten time-to-market, and increase software quality [4]. The artifact most commonly reused in software development is code [16].

Recently, researchers have pointed to the reuse of "Internet code" in commercial software development as a new facet of software reuse [13, 22]. By this term we denote code in the form of components (e.g. a library encapsulating required functionality) or snippets (e.g. containing a synchronization block) that can be downloaded from the Internet for free and without individual agreement with the originator. An important instance of Internet code is publicly available open source software (OSS). Internet code generally comes with the permission to be reused in commercial software development [14], which makes it highly attractive for firms [2]. Therefore, some firms have begun to *systematically* reuse Internet code by including the steps of identification, evaluation, and integration of suitable code into their development processes [e.g. 18]. Alternatively, Internet code can also be reused in *ad-hoc* fashion [e.g. 23]. In this form, individual professional software developers—on their own and typically without telling anybody else—search the Internet for existing code as a short-cut in their work, download this code and integrate it into the software they are currently working on.[1]

Despite its general suitability for reuse in commercial software, Internet code is rarely in the public domain. Usually it comes under licenses which demand compliance with specific conditions as a prerequisite of reuse [8]. These conditions vary widely and may for example demand attribution of the original creators of the reused code. More critical for firms are the obligations demanded by the GNU General Public License (GPL)[2] as the most common license [11]. The GPL is an OSS license and requests that other code tightly integrated with code governed by it also be licensed under its terms [9]. These terms require that users of GPL licensed software may access, modify, and redistribute the source code of the software [19]. For firms trying to protect their source code as intellectual property this obligation may be difficult to comply with. However, if a firm integrates code under the GPL into its software without adhering to the license terms and is then found out, it can be legally forced to either replace the GPL'ed

---

[1] Typical places to search for such code are OSS repositories (e.g. SourceForge.net) , code search engines (e.g. Koders.com) or code bases of related OSS projects. See [20] for a detailed overview and quantitative analyses.

[2] More precisely, the GPL is a family of licenses, with versions 1, 2, and 3. Since all versions share the characteristic that is most relevant in our context, we refer in the following for simplicity to "the GPL."

code, or license the entire program under the GPL. Either option can have serious legal and economic consequences [19].

Other license conditions that can turn out to be problematic for firms are obligations like reusing the code only in non-commercial settings or in certain application types, employing it only for a certain period of time, or not exporting it to certain geographies [17].[3] Finally, some code available from the Internet does neither explicitly state a license nor reuse obligations, still—since it is nonetheless protected by copyright—proper reuse handling necessitates contacting the creator and asking for permission to reuse.

It seems feasible to weigh the benefits and risks of Internet code reuse and manage potential license issues when Internet code is reused systematically. Yet, on the side of ad-hoc Internet code reuse colloquial evidence suggests that individual professional software developers sometimes do not treat the license obligations of the code they reuse properly [12, 15]. Thus, while their ad-hoc Internet code reuse might still result in effectiveness, efficiency, and quality benefits for their firms, their behavior might also lead to legal and economic trouble for their employer.

Most existing research addressing Internet code reuse has largely been theoretical or based on industrial case studies. As an exception, German and co-authors [6-9] quantitatively investigate license issues from OSS code reuse through the analysis of code bases and software distributions.

Complementing this work by taking the perspective of individual professional software developers, our study employs quantitative data obtained from a global survey among 869 professional software developers to scrutinize ad-hoc Internet code reuse with a particular focus on license issues. Our findings should provide firms with starting points to assess their exposure to license risks from their developers' ad-hoc Internet code reuse and to devise measures to avoid potential issues.

Survey

The questionnaire employed to gather our data was developed after a literature review and 20 interviews with industry experts.[4] Before conducting the survey, the questionnaire was pre-tested by four academic peers and 113 software developers. We chose a survey-based

[3] Such restrictions are however not contained in OSS licenses.

[4] full questionnaire employed has been provided in the Appendix 1.

research approach over an analysis measuring the share of reused Internet code in commercial software code bases. While this setup does not allow us to report precise percentages of Internet code reuse in commercial software development, it allows us to include a greater number of professional software developers. Moreover, if deviations between developers' actual and their survey-reported reuse should arise, they are unlikely to be systematic and thus should not affect the results of our multivariate analyses.

Since our study is among the first to investigate ad-hoc Internet code reuse on the level of individual professional software developers, we opted not to choose a limited sample of developers from only one firm, but rather selected the broad and heterogeneous group of professional software developers active in Internet newsgroups as our survey population.[5] During fall 2009 we extracted a total of 93,541 unique email addresses from more than one million messages posted throughout the previous three years in a total of 528 newsgroups dealing with software development.[6] After cleaning these addresses, we selected a random sample of 14,000 addresses from our list and via email invited these newsgroup participants to take our survey. We received 1,133 fully filled-in responses, yielding a response rate of 9.9% which is consistent with other Internet surveys.[7] Of the 1,133 responses, 869 had been submitted by current or former professional software developers who are the focus of the following analyses.

The vast majority of the professional software developers surveyed (98%) is male with an average age of 35.6 years. They live in Europe (53%), North America (28%), Asia (12%), and South America (4%). 56% of them have contributed to OSS in the past.

79% of the developers currently work as professional software developers. The others have done so in the past but quit.[8] On average the developers have 9.7 years of work experience as professional software developers. Most of them are employed as programmers (51%), software

---

[5] Potential limitations of this approach are discussed in the "threats to validity" section at the end of the paper.

[6] These 528 newsgroups included all main and high traffic groups such as e.g. comp.lang.c++ or comp.lang.java.programmer. A more detailed overview of these newsgroups and the sampling process has been provided in the Appendix 2.

[7] To calculate the response rate we adjusted the number of invitations sent by the number of emails that did not reach their designated recipients.

[8] For those developers who have quit creating software in the past, in the following the characteristics of their last software development activities are reported.

architects (28%), or project managers (4%). 23% work as freelancers, the others are permanently employed.

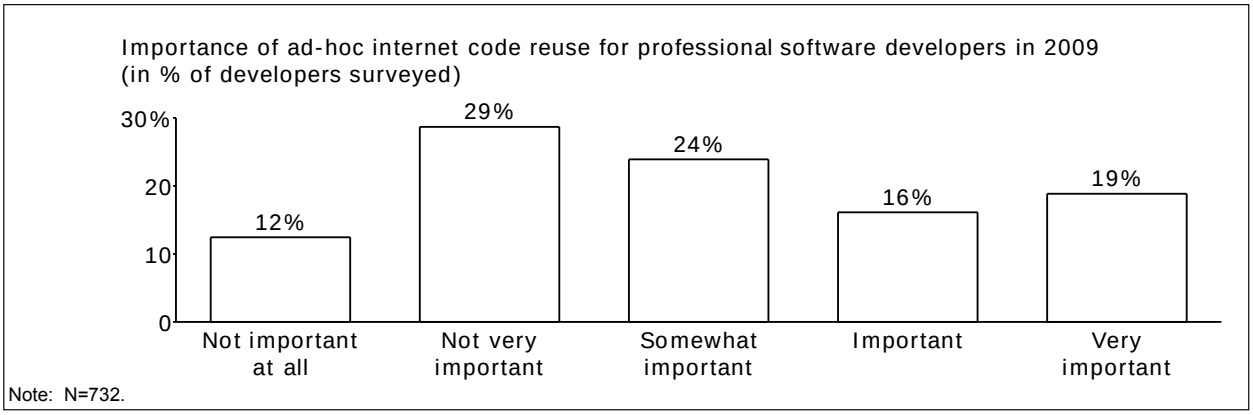
54% of the developers work for firms for which software development is their main business; 68% develop software for external customers of their firms, while the others create internal-use software. Of those developers writing software for external customers, 62% create off-the-shelve software for multiple customers while the others develop custom-built software. These distinctions are important because the license risks resulting from Internet code reuse are typically more severe for software developed for multiple external customers.

Extent of ad-hoc Internet code reuse

To quantitatively assess the extent of ad-hoc Internet code reuse in commercial software development, survey participants were asked to indicate how important reusing Internet code (components and snippets) in ad-hoc fashion is for their individual work.

Figure 1, depicting the perceptions of those professional software developers still active in 2009, points out that ad-hoc Internet code reuse is an essential part of their work for many professional software developers today. More than half of the developers (59%) considers ad-hoc Internet code reuse at least as "somewhat important" for their work. Only 12% apparently do not reuse any Internet code in ad-hoc fashion. This finding is in contrast to the prevailing assumption of many firms that their code base does not or only to a small and controlled degree contain Internet code [15].

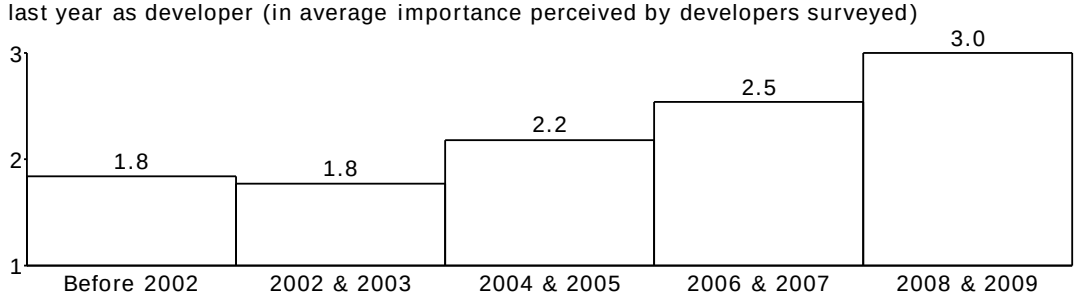
Figure 1. Extent of ad-hoc Internet code reuse in 2009



In addition to analyzing the extent of ad-hoc Internet code reuse in 2009, we also investigate its historic development. The results presented in Figure 2 also show perceptions of professional software developers who quit creating software before 2009. Since we asked these participants about their last year as active developers, their responses are informative about the

respective year. The data show that starting with 2004 the importance of ad-hoc Internet code reuse for professional software developers' work has increased, rising from a mean importance value of 1.8 ("not very important") in 2002 and 2003 to 3.0 ("somewhat important") in 2008 and 2009.

Figure 2. Evolution of the extent of ad-hoc Internet code reuse



	Before 2002	2002 & 2003	2004 & 2005	2006 & 2007	2008 & 2009
S.D.	1.2	1.3	1.3	1.2	1.3
Number of developers in class	32	13	17	28	779

Notes: Average values displayed for multi-year groups; S.D. = standard deviation; importance scale: 1=Not important at all, 2=Not very important, 3=Somewhat important, 4=Important, 5=Very important; N=869.

A possible interpretation of this finding is that before 2004, code available from the Internet might have been suited for reuse in commercial software development only rarely because it was not mature enough and did only cover a few functional areas. However, resulting from the strong recent growth of OSS [3], both the quality and the fields for which there exists code should have increased strongly which made Internet code reuse much more attractive.

Individual-level determinants of ad-hoc Internet code reuse

To understand which factors influence the importance professional software developers attribute to ad-hoc Internet code reuse we conducted an exploratory regression analysis. The model (see Table 1) employs an ordered logistic regression [10] and uses the perceived importance of ad-hoc Internet code reuse for the individual work of professional software developers measured on a five-point scale as dependent variable. As independent variables multiple characteristics of professional software developers have been included, some as dummy variables. Regression coefficients are not standardized, such that the range or the standard deviation of a variable must be taken into account when assessing the variable's effect size.

First, the model results point out that developers' ad-hoc reuse of Internet code seems to be independent of the "license risk level".[9] That is, developers creating software to be sold to multiple external customers do not deem ad-hoc Internet code reuse less important than developers working on custom-built software or software for firm-internal use. A possible interpretation of this is that developers, in their decision to reuse Internet code in ad-hoc fashion, do not take into account the likelihood of negative legal and economic consequences their employer would face in the case of license violations. However, there might also be alternative explanations: First, one could assume that there exists less reusable code for internal-use or custom-built software due to its tailored nature. Second, one could imagine that while not considering ad-hoc Internet code reuse less important, professional software developers still are more careful when reusing Internet code in software development projects for multiple external customers.

Also pointing out risks, developers who have never received any form of training or information on Internet code reuse and thus should be more likely to create license issues do not differ significantly[10] in their perceived importance of ad-hoc Internet code reuse from developers who have been trained or have received information. Adding to that, while developers who *self-assess* their knowledge about Internet code licenses better also deem ad-hoc Internet code reuse more important, this relationship does not hold for an *objective* assessment of developers' proficiency regarding Internet code licenses.[11] If we—plausibly—assume that the results of our objective assessment are more informative about developers' license-related knowledge level than their self-assessment, this finding implies that developers on average do not correctly account for their own knowledge about Internet code licenses or lack thereof when considering the ad-hoc reuse of Internet code.

---

[9] We set "license risk level" to 1 if the developer is working on internal-use projects, to 2 if he/she is working on external projects for only one customer, and to 3 if he/she is working on projects for multiple external customers.

[10] Throughout the paper, we use the term "significant" exclusively in the sense of "statistically significant."

[11] This objective assessment of developers' knowledge about Internet code reuse obligations is based on a quiz contained in our survey. The quiz was developed after our 20 interviews with Internet code reuse industry experts. It covers five typical situations in which professional software developers may violate license obligations when reusing Internet code. Developers received one point for each correct answer to the five quiz questions. The actual quiz questions and descriptive statistics of the results are provided in Appendix 4. One might conjecture that the insignificance of the objectively assessed knowledge is caused by the fact that it is correlated with the self-assessed knowledge. However, this insignificance persists when the self-assessed knowledge level is dropped from the list of explanatory variables.

In addition, the model results indicate that developers who have been active in OSS projects and those with longer experience as professional software developers consider ad-hoc Internet code reuse significantly more important.[12] A plausible interpretation of these findings, consistent with Sojer and Henkel [21], is that for OSS-savvy developers the costs of searching, evaluating, and understanding Internet code should be lower. Similarly, more senior developers should face lower costs of reusing due to a larger personal network and since they can turn to their own reuse experiences.

Furthermore, the multivariate model substantiates the result of Figure 2, showing that the perceived importance of ad-hoc Internet code reuse has grown significantly over time in the past.

Beyond that, developers with different software development roles perceive different levels of importance of ad-hoc Internet code reuse. Programmers and database developers attribute a significantly lower importance to it than the reference group, architects. For all other roles, the difference to "architects" is insignificant on a 10% level. The finding that architects deem ad-hoc Internet code reuse significantly more important then programmers is startling since architects should be concerned with systematic rather than ad-hoc Internet code reuse. However, especially in smaller and medium-sized firms architects might also take over programmer responsibilities and leverage their greater architectural latitude to reuse Internet code in ad-hoc fashion. The architecture of a piece of software influences how easy it is to reuse external code [5]. Since it is architects who shape the architecture, they might have more control over reusing Internet code than programmers for whom the architecture of the software they develop is often exogenous. In addition to that, greater architectural latitude may also allow developers to integrate Internet code in such a way that license violations can be avoided [9]—assuming that developers are aware of the relevant issues. Supporting this line of thought, we find that architects are significantly more knowledgeable regarding licensing topics than other developers such as programmers. Thus, architects might still be able to reuse Internet code properly while programmers would only have the choice between reusing the code in a way violating its obligations and not reusing it at all.

[12] Note that, as mentioned above, regression coefficients are not standardized. Since "experience as professional software developer" is measured in years and ranges from 0.5 to 45, its effect is comparable in size to that of the dummy variable "developer has OSS experience." While the coefficient of the latter variable is much larger (0.39 vs. 0.017), its range is much smaller (1 vs. 44.5).

