

DOI:10.1145/2043174.2043193

Software developers' reuse of code from the Internet bears legal and economic risks for their employers.

BY MANUEL SOJER AND JOACHIM HENKEL

License Risks from Ad Hoc Reuse of Code from the Internet

REUSING EXISTING SOFTWARE artifacts when developing new software is an attractive way to reduce development costs and time to market while improving software quality.⁴ Code is the artifact most commonly reused in software development.¹⁶ Researchers have identified such reuse in commercial software development as a new facet of software reuse.^{13,22} Here, “Internet code” means code in the form of components (such as a library encapsulating required functionality) and snippets (such as containing a synchronization block) that can be downloaded from the Internet for free and without individual agreement with the originator; an important instance of such code is publicly available open source software (OSS). Internet code generally

includes permission to be reused in commercial software development,¹⁴ making it highly attractive for firms.² Therefore, some firms systematically reuse it by including identification, evaluation, and integration of suitable code in their development processes.¹⁸ Alternatively, Internet code can also be reused in ad hoc fashion, as described in Umarji et al.,²³ with individual professional developers, on their own and typically without telling anybody, searching the Internet for existing code as a shortcut in their work, downloading and integrating it into the software they develop.³

Despite its general suitability for reuse in commercial software, Internet code is rarely in the public domain and usually under licenses that demand compliance with specific conditions as a prerequisite for 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)^b

a Places to search for code include OSS repositories (such as SourceForge.net), code search engines (such as Koders.com), and code bases of related OSS projects; for a detailed overview and quantitative analyses, see Sojer.²⁰

b The GPL is a family of licenses, including versions 1, 2, and 3; since all versions share the “copyleft” obligation, we refer to the whole family as “the GPL” throughout this article.

» key insights

- Professional software developers reuse code freely available on the Internet (such as open source code) in their commercial projects in ad hoc fashion.
- Such code often comes with license obligations; noncompliance can mean legal and economic risk, but developers are often not sufficiently knowledgeable in these matters.
- Firms should establish clear policies regarding reuse, leveraging reliable information resources on the Internet and complementing them with internal training, lobby universities to include the topic in their curricula, and acknowledge the interdisciplinary nature of the issue.



ILLUSTRATION BY METROPOLIS

as the most common license.¹¹ The GPL is an OSS license, requesting that other code tightly integrated with the code it governs is also licensed under its terms.⁹ These terms allow users of GPL-licensed software to access, modify, and redistribute the source code of the software.¹⁹ For firms trying to protect their source code as proprietary intellectual property, complying with this requirement may be difficult. However, firms that integrate code under the GPL into their software without complying with the license terms and are then found out can be legally forced to replace the GPLed code or license the entire program under the GPL. Either option could produce costly legal and eco-

nomic consequences.¹⁹

Other license conditions that can be problematic for firms include reusing the code only in non-commercial settings, only in certain application types, only for a certain period of time, and only when not exporting it to certain geographic locations.^{17,c} Finally, some code available from the Internet does not explicitly spell out license or reuse conditions, though since it is protected by copyright, proper reuse necessitates contacting the creator and other rights holder(s) and asking permission.

When Internet code is reused systematically it seems feasible for firms

to weigh the benefits and risks of reusing and manage potential license issues properly. Yet colloquial evidence of the reuse of Internet code in ad hoc fashion—as opposed to systematic reuse—suggests individual professional software developers do not always address the license obligations of the code they reuse.^{12,15} Thus, while their ad hoc reuse of Internet code might still result in greater effectiveness, efficiency, and quality for their firms, their behavior might also produce legal and economic trouble.

Most previous published research addressing reuse of Internet code is largely theoretical or based on industrial case studies. As an exception, German and various co-authors⁶⁻⁹ quanti-

c Such restrictions are not in OSS licenses.

tatively investigated license issues from OSS code reuse through the analysis of code bases and software distributions.

To complement this work, we employed quantitative data obtained from a global survey we conducted in 2009 involving 869 professional software developers to explore ad hoc reuse of Internet code, with a special focus on license issues. Our findings should provide firms with a starting point for assessing their exposure to license risks from their developers' ad hoc reuse of Internet code and devising measures to avoid potential related liabilities.

