Decision Criteria for Software Component Sourcing
An Initial Framework on the Basis of Case Study Results

Jos. J. M. Trienekens[1], Rob J. Kusters[1], Jan van Moll[2] and Casper Vos[3]
[1]*Faculty Management, Science and Technology, Open University, Valkenburgerweg 177, Heerlen, The Netherlands*
[2]*Philips Health Tech, Veenpluis 4-6, Best, The Netherlands*
[3]*Faculty of Industrial Engineering, Eindhoven University of Technology, The Netherlands*

Keywords: Decision Criteria, Software Components, Sourcing, Framework.

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 aims at structured decision support in this type of decision. As a basis for this study an initial set of criteria is taken, that has been identified and validated in a particular software development environment in a previous study on the subject (Kusters et al, 2016). In the paper at hand we report on the results of the application and validation of the initial set of sourcing criteria in a completely different case study environment, namely a software environment in that. medical embedded software is being developed. In addition, and based on the outcomes of our case study, a further step is made towards structured decision support for sourcing decisions by the development of an initial sourcing criteria framework.

1 INTRODUCTION

Companies who develop software have a lot of challenges nowadays. "They need to deliver software in time, within the budget, and within the quality and functional requirements" (Kusters et al, 2016). The traditional way of software development is not suitable for the development of large scale and complex systems (Jha et al., 2014). Component-based software development is often used to deal with this challenge but the selection of appropriate software components then becomes an important decision. A component could 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 large system (D'souza & Wills, 1997). Companies who develop component-based software have several options when it comes to software development sourcing. In a previous research project we identified 6 possible sourcing options (Kusters et al, 2016):

- In-house development
- Buying of the shelf software components
- Buying software components according to specific requirements
- Re-use of software components
- Using open source software in the software components
- The use of adjusted open source software in the software components.

Choosing between these sourcing options is not easy, a software development team has different interests concerning to the software development. For example, a software developer could base a decision on meeting the functional and non-functional requirements, where a software manager could base a decisions on meeting the project budget (Cortellessa et al, 2008).

In this paper we elaborate on a list of 34 possible decision criteria as identified in previous research (Kusters et al., 2016). These criteria were obtained by literature research and were validated in a case study in an e-commerce application and web application development company. Obviously, further research was needed to further extend and validate this set of criteria and to add more insights from practice. We believe that more research effort in different types of organisations is needed to discover more potentially useful criteria, and to strive at a set of criteria with a higher level of saturation. On that basis next steps can

Trienekens, J., Kusters, R., Moll, J. and Vos, C.
Decision Criteria for Software Component Sourcing - An Initial Framework on the Basis of Case Study Results.
DOI: 10.5220/0006294302790286
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 2*, pages 279-286
ISBN: 978-989-758-248-6

be taken towards advanced structured support for sourcing decisions.

The purpose of our new study is three-fold. First we will check the usage of the sourcing options as give in the foregoing, and subsequently we will investigate the sourcing criteria that are being used in a type of software environment which is completely different from the software environment in that we previously carried out a case study. The software environment in this study is a company which develops medical embedded software. Second we will (eventually) elaborate our initial list with the newly identified sourcing criteria, and we will apply and validate the resulting list in a case study. Finally, we will develop an initial framework of sourcing criteria to support to support the sourcing decision making process.

In section 2 related work will be discussed which consists mainly of the results of our previous study and a summary of its theoretical background. The methodology used in the research is described in section 3, and the execution of the research and the results in section 4. The paper ends with conclusions and a discussion in section 5.

2 RELATED WORK

In previous studies we found that although there is already a significant body of literature available on management of software development in general, most of the literature is only indirectly related to software component sourcing (Kusters et al, 2016). Sometimes the importance of particular open source characteristics is stressed (Ruffin and Ebert, 2004), such as the adherence to license conditions. In other research the quality of open source components is discussed as being an issue in sourcing decisions. Ruffin and Ebert (2004) argue that open source software components may increase security. But others stress the reputation of an open source provider can play a role (Li et al., 2006), or the uncertainties for the developers, e.g. regarding the need for (near future) customization of components, or their implementation complexity (Xin and Levina, 2008), (Benlian and Hess, 2011). In (Kusters et al, 2016) a first attempt has been made towards a structured overview of criteria. Based on a structured literature review and a subsequent in-depth case study a in a particular software development company a list of 34 criteria has been developed, see Table 1.

