<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper published in *IEEE Transactions on Software Engineering*. This paper has been peer-reviewed but does not include the final publisher proof-corrections or journal pagination.

Citation for the original published paper (version of record):

Petersen, K., Badampudi, D., Ali Shah, S M., Wnuk, K., Gorschek, T. et al. (2017)
Choosing Component Origins for Software Intensive Systems In-house, COTS, OSS or
Outsourcing?: A Case Survey
IEEE Transactions on Software Engineering, 39(12)
<https://doi.org/10.1109/TSE.2017.2677909>

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:bth-15909>

Choosing Component Origins for Software Intensive Systems: In-house, COTS, OSS, Outsourcing or Services? – A Case Survey

Kai Petersen ^{*,a}, Deepika Badampudi^a, Syed Muhammad Ali Shah^b, Krzysztof Wnuk^a, Tony Gorschek^a, Efi Papatheocharous^b, Jakob Axelsson^b, Séverine Sentilles^c, Ivica Crnkovic^c, Antonio Cicchetti^c

^aBlekinge Institute of Technology, Karlskrona, Sweden

^bSICS Swedish ICT AB, Kista, Sweden

^cMälardalen University, Västerås, Sweden

Abstract

Background: When choosing components for software intensive systems decisions are made on multiple levels. On the highest level, the component's origin or source is chosen (e.g. In-house versus Components Off-The-Shelf), which answers the question where to get a component from. After deciding the origin, the vendor or supplier has to be chosen, followed by choosing the actual component and how to implement and/or integrate it. While vendor and component selection have been intensively explored, the trade-off between component sourcing options is not well investigated.

Objective: In this study we investigate how practitioners choose among the component sourcing options with respect to stakeholders involved, criteria used for making the decision, and the decision making method used. Moreover, we focus in this work on determining the outcome of the decision making with respect to the chosen component sourcing options, the effort invested in decision making, the criteria being most important in the final decision, and the evaluation of the perceived success degree of the choice.

Method: The case survey method has been used to elicit and synthesize the cases. The case survey method is combines the advantages of surveys (capturing a larger set of cases) and case studies (capturing rich information about each case). Overall, 22 decision cases have been obtained. Descriptive statistics, open coding of qualitative information, and odds ratio have been used to analyze the data.

Results: Decisions were prepared in a group of people, while the actual decision was individually made by managers or decision makers. Cost, performance, and system reliability stand out as the most common criteria. More than two decision criteria are frequently considered, which has to be taken into account when supporting sourcing decisions. The final decision was perceived negatively in nine cases and positively in seven cases, while in the remaining cases it was perceived as neither positive nor negative.

Conclusions: There exists a mismatch between industrial processes and the solution proposed with respect to number of considered criteria and methods utilized for decision making. Furthermore, the need for decision support is emphasized due to the low number of cases where the decision made was perceived as positive.

Keywords:

Decision making, In-house, COTS, OSS, Outsourcing, Services;

1. Introduction

Decision making of which component for software-intensive systems and where to get them from is a complex process on several levels. On the highest level, it is needed to choose whether to develop the components internally (in-house development) or to obtain them externally, from henceforth referred

to as the component sourcing option (CSO). Multiple CSO options are possible, such as obtaining off-the-shelf components (COTS), components developed open source (OSS), to outsource the development to a supplier, or to obtain an external service. After deciding on the sourcing option, a supplier needs to be chosen. Thereafter, a choice for the actual component has to be made when the supplier has several offers. From a research perspective the choice of suppliers and actual components is largely investigated, while *how the trade-off between alternative sourcing options (In-house, COTS, OSS, Outsourcing and Services) is made in the industry is not well explored*. The component sourcing option decision is of strategic nature and has implications for properties (such as performance, reliability, and security) [1, 2, 3]. Beyond that, it has implications on lead-time, cost and effort [1, 2], the level of control over the

*Corresponding author

Email addresses: kai.petersen@bth.se (Kai Petersen *), deepika.badampudi@bth.se (Deepika Badampudi), syed.shah@sics.se (Syed Muhammad Ali Shah), krzysztof.wnuk@bth.se (Krzysztof Wnuk), tony.gorschek@bth.se (Tony Gorschek), efi.papatheocharous@sics.se (Efi Papatheocharous), jakob.axelsson@sics.se (Jakob Axelsson), severine.sentilles@mdh.se (Séverine Sentilles), ivica.crnkovic@mdh.se (Ivica Crnkovic), antonio.cicchetti@mdh.se (Antonio Cicchetti)

evolution of a component [4], or the effect on the development lifecycle (such as how to test the system).

Five CSOs are commonly considered in the literature: in-house developed components, components off-the-shelf (COTS), open source components (OSS), outsourced development of components, and obtaining services. Moreover, the literature suggests three ways of supporting this decisions: (cf. [5]): (1) Models to support the decision making, such as optimization models to maximize the reliability with constraints on time and cost [6, 7, 8]; (2) Proposals of criteria that have to be considered [1, 9, 10]; (3) Comparisons of the alternative CSOs with respect to [10, 11, 12, 13] decision outcomes such as time, cost, and effort. We argue that, beyond these contributions, other aspects play an important role. This includes an understanding of how industry makes the decisions related to CSOs, and what were the outcome and impact of the decisions made. Furthermore, if we would understand how industry makes CSO related decisions, we could reflect to which extent the models proposed from the academic side are applicable in the industrial context.

In this paper we provide a comprehensive analysis of 22 cases about *how decision making took place when choosing among CSOs for adding components in industrial software-intensive systems*. We identified the CSOs considered, the stakeholders involved, the criteria considered, and the actual decision taken. Furthermore, reflections on the decision reached were obtained, such as whether the “right” decision was reached. From a research perspective it is important to know how practitioners make the decisions related to CSOs in order to address industry relevant problems. For example, the number of criteria considered may influence the solutions needed to support CSO decision making. In addition, practitioners can use the findings of this study to reflect on their own decision situation with respect to CSOs. Thus, this study can be characterized as descriptive [14].

The research method used was case survey [15]. The cases were gathered in the context of the ORION project¹. The results from this study bring an important cornerstone towards the goal of the project to develop a decision making framework and knowledge base to facilitate the choice of component origins in practice. Hence, the project includes experts with experiences in sourcing decision making and close connections to industry where decision making experience related to sourcing decisions was available. The cases were collected based on the researchers’ experiences in industrial settings (11 cases) and based on interviews with experts from companies (11 cases).

The remainder of the paper is structured as follows. Section 2 presents the related work. Section 3 describes the research method used. Section 4 explains the results, followed by their discussion (Section 5). Section 6 concludes the paper.

2. Related work

Decisions about selecting components are made on multiple levels. Figure 1 depicts the different levels at which decisions

are made. On the top (strategic) level, the CSO is selected. On the provider selection level, depending on the option, either vendors, suppliers, or communities may be chosen. Finally, on the lowest level a concrete component is chosen. Note that we do not imply a particular order in which the decisions are made. For example, one may be aware of one particular COTS component, but then trades that off with an outsourcing decision where the supplier has not yet been chosen. We discuss the choice of CSOs based on a systematic literature review [5] with similar research questions as those raised in this case survey.

2.1. Choosing among CSOs

In an earlier work we conducted a systematic literature review [5] on how to decide among CSOs. In particular, the systematic review investigated decision criteria, methods for decision making, and evaluations of decision outcomes. All of these are also subject of this case survey, though with a focus on industrial cases.

Criteria for decision making: Decision criteria represent the elements to be considered when making trade-off between the component sourcing options (CSOs). Table 1 provides an overview of the quality criteria that are considered in the literature [1, 9, 10, 11, 12, 13]. The recent literature review provides the following important aspects relevant for this study.

Methods for decision making for CSOs: To decide between in-house and COTS development, the most common solutions reported in literature are simulation models [6, 7, 16, 10, 3]. As an example, an optimization model may be used to make a choice of components to maximize reliability, while minimizing delivery time. Different constraints can be defined, such as costs. For making a trade-off between in-house development and outsourcing Kramer and Eschweiler [17] propose to utilize clustering to group components and utilize requirements dependencies and priorities to make the choice. Kramer et al [18]. propose utilizes outsourcing potential, knowledge specificity, and interdependencies to make a decision utilizing decision tables.

Evaluation of decision outcomes for CSOs: As input to the decision making, the knowledge of how a choice influences the criteria is important. Consequently, a number of studies [1, 2, 19, 20, 21, 22, 23, 3, 24, 4] elaborate on the effect of the choices.