Survey

We developed the questionnaire following our literature review and 20 interviews with industry experts.^d Before conducting the survey, we enlisted four academic peers and 113 software de-

velopers to pre-test the questionnaire. We chose a survey-based research approach over an analysis measuring the share of reused Internet code in commercial software code bases. While this setup did not allow us to calculate a precise percentage of reuse of Internet code in commercial software development, it did allow us to include more professional software developers. Moreover, if deviations between developers' actual and survey-reported reuse would arise, they would be unlikely to be systematic and thus should not affect the results of our multivariate analyses.

Since we were among the first to investigate ad hoc reuse of Internet code by individual professional software developers, we opted not to use a limited sample of developers from a single firm but rather a broad and heterogeneous group of professional software developers active in Internet newsgroups as our survey popu-

lation.^e We extracted a total of 93,541 unique email addresses from more than one million messages posted over the previous three years in 528 newsgroups dealing with software development.^f After cleaning the addresses, we selected a random sample of 14,000 addresses and invited the newsgroup participants via email messages to take our online survey. We received 1,133 fully filled-in responses, yielding a response rate of 9.9% (consistent with other Internet surveys).^g Of them, 869 responses were submitted by current or former professional software developers who are the focus of the analyses discussed in the following sections.

The vast majority (98%) of the 869 professional software developers we surveyed was male, with average age 35.6, living in Europe (53%), North America (28%), Asia (12%), and South America (4%); 56% had previously contributed to OSS. At the time of the survey, in 2009, 79% of the developers were employed as professional software developers; the others had been working as professional developers but had quit before 2009.^h On average, survey participants had 9.7 years of work experience as professional developers in 2009, most as programmers (51%), others as software architects (28%) and project managers (4%); 23% were employed as freelancers in 2009, and the others worked on permanent contracts.

Also at the time of the survey, 54% of the developers worked for firms for which software development was the main business, with 68% developing software for external customers, the rest for internal use in their firms. Among the 68% writing software for external customers, 62% were creating off-the-shelf software for multi-

e Potential limitations of our approach are discussed later in the section on threats to validity.

f The 528 newsgroups included all main and high-traffic groups (such as comp.lang.c++ and comp.lang.java.programmer).

g To calculate response rate, we adjusted the number of invitations we sent to potential survey participants by the number of email messages that did not reach their designated recipients.

h In the following sections, we report the characteristics of the last software development activities of developers who quit creating software.

d Full questionnaire available from the authors.

Figure 1. Extent of ad hoc reuse of Internet code, 2009.

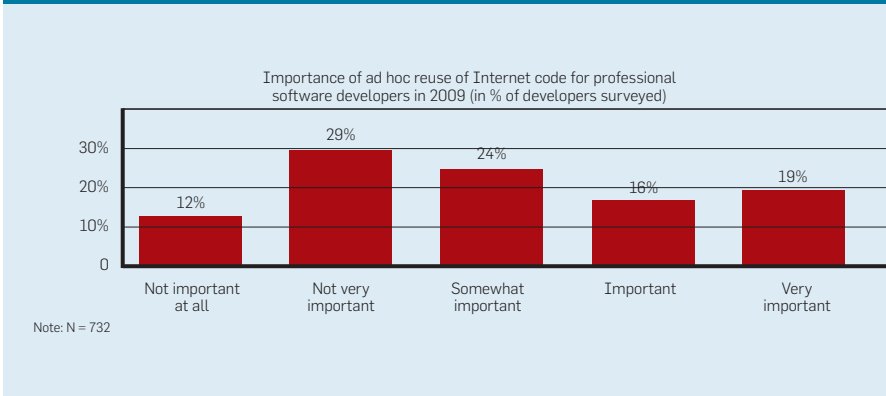
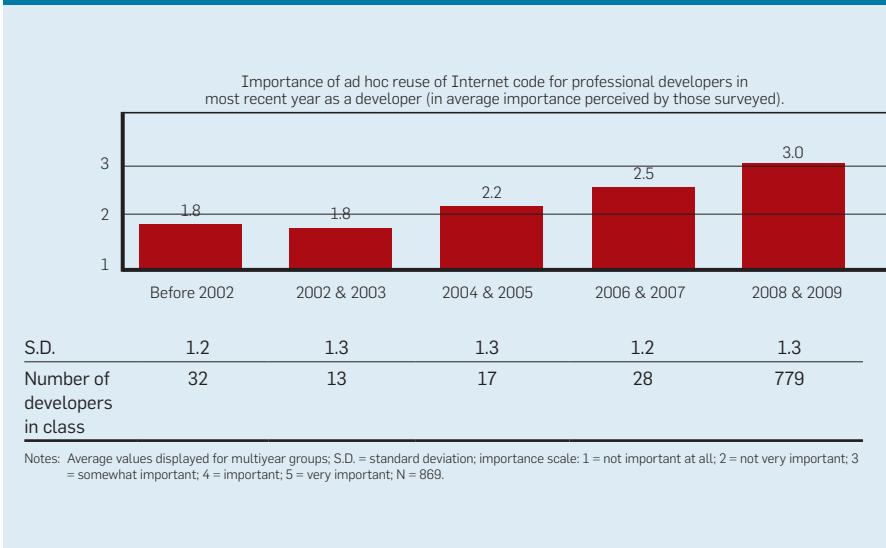


