Decision Criteria for Software Component Sourcing
Steps towards a Framework

Rob J. Kusters, Lieven Pouwelse, Harry Martin and Jos Trienekens
Faculty MST, Open University, Valkenburgerweg 177, Heerlen, The Netherlands

Keywords: Make-or-buy, Software Components, Open Source, Closed Source, in House Development.

Abstract: Software developing organizations nowadays have a wide choice when it comes to sourcing software components. This choice ranges from developing or adapting in-house developed components via buying closed source components to utilizing open source components. This study seeks to determine criteria that software developers can use to make this choice. Answering this question will result in a list of criteria that can, after further validation, be used to develop structured decision support in this type of decision. A first step is a literature search resulting in an initial list. Since the literature used was not specifically targeted at the question at hand, it was decided to separately conduct interviews to obtain an independently derived list of criteria. In a second part of the interview the respondents were confronted with the list resulting from literature. Together this resulted in a preliminary proposal for decision criteria for software sourcing.

1 INTRODUCTION

Delivery in time and within budget of (business) software that meets the functional and quality requirements is often a challenge. Component-based software development is often used to deal with this challenge but the selection of appropriate software components then becomes an important decision (Jha et al., 2014). A component can be defined as a coherent package of software that can be independently developed and delivered as a unit, and that offers interfaces by which it can be connected, unchanged, with other components to compose a larger system (D'Souza and Wills, 1997). When developing component-based software, an organization nowadays has a wide choice of sourcing options. The main choices are:

- In-house development
- Re-use (possibly with adaption) of earlier in-house developed components
- Acquisition of commercial components
- Usage of open source components
- Adaption of open source components.

Choosing between these options is not obvious (Cortellessa et al., 2008). However, we were unable to find a good overview of criteria that could be used for such a decision. In this paper we propose a first attempt at filling this gap. A two-fold approach is taken. First, in a literature survey we try to identify a basic list of criteria. After this we conducted a series of interviews with experienced software developers and managers without using the results obtained from literature. From this, a list of criteria derived from practice is extracted. In a second part of the interviews, the results from literature are discussed explicitly. We expect that the results can be used as the basis for the development of structured decision process support for sourcing software components.

In section 2 related work is discussed. The methodology used in the research is described in section 3, and execution of the research and the results in section 4. The paper ends with conclusions and a discussion of results in section 5.

2 RELATED WORK

A significant body of literature is already available on management of software development in general. However most of the literature found is only indirectly related to software component sourcing. No specific literature on software component sourcing decision criteria was found. We did find however literature on relevant aspects, focusing at the basic make-or-buy decision, at the consequences of organizing re-use of in-house developed components, or on the advantages and disadvantages of using open

Kusters, R., Pouwelse, L., Martin, H. and Trienekens, J.
Decision Criteria for Software Component Sourcing - Steps towards a Framework.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 580-587
ISBN: 978-989-758-187-8

source which will be a topic of discussion in this article as well.

Morisio et al., (2002) shows the main issues of a 'make or buy' decision. They also state that each product variant has its specific considerations regarding appropriate requirements, risks and costs. Cortellessa et al., present a framework supporting the choice between selection of commercial component software and development in-house (Cortellessa et al., 2008). Daneshgar et al have examined, in relation to the 'make or buy' decision on the basis of 10 existing decision criteria, additional criteria that affect small and medium businesses (Daneshgar et al., 2013). Boehm and Bhuta (2008) also identified advantages and disadvantages of commercial off-the-shelf products in their study. The choice between the use of existing software components rather than to fully develop in-house is often made implicitly. Also, in projects that have been studied, it is implicitly expected that the development time and effort required can be reduced by making use of software components. However, convincing evidence is not yet available (Morisio et al., 2002). They also note that when using commercial off the shelf products new types of activities and their associated costs have to be taken into account (Morisio et al., 2002). The suitability of a component-based software system is highly dependent on the architecture of the system. Consistency and coupling play an important role in determining the quality of the system in terms of reliability and the effect of the component on the maintainability and availability of the system as a whole. This is also an area that should be taken into account when making sourcing decisions (Jha et al., 2014).