Figure 1: Decision Levels when Choosing Components

¹<http://orion-research.se>

For example, with regard to contextual factors Conradi et al. [1] highlight the benefits of COTS over OSS with regard to the market and vendor and community support. On the other hand, OSS has advantages over COTS with regard to licence and source code availability. Looking at the quality criteria, COTS is preferred over OSS in relation to performance and modifiability as well as reliability, while there are shortcomings in regard to security. The shortcomings of COTS for security are also highlighted by [2]. When comparing in-house development with OSS, Norris [3] indicates that OSS is preferred over in-house developed software. Regarding project metrics, Sharifah and McBride [4] account for possible time savings.

Our previous systematic review [5] has shown that the total number of studies investigating decisions on the CSO level is limited. Only a few case studies are presented. Hence, this case survey complement existing works by adding the insights from 22 additional cases to the body of knowledge on choosing among CSOs.

2.2. Choosing Vendors and Suppliers

Vendor and supplier selection has been considered in several studies. A secondary study has been presented by [25], which identifies the success factors in the selection of offshore outsourcing vendors. In total 22 factors were identified from the primary studies. Cost-saving, skilled human resource, appropriate infrastructure and quality of products and services were among the factors that were identified in more than 50% of the primary studies. Studies [26, 27] present findings related to vendor selection and community selection, respectively. The relationship between component integrator and external vendor (COTS vendor and OSS community) is considered important to minimize the time and effort on technical, legal and business negotiations [26]. In addition, the collaboration between the OSS developers and active users is considered important to maintain good communication and relationship with the OSS community.

2.3. Choosing Components

Multiple studies have investigated component selection, such as [28, 29]. The best practices for COTS selection in literature and in industry were identified by Rikard et al. [28] and

three COTS selection methods (Comparative evaluation process, COTS-based requirements engineering and framework of COTS selection) were compared by Wanyama and Far [29]. Risk factors considered during component selection, integration and maintenance were identified by Morandini et al. [30]. The risk factors were related to ill-estimation of selection, integration and maintenance effort, wrong component selection, component integration and maintenance failure. In addition, legal risk with respect to intellectual property and license were identified.

3. Method

We first present the research questions, and thereafter provide details on the case survey method and how it has been used in this study. We utilized the guidelines by Larsson [15] during the design of the research.

3.1. Research Questions

The first research question was concerned with how CSOs were selected in the context of software-intensive system development. In order to determine how the decisions were made, several sub-questions needed to be answered:

- *RQ1: How are CSOs chosen?*
 - *RQ1.1: Which CSOs were considered (In-house, COTS, OSS, Outsource or Services)?* The first sub-question provides an insight of the decision making problem formulated by the companies. Knowing the combinations frequently considered by companies helps to guide future research, in particular it shows which CSOs should be compared empirically so that companies may use this information as input to their decision making process.
 - *RQ1.2: Which stakeholders were involved in the decision process?* Whether a decision is taken individually or in a group is an important aspect when designing decision support for CSOs. Also, understanding the different roles involved in the stages of

Table 1: Decision criteria

Criteria	Description
Performance	System performance in terms of response time in relation to system resources
Maintainability	Ease to evolve (update, bug-fix) the system
Reliability	System reliability (e.g. measured by Mean Time To Failure, and other reliability measures, such as reliability growth)
Security	Security of the system (protection of data and resources of systems)
Quality	Software quality in general, attribute not specified.
Time	This refers to the lead-time of development in terms of duration. Example: using COTS may allow to shorten the development time by 80 % (from 100 days to 20) in comparison to write the component from scratch
Cost	Monetary cost for the CSO
Market and contract	These are properties related to being able to access the component (source code availability) and people to handle the component (internally, e.g. in relation to availability of experts, and externally, e.g. community support), and controlling its evolution (e.g. control on stability and system evolution).
Access and control	Access and control refers to source code availability, availability of experts to support the usage of the component, control on stability and system evolution, development uncertainty, community support, and organization.
Component usage in system	This category refers to issues of actually using the component in the system, such as ease of use, compatibility, dependencies to other components, and the system structure in relation to cohesion and coupling affecting the development option choice.
Component history	This category relates to the history of the component in terms of maturity expressed in its change history.

decision making provides practitioners with the possibility to reflect on whether the identified roles are also relevant in their decision scenarios.

- *RQ1.3: Which criteria were considered for making the decision?* Similar to RQ1.1, the criteria show which variables need to be studied when comparing CSOs empirically.
- *RQ1.4: Which decision making approach/model was used?* In the literature several methods are proposed (such as optimization models). Understanding the methods used in practice allows to determine whether the solutions proposed in research found their way into practice, and which methods used in practice need to be integrated into decision support systems.

The second research question aims at understanding the outcome of the decision making process:

- *RQ2: What were the decision outcomes?*

- *RQ2.1: Which CSOs among those considered (RQ1.1) were chosen?* We investigate whether trends are visible of one CSO being preferred over another, which provides early indications of preferences between CSOs with respect to their advantages and disadvantages.
- *RQ2.2: What was the effort invested in the decision making process?* When looking at solutions in software engineering (in this case how CSOs are chosen) the cost factor has to be considered. Thus, we investigated the estimated effort spent on decision making.
- *RQ2.3: Which criteria initially considered (RQ1.3) ended up as significant for the final decision?* Criteria considered in the preparation for the decision making, though not considered as relevant in the final decision, point to a potential for improvements in decision making processes. Thus, it is of interest which criteria are actually those that were essential in the final decision.
- *RQ2.4: Were the CSOs chosen considered the “right” choice retrospectively?* In this question we reflect on the selected options for making a decision from the retrospective point of view: finding that decisions are mostly positive indicates a success of CSO decision practices. Negative results on the other hand point to the need for decision support and adjustments in the practices.

3.2. The case survey method

We provide an introduction to the case survey method as it has not been widely applied in the software engineering context.

Studies often report only a small number of cases in a single publication. On the other hand, surveys focus on a large number of data points and are mostly quantitative. The case survey method is a compromise of the two approaches [15], as Larsson

points out “it can overcome the problem of generalizing from a single case study and at the same time provide more in-depth analysis of complex organizational phenomena than questionnaire surveys” (cf. [15], p. 1566). According to Larsson there are multiple benefits of using the case survey method. As cases are synthesized the case survey adds value to previous individual cases, and the richness of case studies can be incorporated to draw conclusions in the quantitative synthesis. A data extraction form is used for extracting data from the cases, hence it is possible to easily extend the case survey by adding further cases. Overall, Larsson concludes that the case survey is a means to bridge the gap between positivist approaches (such as surveys) and humanistic/interpretivist approaches (such as case studies).

The process of the case survey method comprises of four different steps:

1. Select the cases of interest.
2. Design the data extraction form for elicitation
3. Conduct the coding
4. Use statistical approaches to analyze the coding

The process steps as executed in this study are outlined in Sections 3.2.1 to 3.2.4.

3.2.1. Step 1: Select the cases of interest

The focus of this paper is on choosing CSOs, such as choosing between in-house development versus open source for software-intensive systems. The components chosen should be utilized as parts of the system-intensive system developed. For example, if an automotive component is developed, and the choice is where to obtain the integrated development environment (IDE), this case would not be included in the study. On the other hand, if the focus of the software development is the IDE itself, then this case would be included. The inclusion criteria thus can be summarised as follows:

- The case provides information of how the decision making between at least two CSOs has been taking place where the component should become part of a software-intensive system. For example, a database component becomes part of the system, while the development environment does not.
- The system for which the CSO decision is made is industrial (can involve academics if they are supporting the industry).
- Cases should at least be explicit about the CSOs considered, the persons involved in the decision making process, the CSO chosen, the methods used in decision making, and the criteria used when preparing and making the decision.
- The source (person) reports the case based on his or her own experiences (e.g. as a consultant, participant in the decision making process, etc.), or the case has been elicited from an industry representative through an interview.

The sampling strategy for the reported cases has been convenience sampling. In order to obtain the data, two approaches

were used, namely the experiences from the research participants in the project, and interviews with practitioners (see Table 2). The research participants in the ORION project utilized their experience as well as networks to obtain the cases and filled them in using the data extraction scheme (see Section 3.2.2). As the researchers had access to different networks a diverse sample could be obtained focusing on different domains and including experience from industrial and academic environments. The experiences of the research participants originate from knowledge obtained about relevant cases during former research projects (3 cases), the work done in an own company or work (3 cases), as well as consultancy work (5 cases). The interviewees were identified through the researchers’ networks. The introduced data extraction form (Section 3.2.2) served as the basis for the interviews.