Table 1. Multivariate analysis of the importance of ad-hoc Internet code reuse[13]

	Ordered Logistic Regression	
	Coef.	Std. Err.
License risk level of developer's work	0.111	0.085
Developer has never received any form of training or information on Internet code reuse (dummy)	-0.258	0.167
Developer's self-assessed knowledge about Internet code licenses	**0.442*** **	0.099
Developer's objectively assessed knowledge about Internet code licenses	-0.032	0.057
Developer has OSS experience (dummy)	**0.391*** **	0.143
Experience as professional software developer (in years)	**0.017***	0.009
Last year as professional software developer	**0.197*** **	0.043
Software development role (dummies, reference group: architect)		
Project manager	0.155	0.358
Programmer	**-0.356** **	0.149
Analyst	-0.943	0.969
Tester	-1.176	0.717
Database developer	**-0.751** **	0.350
Other	-0.281	0.241
Primary programming language (dummies, reference group: Ruby)		
Python	-0.284	0.276
Perl	**-0.861** **	0.435
Java	**-1.015*** **	0.268
PHP	**-1.533*** **	0.381
C	**-1.550*** **	0.333
C++	**-1.808*** **	0.269
Visual Basic	**-2.001*** **	0.516
C#	**-1.957*** **	0.315
Other	**-1.842*** **	0.258
Developer lives in … (dummies, reference group: Europe)		
… North America	0.016	0.164
… South America	**0.727** **	0.337
…Asia or rest of world	-0.206	0.210
Developer is working as a freelancer (dummy)	0.041	0.163
Education (dummies, reference group: engineering)		
Computer science or related subject	-0.223	0.158
Mathematics or physics	-0.300	0.251
Business administration	-0.222	0.421
Other subject	0.147	0.258
Developer is working on embedded software projects (dummy)	-0.159	0.185
Developers' self-assessed software development skill level	-0.048	0.087
Observations	807	
Pseudo R^2	0.09	
Wald test	$X^2(32)=208.94$, $p<0.0001$	
Cuts	393.318, 395.001, 396.161, 397.153	

* significant at 10%, ** significant at 5%, *** significant at 1%

Notes: Significant coefficients are bolded; reported standard errors are robust standard errors.

Also the main programming language developers use influences the importance of ad-hoc Internet code reuse for their work. Developers relying mainly on Ruby or Python find ad-hoc Internet code reuse most important. These developers are followed by a group working with

[13] Descriptive statistics and the correlation table of the explanatory variables used are provided in Appendix 3.

languages such as Perl, Java or PHP. Developers using more traditional programming languages such as C or C++, less common ones such as Visual Basic or C#, and various others form the last group.

While one could conjecture that diverse legal systems (e.g. common law vs. civil law), cultural variations, or different availability levels of Internet code in local language lead to different perceptions of the importance of ad-hoc Internet code reuse in different geographies, we do not find substantial support for such reasoning: North American, Asian, and European developers do not differ significantly in their perceived importance of ad-hoc Internet code reuse; only South American developers deem ad-hoc Internet code reuse significantly more important. Yet, since only 33 developers from this region have participated in our survey, this finding may very well not be representative.

Finally, we do not find significant differences in professional software developers' perceived importance of ad-hoc Internet code reuse based on their education, their software development skills, whether they develop embedded or traditional software, and whether they are employed in time-limited contracts (e.g. freelancers) or as permanent employees.

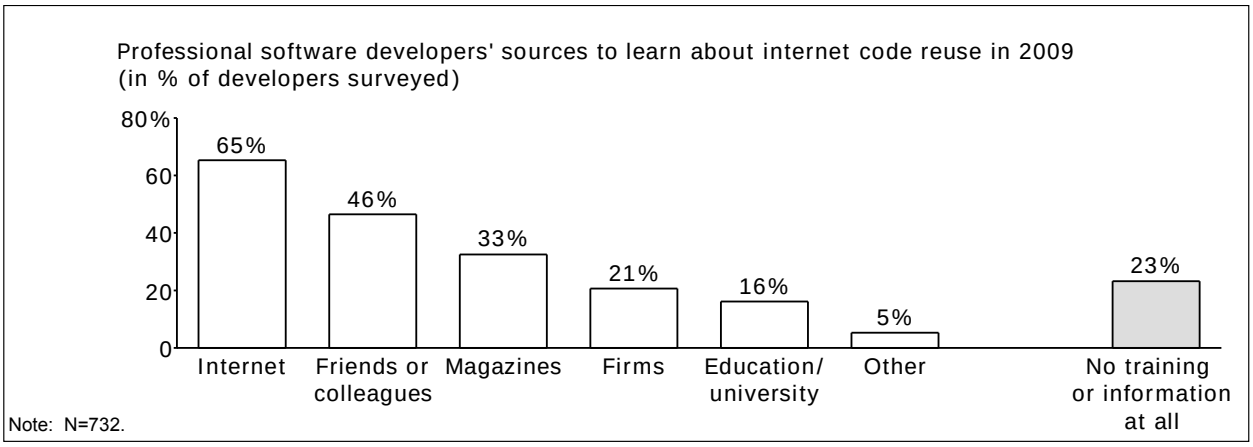
Developers' knowledge about Internet code reuse and resulting risks for firms

We now turn to the question of how well professional software developers are prepared to deal with the licenses and obligations coming with ad-hoc Internet code reuse.

It seems reasonable to assume that professional software developers who are more aware of the particularities of Internet code such as its licenses will also account for license obligations more properly. Thus, our first analysis investigates if professional software developers have received training or information on Internet code reuse and from which sources (see Figure 3).

With the Internet (65%) and friends and colleagues (46%), two rather informal channels are developers' main sources of information. Comparatively unimportant are firms (21%) and institutions of education such as universities (16%). 23% of the surveyed developers have never received any form of training or information on Internet code reuse. We interpret these findings as indications that conveying knowledge about Internet code reuse is not high on the agenda of firms and universities. Moreover, the fact that nearly a third of our respondents has received no training or information at all strikes us as surprising.

Figure 3. Software developers' sources to learn about Internet code reuse in 2009



Given the high number of developers who have either never received any training or information on Internet code reuse or who rely on information from non-official channels, it is interesting to assess developers' knowledge about Internet code licenses directly. When self-assessing their knowledge, two thirds of the developers claim that they are "familiar" or even "very familiar" with nearly all obligations from Internet code licenses and can deal with them well (see Table 2). Contrasting this self-assessment with the results of a five-question quiz about license obligations resulting from Internet code reuse (described in footnote 11) suggests that developers overestimate their knowledge. Even those developers considering themselves "very familiar" with pertaining license obligations on average failed on two questions in the quiz, obtaining a mean score of 3.11 out of a maximum of 5 (see Table 2).[14] Moreover, while positive and statistically significant ($p<0.001$) the correlation between self-assessment and quiz score is weak, with a correlation coefficient of 0.345.

Table 2. Software developers' familiarity with Internet code license obligations in 2009

	Not familiar at all	Not very familiar	Somewhat familiar	Familiar	Very familiar
Share of developers self-assessing their familiarity with Internet code license obligations in the respective groups	2%	3%	29%	50%	16%
Developers' average score in quiz on license obligations from Internet code reuse (max. score attainable: 5, average score across all groups: 2.54)	0.88	1.50	2.08	2.74	3.11

Note: N=732.

[14] We pre-tested our quiz questions to make sure, as good as possible, that they are of comparable difficulty and relevance. Still, there will be some variation between them, and it is possible that respondents who described themselves as "very familiar" with Internet code license obligations (and who failed on average in 1.89 questions), tended to fail on those license issues that appear less frequently and are thus less critical for firms.

Going one level deeper, we investigate which factors influence developers' objectively assessed knowledge about Internet code licenses and their obligations. The exploratory Tobit [10] regression model (see Table 3) employed uses developers' score in the license quiz as the dependent variable. The results point out that developers with OSS experience are significantly more knowledgeable about Internet code licenses than other developers. Furthermore, most forms of training and information about Internet code reuse (from firms, friends or colleagues, magazines, and other sources) do not exhibit a significant influence on developers' knowledge. Developers who have received training or information in institutions of education are even significantly less proficient than other developers. Only information acquired from the Internet has a significant positive effect on developers' knowledge.

Besides these factors, developers from North America or Asia seem to know less about Internet code licenses than their European or South American counterparts. Regarding educational backgrounds, developers with degrees in computer science and related subjects or engineering are more proficient regarding Internet code licenses than other developers.

In the situation described above where ad-hoc Internet code reuse seems to be prevalent but also opening up license risks it would seem reasonable for firms to introduce explicit policies providing close guardrails to developers considering to reuse Internet code.

Despite this, only about one third of the developers surveyed works in firms with such policies. More detailed analysis of this matter points out that large firms with more than 5,000 employees are 31% more likely to have such policies while there is no significant difference between smaller firms of various sizes.[15] Furthermore, firms for which software development is their main business have a 19% higher probability of having such policies while there is no consistently significant effect of firm age.

[15] These findings result from exploratory logistic regression analyses and resulting marginal effects not depicted here due to space constraints. The full regression tables are provided in Appendix 6.

Table 3. Multivariate analysis of developers' knowledge about Internet code licenses[16]

	Tobit Regression	
	Coef.	Std. Err.
Developer has OSS experience (dummy)	**0.835*** **	0.098
Developer has received training or information on Internet code from… (dummies)		
… firms	0.124	0.120
… institutions of education	**-0.243***	0.126
… friends or colleagues	0.080	0.112
… the Internet	**0.390*** **	0.122
… magazines	0.089	0.112
… other sources	-0.091	0.213
Developer lives in … (dummies, reference group: Europe)		
… North America	**-0.238****	0.117
… South America	-0.119	0.222
… Asia or rest of world	**-0.297****	0.142
Education (dummies, reference group: computer science of related subject)		
Engineering	0.073	0.124
Mathematics or physics	**-0.320***	0.170
Business administration	**-0.751****	0.354
Other subject	**-0.385****	0.184
Experience as professional software developer (in years)	0.002	0.007
Constant	**1.890*** **	0.141
Observations	869	
Pseudo R^2	0.04	
F test	F(15, 854)=8.62, p<0.0001	
Σ	1.376	

* significant at 10%, ** significant at 5%, *** significant at 1%

Notes: Significant coefficients are bolded; reported standard errors are robust standard errors.

Of the developers working in firms with policies nearly a quarter claims not to have read these policies. Programmers are less likely to have read policies than architects; also, developers unhappy with their job are significantly less likely to read the policies of their employers.[17] Additionally, developers who are not involved in software development projects for multiple external customers are significantly less likely to have read the policies.

As a consequence of the ad-hoc Internet code reuse situation detailed, it is not surprising to find that 21% of the developers creating software in 2009 have at least once not checked thoroughly for Internet code license obligations when reusing snippets. 16% have done the same when reusing components and 14% have even ignored license obligations they were aware of when reusing snippets.

[16] Descriptive statistics and the correlation matrix of the explanatory variables used are provided in Appendix 5.

[17] These findings result from exploratory logistic regression analyses and resulting marginal effects not depicted here due to space constraints. The full regression tables are provided in Appendix 7.

Possible threats to validity and limitations

Given the multiple variables in our regression models, the size of our sample, and significance levels reported, our results should possess statistical validity. However, we are aware of threats to internal, construct and external validity of our work which should be addressed in future research.

In terms of internal validity the explanatory and control variables in our models should make sure that no omitted variable biases influence our results. However, since our questionnaires were completed in anonymous fashion by software developers identified based on their email addresses we can not be sure of the accuracy and truthfulness with which our questions were answered.

Regarding construct validity, the main dependent variable of our research is the perceived importance of ad-hoc Internet code reuse for developers' individual work. While we belief that this variable is a suitable proxy for the extent to which professional software developers practice ad-hoc Internet code reuse, future research might want to employ more direct measures to check robustness of our findings. Moreover, despite our extensive pre-test with more than 100 developers it might be possible that some survey participants misunderstood the meaning of some survey questions.

Addressing external validity, there is the threat that our survey population of professional software developers active in Internet newsgroups is not representative of professional software developers in general. Since this research is among the first to quantitatively investigate ad-hoc Internet code reuse on the level of individual developers, we consciously chose software developers from newsgroups to ensure broad heterogeneity in our sample. Moreover, the comparison of the demographics of our sample with that of other recent studies among professional software developers [e.g. 1] gives us confidence in the representativeness of our sample. Still it seems worthwhile to repeat our study in a more homogeneous single-firm setting.

Conclusion

Analyzing ad-hoc Internet code reuse of components and snippets in commercial software development, we find that its importance has increased over time, and that today more than half of all professional software developers surveyed deem ad-hoc Internet code reuse at least "somewhat important" for their own work. This result is in contrast to the prevailing assumption of many firms that their code base does not or only to a small and controlled degree contain Internet code [15].

Addressing professional software developers' knowledge about Internet code licenses and their obligations we find that about a quarter of the developers has never received any form of training or information on this topic. Moreover, only a small fraction has received training or information from firms or institutions of education. Furthermore, many existing forms of training and information seem not to be effective.

As a consequence of this, many developers lack detailed knowledge about obligations potentially resulting from Internet code reuse. Only a minority of firms has deployed policies addressing Internet code reuse. Consequently, a considerable share of developers—14% to 21% of our sample, depending on the scenario—has at some point either not checked thoroughly for license obligations or even knowingly ignored them when ad-hoc reusing Internet code in the past.

As practical implications of our results firms need to acknowledge the existence of Internet code in their code bases. Given our findings, they should further take into consideration that some of the Internet code reused in their software might also violate resulting license obligations.

To address this situation, our study offers multiple levers for firms to mitigate the economic and legal risks from ad-hoc Internet code reuse. First, the topic of Internet code reuse needs to be positioned more prominently on firms' agendas. Firms should play an active role in making developers aware of the potential license issues resulting from Internet code reuse. They should select and leverage existing, reliable information resources on the Internet and complement them with useful mandatory internal trainings and other practical information. Second, firms should also lobby institutions of education such as universities to effectively include this topic into their curricula. Third, firms should establish easy to read and understand policies providing guidance to their employees on how to deal with Internet code. Moreover, firms need to make sure that developers are aware of these policies, read them and understand them. Finally, firms need to acknowledge the interdisciplinary nature of license risks from Internet code reuse, relating to developers and engineers as well as to lawyers. In this matter, they should e.g. facilitate fast unbureaucratic communication between developers and legal experts such that clearance for specific instances of Internet code reuse can be obtained quickly. Otherwise, developers under time pressure effectively face the choice between practicing such reuse on their own responsibility or abandoning it altogether—an option that would leave a valuable source of efficiency and quality gains unexploited.

References

1. Alexy, O. Free Revealing: How Firms Can Profit from Being Open. Gabler, Wiesbaden, 2009.
2. Chen, W., Li, J., Ma, J., Conradi, R., Ji, J. and Liu, C. An Empirical Study on Software Development with Open Source Components in the Chinese Software Industry. Software Process Improvement and Practice, 13, 1 (2008), 89-100.
3. Deshpande, A. and Riehle, D. The Total Growth of Open Source. in 4th International Conference on Open Source Systems, (Milan, Italy, 2008), 197-209.
4. Frakes, W.B. and Kang, K. Software Reuse Research: Status and Future. IEEE Transactions of Software Engineering, 31, 7 (2005), 529 - 536
5. Garlan, D., Allen, R. and Ockerbloom, J. Architectural Mismatch: Why Reuse Is Still So Hard. IEEE Software, 26, 4 (2009), 66-69.
6. German, D.M., Di Penta, M. and Davies, J. Understanding and Auditing the Licensing of Open Source Software Distributions. in 18th International Conference in Program Comprehension, (Braga, Portugal, 2010).
7. German, D.M., Di Penta, M., Guéhéneuc, Y.-G. and Antoniol, G. Code Siblings: Technical and Legal Implications of Copying Code between Applications. in 6th International Workshop on Mining Software Repositories, (Vancouver, Canada, 2009).
8. German, D.M. and Gonzalez-Barahona, J.M. An Empirical Study of the Reuse of Software Licensed under the GNU General Public License. in 5th International Conference on Open Source Systems, (Skövde, Sweden, 2009), 185-198.
9. German, D.M. and Hassan, A.E. License Integration Patterns: Dealing with License Mismatches in Component-Based Development. in 31st International Conference on Software Engineering, (Vancouver, Canada, 2009), 188-198.
10. Greene, W.H. Econometric Analysis. Prentice Hall, Upper Saddle River, NJ, 2007.
11. Lerner, J. and Tirole, J. The Scope of Open Source Licensing The Journal of Law, Economics, and Organization, 21, 1 (2005), 20-56.
12. Levi, S.D. and Woodard, A. Open Source Software: How to Use It and Control It in the Corporate Environment. Computer & Internet Lawyer, 21, 8 (2004), 8-13.
13. Li, J., Conradi, R., Bunse, C., Torchiano, M., Slyngstad, O.P.N. and Morisio, M. Development with Off-the-Shelf Components: 10 Facts. IEEE Software, 26, 2 (2009), 80-87.
14. Madanmohan, T.R. and De, R. Open Source Reuse in Commercial Firms. IEEE Software, 21, 6 (2004), 62-69.
15. McGhee, D.D. Free and Open Source Software Licenses: Benefits, Risks, and Steps toward Ensuring Compliance. Intellectual Property & Technology Law Journal, 19, 11 (2007), 5-9.
16. Morisio, M., Ezran, M. and Tully, C. Success and Failure Factors in Software Reuse. IEEE Transactions on Software Engineering, 28, 4 (2002), 340-357.
17. Murray, G.F. Categorization of Open Source Licenses: More Than Just Semantics. Computer & Internet Lawyer, 26, 1 (2009), 1-11.