Table 1: Sourcing criteria (Kusters et al, 2016).

ID	Criterion
L01	Because the source code is publicly available the risk of stopping vendor support is reduced because it is possible to switch to another supplier
L02	Developing an application on a de facto standard API protects the application against changing supplier conditions
L03	The risk of having to provide compensation to the licensor for the breach of license, patent or proprietary rights
L04	The number of interactions between different components
L05	The scale and complexity of a software component
L06	Appropriate requirements – the extent to which the component standard meets user needs
L07	The number of discovered vulnerabilities
L08	Lead time required to fix discovered vulnerabilities
L09	Reliability – maturity, fault tolerance and recoverability
L10	Maintainability – analysability, changeability, stability and testability
L11	Effect of the software component on the availability of the system as a whole
L12	Flexibility in the use of the component
L13	Delivery time
L14	Development costs
L15	Life cycle / maintenance costs
L16	The number of functional additions per release
L17	Freedom to adapt code
L18	License of the component
L19	Intellectual property
L20	Government requiring usage of specific accounting software
L21	Wish to maintain a broad technical vision across the entire product
L22	Wish to use knowledge and business expertise efficiently across projects
L23	Desire to systematically manage parts which allow flexible reaction to changing market conditions
L24	Availability of capable staff for development
L25	Maintaining and keeping available reusable software components

Table 1: Sourcing criteria (Kusters et al, 2016) (cont.).

L26	Available financial means to organize re-use
L27	Experience with the software component within the organization
L28	Availability of documentation
L29	Interoperability and compatibility with plug-ins and / or frameworks
L30	The wish of the customer
L31	Expected life of the software component
L32	Software component is widely accepted by the community
L33	Evaluation of the software component by the community
L34	Connect with market demand / increase commercial opportunities

Although we believe that the list contributes significantly to the software sourcing decision research area, we are convinced that further research is needed. For instance the validation of the list of criteria in other software development environments than in our previous study, and the development of a framework of sourcing criteria to support structured decision making are considered as interesting challenges. Therefore we decided to carry out an in-depth case study in an embedded software development environment, to validate the list of criteria and to develop an initial framework of sourcing criteria.

3 METHODOLOGY

To extend our insight into the way sourcing criteria are being interpreted and used in practice, and to strive towards further decision support for software sourcing, we opted just as in our previous study, for an in-depth case study. We selected a single organization, which we will call Company X, where component based development is in use already for years, where sourcing decisions are made routinely and where alternate sourcing options are considered. Contrary to the company in our previous study, Company X does not work with small projects of which the products are delivered to a customer. At Company X medical embedded software development is carried out in large evolutionary software development projects.

In our case study in Company X we will use a two-fold inductive approach, followed by a design step towards an initial sourcing criteria framework. First, explorative unstructured interviews are carried out to get an overview of the software development processes, and to get insight in the process aspects where sourcing decisions are made. These open interviews should eventually also lead to the identification of particular (new) sourcing criteria being used at Company X. Second, semi-structured interviews are carried out to discuss separately and in detail an extended list of sourcing criteria, i.e. consisting of the list of 34 criteria from previous research as well as the newly identified criteria. In these interviews we will investigate in particular the usage and the importance of the sourcing criteria at Company X. During the interviews the experts will be asked to recall projects and process aspects, in which sourcing decisions have been made. The disadvantage of limited participation will be off-set by the depth and quality of the interviews. For the interviews highly skilled and experienced experts from the software development environment are selected, who have an overview of the processes, and who had expertise in sourcing decisions. For the open (unstructured) interviews, as well as the semi-structured interviews, five interviewees were selected, respectively a software integrator, a software manager, two software architects, and a group leader. Together these experts cover with their expertise and knowledge all of the aspects of the software development processes.