Table 2: Sources for cases

Source	Number of cases	Case IDs
Own experience	11	Cases 1, 2, 3, 4, 5, 6, 7, 8, 9, 16, 19
Interview	11	Cases 10, 11, 12, 13, 14, 15, 17, 18, 20, 21, 22

3.2.2. Step 2: Design data extraction scheme for data elicitation

There is a risk that the participants in the study eliciting the cases misinterpret the items in Table 3. Thus, the data extraction scheme was reviewed by all ORION project participants. The quality of the form and its understandability were essential in the extraction process, and was important for the validity of the results. An initial form was designed by the first author of the study. The co-authors reviewed the form and submitted change requests to the first author, each reviewer could see the comments already submitted. Each review was considered and a rejoinder was written to keep track of changes, including a response motivating the change, and an action specifying what was changed. After no further comments had been received, the form was considered of sufficient quality to start the data extraction.

The initial version of the data extraction form was defined based on the literature on decision making for CSOs (see related work), and ongoing work to create a taxonomy to describe the decision making for choosing among CSOs, the so-called GRADE taxonomy [31]. The taxonomy defines criteria, roles, decision making methods, environments, and decision making goals.

3.2.3. Step 3: Conduct the coding

Not all information in the form needed to be coded. As can be seen in Table 3 many items were either integer values, enumerations, or boolean (present or not present in the case). That is, only items 12, 13, 14, 26, 27, 28, 32, and 33 in Table 3 needed to be coded. These concern, among others, decision outcomes, lessons learned, or decision criteria not captured in the initial version of the extraction form. The initial coding was conducted by the first author. An open coding strategy was

followed. The coding was reviewed by two co-authors of the paper, Deepika Badampudi and Syed Muhammad Ali Shah.

3.2.4. Step 4: Analysis

Yin and Heald [32] report the results of the case survey in terms of vote counting. A similar approach is utilized in this case survey (e.g. the number of cases considering a CSO).

Odds ratio is used as a statistical analysis method to quantify how strongly the presence or absence of property A is associated with the presence or absence of property B in a given population. In this case, the association between the presence or absence of a decision criterion and the presence or absence of a decision option is measured through odds ratio.

4. Results

4.1. Overview of cases and context

Table 4 provides an overview of the 22 cases included in the case survey. The cases were characterized by the size of the company, the development unit where the component should be used. Furthermore, the domain, application type, and development methodology were specified. Half of the cases were in the context of the automotive domain (11 of 22), while other contexts have been considered as well. Given the proportionally high number of automotive cases, the most frequently reported application type was embedded systems. A variety of software development methodologies has been used, including agile and plan-driven processes. Furthermore, multiple cases reported hybrid processes combining agile and plan-driven concepts. With regard to sizes a range of different company sizes as well as sizes of the development units concerned have been reported. Overall, cases with varying sizes, domains, and development methodologies have been obtained.

4.2. RQ1: How are CSOs chosen?

In the context of RQ1 we investigated the CSOs considered, the involved stakeholders, the decision criteria, and the decision model used.

4.2.1. RQ1.1: Which CSOs were considered (In-house, COTS, OSS, Outsource or Services)?

The decision making problem constitutes the choice of CSOs for a software intensive system, the CSOs observed in the cases were:

- In-house: The component is developed within the same company. Badampudi highlights that it is still considered in-house development when the development is distributed, as long as it takes place within the company [5].
- COTS: This option stands for “components off-the-shelf” or “commercial-off-the-shelf”, which are already developed (pre-built) and for which the source code is not available to the buyer.
- OSS: Open source components are also pre-built, but the source code is available. OSS components are commonly built by a community.

Table 3: Extraction scheme for the case elicitation

Item ID	Category	Item	Description	Type	Research question	Comments
1	Meta-information	Code	Unique identifier for case	Integer	X	X
2	Meta-information	Author	ORION research participant providing the case	String	X	X
3	Meta-information	Company	Case company	String	X	X
4	Meta-information	Source	Origin of the case	String	X	X
5	Meta-information	Decision scenario	A short summary of the decision case	String	X	X
6	Context	Domain	Domain in which the decision was taken (e.g. automotive, avionics)	String	X	X
7	Context	Application type	Type of the application developed (e.g. embedded, information system)	String	X	X
8	Context	Company size	Size of the whole company	Integer	X	X
9	Context	Development unit size	Size of the development unit where the component was used	Integer	X	X
10	Context	Development methodology	Software Development Methodology used	String	X	X
11	CSOs	CSOs considered in the decision	CSOs = In-house, COTS, OSS, Outsource, Services	Enumeration	RQ1.1	X
12	Stakeholders	Decision initiator	Stakeholders involved in the initiation of the decision (identification of the need to make a decision and formulation of the decision problem)	String	RQ1.2	requires coding
13	Stakeholders	Stakeholders in decision preparation	Stakeholders preparing the information and reflections needed to make a decision	String	RQ1.2	requires coding
14	Stakeholders	Decision makers	Stakeholders taking the decision	String	RQ1.2	requires coding
15	Decision criteria	Performance	Response time, timing behaviour of the system	Boolean	RQ1.3	X
16	Decision criteria	Maintainability	Ease of updating the system (corrective, enhancements)	Boolean	RQ1.3	X
17	Decision criteria	Reliability	Reliability of the system	Boolean	RQ1.3	X
18	Decision criteria	Security	Security of the system	Boolean	RQ1.3	X
19	Decision criteria	Time	Time (duration) to develop the system	Boolean	RQ1.3	X
20	Decision criteria	Cost	Cost to develop the system	Boolean	RQ1.3	X
21	Decision criteria	Market and contract	Information about the market (mass market or bespoke development)	Boolean	RQ1.3	X
22	Decision criteria	Access and control	Ability to access the code and control the evolution of the component	Boolean	RQ1.3	X
23	Decision criteria	Component usage in the system	Ease of use when using the component in the system	Boolean	RQ1.3	X
24	Decision criteria	Component history	Evolution of the component in terms of maturity and change history	Boolean	RQ1.3	X
25	Decision criteria	Certification	Certification of the component, certification of the company offering a component	Boolean	RQ1.3	X
26	Decision criteria	Other	Other criteria not mentioned	String	RQ1.3	requires coding
27	Decision method	Decision model	Method used to make the decision	String	RQ1.4	requires coding
28	Decision method	Property model	Method used to estimate the impact of the decision	String	RQ1.4	requires coding
29	Decision results	Decision outcome	CSOs chosen	Enumeration	RQ2.1	X
30	Decision results	Decision preparation effort	Effort in preparing the decision (estimated)	Integer	RQ2.2	X
31	Decision results	Decision making effort	Effort in making the decision (estimated)	Integer	RQ2.2	X
32	Decision evaluation	Evaluation of the decision and the decision impact	Important criteria of the decision and reflections on the success/failure	String	RQ2.3/RQ2.4	requires coding
33	Other information	Noteworthy comments	Remarks considered important by the person extracting the case	String	X	requires coding

- **Services:** Services are components that are pre-built and can be called/obtained over a network, such as is the case for web services.
- **Outsource:** Another company is developing the component and is given the contract by the company wanting to obtain the component.

Table 5 shows the most frequently considered CSOs. As is evident from the table the most frequently considered CSOs were In-house, COTS and Outsource. This information is important to, for example, decide which options have to be well supported by evidence, e.g. with regard to benefits and limitations of one alternative over another (e.g. COTS in comparison to In-house).

To understand which CSOs are traded off against each other we analyzed the frequency of combinations for comparisons be-

tween the alternatives, which is shown in Table 6. The most frequent trade-offs were between In-house vs. COTS, In-house vs. Outsource, and COTS vs. OSS. It is no surprise that the most frequent comparisons comprise of the CSOs being most frequently considered (see Table 5).

The number of options considered has an implication on decision support methods, i.e. one needs to determine whether solutions can support a trade-off between two or more alternatives. The number of options considered in the cases are thus illustrated in Figure 2. The figure shows that it is common to consider two options, while there are still a number of cases that involve three options and more.