18. Norris, J.S. Mission-Critical Development with Open Source Software: Lessons Learned. IEEE Software, 21, 1 (2004), 42-49.

19. Rosen, L. Open Source Licensing: Software Freedom and Intellectual Property Law. Prentice-Hall, Englewood Cliffs, NJ, 2004.

20. Sojer, M. Reusing Open Source Code. Gabler Verlag, Wiesbaden, 2010.

21. Sojer, M. and Henkel, J. Code Reuse in Open Source Software Development: Quantitative Evidence, Drivers, and Impediments. Journal of the Association for Information Systems, 11, 12 (2010), 868-901.

22. Spinellis, D. and Szyperski, C. How Is Open Source Affecting Software Development? IEEE Software, 21, 1 (2004), 28-33.

23. Umarji, M., Sim, S.E. and Lopes, C. Archetypal Internet-Scale Source Code Searching. in Russo, B., Damiani, E., Hissam, S., Lundell, B. and Succi, G. eds. Open Source Development, Communities and Quality, Springer, Boston, MA, 2008, 257-263.

Appendix

Appendix 1: Survey questionnaire

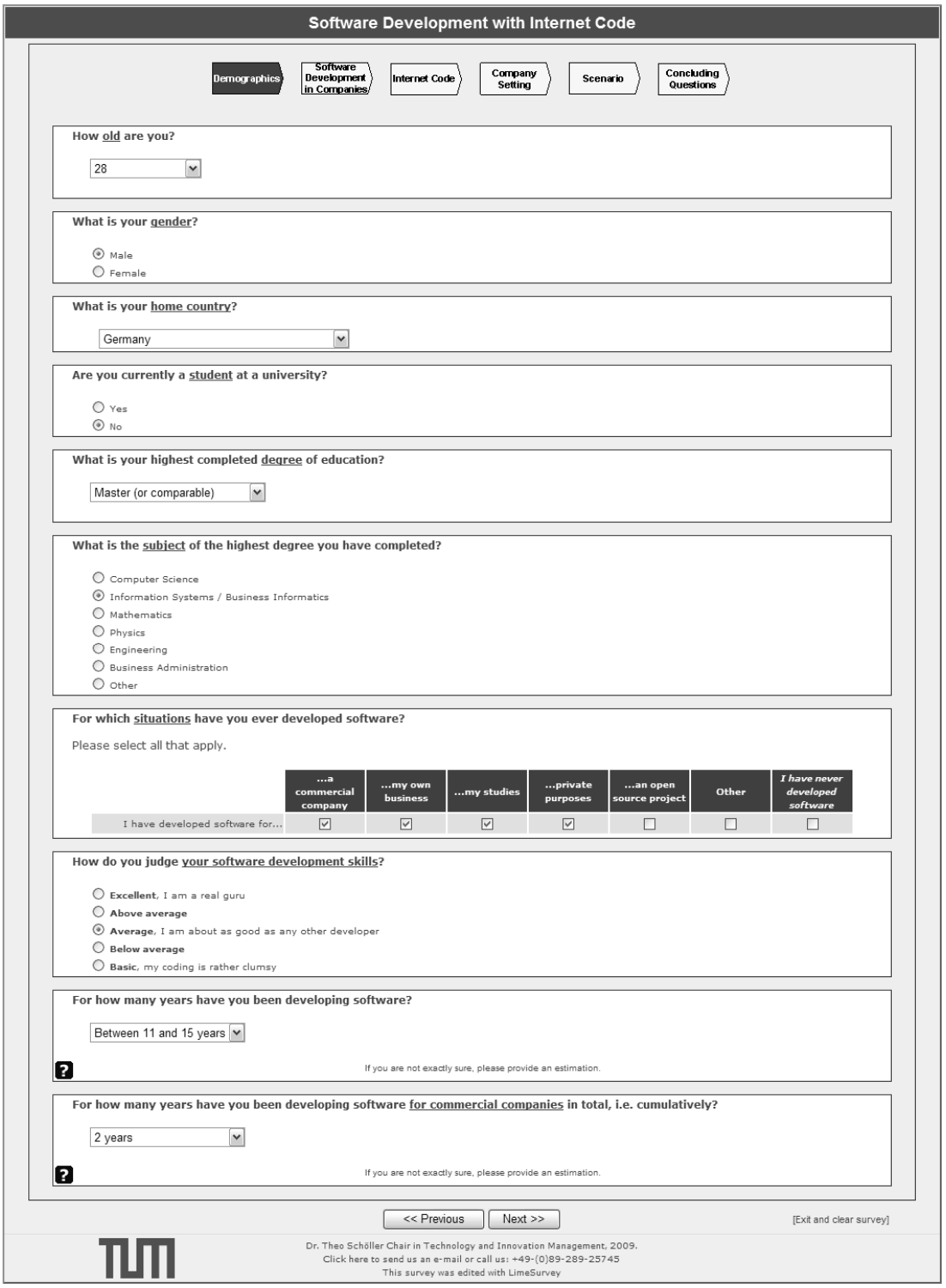
Three different versions of the questionnaire were employed and allocated to survey participants at random (see Figures A1.1, A1.2 and A1.3). The three different versions contain identical questions in all survey parts with the exception of the "scenario" part. Here each version presents a different scenario of a professional software developers reusing code from the Internet in a way potentially violating its license:

- In scenario 1 the developer reuses a snippet and does not check thoroughly for the license obligations that come with it.
- In scenario 2 the developer reuses a component and does not check thoroughly for the license obligations that come with it.
- In scenario 3 the developer reuses a snippet and is aware of GPL-like license obligations coming with the snippet. However, he consciously chooses to ignore these obligations.

Figure A1.1. Survey questionnaire – Scenario 1
(Full questionnaire with scenario 1: Not checking thoroughly for snippet reuse obligations)[18]

[18] See Figure A1.2 and Figure A1.3 for the other scenarios employed.

Appendix



Software Development with Internet Code

Demographics | Software Development in Companies | Internet Code | Company Setting | Scenario | Concluding Questions

How old are you?

28

What is your gender?

- (•) Male
- ( ) Female

What is your home country?

Germany

Are you currently a student at a university?

- ( ) Yes
- (•) No

What is your highest completed degree of education?

Master (or comparable)

What is the subject of the highest degree you have completed?

- ( ) Computer Science
- (•) Information Systems / Business Informatics
- ( ) Mathematics
- ( ) Physics
- ( ) Engineering
- ( ) Business Administration
- ( ) Other

For which situations have you ever developed software?

Please select all that apply.

	...a commercial company	...my own business	...my studies	...private purposes	...an open source project	Other	*I have never developed software*
I have developed software for...	☑	☑	☑	☑	☐	☐	☐

How do you judge your software development skills?

- ( ) **Excellent**, I am a real guru
- ( ) **Above average**
- (•) **Average**, I am about as good as any other developer
- ( ) **Below average**
- ( ) **Basic**, my coding is rather clumsy

For how many years have you been developing software?

Between 11 and 15 years

? If you are not exactly sure, please provide an estimation.

For how many years have you been developing software for commercial companies in total, i.e. cumulatively?

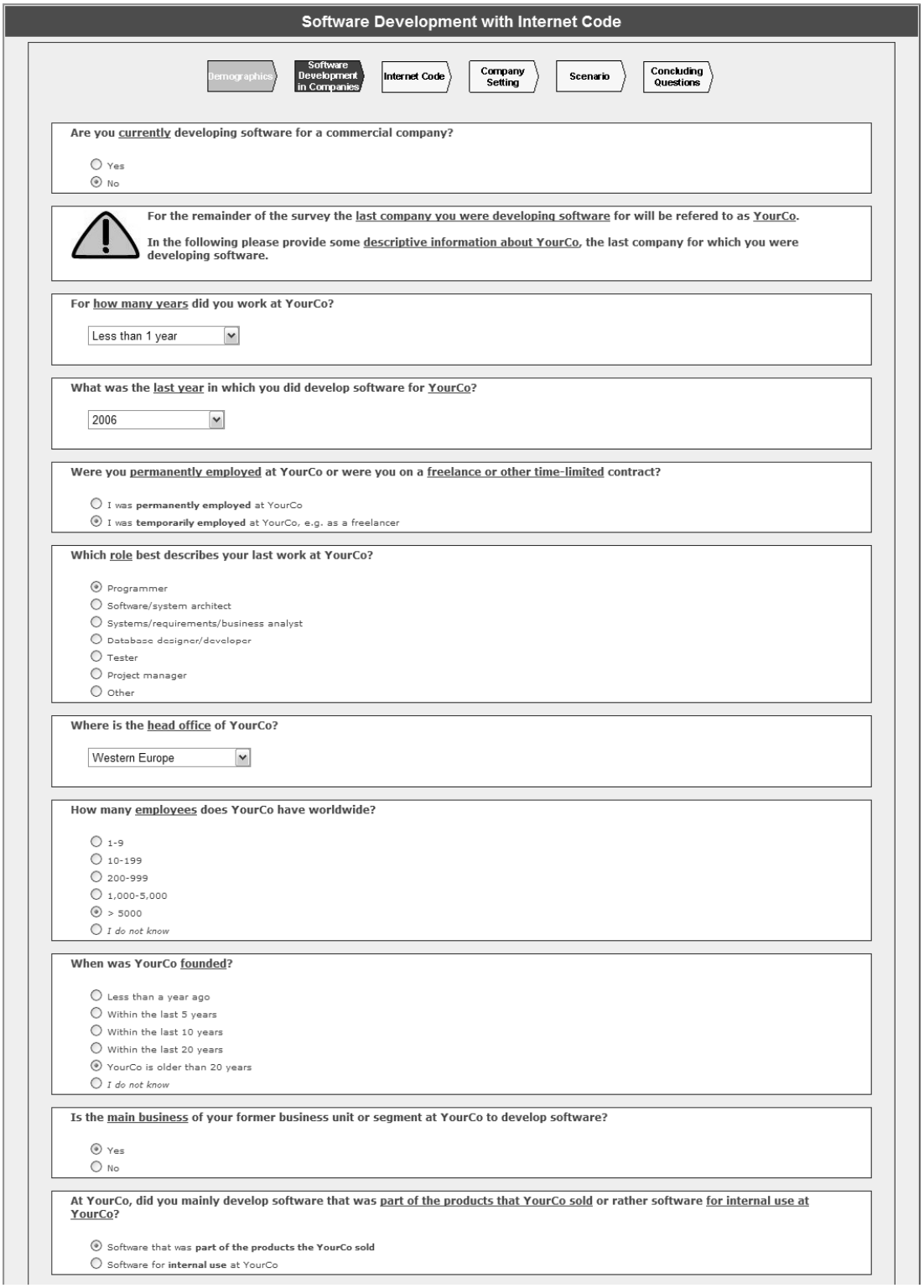
2 years

? If you are not exactly sure, please provide an estimation.

<< Previous | Next >> [Exit and clear survey]

Dr. Theo Schöller Chair in Technology and Innovation Management, 2009.
Click here to send us an e-mail or call us: +49-(0)89-289-25745
This survey was edited with LimeSurvey

Appendix



Software Development with Internet Code

Demographics | Software Development in Companies | Internet Code | Company Setting | Scenario | Concluding Questions

Are you currently developing software for a commercial company?

- ○ Yes
- ◉ No

For the remainder of the survey the last company you were developing software for will be refered to as YourCo.

In the following please provide some descriptive information about YourCo, the last company for which you were developing software.

For how many years did you work at YourCo?

Less than 1 year

What was the last year in which you did develop software for YourCo?

2006

Were you permanently employed at YourCo or were you on a freelance or other time-limited contract?

- ○ I was **permanently employed** at YourCo
- ◉ I was **temporarily employed** at YourCo, e.g. as a freelancer

Which role best describes your last work at YourCo?

- ◉ Programmer
- ○ Software/system architect
- ○ Systems/requirements/business analyst
- ○ Database designer/developer
- ○ Tester
- ○ Project manager
- ○ Other

Where is the head office of YourCo?

Western Europe

How many employees does YourCo have worldwide?

- ○ 1-9
- ○ 10-199
- ○ 200-999
- ○ 1,000-5,000
- ◉ > 5000
- ○ *I do not know*

When was YourCo founded?

- ○ Less than a year ago
- ○ Within the last 5 years
- ○ Within the last 10 years
- ○ Within the last 20 years
- ◉ YourCo is older than 20 years
- ○ *I do not know*

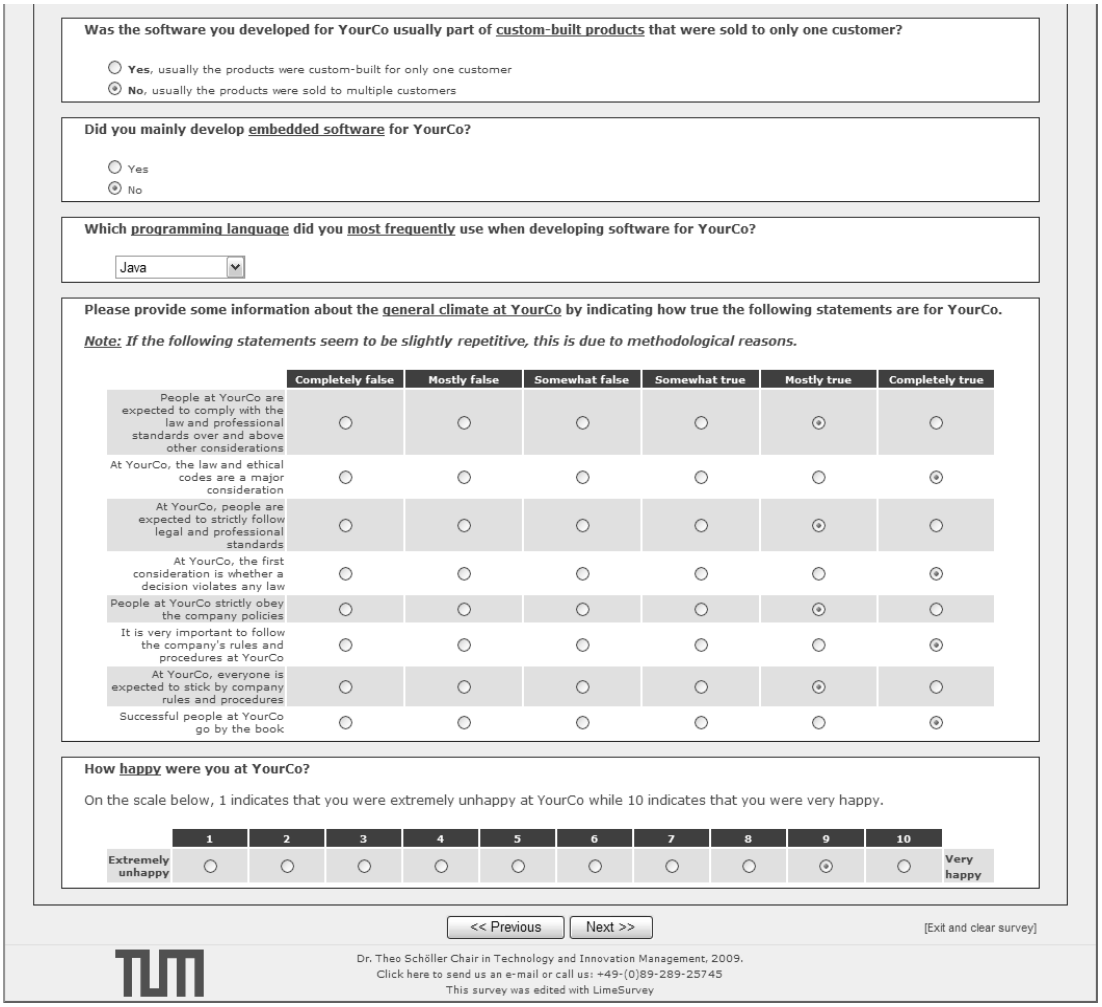
Is the main business of your former business unit or segment at YourCo to develop software?