Figure 2. Evolution of extent of ad hoc reuse of Internet code, through 2009.



ple customers, and the rest developed custom software. These distinctions are important because the license risks resulting from reuse of Internet code are typically more severe for software developed for multiple external customers.

Extent of Code Reuse

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

Outlining the perceptions of professional software developers active in 2009, Figure 1 reflects that ad hoc reuse was an essential part of the work of many professional developers. More than half of those we surveyed (59%) considered ad hoc reuse of Internet code at least “somewhat important” for their work, while only 12% apparently did not reuse any Internet code in ad hoc fashion. This finding contrasts with the prevailing assumption of many firms that their code base does not or only to a small, controlled degree contain Internet code.¹⁵

In addition to analyzing the extent of ad hoc reuse of Internet code, we also investigated the historic development of such reuse. Figure 2 includes the perceptions of professional software developers who quit creating software before 2009. Since we asked survey participants about their last year as active developers, their responses are informative about the respective year. Our survey data shows that starting with 2004 the importance of ad hoc reuse of Internet code for professional software developers had 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.

A possible interpretation is that before 2004, code available from the Internet might have only rarely been suited for reuse in commercial software development because it was not mature enough and covered only a few functional areas. However, resulting from the strong recent growth of OSS,³ both the quality and the fields for which code exists should have increased strongly, thus making Internet

code reuse much more attractive to professional developers.

Determinants of Code Reuse

To understand which factors most influence the importance professional

software developers attribute to ad hoc reuse of Internet code we conducted an exploratory regression analysis with the data collected in our survey. The model (see Table 1) employs an ordered logistic regression¹⁰ and the

Table 1. Multivariate analysis of the importance of ad hoc reuse of Internet code.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 in bold type; reported standard errors are robust standard errors. Descriptive statistics and the correlation table of the explanatory variables used are available from the authors on request.

perceived importance of ad hoc reuse for the individual work of professional developers measured on a five-point scale as a dependent variable. As independent variables we included multiple characteristics of professional developers, some as dummy variables. Regression coefficients are not standardized, such that the range or standard deviation of a variable must be taken into account when assessing the variable's effect on the importance professional developers attribute to ad hoc reuse in their work.

First, the model results point out that developers' ad hoc reuse seemed to be independent of the "license risk level"; that is, developers creating software to be sold to multiple exter-

nal customers did not deem ad hoc reuse as less important than developers working on custom software or software for internal firm use. A possible interpretation is that developers, in deciding to reuse Internet code, did not acknowledge the real possibility of negative legal and economic consequences their employers might face due to license violations. However, we can also think of two alternative explanations: One could assume less reusable code was available for internal use or custom software due to its tailored nature; and one could also imagine that while not considering ad hoc reuse less important, professional developers were still more careful when reusing such code in development projects for multiple external customers.