Table 4: Overview of the cases

Case ID	Company Size	Size development unit	Domain	Application type	Development methodology
Case 1	100.000	5.000	Automotive	Embedded systems	Iterative development, Lean manufacturing
Case 2	1.200	350	Utilities	Embedded + Software + Apps	Agile SCRUM variant
Case 3	40	32	Human resource management	Software	Hybrid Plan driven, Agile
Case 4	21	20	Trade, Security, Consumer	Software, Hardware, apps	Hybrid Plan driven, Agile
Case 5	500	6	Financial	Software + Hardware (servers)	Hybrid Plan driven, Agile
Case 6	17.000	1.300	Surveillance/Security	Software + Hardware	Plan driven, iterative
Case 7	NA	NA	Telecommunication	Embedded system	NA
Case 8	NA	NA	Automotive	Embedded control systems	Iterative, informal process
Case 9	100.000	50	Automotive	Interactive tool for calibration	N/A
Case 10	100.000	4.000	Automotive	Embedded software architecture	Iterative development, time-boxed deliveries, incremental
Case 11	100.00	4.000	Automotive	Embedded system architecture	Waterfall
Case 12	100.00	4.000	Automotive	Embedded system architecture	Waterfall
Case 13	20	4	Defense	Search engine	Agile
Case 14	20	3	Marketing & Advertising	Analytics tool	Waterfall
Case 15	14.000	180	Defense	Mission critical	Streamlined development, Hybrid process using the most appropriate development methods and techniques.
Case 16	140 000	300	Process automation and robotics for several industries	Embedded control system	Waterfall with safety certification and extensive change impact analysis process.
Case 17	100.000	30	Automotive	Embedded control system	Each of the department uses its own development methodology but there is a global process (V model). The components used SCRUM
Case 18	10.000	10	Automotive	Control-dominant software	Agile
Case 19	N.A. (large scale)	1.000	Telecom	Information system	Hybrid Plan driven, Agile
Case 20	100.000	100	Telecom	Charging system	Scrum
Case 21	100.000	5	Telecom	Embedded system	Scrum
Case 22	3	3	Telecom	NA	NA

4.2.2. RQ1.2: Which stakeholders were involved in the decision process?

We distinguish three groups of stakeholders in the decision making. According to Strydom [33] the key stakeholders in decision making are (1) the decision initiator who is raising the need to make a decision to solve a specific problem (in this case the selection of CSOs), (2) the decision preparers (people involved or supporting the decision making and in the discussions/analysis to reach a decision, such as experts and stake-

holders having an interest in the decision or influencers) and (3) the decision makers (taking the decision and thus “making the final choice between alternatives” (cf. [33])).

Table 7 provides an overview of the roles involved in the decision making, grouped by initiation, preparation, and decision making. Table 8 shows the distribution for the management roles involved.

Decision initiation: With regard to decision initiators software management is the most frequent initiator for the decision making cases. Non-managerial roles have initiated the decision

Table 5: Decision making options considered (frequency)

CSO	Frequency of consideration	%
In-house	17	32.08
COTS	14	26.42
Outsource	11	20.75
OSS	6	11.32
Services	5	9.43
Total	53	100.00

Table 6: Decision making option trade-offs

	In-house	COTS	OSS	Services	Outsource
In-house	9	1	4	11	
COTS		6	3	4	
OSS			1	1	
Services				2	
Outsource

Figure 2: Number of options considered

making process in three cases (software design/construction).

In three cases (16, 17 and 19) multiple roles initiated the decision. For software management the roles could be further refined. There it is noteworthy that in six cases executive management (CTO/CEO) have initiated the decision making. Other roles initiating the decision making were project leaders/manager, line manager, and product manager. When multiple roles were involved, it was a combination of management and technical roles.

Decision preparation: Compared with the decision initiation, more distinct roles were involved as stakeholders in the process of supporting the decision with discussion and expert input. Table 7 shows that the most common roles in decision preparation were software management and software design/construction. It was evident that expert decision support was obtained in twelve cases from either researchers or consultants. Additionally, customer relations/sales and software designers were involved frequently. Only in a few cases software test (cases 6 and 7), the actual customers (cases 2, 4 and 9) and sub-contractors (cases 5, 9 and 16)

Decision making: The decision makers were primarily managers. In comparison, only a few decision makers were in the category of software construction. In two cases (4 and 8) a consensus decision between management and software design/construction was made.

Number of roles per decision making process: Figure 3 shows the number of roles involved in the decision making process related to initiation, preparation, and decision making. Understanding the number of roles involved has important impli-

Table 7: Decision making roles involved

Roles	Initiation		Preparation		Deciding	
	Abs	%	Abs	%	Abs	%
Software management	15	62.5	32	43.24	21	80.77
Software construction/dev.	7	29.17	9	12.16	4	15.38
External support	1	4.167	8	10.81	0	0.00
Software test/quality control	1	4.167	2	2.70	0	0.00
Customers	0	0.00	3	4.05	0	0.00
Expert group (group of roles, unspecified)	0	0.00	5	6.75	0	0.00
Legal	0	0.00	2	2.70	1	3.84
Sales (business/customer relations)	0	0.00	4	5.40	0	0.00
Software design	0	0.00	6	8.10	0	0.00
Sub-contractors (providers of components)	0	0.00	3	4.0554	0	0.00
Total	24	100.00	74	100.00	26	100.00

Table 8: Decision making roles involved (Refinement for management roles)

Roles	Initiation		Preparation		Deciding	
	Abs	%	Abs	%	Abs	%
Executive management (CEO/CTO)	6	30.00	11	31.43	8	32.00
Management (type unspecified)	7	35.00	9	25.71	9	36.00
Product management	1	5.00	7	20.00	2	8.00
Project management	4	20.00	7	20.00	4	16.00
Line management	2	10.00	1	2.86	2	8.00
Total	20	100.00	35	100.00	25	100.00

Table 9: Criteria for decision making considered (frequency)

Group	Criterion	Cases	%
Product	Performance	17	77.27
	Reliability	13	59.09
	Maintainability	13	59.09
	Certification	12	54.55
	Level of openness and access/control	11	50.00
	Security	9	4.91
	Compatibility	9	4.91
	Functionality	7	31.82
	Architectural dependencies	4	18.18
	Compliance to standards and regulations	3	13.64
	Licensing rules	3	13.64
	Portability	2	9.09
	Quality (general)	3	13.64
	Safety	1	4.55
	Stability	2	9.09
	Ease of integration	1	4.55
	Extendability	2	9.09
	Longevity of asset	2	9.09
	Scalability	2	9.09
	User experience	1	4.55
	Availability	1	4.55
	Fitness for purpose	1	4.55
	Number of users	1	4.55
	Robustness	1	4.55
Financial	Cost - general (time, effort, resources)	17	86.36
	Cost - buy/rent	3	1.62
	Cost - acquisition	1	4.55
	Cost - Adaptation	1	4.55
	Cost - Licensing	1	4.55
	Cost - risks incurred	1	4.55
	Cost - total cost of ownership	1	4.55
	Cost - maintenance	1	4.55
Project	Level of support	5	22.73
	Familiarity with technology	1	4.55
Business	Ecosystems	1	4.55
	Market trend	1	4.55
	time to market	1	4.55
Total		158	

cations on how to support the decision making process. For example, as soon as multiple roles are involved there is a need to support consensus building and incorporating multiple points of view during the process. Only a few roles (primarily management) are involved in the initiation (one individual role in 19 cases) and making the decision, while a larger set of roles is involved in the preparation (in average three roles, see Figure 3).

4.2.3. RQ1.3: Which criteria were considered for making the decision?

Criteria considered: The criteria based on which the decision options are evaluated are shown in Table 9 that indicates the number of cases that have considered the criteria and also the percentage of the total number of cases that consider the criteria.

Frequently considered criteria related to the product are quality criteria, such as performance, reliability, maintainability, compatibility and security. Further product-related characteristics were considered frequently — the most frequently mentioned ones being certification, level of openness and access/control. Furthermore, cost was an important criterion. A sub-set of cases further specifies the type of cost (e.g. to buy/rent, licensing, etc.).

Criteria considered together: As seen in Table 9, some of the criteria are considered more frequently. We investigated the criteria that are considered together among the frequently considered criteria (frequencies greater than 10) as shown in Fig-

Figure 3: Number of roles involved in the decision process

ure 4. The percentages are calculated by dividing the number of times the criterion is considered together with another criterion by the total number of times a criterion is considered. For example performance is considered 17 times in total and out of the 17 times, it is considered 13 times together with reliability. Therefore the percentage of performance and reliability considered together and in this order is $(13/17) \times 100 = 76.47\%$. Some values in Figure 4 are 100% which means that every time the criterion A is considered, it is always considered together with the criterion B. Interestingly, this is the case for reliability and performance. That is, every time reliability is considered, performance is also considered. Higher percentages indicate that possible trade-offs between the criteria might need to be considered in the decision. For example, improving reliability might be done under performance constraints.

Number of criteria: The numbers of criteria considered are shown in Figure 5. The number of criteria taken into consideration impacts the requirements on the solution, e.g. with respect to optimization this means to choose an approach for multi-objective optimization. When conducting an analysis of trade-offs between criteria (i.e. how they positively or negatively influence each other) the complexity of the analysis increases.