- ◉ Yes
- ○ No

At YourCo, did you mainly develop software that was part of the products that YourCo sold or rather software for internal use at YourCo?

- ◉ Software that was **part of the products the YourCo sold**
- ○ Software for **internal use** at YourCo

Appendix



Was the software you developed for YourCo usually part of custom-built products that were sold to only one customer?

- ○ **Yes**, usually the products were custom-built for only one customer
- ◉ **No**, usually the products were sold to multiple customers

Did you mainly develop embedded software for YourCo?

- ○ Yes
- ◉ No

Which programming language did you most frequently use when developing software for YourCo?

Java

Please provide some information about the general climate at YourCo by indicating how true the following statements are for YourCo.

***Note:** If the following statements seem to be slightly repetitive, this is due to methodological reasons.*

	Completely false	Mostly false	Somewhat false	Somewhat true	Mostly true	Completely true
People at YourCo are expected to comply with the law and professional standards over and above other considerations	○	○	○	○	◉	○
At YourCo, the law and ethical codes are a major consideration	○	○	○	○	○	◉
At YourCo, people are expected to strictly follow legal and professional standards	○	○	○	○	◉	○
At YourCo, the first consideration is whether a decision violates any law	○	○	○	○	○	◉
People at YourCo strictly obey the company policies	○	○	○	○	◉	○
It is very important to follow the company's rules and procedures at YourCo	○	○	○	○	○	◉
At YourCo, everyone is expected to stick by company rules and procedures	○	○	○	○	◉	○
Successful people at YourCo go by the book	○	○	○	○	○	◉

How happy were you at YourCo?

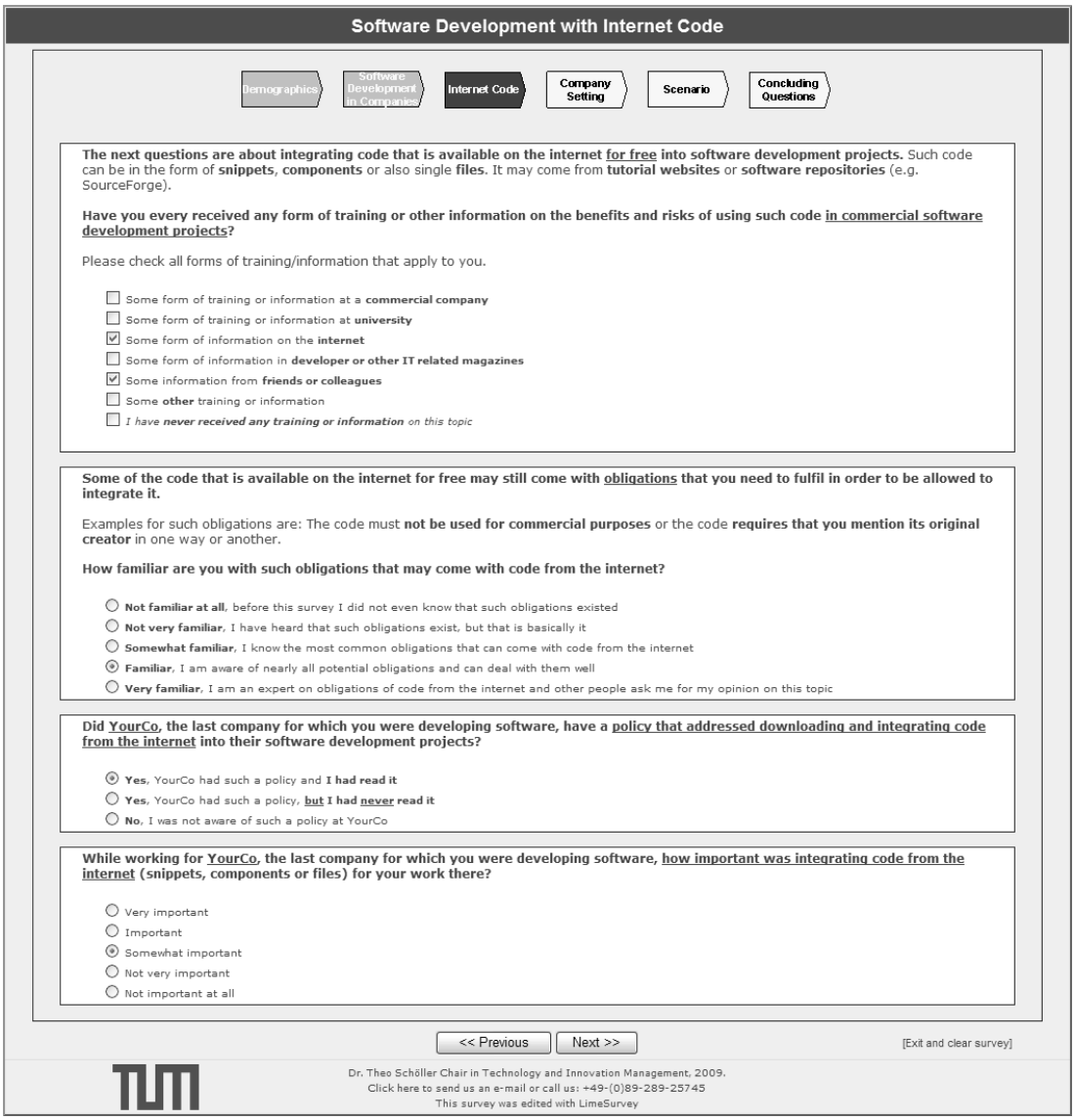
On the scale below, 1 indicates that you were extremely unhappy at YourCo while 10 indicates that you were very happy.

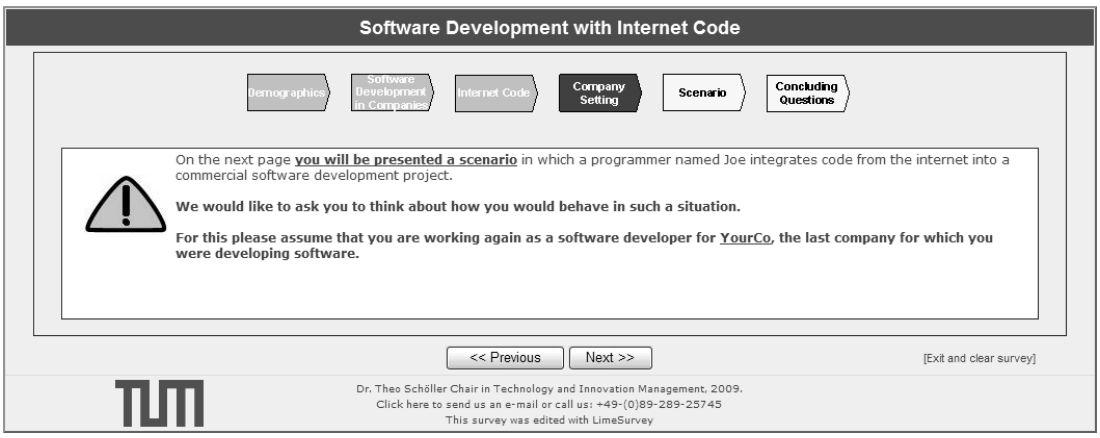
	1	2	3	4	5	6	7	8	9	10	
Extremely unhappy	○	○	○	○	○	○	○	○	◉	○	**Very happy**

<< Previous | Next >> [Exit and clear survey]

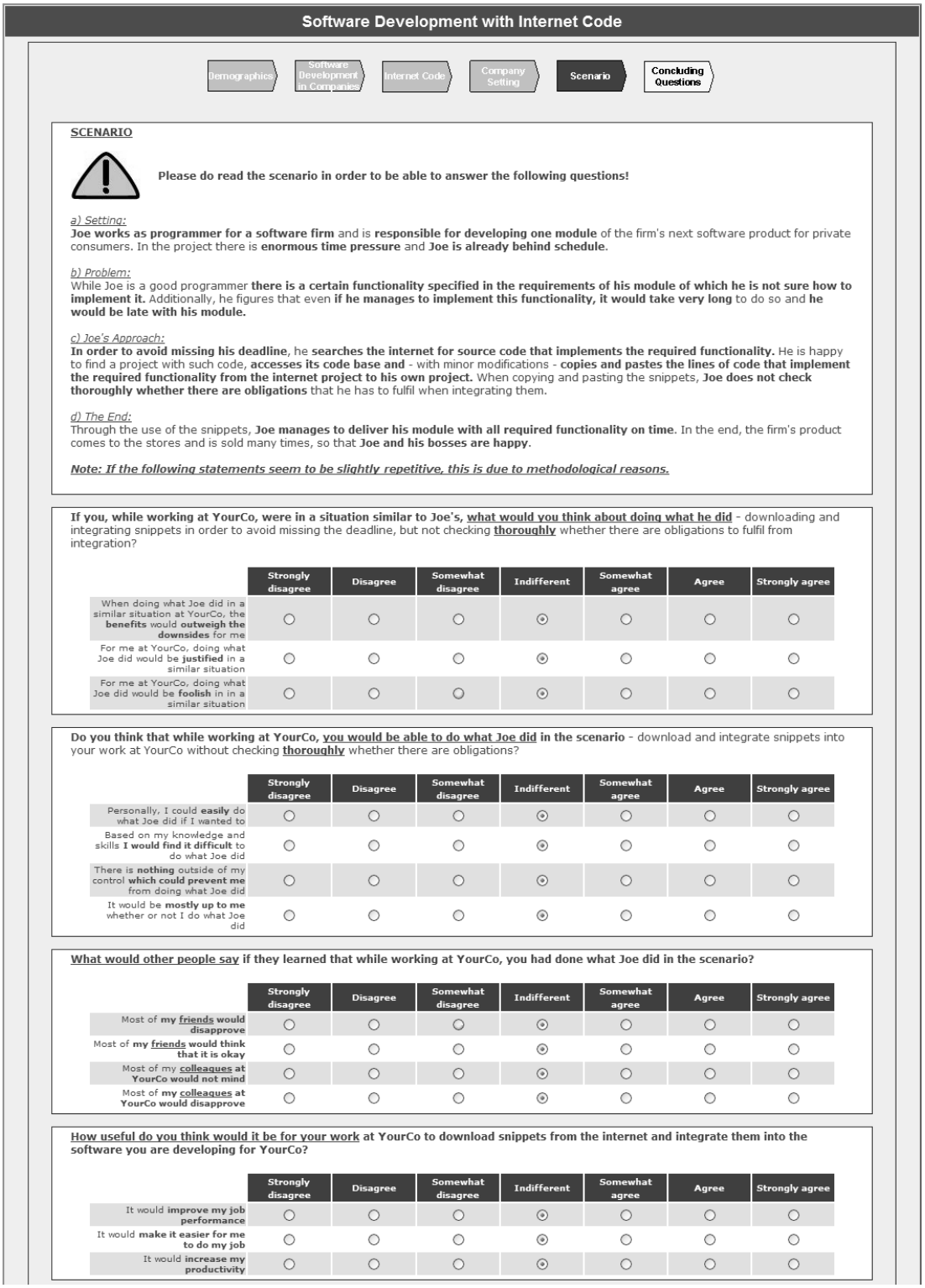
Dr. Theo Schöller Chair in Technology and Innovation Management, 2009.
Click here to send us an e-mail or call us: +49-(0)89-289-25745
This survey was edited with LimeSurvey

Appendix


Software Development with Internet Code
Demographics
Software Development in Companies
Internet Code
Company Setting
Scenario
Concluding Questions
The next questions are about integrating code that is available on the internet for free into software development projects. Such code can be in the form of snippets, components or also single files. It may come from tutorial websites or software repositories (e.g. SourceForge).
Have you every received any form of training or other information on the benefits and risks of using such code in commercial software development projects?
Please check all forms of training/information that apply to you.
Some form of training or information at a commercial company
Some form of training or information at university
Some form of information on the internet
Some form of information in developer or other IT related magazines
Some information from friends or colleagues
Some other training or information
I have never received any training or information on this topic
Some of the code that is available on the internet for free may still come with obligations that you need to fulfil in order to be allowed to integrate it.
Examples for such obligations are: The code must not be used for commercial purposes or the code requires that you mention its original creator in one way or another.
How familiar are you with such obligations that may come with code from the internet?
Not familiar at all, before this survey I did not even know that such obligations existed
Not very familiar, I have heard that such obligations exist, but that is basically it
Somewhat familiar, I know the most common obligations that can come with code from the internet
Familiar, I am aware of nearly all potential obligations and can deal with them well
Very familiar, I am an expert on obligations of code from the internet and other people ask me for my opinion on this topic
Did YourCo, the last company for which you were developing software, have a policy that addressed downloading and integrating code from the internet into their software development projects?
Yes, YourCo had such a policy and I had read it
Yes, YourCo had such a policy, but I had never read it
No, I was not aware of such a policy at YourCo
While working for YourCo, the last company for which you were developing software, how important was integrating code from the internet (snippets, components or files) for your work there?
Very important
Important
Somewhat important
Not very important
Not important at all
<< Previous
Next >>
[Exit and clear survey]
Dr. Theo Schöller Chair in Technology and Innovation Management, 2009.
Click here to send us an e-mail or call us: +49-(0)89-289-25745
This survey was edited with LimeSurvey


Software Development with Internet Code
Demographics
Software Development in Companies
Internet Code
Company Setting
Scenario
Concluding Questions
On the next page you will be presented a scenario in which a programmer named Joe integrates code from the internet into a commercial software development project.
We would like to ask you to think about how you would behave in such a situation.
For this please assume that you are working again as a software developer for YourCo, the last company for which you were developing software.
<< Previous
Next >>
[Exit and clear survey]
Dr. Theo Schöller Chair in Technology and Innovation Management, 2009.
Click here to send us an e-mail or call us: +49-(0)89-289-25745
This survey was edited with LimeSurvey

Appendix



Software Development with Internet Code

Demographics | Software Development in Companies | Internet Code | Company Setting | Scenario | Concluding Questions

SCENARIO

Please do read the scenario in order to be able to answer the following questions!

a) Setting:
Joe works as programmer for a software firm and is **responsible for developing one module** of the firm's next software product for private consumers. In the project there is **enormous time pressure** and **Joe is already behind schedule**.

b) Problem:
While Joe is a good programmer **there is a certain functionality specified in the requirements of his module of which he is not sure how to implement it.** Additionally, he figures that even **if he manages to implement this functionality, it would take very long** to do so and **he would be late with his module.**

c) Joe's Approach:
In order to avoid missing his deadline, he **searches the internet for source code that implements the required functionality.** He is happy to find a project with such code, **accesses its code base and** - with minor modifications - **copies and pastes the lines of code that implement the required functionality from the internet project to his own project.** When copying and pasting the snippets, **Joe does not check thoroughly whether there are obligations** that he has to fulfil when integrating them.

d) The End:
Through the use of the snippets, **Joe manages to deliver his module with all required functionality on time**. In the end, the firm's product comes to the stores and is sold many times, so that **Joe and his bosses are happy**.

Note: If the following statements seem to be slightly repetitive, this is due to methodological reasons.

If you, while working at YourCo, were in a situation similar to Joe's, what would you think about doing what he did - downloading and integrating snippets in order to avoid missing the deadline, but not checking **thoroughly** whether there are obligations to fulfil from integration?

	Strongly disagree	Disagree	Somewhat disagree	Indifferent	Somewhat agree	Agree	Strongly agree
When doing what Joe did in a similar situation at YourCo, the **benefits** would **outweigh the downsides** for me	○	○	○	◉	○	○	○
For me at YourCo, doing what Joe did would be **justified** in a similar situation	○	○	○	◉	○	○	○
For me at YourCo, doing what Joe did would be **foolish** in in a similar situation	○	○	○	◉	○	○	○

Do you think that while working at YourCo, you would be able to do what Joe did in the scenario - download and integrate snippets into your work at YourCo without checking **thoroughly** whether there are obligations?

	Strongly disagree	Disagree	Somewhat disagree	Indifferent	Somewhat agree	Agree	Strongly agree
Personally, I could **easily** do what Joe did if I wanted to	○	○	○	◉	○	○	○
Based on my knowledge and skills **I would find it difficult** to do what Joe did	○	○	○	◉	○	○	○
There is **nothing** outside of my control **which could prevent me** from doing what Joe did	○	○	○	◉	○	○	○
It would be **mostly up to me** whether or not I do what Joe did	○	○	○	◉	○	○	○

What would other people say if they learned that while working at YourCo, you had done what Joe did in the scenario?