In the open (unstructured) interviews, the following questions are central: which software sourcing options are used at Company X (see the list of six options given in the Introduction)? Is there a standardized process for the software development sourcing decisions? And: what are important decision criteria when choosing for a particular sourcing option? The unstructured interviews are also used to check if there were corresponding documents which could give indications for particular decision criteria. Participants were encouraged to recollect arguments which they actually used in their specific projects. The respondents were not shown the results of the initial list of criteria to prevent any unintentional bias.

The purpose of the semi-structured interviews was to validate the findings of the unstructured interviews, e.g. the usage of the additional criteria, as well as the list of decision criteria from the previous study (Kusters et al., 2016), but also to investigate the possibility of ranking the decision criteria on importance. For each decision criterion the following questions were asked: can you remember that this decision criterion was taken into account when making a software sourcing decision? Could you tell me what kind of project this was? To what extend (low, medium, high) has this criterion contributed to

the final choice for a sourcing option? Can you mention a software sourcing decision where this criterion should have been taken into account? In these semi-structured interviews, interviewees could ask further explanation and give additional information to clarify their answers. At the end of the interview they were asked if they could look again through the list and if they could tell if something triggered them, which could lead to insights that the researcher was not yet aware off. So, in these interviews the relation with actual decision making practice was maintained, because questions were aimed at actual experience.

In the design step of this study, a Metaplan approach will be followed to develop a sourcing criteria framework. Metaplan was developed in the early 1970s by several researchers to improve classifications of (in)dependent issues collected from business situations (Howard M, 1994). The Metaplan method is a card sorting method where, during a structured meeting, cards are sorted into clusters. In our Metaplan session each card will represent a particular identified criterion of our extended list of sourcing criteria, and the clustering process should lead to an initial sourcing criteria framework to support sourcing decision making.

Regarding the quality of our case study we will reflect on its validity (internal, construct and external) and reliability (Yin, 2014). 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 (see the six options given in the Introduction). 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, just like the organization in our previous study, 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 in a different software environment, experienced practitioners have found them useful, hinting that others may value the use of explicit component sourcing criteria as well. Reliability is also supported by the careful design of the interviews. This resulted in the development of an extended interview guideline that allowed to a large degree repeatable interviews.

4 EXECUTION AND RESULTS

In the unstructured interviews the five interviewees were asked which software development sourcing options are used at Company X. In all of the interviews the same five sourcing options were found, which were the first five given in the list of options in the Introduction section in this paper. The interviewees not only recognized these five sourcing options, but also had experience in choosing them. The sixth possible sourcing option: "the use of adjusted open source software in software components" is not used at Company X. The interviewees had the following reasons for this. "If we adjust open source software we have to really understand the code. This will cost a lot of time of the available software engineers which in turn will cost a lot of money. That will cut the main advantages of the use of open source software, which are speed and cost reduction". Another reason for not using this option was: "Using open source could have licencing risks which are checked by the IP&S (Intellectual Property & Software) department who decide if the software could be used or not. If we adjust open source software and use it in a commercial product it could lead to the risk that we have to publish the adjusted open source software". This will limit strongly the benefits of this option of adjusting open source software.

All the respondents were asked who is responsible for the sourcing decision and how this process is executed. From these open interviews eight new sourcing criteria were identified, see Table 2. It became also clear during the interviews that the sourcing decision process is quite implicit: "If such a decision takes place, an engineer has to fill out a form with certain conditions which the proposal has to meet. This form has to be approved by the software manager and the lead software integrator". Unfortunately, investigated filled-in forms did not contain much criteria on which sourcing decisions were based. Only one of decision criterion could be taken from these forms: "Previous development choices", see P02 in Table 2.

Table 2: New identified criteria (results from the unstructured interviews).

ID	Criterion
P01	Physical size of the hardware
P02	Previous development choices
P03	Corporate requirements
P04	Regulatory requirements
P05	Reliability of the vendor
P06	Security of the vendor
P07	Competitive reasons
P08	Performance hardware

The newly identified sourcing criteria will be clarified in the following.

P01 Physical size of the hardware
This criterion was considered as a possible decision criterion because of the fact that embedded software is being developed. In this type of software development there is always the possibility that certain hardware is needed to test the software. Some parts of this hardware may be very large, which may influence the sourcing decision, since not all software development vendors can handle such hardware.