4.2.4. RQ1.4: Which decision making approach/model was used?

Table 10 shows the approaches used in decision making. Expert opinion/judgment has been utilized in all cases. Expert judgment was supported by a number of approaches. Prioritizing and ranking the alternatives with respect to weighted criteria (4 cases), listing the Pros and Cons (5 cases), and Pugh analysis (4 cases) were identified. More formal approaches for estimating (e.g. COCOMO) or the use of models for decision making

Table 10: Approaches used for decision making (frequency)

Decision making approaches used	Cases
Expert opinion / expert judgment	22
Pros and Cons	5
Prioritization, ranking, weighted criteria	4
Pugh analysis (Decision-matrix method)	4
Total	22

(e.g. optimization) was not observed.

4.3. RQ2: What was the result of the decision making process?

The second research question focuses on the description of the decision outcomes while following the decision making processes characterized in Section 4.2.

Figure 5: Number of criteria considered

	Performance	Reliability	Maintainability	Cost	Certification	Level of openness
Performance	X	76,47	58,82	70,59	58,82	58,82
Reliability	100	X	61,54	92,31	69,23	46,15
Maintainability	76,92	61,54	X	92,31	53,85	46,15
Cost	73,68	63,16	63,16	X	57,89	47,37
Certification	91,67	83,33	58,33	91,67	X	50
Level of openness	90,91	81,82	54,55	63,64	54,55	X

Figure 4: Criteria Trade-offs

4.3.1. RQ2.1: Which CSOs among those considered (RQ1.1) were chosen?

Figure 6 shows the options considered for each case as well as the options chosen. The elements in the figure are to be interpreted as follows:

- CSOs that were considered, but not chosen, are represented as black circles. For example, in case 12, In-house has been considered, but it was not chosen.
- Decisions that deviated from the recommended choice based on the investigation during the decision preparation are represented as black diamonds. For example, in Case 1 the recommendation was to go with COTS based on the decision preparation, but then the choice made was Outsource, which was also a CSO considered from the beginning in the decision making process.
- CSOs that were considered, chosen, and did not deviate from the recommended choice, are illustrated as circles with a white diamond inside.

In 7 of 22 cases in-house development has been chosen among the alternatives. Also in 7 cases components off-the-shelf (COTS) was the CSO chosen. In-house development has been considered in 17 out of 22 cases. Thus, the reflection of developing internally, or obtaining a component externally from different sources seems to be a key decision in practice. In a few cases only, the recommended alternative has not been chosen, which is further reflected when looking at RQ2.3 (Section 4.3.3).

4.3.2. RQ2.2: What was the effort invested in the decision making process?

The effort spent on preparing the decision for the decision maker is shown in Figure 7, the x-axis shows effort spent in person months. The effort data was available in 16 of 22 cases. The average time spent on preparation was around 780 person hours.

In comparison Figure 8 shows the effort spent on decision making, but it is available for 16 cases, only. This indicates only a small fraction of the effort spent in preparation (on average 30 person hours) is spent on decision making.

4.3.3. RQ2.3: Which criteria initially considered (RQ1.3) ended up as significant for the final decision?

The association between a criterion and a CSO is evaluated in order to determine if the decision options are more favorable when a particular criterion is considered. To evaluate the association we considered the odds ratio (OR) between the criterion and decision option. Table 11 presents the odds ratio which is a measure of association between the most frequently considered criteria and decision options. The OR is calculated using a two-by-two frequency table as shown in Table 11.

Where

a = Number of cases where the criterion is considered and the decision option is chosen

b = Number of cases where the criterion is considered and the

Figure 6: Decision outcomes

Table 11: Odds ratio frequency table

Criteria	Decision option	
	Selected	Not selected
Considered	a	b
Not considered	c	d

decision option is not chosen
c = Number of cases where the criterion is not considered and the decision option is chosen

Figure 7: Effort in decision preparation

Figure 8: Effort in decision making

d = Number of cases where the criterion is not considered and the decision option is not chosen

$$OR = \frac{a/c}{b/d} = \frac{ad}{bc} \quad (1)$$

If any values of a , b , c or d equals zero then the OR is either 0 or results in division by 0 problem, depending of the value that is 0. The OR values are interpreted as follows:

- $OR = 1$ The criterion does not affect odds of decision option being chosen.
- $OR > 1$ indicates that the criterion is associated with higher odds of the decision option being chosen.
- $OR < 1$ indicates that the criterion is associated with lower odds of the decision option being chosen.

According to the values in Table 12, in-house seems to be a favorable decision option when all the frequently considered criteria (excluding general cost) are considered. OSS seems to be a suitable option when cost is considered as criteria. Also, some criteria have lower association with decision options such as the level of openness has lower association with COTS and outsourcing. However, the confidence intervals shown in Table 12 range from 0 to a positive number, which means it also includes the value 1 indicating that criteria do not have any effect on the decision outcome. For odds ratio the values of confidence interval are considered to determine if the results are statistically significant. If the range of confidence intervals include the value 1 then the result is determined as not statistically significant. This may be due to the low number of cases. There might be other confounding factors such as political reasons or tradition which might influence the decision. This means that, even though there is a set of criteria considered as important, the actual decision might be taken based on only a subset or a new criterion. Hence, we separately evaluate the criteria that were considered in the preparation and the criteria that actually led to the decision, as shown in Table 13.

Table 12: Decision outcomes

Criteria	In-house		COTS		OSS		Services		Outsourcing	
	OR	Conf. Int.	OR	Conf. Int.	OR	Conf. Int.	OR	Conf. Int.	OR	Conf. Int.
Performance	2,18	0,20	0,82	0,11	1,17	0,09	0,00	0,00	0,81	0,07
		24,21		6,34		14,52				9,52
Reliability	2,19	0,32	0,56	0,10	0,64	0,07	0,00	0,00	1,05	0,14
		15,04		3,25		5,61				8,02
Maintianability	2,19	0,32	1,25	0,21	2,40	0,21	0,67	0,04	1,05	0,14
		15,04		7,41		27,72		12,27		8,02
Cost	0,399	0,04	0,16	0,00	0,39	0,00	0,56	0,00	0,64	0,13
		3,52		0,00						26,32
Certification	6,86	0,66	1,25	0,21	0,17	0,01	0,18	0,00	0,78	0,09
		71,72		7,41		1,96				5,69
Level of openness	3,75	0,54	0,45	0,08	3,75	0,32	0,00	0,00	0,18	0,02
		26,04		2,67		43,31				1,92

The table shows that a wide range of criteria has been considered in preparation. Also, in most cases five and more criteria have been considered, see Figure 5. For each choice, we look into the criteria that were important and led to favouring a specific alternative. For example, when in-house was chosen, we looked in the reasons why it was preferred over another alternatives. For the different choices of CSOs, we analysed the criteria considered in preparation, the criteria that were important in the actual decision, and the assessment of the impact of the decision (positive or negative). Table 13 provides an overview of the results. The cases are grouped according to the decision made (In-house, COTS, etc.). Furthermore, the criteria considered in preparation, the criteria actually important in the final decision, and an evaluation of the outcome of the decision are considered. The criteria that were important in the final decision are divided into two groups: (1) Criteria that were considered in the preparation and are a sub-set chosen as important by the decision maker, (2) New criteria not considered in decision preparation, but in decision making.

Several interesting observations can be made from the table. In several cases only a small sub-set of criteria was considered in decision making compared to the preparation (see e.g. Cases 4, 5, 8, 17, and 19). That is, if the relevant decision criteria could have been identified earlier, this has a potential to save investigation effort in the preparation phase (such as pre-studies).

An undesirable case is where the input from the preparation is not considered in the final decision, and new criteria become important. That is, the effort spent in preparation is wasted (see, for example, Cases 1, 6, 11, 18, and 19). For example, in Case 1 15 criteria were considered, namely *performance*, *reliability*, *cost (general)*, *acquisition cost*, *adaptation cost*, *maintenance cost*, *functionality*, *quality (general)*, *familiarity with technology*, *ecosystem*, *architectural dependencies*, *longevity of the component*, *extensibility*, *level of openness and access/control*, *compatibility*: a large car manufacturer performed a pre-study to investigate pros and cons as a basis for opting between In-house, COTS, and OSS. However, the best option with respect to the criteria considered in preparation has not been chosen. Rather a political decision was made to maintain a good relationship with a supplier company.