From the point of view of re-use as a sourcing option, re-use has the potential to shorten lead times, improve quality and reduce development costs. The studies conducted by (Lim, 1994) and (Kakarontzas et al., 2013) suggest that reuse of software results in higher quality. Also (Favaro et al., 1998) indicate that the economic benefits of software reuse are substantial. However, the reuse of software has shown to be challenging for many organizations on both a technical and organizational level (Kakarontzas et al., 2013). Results from the study of (Lim, 1994) are broadly consistent with research from (Kakarontzas et al., 2013). Lim mentions the following arguments for reuse: reduced delivery times; lower development costs and higher quality by fixing bugs in the product, but balances this with the need for sufficient funding for developing, maintaining and keeping components for re-use available. Frakes (2005) further elaborates the organizational issues that need to be dealt with for facilitating reuse.

Also, work is available focusing on the adoption and the adaption of open source components. Such components have numerous benefits including free customizable source code. On the other hand, the use of (open source) software components may present various challenges concerning selection, testing and integration. If a system is being distributed or sold, it is e.g. important that a component with an appropriate license is selected (Chen et al., 2007). Ruffin and Ebert (2004) also state that, depending on the product, use, and market conditions, certain open source properties may be advantageous. One example is the existence of a large user community which results in a de-facto standard. Additionally, like (Chen et al., 2007) they emphasize the importance of adhering to license conditions. There seems to be general disagreement on the added quality open source can provide. The study by (Paulson et al., 2004) suggests that defects are generally found and resolved faster in open source than in closed source software. Ruffin and Ebert (2004) argue that open source software may increase security. However, (Schryen and Kadura, 2009) state that this conclusion requires further research, since a solid basis for this conclusion has yet to be established. For users of commercial off the shelf software components, it is more difficult to track changes than for open source software users. Open source users are also more concerned about the reputation of their support provider (Li et al., 2006).

In the related field of package software selection a good overview is provided by (Jadhav and Sonar, 2011) which gives an interesting and very possibly relevant overview of package selection criteria. However, the field is sufficiently distinct to prevent us in this stage from accepting these results as-is. They can however be looked into in a further stage of the research.

All together we see significant contributions, often focusing on specific but related aspects of the sourcing criteria issue without, as of yet, resulting in a well-structured overview of criteria. Based on this conclusion, we decided to investigate sourcing criteria in a dedicated study.

3 METHODOLOGY

We started with a literature search using relevant combinations of the search terms “advantages, disadvantages, open source software, closed source software, software components, software component

selection, software reuse, and software 'make or buy'". We used the generic search engine scholar.google.com and also the following online databases:

- ACM Digital Library
- IEEE Digital Library
- JSTOR Business, Biological, Mathematics & Statistics Collection
- ScienceDirect (Elsevier)
- SpringerLink

Relevancy of papers was assessed first on title and abstract. Resulting interesting papers were then investigated in detail. Selected papers were also used as the basis for a further reference based search (forward and backward) to find additional related papers. In total 94 papers were selected of which after further study 11 were eventually used.

The results of the literature search were not convincing. Meaning that many criteria had to be derived from a literature base that was not specifically written for our purpose. So although the first resulting list of criteria might look plausible, we felt it had insufficient justification to serve as the sole basis for this research. Straightforward validation of such a list (e.g. in a survey) could provide information on the relevance of items already on the list, but it would be unlikely to lead to adding missing items to this list. To substantiate our first impression of the literature search and to gain insight into the way in which these and local criteria are being interpreted in practice, we opted for an in-depth case study.

Since both relevance and completeness are relevant objectives for such a type of research, we decided on a twofold approach. In an interview, first an open part took place aimed at independently identifying a set of criteria that can provide a reference set for discussion and valuation of the set derived from literature. This was followed by a second, semi-structured part. Here, explicitly based on the list resulting from literature, we made a first attempt to assess the potential relevance of the literature set.

The choice was made for open in-depth interviews since we felt the questions were too complex to allow sufficient quality results and to provoke sufficient response from a survey. We felt the disadvantage of limited participation was off-set by the depth and quality of the results which we could expect from in-depth interviews.

We looked at a single organization where component based development had been in use for several years and where the sourcing decision is therefor made routinely and where alternate sourcing options are considered. The organization is a provider of e-commerce applications and web applications for SME's and (semi-) government organizations. The organization was founded 16 years ago, and currently comprises 48 employees of which 35 are developers. The organization is relatively young with ages of employees varying from 20 to 40 years.