	Strongly disagree	Disagree	Somewhat disagree	Indifferent	Somewhat agree	Agree	Strongly agree
Most of **my friends would disapprove**	○	○	○	◉	○	○	○
Most of **my friends would think that it is okay**	○	○	○	◉	○	○	○
Most of my **colleagues at YourCo would not mind**	○	○	○	◉	○	○	○
Most of **my colleagues at YourCo would disapprove**	○	○	○	◉	○	○	○

How useful do you think would it be for your work at YourCo to download snippets from the internet and integrate them into the software you are developing for YourCo?

	Strongly disagree	Disagree	Somewhat disagree	Indifferent	Somewhat agree	Agree	Strongly agree
It would **improve my job performance**	○	○	○	◉	○	○	○
It would **make it easier for me to do my job**	○	○	○	◉	○	○	○
It would **increase my productivity**	○	○	○	◉	○	○	○

Appendix

How serious do you think would be the consequences for YourCo if it became public that their software includes snippets from the internet, but does not fulfil the obligations of these snippets?

	Strongly disagree	Disagree	Somewhat disagree	Indifferent	Somewhat agree	Agree	Strongly agree
There would be **no or very low consequences for YourCo**	○	○	○	◉	○	○	○
YourCo would incur **major financial losses**	○	○	○	◉	○	○	○
YourCo would be in **serious legal trouble**	○	○	○	◉	○	○	○

How serious do you think would be the consequences for you personally if you were caught integrating snippets into YourCo's software without fulfiling the obligations as required by the snippets?

	Strongly disagree	Disagree	Somewhat disagree	Indifferent	Somewhat agree	Agree	Strongly agree
It would **not affect my future much**	○	○	○	◉	○	○	○
There would be **major negative consequences** for me	○	○	○	◉	○	○	○
It would **not hurt my career much**	○	○	○	◉	○	○	○

How easy do you think would it be to detect that YourCo's software contains snippets from the internet?

	Strongly disagree	Disagree	Somewhat disagree	Indifferent	Somewhat agree	Agree	Strongly agree
Scanning for snippets from the internet in YourCo's software **is virtually impossible**	○	○	○	◉	○	○	○
The **probability** that anybody would find out **is very high**	○	○	○	◉	○	○	○
It would **very difficult** for anybody to find out	○	○	○	◉	○	○	○

How serious do you think would it be for you personally if you failed to deliver required functionality on time at YourCo?

	Strongly disagree	Disagree	Somewhat disagree	Indifferent	Somewhat agree	Agree	Strongly agree
It would **not hurt my career much**	○	○	○	◉	○	○	○
It would **not affect my future much**	○	○	○	◉	○	○	○
There would be **major negative consequences** for me	○	○	○	◉	○	○	○

How easy would it be for you to check thoroughly for potential obligations that come with snippets from the internet that you want to integrate when working at YourCo?

	Strongly disagree	Disagree	Somewhat disagree	Indifferent	Somewhat agree	Agree	Strongly agree
It **would be very difficult for me** to check for all potential obligations of the snippets	○	○	○	◉	○	○	○
It **would take very long for me** to thoroughly check for all obligations that come with the snippets	○	○	○	◉	○	○	○

How likely is it that while working at YourCo, you will do what Joe did as described in the scenario?

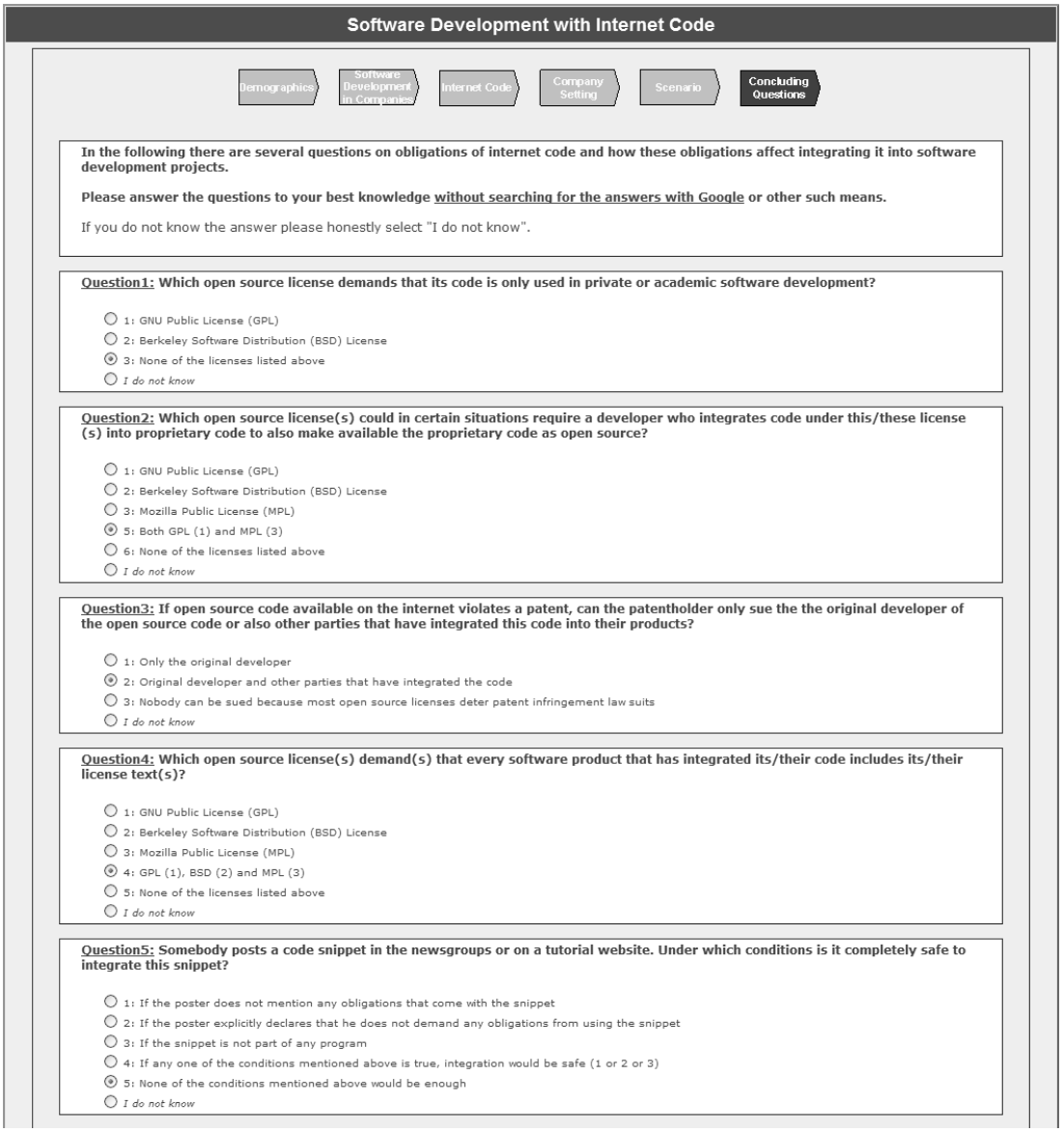
	Strongly disagree	Disagree	Somewhat disagree	Indifferent	Somewhat agree	Agree	Strongly agree
It is **likely** that I will do what Joe did in the future	○	○	○	◉	○	○	○
I may do what Joe did in the future	○	○	○	◉	○	○	○
I would never do what Joe did	○	○	○	◉	○	○	○

How frequently have you done what Joe did as described in the scenario during your past work at YourCo?

- ◉ Never
- ○ Once
- ○ Rarely
- ○ Sometimes
- ○ Often

<< Previous | Next >> [Exit and clear survey]

Appendix



Software Development with Internet Code

Demographics | Software Development in Companies | Internet Code | Company Setting | Scenario | Concluding Questions

In the following there are several questions on obligations of internet code and how these obligations affect integrating it into software development projects.

Please answer the questions to your best knowledge without searching for the answers with Google or other such means.

If you do not know the answer please honestly select "I do not know".

Question1: Which open source license demands that its code is only used in private or academic software development?

- ○ 1: GNU Public License (GPL)
- ○ 2: Berkeley Software Distribution (BSD) License
- ◉ 3: None of the licenses listed above
- ○ *I do not know*

Question2: Which open source license(s) could in certain situations require a developer who integrates code under this/these license (s) into proprietary code to also make available the proprietary code as open source?

- ○ 1: GNU Public License (GPL)
- ○ 2: Berkeley Software Distribution (BSD) License
- ○ 3: Mozilla Public License (MPL)
- ◉ 5: Both GPL (1) and MPL (3)
- ○ 6: None of the licenses listed above
- ○ *I do not know*

Question3: If open source code available on the internet violates a patent, can the patentholder only sue the the original developer of the open source code or also other parties that have integrated this code into their products?

- ○ 1: Only the original developer
- ◉ 2: Original developer and other parties that have integrated the code
- ○ 3: Nobody can be sued because most open source licenses deter patent infringement law suits
- ○ *I do not know*

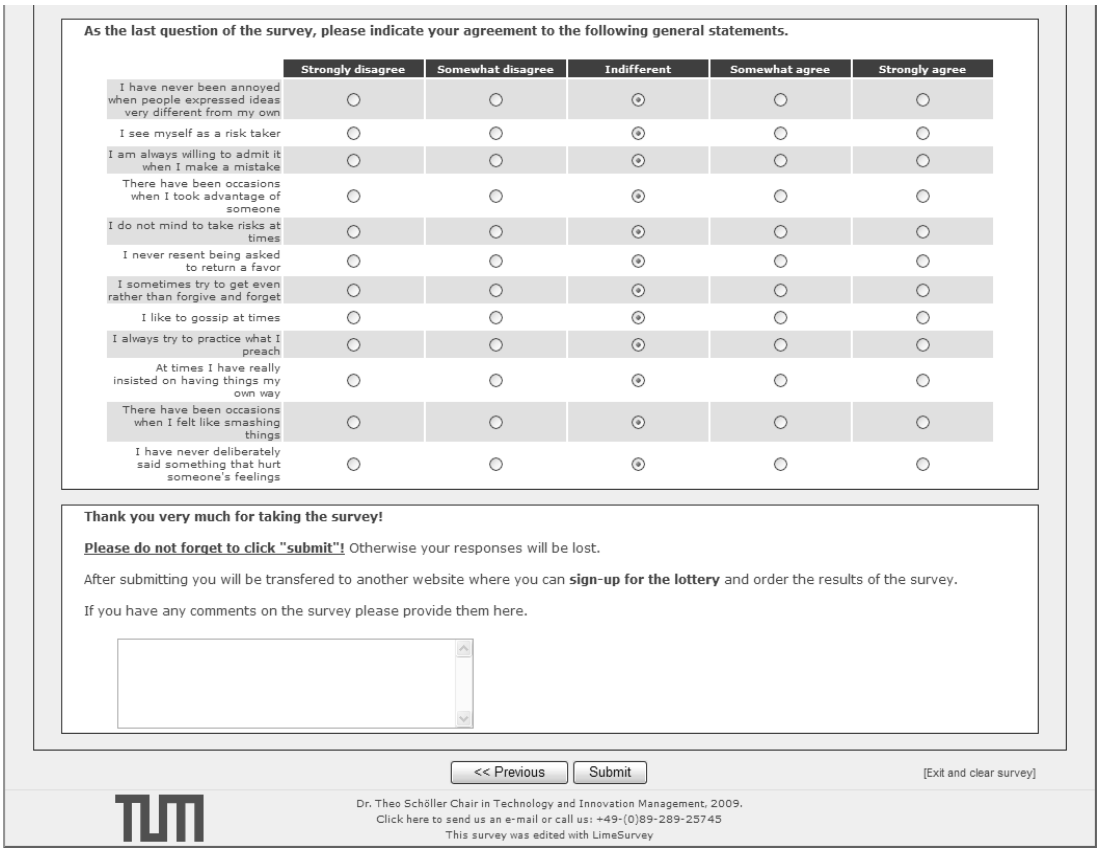
Question4: Which open source license(s) demand(s) that every software product that has integrated its/their code includes its/their license text(s)?

- ○ 1: GNU Public License (GPL)
- ○ 2: Berkeley Software Distribution (BSD) License
- ○ 3: Mozilla Public License (MPL)
- ◉ 4: GPL (1), BSD (2) and MPL (3)
- ○ 5: None of the licenses listed above
- ○ *I do not know*

Question5: Somebody posts a code snippet in the newsgroups or on a tutorial website. Under which conditions is it completely safe to integrate this snippet?

- ○ 1: If the poster does not mention any obligations that come with the snippet
- ○ 2: If the poster explicitly declares that he does not demand any obligations from using the snippet
- ○ 3: If the snippet is not part of any program
- ○ 4: If any one of the conditions mentioned above is true, integration would be safe (1 or 2 or 3)
- ◉ 5: None of the conditions mentioned above would be enough
- ○ *I do not know*

Appendix



As the last question of the survey, please indicate your agreement to the following general statements.

	Strongly disagree	Somewhat disagree	Indifferent	Somewhat agree	Strongly agree
I have never been annoyed when people expressed ideas very different from my own	○	○	◉	○	○
I see myself as a risk taker	○	○	◉	○	○
I am always willing to admit it when I make a mistake	○	○	◉	○	○
There have been occasions when I took advantage of someone	○	○	◉	○	○
I do not mind to take risks at times	○	○	◉	○	○
I never resent being asked to return a favor	○	○	◉	○	○
I sometimes try to get even rather than forgive and forget	○	○	◉	○	○
I like to gossip at times	○	○	◉	○	○
I always try to practice what I preach	○	○	◉	○	○
At times I have really insisted on having things my own way	○	○	◉	○	○
There have been occasions when I felt like smashing things	○	○	◉	○	○
I have never deliberately said something that hurt someone's feelings	○	○	◉	○	○

Thank you very much for taking the survey!

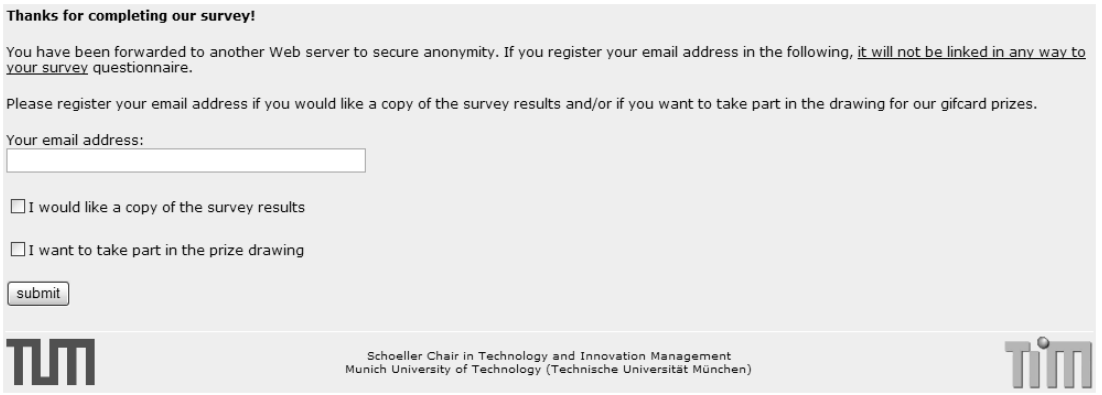
Please do not forget to click "submit"! Otherwise your responses will be lost.

After submitting you will be transfered to another website where you can **sign-up for the lottery** and order the results of the survey.

If you have any comments on the survey please provide them here.

<< Previous | Submit | [Exit and clear survey]

Dr. Theo Schöller Chair in Technology and Innovation Management, 2009.
Click here to send us an e-mail or call us: +49-(0)89-289-25745
This survey was edited with LimeSurvey



Thanks for completing our survey!

You have been forwarded to another Web server to secure anonymity. If you register your email address in the following, it will not be linked in any way to your survey questionnaire.

Please register your email address if you would like a copy of the survey results and/or if you want to take part in the drawing for our gifcard prizes.