Table 13: Evaluation of decisions made

Case	Outcome of the decision	Attributes considered in preparation	Attributes considered important in decision making (subset of preparation)	Attributes considered important in decision making (new)	Evaluation of the final decision $\sqrt{}$ = positive, o = indifferent, $\dagger$ = negative
In-house					
Case 4	In-house over COTS and Services	Performance, reliability, security, cost, user experience, compatibility, level of support, level of openness and access/control, stability, and certification	Security		$\dagger$: company needed to build themselves, as security requirements could not be met by other options, i.e. other options would have been preferred if possible
Case 6	In-house over COTS and outsource	Performance, reliability, security, cost, user experience, compatibility, level of support, level of openness and access/control, stability, and certification		Tradition of developing in-house	$\dagger$: huge effort invested in the investigation of vendors, and in the end it was done in-house anyways
Case 18	In-house over outsource	Performance, maintainability, reliability, cost, and certification.		Competence	$\sqrt{}$: perception that the best decision was made
Case 21	In-house over outsource	Performance, reliability, and certification	Performance, reliability		$\sqrt{}$: reliability was improved
COTS					
Case 11	COTS over in-house	Performance, maintainability, reliability, security, cost, safety, robustness, compatibility, and certification	Cost	Ease to provide solutions to customers	$\sqrt{}$: no issues later on, decision was considered a success
Case 13	COTS over OSS	Performance, cost, level of support, market trend, compatibility, architectural dependencies, level of openness/access and control, component history, certification	Component history, market trend, performance, (most important), compatibility, cost, level of support, architectural dependencies, level of openness and access/control		$\dagger$: issues arose in terms of computing performance that was not seen in advance, a larger pre-study could have helped
Case 16	COTS over in-house	Maintainability, cost, fitness for purpose	Cost, fitness for purpose		o
Case 22	COTS over in-house	Cost, functionality	Cost, functionality		o
OSS					
Case 3	OSS over COTS	Performance, reliability, security, cost, scalability, level of openness/access and control	Scalability, functionality		$\sqrt{}$: cost of the chosen solution was low
Case 7	OSS over COTS	Maintainability, cost	Maintainability, cost		o
Case 15	OSS over COTS and in-house	Performance, maintainability, reliability, security, cost, architectural dependencies, level of openness and access/control	Architectural dependencies, reliability		$\sqrt{}$: very successful with regard to cost reduction by a factor of 10
Outsource					
Case 1	Outsource over COTS and in-house	Performance, reliability, cost (general), acquisition cost, adaptation cost, maintenance cost, functionality, quality (general), familiarity with technology, ecosystem, architectural dependencies, longevity of the component, extensibility, level of openness and access/control, compatibility		Relationship with supplier company (maintain)	$\dagger$: sub-optimal solution with respect to criteria considered in preparation
Case 12	Outsource over in-house	Performance, maintainability, reliability, cost, functionality, compatibility, certification	Performance, cost, maintainability, functionality, compatibility, reliability		$\dagger$: underestimated computing resources needed (performance) of the chosen solution
Case 19	Outsource over in-house and services	Performance, maintenance cost, reliability, security, cost (general), quality (general), time to market, extensibility	Maintenance cost	Adaptability, level of openness and access/control	o
Case 20	Outsource over in-house	Cost, certification	Cost		$\dagger$: quality issues arose
Case 14	Services over in-house	Performance, cost, functionality, availability, level of support, level of openness and access/control	Cost, functionality, performance, availability, level of support		$\dagger$: solution not as successful as hoped for, quality issues cost time to market, service needs to be replaced.
Combinations					
Case 2	COTS and Services over OSS	Performance, maintainability, reliability, security, cost, risks incurred, licensing rules, level of support, portability, compatibility, certification	N.A.		o: tradeoff between level of support (safer solution), but with a higher cost
Case 5	COTS, Services and Outsource over In-house	Performance, reliability, security, cost, compliance to standards and regulations, cost for buying/renting, scalability, component history, certification	Performance, reliability		$\dagger$: challenging to estimate and assess the impact and decision required long lead-time
Case 8	COTS over OSS (first iteration) and OSS over COTS (second iteration)	Performance, portability, cost, licensing, functionality, portability, licensing rules, level of openness and access/control, ease of integration, component history	Level of openness and access/control, component history		$\dagger$: problems in details could not be foreseen (old version of programming language became an issue for chosen mature component)
Case 10	In-house and COTS over an individual option	Performance, maintainability, reliability, security, cost, certification, level of openness and access/control, compatibility, stability, number of users	Cost, maintainability, reliability, certification, level of openness and access/control, performance, security		$\sqrt{}$: perceived as the right decision
Case 17	In-house and Outsource over an individual option	Performance, maintainability, cost, quality (general), compliance to standards and regulations, functionality, level of openness and access/control, compatibility	Cost, functionality, compliance to standards and regulations		$\sqrt{}$: perceived as positive decision with regard to a combination of in-house and outsource in terms of available competence and responsibility

4.3.4. RQ2.4: Were the chosen CSOs considered the “right” choice retrospectively?

Looking at how the decisions were evaluated it is visible that a high number of decisions were perceived as sub-optimal, which applied to a total of 9 cases. In only 7 cases the decision was perceived as clearly positive. This highlights the need for support in the decision making processes to improve the decision making outcomes.

Table 13 also provides an insight on why one option was preferred over another. For example, it is visible that in-house was preferred over COTS and services for security reasons (Case 4), in-house over COTS and outsource for performance and stability reasons (Case 21), etc.

When we compare Table 13 and Table 12 we see that all positive decisions were based on criteria that have high association between the decision option and criteria. The negative decisions were based on criteria that have lower association between the decision option and criteria. For example, in-house is chosen over outsource based on performance and reliability in Case 21. It was perceived as positive decision. Performance (P) and reliability (R) have higher association with in-house (P=3.64, R=3.50) than outsource (P=1.33, R=1.00). On the other hand, the decision to choose in-house over COTS and services based on security was negatively perceived as the security (S) requirements were not met. As shown in Table 12 the odds that in-house (S= 1.33) was chosen when security is a criteria is the lowest as compared to COTS (S= 1.83) and services (S= 7.50). Hence, it is not surprising that the decision was not positive. Even though the odds ratio in Table 12 are not statistically significant, they are consistent with the individual decision outcome in Table 13.

5. Discussion

The discussion is structured along the research questions. In particular, the results of the systematic review [5] and the case survey are compared as both studies had a similar focus.

5.1. Reflections with respect to the research questions

The first research question explored how CSOs are chosen in practice, four research questions were formulated and are discussed in the following. Three of the four questions are shared with the literature review by Badampudi [5], namely options considered (RQ1.1), criteria considered (RQ1.3), and decision making method (RQ1.4). Stakeholder roles were not explicitly discussed in the primary studies included by Badampudi, and hence are new results provided by the case survey in the context of choosing CSOs.

Options considered (RQ1.1): The contributions of existing literature and case survey are mapped as shown in Table 14. The primary studies include empirical comparisons between CSOs and solution proposals of how to choose among them. Even though there are comparisons, no decision making solutions assist the COTS vs. OSS decision. The case survey reported six cases that consider COTS and OSS in the decision.

While there are no primary studies comparing in-house and outsourcing, two papers propose decision making solutions for the selection of CSOs. Overall the number of primary studies (two papers) discussing in-house and outsource is low despite being the most frequent decision as revealed by our case survey (11 of 22 cases).

In-house and COTS have also been compared in primary studies identified in our systematic literature review [5], and decision making solutions are proposed to make in-house vs. COTS decisions. The frequencies of the primary study and case survey are consistent for the in-house vs. COTS options.

COTS vs. outsource and OSS vs. outsource are neither compared nor any decision making solutions are proposed. Therefore, based on the existing literature, it is unknown whether such decisions are even considered by decision makers. However, the case survey identify cases where COTS vs. outsource and OSS vs. outsource decisions are considered. There is thus a need for better support in practice.

Stakeholders (RQ1.2): In most cases management roles initiated the decision, while the decision making preparation usually involved several roles from management, software construction and development, but also software design. Hence, these roles seem to be considered important. Also, external support was acquired in order to prepare the decision, which indicates that additional competencies compared to those in the organization are needed to make the decision.

Criteria (RQ1.3): We first look at trade-offs between criteria, and compare them to the trade-offs discussed in our systematic literature review [5].

Trade-off 1: Trade-offs between market-trend, technical support and maintainability is observed in the literature (cf. [5]). Following market trend indicates frequent updates which involves additional maintenance effort. At the same time, the need for a high pace in releasing new features is required, which may result in technical debt as shortcuts may be taken. Also, if the component is not upgraded to the latest available version, then the support offered from supplier/vendors might not be extended. Hence, a trade-off needs to be made between following market trends, maintaining the system stability and retaining the technical support. In the case survey 14 cases consider maintainability, 5 cases consider technical support, and 1 case considers market-trend. This means that maintainability, technical support, and market-trend are not considered together in the decisions.