All five sourcing options identified above are standard practice in this organization. However, the organization does not have a formal policy regarding software component sourcing decision criteria. Therefore, we expected that developers and managers are forced to contemplate the sourcing decision regularly, resulting in the building up of experience. In a sense, they can be considered as an expert group. We expected this would give us a wider and well informed range of answers. Since the organization had no formal policy, documentation was unlikely to provide relevant information. This also explains our reliance on interviews.

To constrain the interviewees into the sourcing decisions they actually make, rather than to trigger unsubstantiated perceptions and opinions, we focused at the decisions that had been made in the recent past on three specific projects. Within this organization we strived for maximum variation to promote diversity of results.

Thus recent projects were selected representing all the different sourcing options. The projects were each taken from different departments within the organization, to further increase the potential diversity of answers. Similarly, per project different stakeholders were interviewed, to account for role-based bias. Stakeholders having a role in sales, project management and software development, were selected. To increase response quality even further, only staff members with at least three years of experience were interviewed.

Within this setting a detailed design of the two parts of the interview was developed. During the first part, an open interview was conducted where the participants were encouraged to recollect the arguments actually used within the specific projects. The respondents were not shown the results of the literature study to prevent any unintentional bias. Respondents were asked to identify the components of the project. For each component identified they were asked:

- Were in your opinion other alternatives available?
- Did any colleague suggest other alternatives?
- On the basis of what criteria did you choose this option?

The interviews were recorded and crucial parts were transcribed. Using NVivo (Bazeley and Jackson, 2013) the results were organized and labelled. Subsequently the results were compared by the researchers with the list of criteria from the literature search, and matches and mismatches have been identified, of which the latter resulted in the identification of new additional criteria

The second part of the interview was more structured. The basis of this part of the interview is the list of criteria found in the literature. The goal is to see if these criteria identified in literature have been used or could have been used in practice. Also in this interview the relation with actual decision making practice will be maintained, so questions are again aimed at actual experience. Per criterion the following questions were asked:

- Have you used this criterion in an earlier decision?
- If yes:
  - Can you indicate what project this was?
  - To what extent has this criterion really contributed to the decision?
- If no (we did ask for opinions here):
  - Is this a plausible criterion?
  - Can you think of a project where this criterion would have relevant?

For the second part of the interviews transcription was not deemed necessary. Based on the recording, the discussions were sufficiently structured and clear.

The choice was made to do both parts of the interview in a single session. This had as an advantage that people remembered better what they said before and were therefore better able to connect what was mentioned in the first part, to the second part. This strengthened the results. It was also done for pragmatic reasons. It was easier to get participation this way. A drawback of course was that newly identified criteria could not now be tested across the participants.

Internal validity is fostered by a careful research design. Respondents were carefully selected and treated with respect. They were informed on the purpose of the project and were told their input was voluntary, would be treated anonymously and that they could, at any time, refuse an answer or stop their participation. They were also given the option to check our recordings and interpretations derived from their interview. Respondents were informed in advance about the purpose of the research and were also provided with definitions of the sourcing options. This allowed them to prepare the interview and also can prevent misunderstanding as to the object of discussion. This will increase the quality of the information obtained, and thus the validity of the research.

External validity is obtained by the 'factual' context maintained throughout the interviews. Results will show that in the particular organization some criteria have actually been used in the sourcing decision. Naturally, this does not imply relevancy for each and all other software organizations. But it does show that experienced practitioners have found them useful, hinting that others may value the use of explicit component sourcing criteria as well.

Reliability is again supported by the careful design of the interviews. This resulted in the development of an extended interview guide that allowed to a large degree repeatable interviews.

4 EXECUTION AND RESULTS

The literature study resulted in a list of 26 criteria (see table 1).

We selected three recent (within the last year) projects intending to cover all types of sourcing identified above. They were:

- P1: an e-commerce solution based on an internally developed e-commerce platform that uses open source software components and recycled in-house developed software.
- P2: an e-commerce solution based on the open source platform that uses open source software components, adapted open source software components, and in-house developed software.
- P3: an internal application framework that uses open source software components, closed source software components and in-house developed software.