P02 Previous software development choices
This criterion was gathered from documentation provided during one of the interviews, as: "it can influence the choice of a next software sourcing decision". If software was already used from a particular software vendor, and additional software was needed, the same software vendor was chosen rather than choosing in-house development (or another vendor).

P03 Corporate requirements
Two of the interviewees indicated that some of the software was outsourced internally. The clinical platform department designs certain software components, which are used in other medical devices next to the medical devices. "The Company X design department determines how the user interface will look. They will provide toolkits to realise the design which they created". As such the two departments are dealing with a corporate policy.

P04 Regulatory requirements for software (for example: IEC62304)
Company X is producing medical devices for humans. The IEC62304 is an example of a regulation for medical software. Two interviewees stated the importance of regulatory requirements.

P05 Reliability of the software vendor
Making a software development outsource decision will come with certain risks: "When we outsource the software development, the reliability and maintainability is something what should be considered. Bankruptcy of a software vendor could be a big risk for Company X when outsourcing is done".

P06 Security of the vendor software archive
Company X has to protect their intellectual property. "When a software component belongs to our core business, it will be made in-house". Another interviewee: "It is important to ensure the selection process of software vendors is good, we need to be able to trust the software vendor".

P07 Competitive reasons
Two interviewees stated respectively: "From idea to market is a critical decision criterion", and: "If we do not have the correct knowledge to create a certain software component, it could cost a lot of time to make it in-house. A software vendor with the right competency to create the software component leads to a big time advantage". And a third interviewee: "The demand of healthcare will increase over the next five to ten years, which creates competitive issues".

P08 Performance within hardware constrains
The medical devices which are created by Company X have non-functional as well as functional requirements: "Some of the software has to produce images in combination with the hardware within nanoseconds". Another interviewee: "It could be the case that a software vendor can deliver a software component that needs extra available memory. Therefore, interactions between the software components and the hardware, need to meet the requirements but within the hardware constraints".

The additional decision criteria P01 to P08 and the 34 decision criteria of the initial list (see Table 1) are subsequently applied and validated in the semi-structured interviews. As described in section 3 the interviewees were asked whether the decision criteria had been taken into account during a project, based on usage (see first column in Table 3). If this was not the case the interviewees were asked if they thought (i.e. their opinion, see the second column in Table 3) whether this decision criterion could be useful for their projects. In the semi-structured interviews, the interviewees were also asked to what extend (High, Medium, Low) a decision criterion contributed to the final choice for a sourcing option. The latter findings are presented in the three columns to the right in Table

Table 3: Decision criteria: usage, opinion and level of importance (H=High, M=Medium, L=Low).

ID	Usage	Opinion	H	M	L
P01	1	1			2
P02	3	1	1	2	
P03	4		4		
P04	4		2		2
P05	4		3	1	
P06	3	1	4		
P07	4		4		
P08	4		1	1	2
L01	4			3	1
L02	4		1	2	1
L03	4		4		
L04	2	1		2	1
L05	4		1	2	1
L06	4		2	2	
L07	4		2		2
L08	4		2		2
L09	4		4		
L10	4		4		
L11	4		3		1
L12	3		1		2
L13	4		4		

ID	Usage	Opinion	H	M	L
L14	4		4		
L15	4		4		
L16					
L17	3		2	1	
L18	4		3	1	
L19	4			1	
L20	1			1	
L21	2		2		
L22	2		2		
L23	2	1		1	2
L24	4		1	2	1
L25	2	1	1	1	1
L26	3	1	1		3
L27	3	1	1	3	
L28	3	1	1		3
L29	4				
L30	2	1		3	
L31	3	1	1	1	2
L32	3		1	2	
L33	3		1	2	
L34	4		1	3	

3. As shown in the table only decision criterion L16 was not taken into account during a project and was also not considered as useful. All the new decision criteria P01 to P08 from the unstructured interviews, were taken into account by at least one interviewee during a sourcing decision. P02 to P08 were even taken into account as decision criteria by at least three of the interviewees.