Table 14: Mapping of existing literature and case study contributions with respect to CSOs

Architectural asset option	Number of primary studies		Number of Cases
	Comparisons	Solutions	
COTS vs. OSS	6	0	7
In-house vs. COTS	2	6	9
In-house vs. outsource	0	2	13
In-house vs. OSS	1	0	2
COTS vs. outsource	0	0	5
OSS vs. outsource	0	0	2

Trade-off 2: Another trade-off identified in the literature is between source code availability, technical support, and license [5]. The availability of source code might be a criteria for selecting a decision option so that the code can be changed. However, some licenses require changes in the code to be open. Also, technical support might not be extended for the modified code. Although source code availability (12) has high frequencies of cases, the frequencies of license (3) and technical support (5) are comparatively lower, which implies that the trade-off is not considered.

Trade-off 3: Development effort and integration effort trade-off is identified in the literature [5]. Development effort can be saved if the development is not done in-house. However, if the saved development effort is less than the additional integration effort, then the decision is not optimal. No such trade-off is considered in any of the cases.

Overall the trade-offs observed in the cases and literature are not consistent, i.e. the existing studies do not support the processes observed in industry. This indicates a potential gap between industry practice and the focus of research with regard to sourcing decisions when looking at the trade-offs made. Overall, this merits the investigation of trade-off practices in the industry to support researchers in the selection of future research topics and questions with regard to trade-offs.

Decision Making Methods (RQ1.4): The methods for decision making devoted to in-house vs. COTS and in-house vs. outsource exist in the literature [5], both trade-offs being frequently considered in the case survey. All the in-house vs. COTS decision making solutions proposed in the literature consider technical factors, notably time, cost, and reliability. These factors are easy to calculate; probably this is the reason why the methods have considered them. Most of the cases in this case survey considered cost and reliability in their decisions. Time has not been considered that often. However, the case survey identified many other criteria that are considered in the decisions (which also include non-technical criteria), which are not included in the methods proposed in literature.

The methods for choosing between in-house vs. outsource proposed in the literature consider requirement dependencies [5]. However, this is not identified as a criterion in any of the cases in the case survey.

All the decision making methods proposed in literature are automated and mathematical, i.e. they are highly formalized. However, the cases indicate that the most popular techniques used in making decisions are expert based, which involves subjective opinions. This indicates that the methods proposed in the literature are not consistent with the practice in industry. This also suggests that practitioners are looking for decision making methods that aid in decision-making and not the solutions that give the decision/outcome.

Further, according to the case survey, the management takes most decisions. The decision making methods proposed in the literature are quite complex. Due to the complexity and learning curve involved, the managers might not accept or use the solutions proposed in the literature.

Options chosen (RQ2.1): The number of times options were chosen in comparison to the times they were considered in-

dicates that all options are viable choices for the software-intensive systems considered. In-house has been selected in 41% of all cases it was considered (7/17), COTS in 50% (7/14), Outsource in 55% (6/11), OSS in 83% (5/6) and Services in 40% (2/5).

Effort invested (RQ2.2): The effort invested in preparing the decision is quite significant, with the mean effort being 780 person hours, while in several cases the effort was over 1000 hours. In order to understand the factors we investigated the correlation between the effort invested and the complexity of the decision problem (number of criteria and number of options considered). In addition we calculated the correlation between the number of people involved in the preparation and the effort. Table 15 shows the results of the correlation (using the non-parametric method proposed by Spearman). Rubin [34] defines boundaries for the strength of correlations. A strong positive correlation is observed for the number of CSOs, and a moderate positive correlation for the number of decision criteria. No correlation to the number of people is observed; that is more people do not necessarily mean more effort, depending on the time the individuals spent on decision making preparation. Given that the number of CSOs as well as the number of criteria seem to be related to effort we suggest a staged process for choosing among options to not invest effort on more obvious exclusions. Furthermore, criteria need to be prioritized. Similar reflections were presented in the field of requirements engineering, where it was found that more complex decisions take more time [35]. As a consequence requirements triage has been introduced that removes the most obvious options first to avoid investigative effort [36]. Thus, similar ideas may be relevant for choosing among CSOs.

Criteria initially considered ending up as important (RQ2.3): The need for prioritizing the criteria and identifying the important ones also became evident when analyzing the final decisions taken, and the criteria ending up as important among those considered. The decision problem may be simplified if criteria are identified and removed early. Much can be learned from the requirements community in that regard, in particular requirements prioritization techniques may be of value [37]. Barney et al's study gives an overview of approaches for trade-offs between quality attributes, which may also be useful to not only trade-off between quality attributes, but by considering other factors as well [38].

Retrospective reflection on CSO choice (RQ2.4): In seven cases only, the result of the decision was perceived as positive. In particular, if high investments have been made in preparing the decision, and the final decision is not perceived as successful, then the preparation effort can be considered wasted. The methods used for decision making were mostly experience

Table 15: Decision criteria

Criteria	Spearman Correlation to Preparation Effort
Number of CSOs	0.559
Number of Decision Criteria	0.350
Number of People in Preparation	0.075

Figure 9: Hierarchical Clustering of Cases

based, and no decision support systems or methods have been used (such as estimations, existing evidence from research); thus the results indicate that there is an industrial challenge relevant for the research to address. In particular, decision support systems aiding the experts may be of interest to design for this particular decision problem. Early attempts have been made and preliminary results are available in that regard (cf. Wohlin et al. [39]).

5.2. Characterization of decision making processes

According to Larsson [15] case surveys focus on identifying patterns (similarities and differences) between cases. With regard to the research questions each individual question investigated a specific aspect of the decision making case. We used hierarchical cluster analysis, in particular clustering of binary data (presence and absence of variables or attributes in cases) for this purpose. The method to calculate the clusters was Squared Euclidean Distance. The clustering is used as a reflective tool. It takes the values obtained across research questions into account to determine the similarity and differences between the cases. We only included variables related to the decision case (CSOs, stakeholders, and decision criteria) to find commonalities of how decision making has been made. Figure 9 shows the Dendrogram. It is evident that four cases are closely related, namely SPSS 18, SPSS 20, SPSS 21 and SPSS 22. Furthermore, cases SPSS 10 and SPSS 11 are very closely related. Overall, the Dendrogram shows that no clear patterns could be obtained. Using two-step clustering as an alternative approach showed that the shapes of clusters are not clearly identifiable, which is a sign that, besides the groups above, no clear patterns across cases are observable.

It has already been established that the cases share decision making characteristics, and a number of cases (more than two) is concerned, hence it is of interest to take a closer look at them. Hence, their contexts and outcomes are of interest to compare, which may explain why they are in the same cluster. The clus-

ter of cases 18, 20, 21, and 22 is interesting because three of the four cases were perceived as positive. Despite of being in different domains (cases 18 and 22 are in the automotive domain and cases 20 and 21 are in the telecom domain) both of them are similar. The cases characteristics, the context, and the outcome for the four cases in the cluster are briefly summarized in Table 16.

5.3. Validity threats

Larsson [15] highlights a number of limitations related to the case survey method. Furthermore, threats described in Petersen and Gencel [40] are highlighted where applicable.

Generalizability: First, Larsson points out that a limited number of cases could be studied, given that a case survey extracts more detailed information than a survey. Furthermore, the available cases are limited and not easily accessible. In this study, a total of 22 cases have been obtained. As it can be seen from Table 4 different domains and application types were studied. The automotive and telecommunication domains are most frequently represented, which introduces a bias in the dataset.

Descriptive validity - factual accuracy: There is also a risk that the coding is misunderstood and that data are not available/missing. Looking at the literature studies in the related work, even though a number of them report experiences from industry, important information about decision making processes is frequently missing. For example, little is known about the practitioners making the decisions, and how they make the decisions. Furthermore, only a small part of the decision making process is covered (e.g. the computation of an optimal decision), though no information is provided about which criteria were considered, and which ones were actually impacting the final decision made by practitioners. Utilizing the authors reporting their own cases, or utilizing interviews reduces this threat. That is, the coverage of the decision making process with respect to roles, criteria, and decision methods, decision taken, criteria considered in preparation and in the final decision, as well as the evaluation of the decision made could be captured in most cases. We were also able to capture the effort invested in the decision making process for at least half of the included cases. Thus, the threat related to the completeness of data is reduced, even though it could not be completely removed.