Respondents were asked to identify the components in each project. Examples of components mentioned were Magento and Wordpress. This proved to be more complex than originally expected, resulting in some differences in components identified between the respondents. For project P1 the respondents identified three, five, and five components, resulting in the discussion of twelve components. For P2 the numbers were five, seven, and "two + others", also resulting in the discussion of twelve components. For P3 finally, the numbers were eight and nine resulting in discussion of fourteen components.

Table 1: Results from literature.

ID	Criterion
L01	Because the source code is publicly available the risk of stopping vendor support is reduced because there a possible to switch to another supplier (Ruffin and Ebert, 2004)
L02	Developing an application on a de facto standard API protects the application against changing supplier conditions (Ruffin and Ebert, 2004)
L03	The risk of having to provide compensation to the licensor for the breach of license, patent or proprietary rights (Ruffin and Ebert, 2004)
L04	The number of interactions between different components (Jha et al., 2014)
L05	The scale and complexity of software component (Daneshgar et al., 2013)
L06	Appropriate requirements - the extent to which the component standard meets user needs (Daneshgar et al., 2013)
L07	The number of discovered vulnerabilities (Schryen and Kadura, 2009)
L08	Lead time required to fix discovered vulnerabilities (Ruffin and Ebert, 2004)
L09	Reliability - maturity, fault tolerance and recoverability (Lawrence, 1996)
L10	Maintainability - analyzability, changeability, stability and testability (Lawrence, 1996)
L11	Effect of the software component on the availability of the system as a whole (Daneshgar et al., 2013)
L12	Flexibility in the use of the component (Daneshgar et al., 2013)
L13	Delivery time (Lim, 1994)
L14	Development costs (Lim, 1994)
L15	Life cycle / maintenance costs (Boehm and Bhuta, 2008; Favaro et al., 1998)
L16	The number of functional additions per release (Paulson et al., 2004)
L17	Freedom to adapt code (Chen et al., 2007)
L18	License of the component (Chen et al., 2007)
L19	Intellectual property (Daneshgar et al., 2013)
L20	Government requiring usage of specific accounting software (Daneshgar et al., 2013)
L21	Wish to maintain a broad technical vision across the entire product (Frakes, 2005)
L22	Wish to use knowledge and business expertise efficiently across projects (Frakes, 2005)
L23	Desire to systematically manage parts which allow flexible reaction to changing market conditions (Frakes, 2005)
L24	Availability of capable staff for development (Lim, 1994)
L25	Maintaining and keeping available reusable software components (Lim, 1994)
L26	Available financial means to organize re-use (Frakes, 2005)

An option here could have been to provide the component structure as an input for the interviews. However, in that case respondents could have been confronted with components they are not really familiar with. In many cases this would have resulted in additional answers. This would have decreased the reliability of the answers given. The results confirm that this indeed occurred during the interviews, with actually surprisingly little overlap between the components identified. This we feel, justified our design decision.

For each project, respondents were selected according to the roles specified above. Project P3 was an in-house project, so no related sales representative was available. Projects P1 and P2 were managed by the same project manager. This person was interviewed twice for the first part, once for each of the projects. The second part naturally only needed to be carried out once. In total, this resulted in eight interview results for part one and seven for the second part.

At this stage, we had the choice between compromising on the number of respondents or on the diversity of sourcing in the projects. We opted for an optimal diversity of sourcing options, feeling that sufficient interviews were left to give valid and reliable results.

The respondents had on average 8.2 years of experience of which 6.7 in their current organization, providing a solid basis of experience.

Table 2: Addition criteria found.

ID	Criterion	#
P01	experience with the software component within the organization	5
P02	availability of documentation	1
P03	interoperability and compatibility with plug-ins and / or frameworks	5
P04	the wish of the customer	6
P05	expected life of the software component	2
P06	software component is widely accepted by the community	4
P07	evaluation of the software component by the community	1
P08	Connect with market demand / increase commercial opportunities	3

For each interview we reserved four hours in a meeting room, so as to have sufficient time and to avoid being disturbed. On average, part one of the interview took slightly over half an hour, while the second part on average lasted for an hour. With some time required for the introduction and small rests between parts 1 and 2 and sometimes halfway part 2, the average duration was less than two hours. The

respondents had sufficient time to answer questions fully, contributing to the reliability of the answers.

The additional criteria found in the first part of the interviews can be found in table 2. In the column '#' is indicated the number of respondents who identified this criterion without prompting.