Your email address:

☐ I would like a copy of the survey results

☐ I want to take part in the prize drawing

submit

Schoeller Chair in Technology and Innovation Management
Munich University of Technology (Technische Universität München)

Appendix

Figure A1.2. Survey questionnaire – Scenario 2
(Questionnaire excerpt with scenario 2: Not checking thoroughly for component reuse obligations)[19]

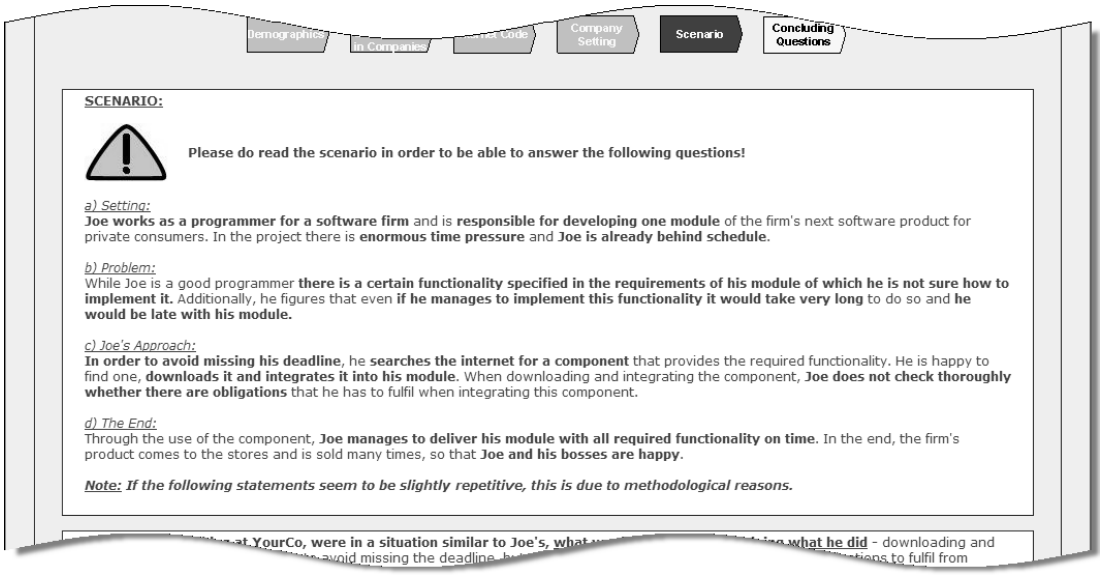
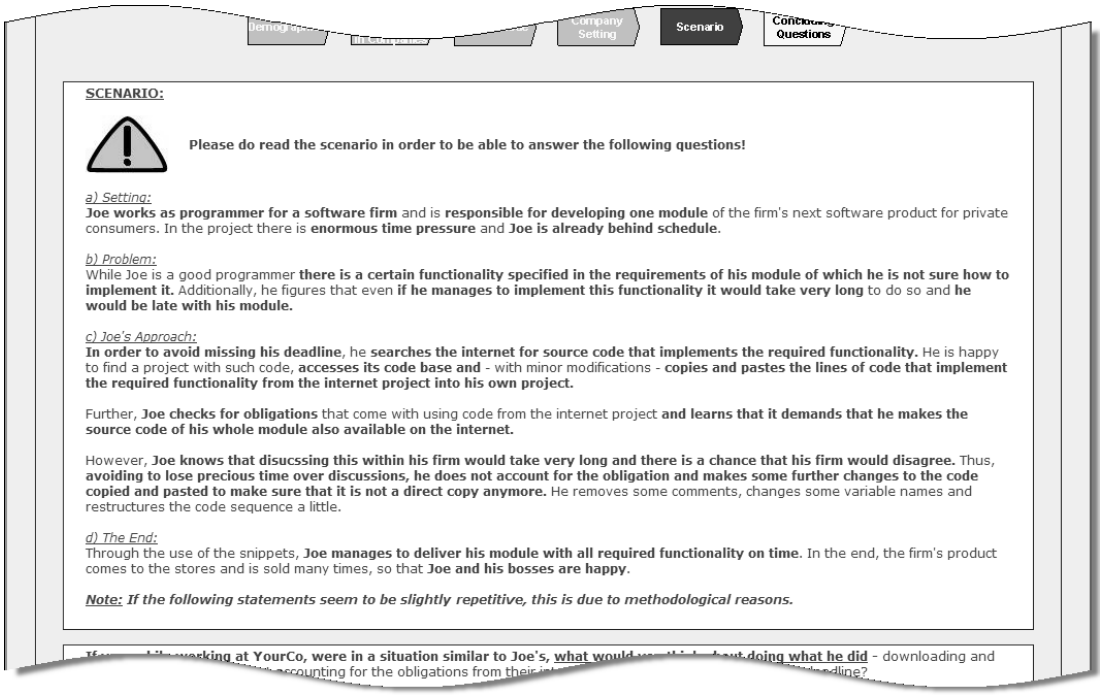


Figure A1.3. Survey questionnaire – Scenario 3
(Questionnaire excerpt with scenario 3: Knowingly ignoring obliations from snippet reuse)[20]



[19] The other parts of the questionnaire do not differ from those presented in Figure A1.1.

[20] The other parts of the questionnaire do not differ from those presented in Figure A1.1.

Appendix

Appendix 2: Construction of survey population

To construct the survey population all newsgroups were identified for which their names suggested that software development was a discussion topic relevant for them. This approach should ensure a rather representative sample of newsgroups relevant to our work. Of the resulting 528 newsgroups all postings available through the free NNTP-server extnews.news.cambrium.nl were downloaded during July 2009.[21] In the course of this process a total number of 1,314,336 postings was retrieved. Analyzing these postings provided us with 93,541 unique profiles (identified through their email addresses) of people active in these newsgroups. Table A2.1 lists all newsgroups identified and the number of postings retrieved for each of them.

Table A2.1. Overview of newsgroups identified and number of postings retrieved

Newsgroup name	# of postings retrieved	Newsgroup name	# of posting retrieved
comp.lang.ruby	84,342	comp.lang.modeling.misc	32
comp.lang.python	83,546	comp.lang.snobol	32
comp.lang.c	82,821	de.comp.lang.pascal.misc	32
comp.lang.c++	81,314	fj.comp.lang.fortran	32
comp.lang.javascript	62,723	it.comp.lang.vrml	32
comp.lang.java.programmer	60,226	japan.comp.lang.visual-c++	32
pl.comp.lang.c	41,118	comp.lang.cplu	31
comp.lang.lisp	39,898	comp.lang.rexx.tso	31
comp.lang.labview	30,828	comp.lang.simula	31
de.comp.lang.java	30,336	de.comp.lang.pascal.delphi	31
comp.lang.perl.misc	30,261	uk.comp.lang.lisp	31
pl.comp.lang.php	30,200	alt.comp.lang.delphi	30
it.comp.lang.visual-basic	28,708	alt.comp.lang.fortran	30
tw.bbs.comp.lang.java	26,908	alt.comp.lang.visualbasic.ver.3	30
comp.lang.forth	20,997	cn.comp.lang.java	30
tw.bbs.comp.lang.perl	16,527	cn.comp.lang.php	30
tw.bbs.comp.lang.basic	16,070	comp.lang.crass	30
tw.bbs.comp.lang.fortran	16,046	comp.lang.esterel	30
it.comp.lang.python	15,655	uk.comp.lang	30
it.comp.lang.delphi	15,435	alt.comp.lang.awk	29
it.comp.lang.javascript	15,308	comp.lang.c++.misc	29
comp.lang.java.help	15,266	comp.lang.rexx.vm	29
comp.lang.c++.moderated	14,530	comp.lang.sigplan	29
pl.comp.lang.delphi	14,332	fj.comp.lang.implementation	29
comp.lang.fortran	14,323	fj.comp.lang.lisp	29
pl.comp.lang.java	14,278	fj.comp.lang.tcl	29
comp.lang.clipper.visual-objects	14,263	comp.lang.asm37	28

[21] The server is not available any more as of writing this paper.

de.comp.lang.delphi.misc	14,261	comp.lang.delphi	28
comp.lang.tcl	14,247	comp.lang.dfl	28
de.comp.lang.php.misc	14,156	comp.lang.for	28
comp.lang.php	13,969	comp.lang.lisp.mcl	28
comp.lang.cobol	10,194	comp.lang.modeling	28
comp.lang.ada	10,168	comp.lang.occam	28
tw.bbs.comp.lang.php	10,153	comp.lang.plb	28
alt.comp.lang.learn.c-c++	10,133	comp.lang.yorick	28
comp.lang.basic.visual.misc	9,393	eug.comp.lang.perl	28
pl.comp.lang.javascript	7,869	fj.comp.lang.cobol	28
alt.comp.lang.borland-delphi	7,831	fj.comp.lang.vhdl	28
comp.lang.idl-pvwave	7,578	comp.lang.java.database	27
pl.comp.lang.python	7,448	de.comp.lang.lisp	27
pl.comp.lang.delphi.bazy-danych	7,412	relcom.comp.lang.basic	27
comp.lang.functional	7,361	relcom.comp.lang.pascal.misc	27
comp.lang.scheme	7,294	tw.bbs.comp.lang.fortr	27
cn.bbs.comp.lang.python	7,217	de.comp.lang	26
fr.comp.lang.c++	7,215	han.comp.lang.misc	26
comp.lang.java.gui	7,128	japan.comp.lang.delphi	26
de.comp.lang.javascript	6,884	relcom.comp.lang.c-c++	26
comp.lang.xharbour	5,443	relcom.comp.lang.pascal	26
fr.comp.lang.c	5,360	cz.comp.lang	25
cn.bbs.comp.lang.perl	5,206	cz.comp.lang.basic	25
de.comp.lang.perl.misc	5,005	de.comp.lang.delphi	25
comp.lang.verilog	4,961	eug.comp.lang	25
fr.comp.lang.javascript	4,914	fj.comp.lang.ada	25
comp.lang.pl1	4,202	han.comp.lang.fortran	25
comp.lang.rexx	4,081	japan.comp.lang	25
comp.lang.postscript	4,065	alt.comp.lang.jcl	24
fr.comp.lang.php	3,908	fj.comp.lang.verilog	24
alt.pl.comp.lang.csharp	3,856	pl.comp.lang	24
fr.comp.lang.python	3,797	relcom.comp.lang.perl	24
fr.comp.lang.tcl	3,756	tw.bbs.comp.lang	24
comp.lang.prolog	3,739	alt.comp.lang.macros.misc	23
de.comp.lang.python	3,737	relcom.comp.lang	23
comp.lang.clipper	3,704	alt.comp.lang.corelscript	22
comp.lang.awk	3,694	alt.comp.lang.lotuscript	22
pl.comp.lang.perl	3,683	alt.comp.lang.ms-dos	22
comp.lang.smalltalk	3,672	alt.comp.lang.pascal	22
de.comp.lang.c	3,633	comp.lang.pascal.mac	22
comp.lang.asm.x86	3,609	fj.comp.lang.pascal	22
de.comp.lang.iso-c++	3,608	gmane.comp.lang.groovy.user	22
alt.comp.lang.php	3,604	relcom.comp.lang.forth	22
it.comp.lang.c++	3,576	alt.comp.lang.forth	21
it.comp.lang.c	3,531	fj.comp.lang.forth	21
fr.comp.lang.java	3,505	japan.comp.lang.rexx	21
comp.lang.misc	3,504	alt.comp.lang.perfectscipt	20
tw.bbs.comp.lang.csharp	3,488	fj.comp.lang.awk	20
pl.comp.lang.vbasic	3,457	fj.comp.lang.functional	20
comp.lang.vhdl	3,419	gmane.comp.lang.javascript.v8.devel	20
comp.lang.pascal.delphi.misc	3,159	gmane.comp.lang.smalltalk.squeak.seaside	20
cn.bbs.comp.lang.java	3,054	han.comp.lang	20
cz.comp.lang.python	2,820	alt.comp.lang.shell.batch.winnt	19
comp.lang.smalltalk.dolphin	2,760	fr.comp.lang	19
han.comp.lang.c	2,558	japan.comp.lang.c	19
comp.lang.perl.tk	2,533	alt.comp.lang.hb++	18
fr.comp.lang.perl	2,402	alt.comp.lang.linoleum	18

comp.lang.perl.modules	2,331	alt.comp.lang.rexx	18
pl.comp.lang.pascal	2,242	comp.lang.clos	18
comp.lang.apl	2,040	comp.lang.sather	18
alt.fr.comp.lang.php	1,780	hun.comp.lang	18
comp.lang.java.advocacy	1,727	user.true.comp.lang.java.programmer	18
comp.lang.basic.misc	1,725	alt.comp.lang.beos	17
it.comp.lang.perl	1,720	alt.comp.lang.shell.unix.csh-tcsh	17
de.comp.lang.php.datenbanken	1,687	alt.fr.comp.lang	17
de.comp.lang.delphi.datenbanken	1,656	fj.comp.lang.objective-c	17
comp.lang.java.databases	1,635	alt.comp.lang.rebol	16
de.comp.lang.delphi.non-tech	1,549	alt.comp.lang.shell.batch.enhancements	16
cn.bbs.comp.lang.visual-basic	1,357	alt.comp.lang.shell.dcl	16
tw.bbs.comp.lang.python	1,354	alt.comp.lang.shell.os2	16
pl.comp.lang.asm	1,300	alt.comp.lang.shell.unix.korn	16
comp.lang.vrml	1,214	comp.lang.limbo	16
comp.lang.c.moderated	1,205	comp.lang.pascal.delphi.components.writing	16
comp.lang.clarion	1,177	free.it.comp.lang.svg	16
alt.comp.lang.applescript	1,135	user.true.comp.lang	16
comp.lang.haskell	1,109	user.true.comp.lang.java	16
han.comp.lang.c++	1,081	alt.comp.lang.ada	15
comp.lang.basic.realbasic	1,068	cn.comp.lang.vb	15
de.comp.lang.misc	1,044	comp.lang.c++.leda	15
comp.lang.logo	1,017	comp.lang.hermes	13
pl.comp.lang.ruby	993	comp.lang.scheme.scsh	13
comp.lang.mumps	898	cn.comp.lang.asp	12
cn.comp.lang.c	821	comp.lang.basic.visual.announce	12
comp.lang.perl.announce	806	fj.comp.lang.asm	12
it.comp.lang.vo-clipper	801	alt.comp.lang.learn.c-c++y	11
comp.lang.objective-c	791	alt.pl.comp.lang.u	11
comp.lang.java.softwaretools	781	free.it.comp.lang.aspnet	11
comp.lang.java	780	free.it.comp.lang.csharp	11
de.comp.lang.perl.cgi	773	gmane.comp.lang.r.geo	11
comp.lang.pascal.borland	740	cn.comp.lang.perl	10
comp.lang.pascal.misc	738	comp.lang.pascal.delphi.advocacy	10
fr.comp.lang.ruby	700	fr.comp.lang.lua	10
hun.comp.lang.java	682	gmane.comp.lang.r.devel	10
de.comp.lang.ruby	653	han.comp.lang.tcltk	10
comp.lang.visual.basic	612	alt.fr.comp.lang.gtk+	9
alt.comp.lang.java	576	alt.comp.lang.bantam	8
alt.comp.lang.javascript	559	alt.comp.lang.c++y	8
de.comp.lang.pascal	538	comp.lang.visual	8
de.comp.lang.assembler	510	free.it.comp.lang.vbnet	8
comp.lang.java.security	483	gmane.comp.lang.c.general	8
comp.lang.basic.visual	458	gmane.comp.lang.c++.root	8
cn.bbs.comp.lang	457	alt.swnet.comp.lang.basic	7
alt.comp.lang.c	418	alt.swnet.comp.lang.c	7
comp.lang.basic.powerbasic	417	alt.swnet.comp.lang.javascript	7
han.comp.lang.lisp	412	comp.lang.pascal.delphi.components.usage	7
cn.comp.lang	408	swnet.comp.lang	7
comp.lang.eiffel	406	cn.comp.lang.delphi	6
de.comp.lang.php.installation	405	comp.languages.visual-basic	6
alt.comp.lang.coldfusion	395	gmane.comp.lang.d.learn	6
comp.lang.modula2	383	hun.comp.lang.madach	6
de.comp.lang.forth	380	it-alt.comp.lang.xml	6
it-alt.comp.lang.asp	376	alt.comp.lang.scripte	5
comp.lang.basic.visual.database	375	cn.bbs.comp.lang.visual-	5
comp.lang.asm370	370	cn.bbs.comp.language	5