Moreover, developers who never had any training or information on reusing Internet code and thus should be more likely to create license issues

did not differ significantly^j in their view of the importance of ad hoc reuse of Internet code from developers who were trained or had received such information. Also, while developers who self-assessed their knowledge about Internet code licenses better also deemed ad hoc reuse of Internet code reuse more important, this relationship does not hold for an objective assessment of developer proficiency regarding licenses for the code.^k If we (plausibly) assume that the results of our objective assessment are more informative about developers' license-related knowledge than their self-assessment, we can also assume that developers, at least as of 2009, on average did not correctly account for their own knowledge about licenses for Internet code when considering ad hoc reuse of Internet code.

The model also indicates that developers who had been active in OSS projects and those with longer experience as professional developers considered ad hoc reuse significantly more important.^l A plausible interpretation of this finding, consistent with Sojer and Henkel,²¹ is that for OSS-savvy developers, the costs of searching, evaluating, and understanding Internet code should be lower than for developers with less OSS experience. Likewise, more senior developers should face lower costs for reuse due to their typically larger personal networks and reuse experience. The

i We set "license risk level" to 1 if developers were working on internal projects, to 2 if they were working on external projects for only one customer, and to 3 if they were working on projects for multiple external customers.

j Throughout this article, "significant" means "statistically significant."

k This objective assessment of developer knowledge of the obligations of reusing Internet code is based on a five-question quiz in our survey. We developed the quiz following 20 interviews with experts in the reuse of Internet code. The quiz covers five typical scenarios in which professional software developers may violate license obligations when reusing Internet code. 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.

l Note regression coefficients are not standardized. Since "experience as professional software developer" is measured in years, ranging from 0.5 to 45, its effect is comparable in size to the dummy variable "developer has OSS experience." The coefficient of the latter variable is much larger (0.39 vs. 0.017), though its range is much smaller (1 vs. 44.5).

Figure 3. Sources for learning about reuse of Internet code, 2009.

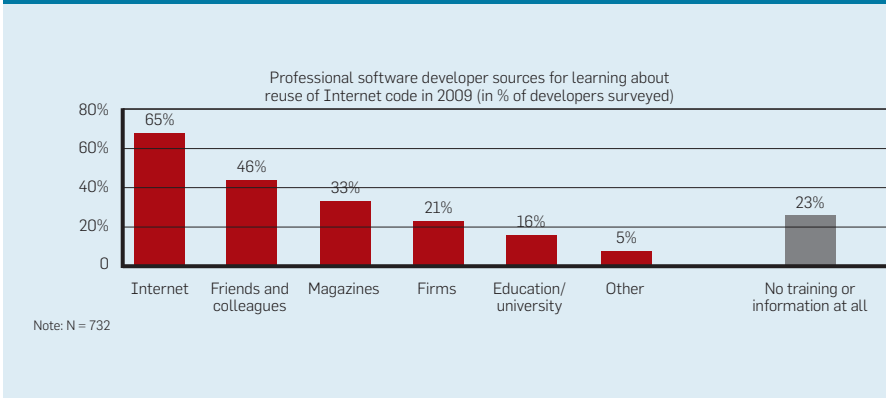


Table 2. Software developer familiarity with license obligations concerning Internet code, 2009.

	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.

multivariate model also supports the result outlined in Figure 2, showing the perceived importance of ad hoc reuse of Internet code grew significantly from 2004 to 2009.

Moreover, the developers we surveyed had different views of the importance of ad hoc reuse of Internet code depending on their development role. Programmers and database developers attributed significantly less importance to it than the architects we defined as a reference group. For all other roles, the difference with the “architects” was insignificant at a 10% level. The finding that architects deemed ad hoc reuse significantly more important than programmers is startling since architects should be concerned with systematic rather than ad hoc reuse. However, architects, especially in smaller and mid-size firms, might also take on programmer responsibilities and leverage their greater architectural latitude to reuse Internet code in an ad hoc fashion. The architecture of a piece of software influences how easy it should be to reuse external code.⁵ Shaping architecture, architects might have more control over reusing Internet code than programmers for whom the architecture of the software they develop is often exogenous. Moreover, greater architectural latitude could also allow developers to integrate Internet code in such a way as to avoid license violations,⁹ assuming developers are aware of the relevant issues in the first place. Supporting this line of thought, our survey found that architects are significantly more knowledgeable regarding licensing topics than other developers, including programmers. Architects should still be able to reuse Internet code properly, while programmers would have to choose between reusing the code in a way that violates the code’s license obligations and not reusing it at all.