Of the 26 criteria identified in literature, eleven were also confirmed in this first part of the interviews. In table 3 the column 'part-1'shows the number of respondents that mentioned criteria (and an associated example from the project) that could be mapped to this list.

Table 3: Results interviews.

ID	Part-1	Part-2	
		based on usage	based on opinion
L01		3	5
L02			7
L03		3	3
L04	1	4	3
L05	3	5	1
L06	3	7	
L07		1	
L08			3
L09	3	6	1
L10	2	6	1
L11	3	5	2
L12	4	7	
L13		2	4
L14	6	6	1
L15	2	5	1
L16		3	2
L17		2	4
L18	5	6	1
L19		1	2
L20		2	5
L21		7	
L22		7	
L23		3	4
L24	2	4	3
L25		6	1
L26		5	

The second part of the interviews only looked at the criteria derived from literature, so no additional confirmation could be obtained for the criteria P1-P8. The results of the second part of the interviews can be found in the column 'part-2' table 3. The column 'based on usage' shows the number of respondents that recognized a criteria as one they had actually used in the past. An additional thirteen criteria from literature were confirmed here. In all cases an actual example was given by the respondents, demonstrating factual knowledge rather than speculation.

We also asked for opinions of respondents in case no actual usage took place. If they had not actually used the criterion they were asked if they found it plausible. The number of respondents who agreed with this can be found in the column 'based on opinion' of table 3.

When discussing criterion L7 (the number of discovered vulnerabilities) no fewer than five respondents stated that instead of the number of vulnerabilities the criterion should in fact consider their (potential) impact. One respondent out of these also provided an example of usage of this criterion in a recent project. This resulted in an unexpected ninth additional criterion:

P09: Impact of discovered vulnerabilities.

5 DISCUSSION AND CONCLUSIONS

In this paper we described research aimed at identifying criteria to support software component sourcing decisions. The literature study resulted in 26 potential criteria. Some criteria were mentioned by several authors, but in principle we saw limited overlap between the authors. This did not inspire confidence as to the completeness of this list. A more complete list would have shown more overlap.

This triggered design of an independent investigation, based on a series of in-depth interviews. Here experts from practice were asked, based on a recently completed project, to indicate criteria used in their sourcing decisions. This resulted in the identification of nineteen criteria, of which eleven could be matched to the list derived from literature and eight were new additions. A ninth addition emerged later from de interviews.

When discussing the quality of the resulting list of criteria we can first look at completeness. Naturally, the current list may be quite incomplete and further research is needed to establish a more complete list of commonly useful criteria. Nine new additions to a list based on the experience of just a single company does suggest that saturation has as yet not been achieved. We are likely to find more when more companies are included in the research.

On the other hand, by combining literature and practice in this way it would seem that at least the most obvious, and maybe then also the most important criteria, will have been identified. It must

be noted that the research conducted only looked at relevance, not at degree of importance.

Apart from completeness we predominantly looked at relevance of the criteria that were identified. There a more positive picture emerges. The nine new criteria have been found without prompting and have been used in practice for a concrete sourcing decision within the target organization. That implies that they are relevant and other organizations can consider using them.

Likewise, eleven criteria emerged from the interviews, which could be easily mapped on the literature. For these a similar degree of confidence can be expressed. Again these were found without prompting and have already been used in a sourcing decision practice. And they also have backing from literature.

In the second part of the interviews we used the list resulting from literature as input to the interviews. Out of the fifteen remaining criteria, for thirteen criteria examples were provided that they had been used in an actual decision process. The evidence can be considered slightly less strong since the respondents required prompting for these criteria but still examples of usage could be given. It is reasonable to conclude that these criteria are also relevant.

That leaves two criteria for which no actual usage could be identified. However, L02 (Developing an application on a de facto standard API protects the application against changing supplier conditions) was seen by all seven respondents to be a plausible criterion nonetheless. This remarkable consensus gives no evident reason to dismiss this criterion. L08 (lead time required to fix discovered vulnerabilities) is also confirmed three times. All in all, there are reasons to qualify the entire result as at least 'plausible'.