Most of the answers on to what extend the decision criterion contributed to a particular sourcing decision confirmed each other. However some of them were opposite, see e.g.: P04, L07, L08, L11, L12, L26, L28. This could have several reasons. For instance it was not always possible to choose for certain projects at Company X where the interviewees participated in, due to the evolutionary type of software development. The interviewees had to recall software development processes where they participated in and those could differ from each other. For some of these differences explanations could be extracted from the additional information given by the interviewees. We will give some clarifications in the following.

Regarding P04 two interviewees said that the extend to that this decision criterion contributes is low for the software development department, since this was handled at other departments such as the procurement department (because of legal agreements). A third interviewee said that it was important to meet the regulatory requirements and the effect of this decision criterion could be high, especially for the people who are managing this. So, it could affect the sourcing decision in an earlier stage of the software development process and indirectly it could have a high impact on their sourcing decision. Regarding L11 one interviewee stated: “You would not outsource the brain of software, this is made in-house so that the consequences of other software components, which are bought or open source, will not influence the availability of the system as a whole”. Another interviewee approached the decision criterion from another angle: “We learned from the past were we had some problems with software we bought from software vendors. Now most of the bought software components are delivered as binaries which are easier to implement into the system”. The reason for the differences regarding L12 are quite similar to the reasons regarding L11. The flexibility in the use of the component has, according to two interviewees, a low contribution to the sourcing decision whereas one interviewee thinks it has a high contribution to the sourcing decision: “There are a lot of guidelines defined, so this is taken care for us. So for us it will have a low impact on the sourcing decision”. Another interviewee said: “The right requirements are very important so if the flexibility in

the use of the component is not correct this could lead to another sourcing choice. So the impact could be high". So, often the differences in answers could be explained by the different viewpoints of the interviewees, and this counts also regarding L07, L08, L26 and L28. At the end of the interviews each of the interviewees were asked if they can look through the criteria list to see if some decision criteria or possible sourcing options triggers them into some additional information which was overseen during the interviews. The four interviewees agreed on both the sourcing options and the presented list of decision criteria as being useful for making software sourcing decisions. One interviewee came up with an extra sourcing option: "Maybe cooperation is also a possible sourcing option. It is possible that companies of other markets have certain software knowledge which Company X does not possess. In this case it could be possible to set up a cooperation between companies to create software components".

In the final design step of this study the Metaplan method was applied to develop an initial framework of sourcing criteria. The Metaplan session has been conducted with a research group of three experts, respectively one from the company and two from the research institute, and additionally one research assistant from the institute. The result is shown in Table 4. During the session some clusters were merged. One large cluster of various software component criteria (right column in Table 4) was divided over more specific clusters but also some clusters remained unchanged, such as vendor related criteria. For various reasons some criteria cards were removed to be placed in another cluster, or to form a new cluster. For instance, some criteria which were first placed in Software component criteria clusters, appeared to fit better in an independent Architecture and governance cluster, see Table 4. Criteria in this cluster address architectural issues such as 'availability of the system as a whole' (L11), or governance issues such as 'desire to systematically manage parts which allow flexible reactions to changing market conditions' (L23). In a final group discussion a small number of criteria were rearranged. The definitions that were added to the clusters were discussed by the research group and revised were necessary. Because the set of clusters stabilized, as is shown in the initial Framework in Table 4, no additional Metaplan sessions were held. The framework reflects currently eight categories of sourcing criteria. Although the framework has some weaknesses, e.g. a cluster with only one criteria (Hardware related, see Table 4), and a mixed cluster of management related criteria such as time and costs (upper left in the Table 4), also interesting findings can be reported. Here, we mention in particular the 'equal' division in the framework between the clusters of criteria that are directly related to software components (right column), and the clusters of criteria that are related indirectly to software components (left column). And also that the total number of criteria (42) is rather equally spread over the two columns (19 in the left column, 23 in the right one). This could be interpreted as a need in practice for a combined management and engineering view on defining and applying software sourcing criteria. Further, we consider the cluster 'Software component internal criteria' as an important (and large) cluster. This cluster reflects in particular various (complementary) software component quality issues that can play a role in sourcing decisions, such as reliability (L09), maintainability (L10) and flexibility (L17). Of course our initial framework also has clusters that should be investigated further, both regarding their content and criteria definitions. For instance the position of the cluster 'Software component legal criteria' in the framework is questionable. Further research should clarify the engineering and management aspects of these criteria, e.g. the existence of a license of a component (L18), and the risk of having to provide compensation to a licensor (L03). Summarising we believe that our initial framework is a next step to structured decision support in software component outsourcing. The framework gives a structured and inclusive overview of criteria to base sourcing decisions on.