The main programming language developers were using influenced how they viewed ad hoc reuse in their work. For example, developers relying mainly on Ruby or Python found ad hoc reuse most important, followed by those working with Perl, Java, PHP, and other such languages. Developers using more traditional programming languages (such as C and C++), less

common ones (such as Visual Basic and C#), and various others formed the last group viewing code reuse as least important.

While one could conjecture that diverse legal systems (such as common law vs. civil law), cultural variations, and the availability of Internet code in local language lead to different views of the importance of ad hoc reuse in different geographies, our survey did not find substantial support for such reasoning; Asian, European, and North American developers did not differ significantly in how they perceived the importance of ad hoc reuse; only South American developers deemed such reuse significantly more important. However, since only 33 South American developers participated in the survey, this finding may not be representative.

Finally, our survey did not find significant differences in professional developers’ perception of the importance of ad hoc reuse based on their education and skills and whether they develop embedded or traditional software or were employed, at the time of the survey, in time-limited contracts (such as freelancers) or as permanent employees.

Developer Knowledge and Risks for Firms

How well are professional software developers prepared to deal with the licenses and obligations associated with ad hoc reuse of Internet code?

It seems reasonable to assume that professional developers who are more aware of the particularities of Internet code (such as its licenses) are less likely to ignore license obligations. Thus, we first investigated whether professional software developers had received training or information on reuse at the time of the survey and the sources of such training and information (see Figure 3).

Two rather informal channels—the Internet (65%) and friends and colleagues (46%)—were developers’ reported main sources of information about Internet code licenses and their particularities. Comparatively unimportant were firms (21%) and educational institutions, including universities (16%). Meanwhile, 23% of the developers we surveyed had not received any form of training or infor-

mation on the reuse of Internet code. Overall, these findings suggest that conveying knowledge about reusing Internet code and potential license risks was not high on the agenda of firms and universities, at least until 2009.

Given the high number of developers surveyed who reported never having received training or information on the reuse of Internet code or who relied on information from unofficial channels (such as the Internet and friends), we were compelled to investigate their knowledge of licenses for such code. When self-assessing their knowledge, two-thirds of surveyed developers reported being “familiar” or “very familiar” with nearly all obligations in Internet code licenses (see Table 2). Contrasting this self-assessment with the results of our five-question quiz about license obligations resulting from the reuse of Internet code (discussed earlier) suggests developers overestimated their knowledge. Even those who viewed themselves as “very familiar” with license obligations on average failed on two questions in our quiz, obtaining a mean score of 3.11 out of a maximum of 5.^m Moreover, while positive and statistically significant ($p < 0.001$), the correlation between self-assessment and quiz score in the survey was weak, with a correlation coefficient of 0.345.

We also sought to identify the factors that influence developers’ objectively assessed knowledge about Internet code licenses and their obligations. The exploratory Tobit¹⁰ regression model (see Table 3) uses developers’ scores in the survey’s license quiz as the dependent variable. The results underscore that developers with OSS experience were significantly more knowledgeable about Internet code licenses than other developers. Furthermore, most forms of training and information about reusing Internet

^m We pre-tested the quiz questions to make sure, as much as possible, they were of comparable difficulty and relevance. However, there was some variation among them, and it is possible that respondents who described themselves as “very familiar” with the obligations of licensing Internet code (and who failed on average to answer 1.89 questions out of 5 in the survey), often struggled with questions on license issues that appear less frequently and are thus less critical for firms.

code (from firms, friends, colleagues, magazines, and other sources) did not exert significant influence on developer knowledge. Developers who had received training or information in educational institutions were significantly less proficient than other developers. Only information acquired from the Internet had a significant positive effect on developer knowledge.