Furthermore, the initial list presented in this study is rather unrefined and needs additional processing. Many criteria are overlapping and differ in the level of abstraction and aggregation. E.g. criterion P04 rather broadly states the importance of "customer wishes". This is a more abstract formulation of the very specifically formulated L20 (Government requiring usage of specific accounting software). L07, L08 and P09 all somehow focus on vulnerabilities. L03 and L18 both consider license issues. Because of this, the current set of criteria cannot be seen as a set of independent criteria. Some further classification is required. We decided against doing so for the results of the literature study for two reasons. One because the number of criteria resulting was manageable and the other because we did not want to run a risk of changing information by our interpretations. The current list can be classified further, but we decided to wait till additional criteria have been identified.

Obviously, further research will be needed to further validate this set of criteria and to add more results and insights from practice. An ongoing effort is required to discover more potentially useful criteria, which may hopefully result in some sort of saturation. After that the resulting list can be classified in a more coherent and manageable form.

There is also the interesting aspect of (relative) degree of importance of criteria. This is probably very much context dependent and therefore local assessment will be needed to make a "common criteria list" operational in decision making practices in software component sourcing. This would open up a new line of research in which the decision making process of the way in which software components are sourced comes into focus.

REFERENCES

Bazeley, P., and Jackson, K. (Eds.). 2013. *Qualitative data analysis with NVivo.* Sage Publications Limited.

Boehm, B., and Bhuta, J. 2008. Balancing opportunities and risks in component-based software development. *IEEE Software*, 25 (6), 56-63.

Chen, W., Li, J., Ma, J., Conradi, R., Ji, J., and Liu, C. 2007. A survey of software development with open source components in China's software industry. *Lecture Notes in Computer Science,* Vol. LNCS 4470, pp. 208-220.

Cortellessa, V., Marinelli, F., and Potena, P. (2008). An optimization framework for "build-or-buy" Decisions in software architecture. *Computers and Operations Research*, 35 (10), 3090 - 3106.

D'Souza D. F. and Wills A.C., 1997. *Objects, Components, And Frameworks with UML – the Catalysis Approach*, Addison-Wesley, Reading, Mass.

Daneshgar, F., Low, GC, and Worasinchai, L. 2013. An investigation of "build vs. buy "decision for software acquisition by small to medium enterprises. *Information and Software Technology*, 55 (10), 1741-1750.

Favaro, JM, Favaro, KR, and Favaro, PF. 1998. Value based software reuse investment. *Annals of Software Engineering,* 5 (1), 5-52.

Frakes, WB 2005. Software reuse research: status and future. *IEEE Transactions on Software Engineering*, 31 (7), 529-536.

Jadhav AS, and Sonar RM 2011. Framework for evaluation and selection of the software packages: A hybrid knowledge based system approach. *Journal of Systems and Software*,. 84 (8) 1394-1407.

Jha, PC, Bali, V., Narula, S., and Kalra, M. 2014. Optimal component selection based on cohesion and coupling

for component-based software system under build-or-buy scheme. *Journal of Computational Science*, 5 (2), 233-242.

Kakarontzas, G. Constantinou, E., Ampatzoglou, A., and Stamelos, I. 2013. Layer assessment of object-oriented software: A metric facilitating white-box reuse. *Journal of Systems and Software*, 86 (2), 349-366.

Lawrence, S. 1996. Software Quality: The Elusive Target. *IEEE Software*, 13 (01), 12-21.

Li, J., Conradi, R., Slyngstad, OPN, Bunse, C., Torchiano, M., and Morisio, M. 2006. An empirical study on decision making in off-the-shelf component-based development. *Proceeding of the 28th International Conference on Software Engineering - ICSE '06*, 897-900.

Lim, W. 1994. Effects of reuse on quality, productivity, and economics. *IEEE Software*, 11 (5), 23-30.

Morisio, M. Seaman, C., Basili, V., Parra, A., Kraft, S., and Condon, S. 2002. COTS-based software development: Processes and open issues. *Journal of Systems and Software*, 61, 189-199.

Paulson, JW, Succi, G., and Eberlein, A. 2004. An empirical study of open-source and closed-source software products. *IEEE Transactions on Software Engineering,* 30 (4), 246-256.

Ruffin, C., and Ebert, C. 2004. Using open source software in product development: A Primer. *IEEE Software*, 21 (1), 82-86.

Schryen, G., and Kadura, R. 2009. Open source vs. closed source software: towards measuring security *Proceedings of the 2009 ACM Symposium on*, 2016 - 2023.