comp.lang.java.machine	366	cn.comp.lang.xml	5
comp.lang.ml	359	alt.comp.lang.learn-c-c++	4
free.it.comp.lang.html	354	comp.lang.ada.sucks	4
comp.lang.java.beans	344	comp.lang.c++sci.electronics.basics	4
comp.lang.oberon	343	comp.lang.java.advoacy	4
fr.comp.lang.ada	332	comp.lang.java.ejb	4
fr.comp.lang.general	316	comp.lang.pascal.delphi.announce	4
comp.lang.perl	312	comp.lang.postacript	4
fr.comp.lang.caml	303	comp.lang.tcl.announce	4
comp.lang.perl.moderated	296	gmane.comp.lang.ml.mlton.devel	4
it-alt.comp.lang.html	286	gmane.comp.lang.ocaml.lib.devel	4
han.comp.lang.perl	278	pl.comp.lang.www	4
comp.lang.java.3d	261	alt.comp.lang.learn.c-c	3
alt.comp.databases.xbase.clipper	259	alt.comp.lang.qbasic	3
comp.lang.dylan	259	alt.comp.lang.visualbasic.ver	3
comp.lang.pascal.ansi-iso	249	comp.lang.c++.perfometer	3
alt.lang.delphi	235	comp.lang.java.bugs	3
it-alt.comp.lang.lazarus	230	comp.lang.java.programming	3
fr.comp.lang.lisp	226	comp.lang.mysql	3
comp.lang.java.corba	217	comp.lang.s-lang	3
it.comp.lang.pascal	216	comp.lang.vhdl;	3
free.it.comp.lang.matlab	210	comp.language.c	3
comp.lang.pascal.delphi.databases	209	free.it.comp.lang.python.learner	3
comp.lang.icon	203	gmane.comp.lang.aldor	3
alt.comp.lang.shell.unix.bourne-bash	201	gmane.comp.lang.as400.rpg	3
de.comp.lang.funktional	195	gmane.comp.lang.erlang.general	3
comp.lang.forth.mac	194	gmane.comp.lang.haskell.cafe	3
pl.comp.lang.tcl	178	gmane.comp.lang.haskell.general	3
alt.comp.lang.visualbasic	173	gmane.comp.lang.haskell.libraries	3
comp.lang.pop	172	gmane.comp.lang.jruby.user	3
comp.lang.java.announce	170	gmane.comp.lang.lambda-prolog	3
comp.lang.idl	167	gmane.comp.lang.lua.general	3
de.comp.lang.assembler.x86	163	gmane.comp.lang.maude.general	3
fr.comp.lang.basic	159	gmane.comp.lang.ocaml.beginners	3
fr.comp.lang.regexp	151	gmane.comp.lang.perl.beginners	3
tw.bbs.comp.lang.cshar	148	gmane.comp.lang.perl.perl6.language	3
alt.comp.lang.perl	138	gmane.comp.lang.ruby.core	3
alt.comp.lang.vb	128	gmane.comp.lang.smalltalk.tweak	3
pl.comp.lang.funkcyjne	121	gmane.comp.lang.tcl.core	3
cn.comp.lang.vc	120	gmane.comp.lang.tcl.mac	3
comp.lang.java.misc	120	it.comp.lang.assembly	3
comp.lang.smalltalk.advocacy	111	alt.comp.lang.borland_delphi	2
de.comp.lang.php.netzprotokolle	111	alt.comp.lang.php;	2
alt.comp.lang.java-games	101	comp.lang.c+	2
cz.comp.lang.php	100	comp.lang.ccomp.lang.c++	2
fr.comp.lang.pascal	100	comp.lang.fpga	2
alt.comp.lang.assembler	98	comp.lang.mh	2
fj.comp.lang.java	93	comp.lang.smalllltalk	2
alt.comp.lang	89	comp.lang.visual-objects	2
alt.comp.lang.c++.help	81	free.it.comp.lang.delphi	2
alt.comp.lang.yorick	79	free.it.comp.lang.python	2
alt.comp.lang.visualbasic.ver3	78	gmane.comp.lang.boo.devel	2
comp.lang.asm	78	gmane.comp.lang.ml.general	2
comp.lang.java.javascript	76	gmane.comp.lang.perl.active-perl	2
comp.lang.beta	75	gmane.comp.lang.perl.modules.html-formfu.general	2
it-alt.comp.lang.ajax	75	gmane.comp.lang.perl.modules.log4perl.devel	2
alt.comp.lang.learn	73	gmane.comp.lang.perl.modules.openinteract.	2

		general	
comp.lang.java.developer	71	gmane.comp.lang.perl.perl5.porters	2
comp.lang.modula3	71	gmane.comp.lang.ruby.cvs	2
de.comp.lang.assembler.misc	71	gmane.comp.lang.ruby.general	2
cn.bbs.comp.lang.delphi	69	gmane.comp.lang.scala.xml	2
fj.comp.lang.st80	69	gmane.comp.lang.sml.smlnj	2
comp.lang.lisp.x	67	tw.bbs.comp.lang.shareware	2
japan.comp.lang.postscript	66	tw.bbs.comp.lang.software	2
alt.comp.lang.scriptease	64	alt.comp.lang.c++.misc	1
comp.lang.visual.basic.misc	64	alt.comp.lang.shell	1
comp.lang.basic.visual.3rdparty	63	alt.comp.language.c	1
gmane.comp.lang.d.general	63	cn.bbs.comp.lang.c	1
alt.comp.lang.reportsmith	61	comp.databases.mysql.comp.lang.php	1
comp.lang.lisp.franz	61	comp.lang.b	1
fj.comp.lang.misc	60	comp.lang.basic;	1
han.comp.lang.java	60	comp.lang.c;comp.arch.embedded	1
de.comp.lang.php	59	comp.lang.compilers	1
it.comp.lang	59	comp.lang.ebonics	1
fj.comp.lang.ruby	58	comp.lang.eisc	1
alt.comp.lang.superlang	57	comp.lang.fpg	1
fj.comp.lang.c++	57	comp.lang.java.awt	1
comp.lang.scheme.c	56	comp.lang.java.databses	1
fj.comp.lang.postscript	54	comp.lang.java.he	1
comp.lang.c-programming	53	comp.lang.java.programer	1
comp.lang.basic	52	comp.lang.java.sercurity	1
comp.lang.java.api	52	comp.lang.modeling.gams	1
fj.comp.lang.basic	52	comp.lang.ms-windows	1
alt.comp.lang.vba	51	comp.lang.pascal.delphi.components.usage.writing	1
alt.lang.vb5.rumors	51	comp.lang.perl.java	1
fj.comp.lang.c	51	comp.lang.phthon	1
fr.comp.lang.postscript	50	comp.lang.pmrd	1
alt.comp.lang.vbscript	48	comp.lang.postscri	1
bln.comp.lang.opal	48	comp.lang.programming	1
comp.lang.clu	48	comp.lang.rubyonrails	1
japan.comp.lang.misc	48	comp.lang.scheee	1
comp.lang	47	comp.lang.smallalk.dolphin	1
fj.comp.lang.perl	47	comp.lang.transputer	1
comp.lang.java.setup	44	comp.lang.visualbasic	1
alt.comp.lang.haskell	43	comp.lang;c	1
comp.lang.pascal	43	de&comp.lang.iso-c++	1
comp.lang.xml	43	de.comp.lang.assemble	1
japan.comp.lang.perl	43	eug&comp.lang.perl	1
comp.lang.java.tech	42	gmane.comp.lang.haskell.cashew-s.engine.auto	1
comp.lang.prograph	42	gmane.comp.lang.haskell.cashew-s.engine.devel	1
japan.comp.lang.javascript	42	gmane.comp.lang.haskell.cashew-s.engine.patches	1
comp.lang.pascal.delphi	41	gmane.comp.lang.perl.modules. apache-asp.general	1
fj.comp.lang.visualbasic	41	gmane.comp.lang.perl.modules.authors	1
japan.comp.lang.visual-basic	41	gmane.comp.lang.perl.modules.cgi-prototype.user	1
bln.comp.lang	40	gmane.comp.lang.perl.modules.formbuilder	1
fj.comp.lang.javascript	40	gmane.comp.lang.perl.modules.openinteract.devel	1
japan.comp.lang.vrml	40	gmane.comp.lang.perl.perl6.compiler	1
comp.lang.pascal.delphi.components.misc	38	gmane.comp.lang.perl.perl6.regexp	1
cz.comp.lang.java	38	gmane.comp.lang.pike.user	1
cz.comp.lang.perl	38	gmane.comp.lang.ruby.rake	1
cz.comp.lang.xml	38	gmane.comp.lang.ruby.wxruby.user	1
japan.comp.lang.basic	38	gmane.comp.lang.smalltalk.smallwiki	1
japan.comp.lang.s	38	gmane.comp.lang.smalltalk.squeak.audio	1

japan.comp.lang.xml	38	gmane.comp.lang.smalltalk.squeak.modules	1
alt.comp.lang.visulabasic.ver3	37	infostrada.comp.lang.c	1
cz.comp.lang.basic.visual	37	it.comp.lang.visual-	1
comp.lang.pascal.delphi.components	36	jipan.comp.lang.perl	1
fj.comp.lang.prolog	36	nj.comp.lang.basic	1
de.comp.lang.c++	35	si.comp.lang	1
relcom.fido.su.c-c++	35	tw.bbs.comp.lang.bas	1
de.comp.lang.perl	34	tw.bbs.comp.lang.fo	1
comp.lang.algol	33	tw.bbs.comp.lang.pe	1
alt.comp.lang.sed	32	tw.bbs.comp.lang.per	1

Appendix

Appendix 3: Descriptive statistics of the explanatory variables used in Table 1

Table A3.1. Descriptive statistics of the explanatory variables used in Table 1

Variable	Frequency of "0"	Frequency of "1"
Developer has never received any form of training or information on Internet code reuse	662 (76%)	207 (24%)
Developer has OSS experience	383 (44%)	486 (56%)
Software development role *project manager*	772 (96%)	35 (4%)
Software development role *programmer*	394 (49%)	413 (51%)
Software development role *analyst*	791 (98%)	16 (2%)
Software development role *tester*	799 (99%)	8 (1%)
Software development role *database developer*	790 (98%)	17 (2%)
Software development role *other*	716 (89%)	91 (11%)
Software development role architect (reference group in regression model)	*580 (72%)*	*227 (28%)*
Primary programming language *Python*	763 (88%)	106 (12%)
Primary programming language *Perl*	842 (97%)	27 (3%)
Primary programming language *Java*	744 (86%)	125 (14%)
Primary programming language *PHP*	824 (95%)	45 (5%)
Primary programming language *C*	795 (91%)	74 (9%)
Primary programming language *C++*	699 (80%)	170 (20%)
Primary programming language *Visual Basic*	845 (97%)	24 (3%)
Primary programming language *C#*	824 (95%)	45 (5%)
Primary programming language *other*	751 (86%)	118 (14%)
Primary programming languages Ruby (reference group in regression model)	*799 (92%)*	*70 (8%)*
Developer lives in *Europe*	412 (47%)	457 (53%)
Developer lives in *North America*	624 (72%)	245 (28%)
Developer lives in *Asia or rest of world*	735 (85%)	134 (15%)
Developer lives in South America (reference group in regression model)	*836 (96%)*	*33 (4%)*
Developer is working as a freelancer	667 (77%)	202 (23%)
Education in *computer science or related subject*	461 (53%)	408 (47%)
Education in *mathematics or physics*	763 (88%)	106 (12%)
Education in *business administration*	850 (98%)	19 (2%)
Education in *other subject*	798 (92%)	71 (8%)
Education in engineering (reference group in regression model)	*717 (83%)*	*152 (17%)*
Developer is working on embedded software projects	699 (80%)	170 (20%)

Variable	Explanation	Min.	Max.	Med.	Mean	S.D.
License risk level of developer's work	Criticality of license risks in developer's work (1=Low since developer is working on internal-use projects, 2=Medium since developer is working on external projects for only one customer, 3=High since developer is working on projects for multiple external customers)	0.0	2.0	1.0	1.1	0.9
Developer's self-assessed knowledge about Internet code licenses	Self-assessment of developer's familiarity with Internet code license obligations (1=Not familiar at all,…, 5=Very familiar)	1.0	5.0	4.0	3.7	0.8
Developer's objectively assessed knowledge about Internet code licenses	Score developer has achieved in five question quiz on Internet code license obligations. Maximum score is 5.	0.0	5.0	2.5	2.5	1.3
Experience as professional software developer (in years)	Number of cumulative years developer has been working as professional software developer	0.5	45.0	7.0	9.7	8.3
Last year as professional software developer	Last year in which developer has still been active as professional software developer.	1985	2009	2009	2008.3	2.6
Developers' self-assessed software development skill level	Self-assessment of developer's software development skills (1=Basic,…, 5=Excellent)	1.0	5.0	4.0	3.9	0.8

Note: N=869.

Appendix

Table A3.2. Correlation matrix of the explanatory variables used in Table 1

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.	21.	22.	23.	24.	25.	26.	27.	28.	29.	30.	31.	32.
1. License risk level of developer's work	1.00																															
2. Developer has never received any form of training or information on Internet code reuse	**-0.07**	1.00																														
3. Developer's self-assessed knowledge about Internet code licenses		**-0.16**	1.00																													
4. Developer's objectively assessed knowledge about Internet code licenses		**-0.14**	**0.34**	1.00																												
5. Developer has OSS experience		**-0.11**	**0.28**	**0.30**	1.00																											
6. Experience as professional software developer (in years)			**0.11**			1.00																										
7. Last year as professional software developer		**-0.06**				**0.07**	1.00																									
8. Software development role project manager				**0.06**			**-0.07**	1.00																								
9. Software development role programmer			**-0.08**			**-0.27**		n.m.	1.00																							
10. Software development role analyst		**-0.08**		**-0.07**	**-0.13**			n.m.	n.m.	1.00																						
11. Software development role tester	**-0.13**				**-0.06**			n.m.	n.m.	n.m.	1.00																					
12. Software development role database developer			**-0.06**	**-0.06**	**-0.08**	**0.07**		n.m.	n.m.	n.m.	n.m.	1.00																				
13. Software development role other	**-0.11**					**0.09**		n.m.	n.m.	n.m.	n.m.	n.m.	1.00																			
14. Primary programming language Python	**-0.10**			**0.08**	**0.13**	**-0.07**	**0.09**							1.00																		
15. Primary programming language Perl	**-0.08**				**0.08**	**0.06**								n.m.	1.00																	
16. Primary programming language Java												**-0.06**	**-0.10**	n.m.	n.m.	1.00																
17. Primary programming language PHP						**-0.11**								n.m.	n.m.	n.m.	1.00															
18. Primary programming language C	**0.07**	**0.06**												n.m.	n.m.	n.m.	n.m.	1.00														
19. Primary programming language C++	**0.18**				**-0.06**		**0.07**		**0.09**	**-0.07**			**-0.10**	n.m.	n.m.	n.m.	n.m.	n.m.	1.00													
20. Primary programming language Visual Basic	**-0.07**			**-0.09**						**0.09**	**0.06**			n.m.	n.m.	n.m.	n.m.	n.m.	n.m.	1.00												
21. Primary programming language C#	**0.06**													n.m.	n.m.	n.m.	n.m.	n.m.	n.m.	n.m.	1.00											
22. Primary programming language other	**-0.08**					**0.20**	**-0.19**	**0.07**	**-0.11**			**0.09**	**0.10**	n.m.	n.m.	n.m.	n.m.	n.m.	n.m.	n.m.	n.m.	1.00										
23. Developer lives in Europe				**0.10**		**-0.12**	**-0.06**						**-0.07**										1.00									
24. Developer lives in North America	**-0.09**		**0.08**			**0.23**		**-0.08**	**-0.08**	**-0.07**			**0.10**				**-0.07**						n.m.	1.00								
25. Developer lives in Asia or rest of world	**0.10**		**-0.11**	**-0.07**	**-0.10**	**-0.11**	**0.08**			**0.06**									**0.06**			**-0.08**	n.m.	n.m.	1.00						
26. Developer is working as a freelancer	**-0.10**				**0.06**	**-0.15**	**-0.08**		**0.16**				**-0.14**				**0.15**		**-0.08**				**0.11**	**-0.07**	**-0.06**	1.00					
27. Education in computer science or related subject	**0.06**					**-0.22**	**0.06**	**-0.07**	**0.19**	**0.08**	**-0.09**		**-0.17**			**0.09**		**-0.06**	**0.07**	**-0.09**		**-0.10**		**-0.07**			1.00					
28. Education in mathematics or physics	**-0.11**					**0.16**			**-0.07**		**0.12**		**0.06**									**0.07**	**0.10**		**-0.07**		n.m.	1.00				
29. Education in business administration				**-0.07**					**-0.08**		**0.07**	**0.09**										**0.06**					n.m.	n.m.	1.00			
30. Education in other subject				**-0.06**			**-0.09**						**0.10**				**0.12**		**-0.10**					**0.09**	**-0.09**	**0.06**	n.m.	n.m.	n.m.	1.00		
31. Developer is working on embedded software projects	**0.12**							**0.06**						**-0.11**		**-0.09**	**-0.06**	**0.29**	**0.14**		**-0.08**				**0.09**		**-0.09**				1.00	
32. Developers' self-assessed software development skill level	**0.12**	**-0.06**	**0.25**	**0.12**	**0.23**	**0.31**	**0.12**		**-0.09**		**-0.08**		**-0.12**				**-0.07**		**0.16**	**-0.09**		**-0.06**		**0.07**		**-0.09**	**0.10**		**-0.06**	**-0.07**		1.00

Notes: Only correlations with $p<0.1$ are shown; n.m. = not meaningful because variables are dummy variables coding the same characteristic.