Along with these factors, the developers from Asia and North America seemed to know less about Internet code licenses than their European and South American counterparts in 2009. Regarding educational backgrounds, developers with academic degrees in computer science and engineering were more proficient regarding Internet code licenses than other developers.

In the situation described earlier in which ad hoc reuse of Internet code

seemed prevalent while also exposing firms to risks, it would seem reasonable for firms to introduce explicit policies providing guardrails to developers considering reuse of Internet code.

However, only about one-third of the developers we surveyed worked in firms with policies regulating such reuse. More detailed analysis of this matter emphasizes that firms with more than 5,000 employees were 31% more likely to have such policies, while there was no significant difference among smaller firms of various sizes.ⁿ Moreover, firms for which software development was the main business had a 19% greater probability of having such policies, while firm age had no consis-

ⁿ These findings result from exploratory logistic regression analyses and resulting marginal effects not covered here; full regression tables are available from the authors.

tently significant effect on whether or not a firm had such policies.

Of the developers working in firms with policies regarding Internet code reuse, nearly one-quarter reported not to have read them. Programmers were less likely to have read policies than architects; also, developers unhappy with their jobs were significantly less likely to have read their employers' policies.^o Additionally, developers who were not involved in development projects for multiple external customers were significantly less likely to have read the policies.

As a consequence of the overall situation regarding the ad hoc reuse of Internet code described here, it is not surprising that our survey found that 21% of the developers creating software in 2009 had at least once not checked thoroughly for Internet code license obligations when reusing snippets; 16% did the same when reusing components; and 14% ignored license obligations they were aware of when reusing snippets.

Threats to Validity

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

In terms of internal validity, the explanatory and control variables in our models should ensure no omitted variable biases influence our survey results. However, since our questionnaires were completed anonymously by developers identified through email addresses, we cannot be sure of the accuracy and truthfulness of the answers to our questions.

Regarding construct validity, the main dependent variable of our research is the perceived importance of ad hoc reuse of Internet code for developers' individual work. While this variable is a suitable proxy for the extent to which professional software developers practice ad hoc reuse, fu-

^o These findings result from exploratory logistic regression analyses and resulting marginal effects not covered here; full regression tables are available from the authors.

Table 3. Multivariate analysis of software developer knowledge concerning Internet code licenses.¹⁷

	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
...educational institutions	-0.243*	0.126
...friends and colleagues	0.080	0.112
...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 or 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 in bold type; reported standard errors are robust standard errors. Descriptive statistics and the correlation matrix of the explanatory variables used are available from the authors on request.

ture research might want to take more direct measures to check the robustness of our findings and conclusions. Moreover, despite our extensive pretest with more than 100 developers, it might be possible that some survey participants misunderstood the meaning of some of our survey questions.

Addressing external validity, there is still the risk that our survey population of 869 developers active in Internet newsgroups is not representative of professional developers in general. Since this research is among the first to quantitatively investigate ad hoc reuse of Internet code by individual developers, we deliberately chose 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 developers (such as Alexy¹) gives us confidence in the representativeness of our sample. Still, it would be worthwhile to repeat our study in a more homogeneous single-firm setting.

Conclusion

Our analyses of ad hoc reuse of Internet code in commercial software development suggest its importance has increased over time; in 2009 over 50% of the developers we surveyed deemed ad hoc reuse at least “some-what important” for their own work. This result differs from the prevailing assumption of many firms that their code base does not or only to a small and controlled degree contains Internet code.¹⁵

Addressing the knowledge of professional developers about Internet code licenses and their legal obligations, we found about one-quarter of them had never received any form of training or information on the topic. Only a small fraction had received training or information from firms or from educational institutions. Moreover, many existing forms of training and information were apparently not effective.

As a consequence of this lack of useful training and information, many developers, at least in 2009, lacked detailed knowledge about their obligations potentially resulting from the reuse of Internet code. Despite this, only a minority of firms

had deployed policies addressing reuse of Internet code in 2009. Consequently, a considerable share of developers—14%–21% of our sample, depending on scenario—had at some point either not checked thoroughly for license obligations or even knowingly ignored them when reusing Internet code in the past.