Table 4: The initial framework of sourcing criteria.

Time, costs, knowledge, experience related criteria (L08, L13, L14, L15, L22, L24, L25, L26, L27)	Software component general criteria (P03, P04, L06, L31)
Vendor-related criteria (P05, P06, L01)	Software component external criteria (L20, L30, L34)
Architecture and governance related criteria (L02, L04, L11, L21, L23, L29)	Software component internal criteria (P02, P08, L05, L07, L09, L10, L12, L16, L17, L28, L32, L33)
Hardware related criteria (P01)	Software component legal criteria (P07, L03, L18, L19)

5 DISCUSSION AND CONCLUSIONS

In this paper we reported on the results of a case study aimed at the identification and classification of criteria to support software component sourcing decisions. The in-depth case study has been carried out at a Company X, were embedded software for large complex medical systems is being developed. Starting point, and empirical basis, for our study was a list of 34 sourcing criteria derived from a previous study on the subject (Kusters et al, 2016). In the first part, in open unstructured interviews, a list of six sourcing options has been checked at Company X. One of these options appeared to be non-valid in the target company. In these open interviews, in that the software development processes were addressed, also eight new sourcing criteria could be identified. These new criteria have been found without prompting and they appeared to be used in practice in concrete sourcing decisions. Together with the existing list of 34 criteria the total list of criteria has subsequently been validated in the second part of the study, in more detail in semi-structured interviews. The criteria were confirmed regarding their relevance in practice, by each of the interviewees. From the 34 criteria of our previous study the interviewees also confirmed the usage of 33 decision criteria in their particular embedded software development environment. In the semi-structured interviews also the degree of importance of the decision criteria has been investigated, with respect to the extent to that a decision criteria contributes to a sourcing decision. Various clarifications from experts, to the degree of importance of criteria, showed interesting examples of the different viewpoints at sourcing criteria in practice. When discussing the quality of the resulting list of 42 criteria we looked at completeness. Eight new additions to a list of 34 criteria (from a previous study) does suggest that saturation of sourcing criteria has not yet been achieved. We are likely to find more when more, and different types of, companies are included in the research. On the other hand, by combining the findings, from our previous study with the findings from this case study (in a completely different software development environment), we believe that a next step has been made towards the identification of important criteria.

In the third part of the study the list of sourcing criteria has been elaborated towards an initial framework of criteria. The Metaplan method approved to be useful to develop this framework in an efficient way. The framework, with its eight clusters of sourcing criteria, almost equally spread over what could be called ‘management’ and ‘engineering’ clusters, offers a structured overview of sourcing criteria. The clusters reflect also some coherence between particular criteria, and the cluster titles point to a particular type of sourcing criteria. Although some framework aspects, such as the overlap between criteria and clusters and the differences in level of abstraction and aggregation, need to be elaborated further, we are convinced that our initial framework is a next valuable step towards support for decision making in software component outsourcing.

REFERENCES

Benlian, A. and T. Hess, Comparing the relative importance of evaluation criteria in proprietary and open-source enterprise application software selection – a conjoint study of ERP and Office systems, *Information Systems Journal*, 21/6, November 2011.

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.

Howard, M. S. (1994). *Quality of Group Decision Support Systems: a comparison between GDSS and traditional group approaches for decision tasks.* Eindhoven: Eindhoven University of Technology.

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.

Kusters, R. J., Pouwelse, L., Martin, H., & Trienekens, J. (2016). Decision Criteria for Software Component Sourcing - Steps towards a Framework. In *Proceedings of the 18th International Conference on Enterprise Information Systems* (pp. 580–587).

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

Yin, R. K. (2014). *Case Study Research Design and Methods*.

Xin, M. and Levina N., Software-as-a-Service Model: Elaborating Client-side Adoption Factors. *Proceedings of the 29th International Conference on Information Systems*, Paris, France, December 14-17, 2008.