Appendix

Appendix 4: Internet code license obligation quiz

The quiz was developed after our 20 interviews with Internet code reuse industry experts. It covers five typical situations in which professional software developers may violate license obligations when reusing Internet code. Developers received one point for each correct answer to the five quiz questions and could mark only one answer as correct for each question. Table A4.1 presents the quiz questions and the answers of those developers who were still creating software in 2009.

Table A4.1. Internet code license obligation quiz

Quiz questions and answers	Percentage
Which open source license(s) could in certain situations require a developer who integrates code under this/these license(s) into proprietary code to also make available the proprietary code as open source?	
GNU General Public License (GPL)*	**56%**
Berkeley Software Distribution (BSD) License	1%
Mozilla Public License (MPL)	1%
Both GPL and MPL	**27%**
None of the licenses listed above	1%
Do not know	14%
Which open source license(s) demand(s) that every software product that has integrated its/their code includes its/their license text(s)?	
GNU General Public License (GPL)	16%
Berkeley Software Distribution (BSD) License	3%
Mozilla Public License (MPL)	1%
GPL, BSD and MPL	**60%**
None of the licenses listed above	1%
Do not know	19%
Which open source license demands that its code is only used in private or academic software development?	
GNU General Public License (GPL)	5%
Berkeley Software Distribution (BSD) License	10%
None of the licenses listed above	**68%**
Do not know	17%
Somebody posts a code snippet in the newsgroups or on a tutorial website. Under which conditions is it completely safe to integrate this snippet?	
If the poster does not mention any obligations that come with the snippet	9%
If the poster explicitly declares that he does not demand any obligations from using the snippet	39%
If the snippet is not part of any program	1%
If any one of the conditions above mentioned is true, integration would be safe	16%
None of the conditions mentioned above would be enough	**19%**
Do not know	16%
If open source code available on the Internet violates a patent, can the patent holder only sue the original developer of the open source code or also other parties that have integrated this code into their products?	
Only the original developer	6%
Original developer and other parties that have integrated the code	**52%**
Nobody can be sued because most open source licenses deter patent infringement law suits	3%
Do not know	39%

*While this answer is not fully correct, developers did still receive 0.5 credits for it in the calculation of the quiz scores.
Notes: Correct answers are bolded; N=732.

Appendix

Appendix 5: Descriptive statistics of the explanatory variables used in Table 3

Table A5.1. Descriptive statistics of the explanatory variables used in Table 3

Variable	Frequency of "0"	Frequency of "1"
Developer has OSS experience	383 (44%)	486 (56%)
Developer has received training or information on Internet code from *firms*	695 (80%)	174 (20%)
Developer has received training or information on Internet code from *institutions of education*	722 (83%)	147 (17%)
Developer has received training or information on Internet code from *friends or colleagues*	484 (56%)	385 (44%)
Developer has received information on Internet code from the *Internet*	305 (35%)	564 (65%)
Developer has received information on Internet code from *magazines*	595 (68%)	274 (32%)
Developer has received information on Internet code from *other sources*	826 (95%)	43 (5%)
Developer lives in *North America*	624 (72%)	245 (28%)
Developer lives in *South America*	836 (96%)	33 (4%)
Developer lives in *Asia or rest of world*	735 (85%)	134 (15%)
Developer lives in Europe (reference group in regression model)	*412 (47%)*	457 (53%)
Education in *engineering*	717 (83%)	152 (17%)
Education in *mathematics or physics*	763 (88%)	106 (12%)
Education in *business administration*	850 (98%)	19 (2%)
Education in *other subject*	798 (92%)	71 (8%)
Education in computer science or related subject (reference group in regression model)	*461 (53%)*	*408 (47%)*

Variable	Explanation	Min.	Max.	Med.	Mean	S.D.
Experience as professional software developer (in years)	Number of cumulative years developer has been working as professional software developer	0.5	45.0	7.0	9.7	8.3

Note: N=869.

Table A5.2. Correlation matrix of the explanatory variables used in Table 3

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
1. Developer has OSS experience	1.00														
2. Developer has received training or information on Internet code from firms		1.00													
3. Developer has received training or information on Internet code from institutions of education		**0.16**	1.00												
4. Developer has received training or information on Internet code from friends or colleagues	**0.10**	**0.14**	**0.23**	1.00											
5. Developer has received information on Internet code from the Internet	**0.10**	**0.10**	**0.20**	**0.43**	1.00										
6. Developer has received information on Internet code from magazines	**0.06**	**0.15**	**0.08**	**0.35**	**0.36**	1.00									
7. Developer has received information on Internet code from other sources	**0.09**					**0.11**	1.00								
8. Developer lives in North America			**-0.08**		**-0.08**		**0.11**	1.00							
9. Developer lives in South America				**0.07**				n.m.	1.00						
10. Developer lives in Asia or rest of world	**-0.10**	**0.10**						n.m.	n.m.	1.00					
11. Education in engineering	**-0.07**							**0.07**		**0.07**	1.00				
12. Education in mathematics or physics			**-0.07**						**-0.07**	**-0.07**	n.m.	1.00			
13. Education in business administration									**0.09**		n.m.	n.m.	1.00		
14. Education in other subject			**-0.06**					**0.09**		**-0.09**	n.m.	n.m.	n.m.	1.00	
15. Experience as professional software developer (in years)			**-0.22**		**-0.07**	**0.07**	**0.10**	**0.23**		**-0.11**	**0.11**	**0.16**			1.00

Notes: Only correlations with $p<0.1$ are shown; n.m. = not meaningful because variables are dummy variables coding the same characteristic.

Appendix

Appendix 6: Multivariate analysis of existence of firm policies on Internet code reuse

Table A6.1. Multivariate analysis of existence of firm policies on Internet code reuse

	Logistic Regression		Logistic Regression		Logistic Regression		Logistic Regression	
	Coef.	Std. Err.	dy/dx	Std. Err.	Coef.	Std. Err.	dy/dx	Std. Err.
Firm size (dummies, reference group: >=5,000 employees)								
<10 employees	**-1.259*** **	0.297	**-0.238*** **	0.045				
<200 employees	**-1.446*** **	0.249	**-0.295*** **	0.045				
<1,000 employees	**-1.295*** **	0.303	**-0.233*** **	0.041				
<5,000 employees	**-1.116*** **	0.293	**-0.205*** **	0.042				
>=5,000 employees					**1.297*** **	0.217	**0.307*** **	0.051
Firm headquarters in … (dummies, reference group: South America)								
… Europe	**-1.000** **	0.421	**-0.221** **	0.091	**-1.001** **	0.421	**-0.222** **	0.091
… North America	-0.650	0.426	-0.140	0.087	-0.641	0.426	-0.138	0.088
… Asia or rest of world	**-0.893** *	0.473	**-0.172** **	0.075	**-0.907** *	0.474	**-0.174** **	0.075
Firm age (dummies, reference group: <1 year)								
>=1 year and <5 years	-0.776	0.528	**-0.158** *	0.096	**-0.853** *	0.517	**-0.182** *	0.091
>=5 years and <10 years	-0.827	0.537	**-0.167** *	0.096	**-0.898** *	0.516	**-0.180** **	0.090
>=10 years and <20 years	-0.634	0.537	-0.132	0.102	-0.696	0.514	-0.144	0.096
>=20 years	-0.671	0.549	-0.145	0.0113	-0.682	0.520	-0.147	0.107
Software development is the main business activity of the firm (dummy)	**0.886*** **	0.174	**0.193*** **	0.036	**0.885*** **	0.172	**0.193*** **	0.036
Constant	**1.414** **	0.675			0.135	0.615		
Observations	818				818			
Pseudo R^2	0.09				0.09			
Wald test	$X^2(12)=83.12$, p<0.0001				$X^2(9)=81.72$, p<0.0001			

* significant at 10%, ** significant at 5%, *** significant at 1%

Notes: Significant coefficients are bolded; reported standard errors are robust standard errors.

Descriptive statistics and the correlation matrix of the explanatory variables used are depicted in Table A6.2 and Table A6.3.

Table A6.2. Descriptive statistics of the explanatory variables used in Table A6.1

Variable	Frequency of "0"	Frequency of "1"
Firm size <10 employees	685 (81%)	165 (19%)
Firm size <200 employees	538 (63%)	312 (37%)
Firm size <1,000 employees	747 (88%)	103 (12%)
Firm size <5,000 employees	764 (90%)	86 (10%)
Firm size >=5,000 employees (reference group in regression)	*666 (78%)*	*184 (22%)*
Firm headquarters in Europe	533 (62%)	333 (38%)
Firm headquarters in North America	559 (65%)	307 (35%)
Firm headquarters in Asia or rest of world	782 (90%)	84 (10%)
Firm headquarters in South America (reference group in regression)	*840 (97%)*	*26 (3%)*
Firm age >=1 year and <5 years	662 (80%)	167 (20%)
Firm age >=5 years and <10 years	666 (80%)	163 (20%)
Firm age >=10 years and <20 years	656 (79%)	173 (21%)
Firm age >=20 years	520 (63%)	309 (37%)
Firm age < 1 year (reference group in regression)	*812 (98%)*	*17 (2%)*
Software development is the main business activity of the firm	396 (46%)	473 (54%)

Note: N=869.

Table A6.3. Correlation matrix of the explanatory variables used in Table A6.1

		1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
1.	Firm size <10 employees	1.00											
2.	Firm size <200 employees	n.m.	1.00										
3.	Firm size <1,000 employees	n.m.	n.m.	1.00									
4.	Firm size <5,000 employees	n.m.	n.m.	n.m.	1.00								
5.	Firm headquarters in Europe					1.00							
6.	Firm headquarters in North America		**-0.08**			n.m.	1.00						
7.	Firm headquarters in Asia or rest of world		**0.09**		**-0.06**	n.m.	n.m.	1.00					
8.	Firm age >=1 year and <5 years	**0.26**	**0.18**	**-0.16**	**0.13**		**-0.08**	**0.10**	1.00				
9.	Firm age >=5 years and <10 years		**0.15**	**0.06**	**0.06**	**-0.09**		**0.07**	n.m.	1.00			
10.	Firm age >=10 years and <20 years		**0.11**						n.m.	n.m.	1.00		
11.	Firm age >=20 years	**-0.28**	**-0.34**		**0.16**	**0.12**	**0.08**	**-0.12**	n.m.	n.m.	n.m.	1.00	
12.	Software development is the main business activity of the firm	**0.10**	**0.12**	**-0.12**			**-0.08**	**0.13**	**0.16**	**0.09**		**-0.026**	1.00

Notes: Only correlations with p<0.1 are shown; n.m. = not meaningful because variables are dummy variables coding the same characteristic.

Appendix

Appendix 7: Multivariate analysis of reading firm policies on Internet code reuse

Table A7.1. Multivariate analysis of reading firm policies on Internet code reuse

	Logistic Regression		Logistic Regression	
	Coef.	Std. Err.	dy/dx	Std. Err.
Software development role (dummies, reference group: architect)				
Project manager	0.400	1.170	0.059	0.152
Programmer	**-1.305*** **	0.415	**-0.218*** **	0.066
Analyst	-0.509	0.880	-0.095	0.182
Database developer	**-1.853** **	0.796	**-0.415** **	0.184
Other	-0.243	0.574	-0.042	0.103
License risk level of developer's work	**0.435** **	0.187	**0.071** **	0.030
Developer is working as a freelancer (dummy)	0.344	0.439	0.053	0.062
Developer is working on embedded software projects (dummy)	-0.297	0.373	-0.051	0.067
Developer happiness in job (measured on ten-point scale)	**0.204*** **	0.072	**0.033*** **	0.012
Developer lives in … (dummies, reference group: South America)				
… Europe	-0.158	0.756	-0.026	0.124
… North America	0.062	0.825	0.010	0.133
…Asia or rest of world	-0.401	0.781	-0.070	0.146
Tenure of developer with firm (in years)	0.025	0.033	0.004	0.005
Constant	-0.022	0.996		
Observations	283			
Pseudo R^2	0.13			
Wald test	$X^2(13)=34.97$, p=0.0009			

* significant at 10%, ** significant at 5%, *** significant at 1%

Notes: Developers with the software development role of "tester" are not included in the model since there are only two observations; significant coefficients are bolded; reported standard errors are robust standard errors.

Descriptive statistics and the correlation matrix of the explanatory variables used are depicted in Table A7.2 and Table A7.3.

Table A7.2. Descriptive statistics of the explanatory variables used in Table A7.1

Variable	Frequency of "0"	Frequency of "1"
Software development role *project manager*	772 (96%)	35 (4%)
Software development role *programmer*	394 (49%)	413 (51%)
Software development role *analyst*	791 (98%)	16 (2%)
Software development role *database developer*	790 (98%)	17 (2%)
Software development role *other*	716 (89%)	91 (11%)
Software development role architect (reference group in regression model)	*580 (72%)*	*227 (28%)*
Developer is working as a freelancer	667 (77%)	202 (23%)
Developer is working on embedded software projects	699 (80%)	170 (20%)
Developer lives in *Europe*	412 (47%)	457 (53%)
Developer lives in *North America*	624 (72%)	245 (28%)
Developer lives in *Asia or rest of world*	735 (85%)	134 (15%)
Developer lives in South America (reference group in regression model)	836 (96%)	33 (4%)

Variable	Explanation	Min.	Max.	Med.	Mean	S.D.
License risk level of developer's work	Criticality of license risks in developer's work (1=Low since developer is working on internal-use projects, 2=Medium since developer is working on external projects for only one customer, 3=High since developer is working on projects for multiple external customers)	0.0	2.0	1.0	1.1	0.9
Developer happiness in job (measured on ten-point scale)	Happiness of developer at last employer (1=Extremely unhappy,…, 10=Very happy)	1.0	10.0	8.0	7.3	2.1
Tenure of developer with firm (in years)	Number of years developer has been working as professional software developer for last employer	0.5	40.0	3.0	4.8	6.0

Note: N=869.

Table A7.3. Correlation matrix of the explanatory variables used in Table A7.3

		1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.
1.	Software development role project manager	1.00												
2.	Software development role programmer	n.m.	1.00											
3.	Software development role analyst	n.m.	n.m.	1.00										
4.	Software development role database developer	n.m.	n.m.	n.m.	1.00									
5.	Software development role other	n.m.	n.m.	n.m.	n.m.	1.00								
6.	Developer is working as a freelancer		**0.16**			**-0.14**	1.00							
7.	Developer is working on embedded software projects	**0.06**						1.00						
8.	Developer lives in Europe					**-0.07**	**0.11**		1.00					
9.	Developer lives in North America	**-0.08**	**-0.08**	**-0.07**		**0.10**	**-0.07**		n.m.	1.00				
10.	Developer lives in Asia or rest of world			**0.06**			**-0.06**	**0.09**	n.m.	n.m.	1.00			
11.	License risk level of developer's work					**-0.11**	**-0.10**	**0.12**		**-0.09**	**0.10**	1.00		
12.	Developer happiness in job (measured on ten-point scale)		**-0.28**		**0.08**	**0.14**	**-0.22**			**0.15**	**-0.12**	**-0.07**	1.00	
13.	Tenure of developer with firm (in years)		**-0.08**			**0.09**				**0.08**			**0.09**	1.00

Notes: Only correlations with $p<0.1$ are shown; n.m. = not meaningful because variables are dummy variables coding the same characteristic.