Firms must recognize and acknowledge the existence of Internet code in their own code bases. Given our findings, they should further consider that some of the Internet code reused in their software might also violate license obligations.

Our study offers multiple levers for firms to mitigate the economic and legal risk from ad hoc reuse of such code. First, the topic itself must be positioned more prominently on their agendas. Firms should actively make developers aware of the potential license issues resulting from their reuse of code. They should leverage reliable information resources on the Internet, complementing them with mandatory internal training and other practical information. Second, they should lobby universities and other educational institutions to include the topic in their curricula. Third, they should establish easy-to-understand policies providing guidance as to how to deal with Internet code. Moreover, they need to ensure that developers are aware of these policies and actually read and understand them. Finally, they need to recognize the interdisciplinary nature of license risks from reuse of Internet code relating to developers and engineers, as well as to lawyers.

They should thus facilitate communication between developers and legal experts such that clearance for specific instances of the reuse of Internet code can be obtained quickly. Otherwise, developers would have to choose between practicing reuse on their own or abandoning it altogether, an option that would ignore a valuable source of efficiency and quality gains. □

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 (Jan. 2008), 89–100.
3. Deshpande, A. and Riehle, D. The total growth of open

- source. In *Proceedings of the Fourth International Conference on Open Source Systems* (Milan, Italy, Sept. 7–10). Springer, Boston, 2008, 197–209.
4. Frakes, W.B. and Kang, K. Software reuse research: Status and future. *IEEE Transactions of Software Engineering* 31, 7 (July 2005), 529–536.
5. Gartlan, D., Allen, R., and Ockerbloom, J. Architectural mismatch: Why reuse is still so hard. *IEEE Software* 26, 4 (July/Aug. 2009), 66–69.
6. German, D.M., Di Penta, M., and Davies, J. Understanding and auditing the licensing of open source software distributions. In *Proceedings of the 18th IEEE International Conference in Program Comprehension* (Braga, Portugal, June 30–July 2). IEEE Press, Los Alamitos, CA, 2010, 84–93.
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 *Proceedings of the Sixth IEEE International Workshop on Mining Software Repositories* (Vancouver, Canada, May 16–17). IEEE Press, Los Alamitos, CA, 2009, 81–90.
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 *Proceedings of the Fifth International Conference on Open Source Systems* (Skövde, Sweden, June 3–6). Springer, Boston, 2009, 185–198.
9. German, D.M. and Hassan, A.E. License integration patterns: Dealing with license mismatches in component-based development. In *Proceedings of the 31st IEEE International Conference on Software Engineering* (Vancouver, Canada, May 16–24). IEEE Press, Los Alamitos, CA, 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 (Apr. 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 (Aug. 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 (Mar. 2009), 80–87.
14. Madanmohan, T.R. and De, R. Open source reuse in commercial firms. *IEEE Software* 21, 6 (Nov. 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 (Nov. 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 (Apr. 2002), 340–357.
17. Murray, G.F. Categorization of open source licenses: More than just semantics. *Computer & Internet Lawyer* 26, 1 (Jan. 2009), 1–11.
18. Norris, J.S. Mission-critical development with open source software: Lessons learned. *IEEE Software* 21, 1 (Jan. 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, 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 (Dec. 2010), 868–901.
22. Spinellis, D. and Szyperki, C. How is open source affecting software development? *IEEE Software* 21, 1 (Jan. 2004), 28–33.
23. Umarji, M., Sim, S.E., and Lopes, C. Archetypal Internet-scale source code searching. In *Open Source Development, Communities and Quality*, B. Russo, E. Damiani, S. Hissam, B. Lundell, and G. Succi, Eds. Springer, Boston, 2008, 257–263.

Manuel Sojer (sojer@wi.tum.de) is a management consultant at Bain & Company, Munich, Germany. This article is part of his Ph.D. work completed in 2010 at Technische Universität München, Munich, Germany.

Joachim Henkel (henkel@wi.tum.de) is a professor of technology and innovation management at Technische Universität München, Munich, Germany.