Theoretical validity - confounding factors and controllability: It would be desirable to make an inference from the characteristics of the decision making process to the success of the decision. Reflections related to the relationship between the two (process and success) are prone to confounding factors that we are not aware of.

Interpretive validity - objectivity of the researcher: Based on the findings from the survey conclusions and recommendations to practitioners are provided. The recommendations follow the data, and there is a risk that an individual researcher draws biased conclusions. The risk is reduced due to the number of authors involved in the study who provided their input to the reflections and key findings.

Interpretation and coding of the data: Distakes and potential bias are probable when interpreting and coding a large dataset.

Table 16: Case Cluster

Attribute	SPSS 18	SPSS 20	SPSS 21	SPSS 28
Context				
Domain	Automotive	Telecom	Telecom	Automotive
Company size	10.000	100.000	100.000	100.000
Development unit size	10	100	5	30
Development method	Agile	Agile (Scrum)	Agile (Scrum)	V-Model/Scrum
Decision case characteristics				
CSOs considered	In-house, Ooutsource	In-house, Ooutsource	In-house, Ooutsource	In-house, Ooutsource
Stakeholders (initiation)	Software construction	Management	Management	Management, Software construction
Stakeholders (preparation)	Management, Expert Group	Management	Management, Sales	Management, Expert Group
Stakeholders (decision making)	Management	Management	Management	Management
Criteria considered	Performance, maintainability, reliability, cost, and certification	Cost, certification	Performance, reliability, and certification	Performance, maintainability, cost, quality (general), compliance to standards and regulations, functionality, level of openness and access /control, compatibility
Method	Expert judgment	Expert judgment	Expert judgment	Expert judgment
Outcome				
Decision impact	Positive	Negative	Positive	Positive
Effort	2400	640	480	N.A.

The coding activity is similar to what would be conducted in a systematic literature review when coding extracted data from papers. Kitchenham et al. [41] recommends to peer-review the coding. Consequently, the second and third authors of the paper reviewed the coding done by the first author.

Repeatability: A data extraction form and the GRADE taxonomy [31] support the data extraction increases its objectivity (e.g. through interviews or extracting one’s own experience), and with that the repeatability of the results. In particular, it is of interest to increase the corpus of cases over time and with that gain further knowledge about the decision making processes for choosing CSOs.

6. Conclusion

In this paper we investigated how CSOs are chosen by conducting a case survey supported by Larsson’s guidelines (cf. [15]). We identified and described 22 decision cases for choosing CSOs in the software-intensive system context. The CSOs were based on the experience of experts participating in a research project for choosing CSOs (ORION), and interviews with industry practitioners. 11 cases were based on experiences of the research project members, and 11 based on interviews.

We aimed to describe and understand how choices among CSOs are made for software-intensive systems. The study had two main parts (RQ1). Secondly, we analyzed the result of the decision making process (RQ2). RQ1 and RQ2 were broken down into four sub-questions. We present the main conclusion with regard to the research questions.

RQ1.1: Which CSOs were considered? The most frequently considered options were In-house, COTS, and Outsourcing. In-house development was considered in 17 of 22 cases, which indicates that the most common trade-off is to either make (in-house) or obtain the components externally (COTS, Outsource, OSS or Services). In-house development was most commonly traded off with Outsourcing and COTS. Hence, empirical comparisons of these combinations of sourcing options are of use, and should be in the focus of future work.

Interestingly enough, despite being more and more common through web-services for instance, services are seldom considered as a possible CSO.

RQ1.2: Which stakeholders were involved in the decision process? In most cases the management initiated the decision making process. The identification of the need for making the decision mostly originates from a single role and is mostly decided by a single role. Decision preparation is done by a group of stakeholders. The most commonly involved roles in preparing the decision were software management, software design/construction, external decision supporters and sales. The decision for the CSO primarily lies with the managers. Decision makers are commonly involved in the initiation and preparation activities.

RQ1.3: Which criteria were considered for making the decision? Cost, performance, and system reliability stand out as the three most commonly considered criteria. It is highly likely that when reliability is considered as a criterion, it is considered along with performance. Similarly it is highly likely that when cost is considered it is also considered with performance and reliability. These results may be biased by the convenience sample obtained, as 10 cases were coming from the automotive domain. Commonly more than two criteria are considered, thus solutions aiding in decision making have to provide opportunities for multi-criteria analysis and decision support. The trade-offs between industrial cases and research studies identified in the literature are not consistent. Hence, the companies in this study would require additional evidence that compares the frequently compared options (RQ1.1) with regard to relevant combinations of criteria (RQ1.3).

RQ1.4: Which criteria were considered for making the decision? Expert-based decision making was used in all cases. Methods used to support the experts were prioritization, weighing, and ranking of alternatives and Pugh analysis. There exists an opportunity for research to provide decision support due to complexity of the decision problems reflected in the consideration of a multiple decision options and criteria. The meth-

ods found in the literature were not used in the practical cases, which indicates a gap between industry and academia.

RQ2.1: Which CSOs among those considered (RQ1.1) were chosen? All options were considered viable. In-house has been selected in 41% of all cases it was considered (7/17), COTS in 50% (7/14), Outsource in 55% (6/11), OSS in 83% (5/6) and Services in 40% (2/5).

RQ2.2: What was the effort invested in the decision making process? The largest amount of effort (in average 780 hours) was invested in the decision preparation phase. We conducted a correlation analysis to learn how the complexity of the decision problem (number of options and criteria) affects the preparation effort. The number of options is strongly correlated with the effort, indicating that a staged decision making process may be useful, where more obvious choices are removed as early as possible. Criteria were moderately associated with effort indicating the importance of criteria prioritization.

RQ2.3: Which criteria initially considered (RQ1.3) ended up as significant for the final decision? Only a small sub-set of the criteria considered in preparation is considered in final decision. In combination with the findings related to RQ2.2 this further stresses the importance for criteria prioritization. We identified cases where the recommended decisions were discarded due to new criteria considered in the final decision making (e.g. political reasons). Decision support systems need to take such situations into consideration to avoid wasted effort, and provide transparency with regard to the criteria that really matter to the decision makers.

RQ2.4: Were the CSOs chosen considered the “right” choice retrospectively? Only in 7 cases the decision made was perceived as clearly positive, while in 8 cases the outcome was perceived as negative, indicating a clear need for decision support to improve on the situation in practice. The criteria that were considered important in the positive decisions have higher observed association with the chosen decision option. Correspondingly, in the negative decisions the selected criteria have lower observed association with the chosen decision option.

Future work: Given the low success rate and reliance on expert opinion, we argue the need for decision support in choosing among CSOs. This case survey provides valuable input to such systems, for instance highlighting important CSOs, and criteria. To further strengthen the findings as input for such a system, we compared the findings with the systematic review on CSO selection presented in the related work. We found multiple deviations with regard to criteria and methods used for decision making. Hence, this points to a gap between what is done in industry and what is proposed by academia, at least for the cases studied. Thus, more collaborative studies with industry participation are needed where the problem of CSO choice is investigated.

References

- [1] J. Li, R. Conradi, O. P. N. Slyngstad, C. Bunse, M. Torchiano, M. Morisio, An empirical study on decision making in off-the-shelf component-based development, in: *Proceedings of the 28th international conference on Software engineering*, ACM, 2006, pp. 897–900.
- [2] P. Di Giacomo, Cots and open source software components: are they really different on the battlefield?, in: *COTS-Based Software Systems*, Springer, 2005, pp. 301–310.
- [3] J. S. Norris, Mission-critical development with open source software: Lessons learned, *Software*, IEEE 21 (1) (2004) 42–49.
- [4] S. M. Syed-Mohamad, T. McBride, A comparison of the reliability growth of open source and in-house software, in: *15th Asia-Pacific Software Engineering Conference (APSEC 2008)*, 2008, pp. 229–236.
- [5] D. Badampudi, C. Wohlin, Software component decision-making: In-house, open source, cots or outsourcing - a systematic literature review., in revision after submission for *Journal of Systems and Software*.
- [6] V. Cortellessa, F. Marinelli, P. Potena, Automated selection of software components based on cost/reliability tradeoff, in: *Software Architecture*, Springer, 2006, pp. 66–81.
- [7] P. Potena, Composition and tradeoff of non-functional attributes in software systems: research directions, in: *The 6th Joint Meeting on European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering: companion papers*, ACM, 2007, pp. 583–586.
- [8] P. Jha, S. Bali, U. D. Kumar, H. Pham, Fuzzy optimization approach to component selection of fault-tolerant software system, *Memetic Computing* 6 (1) (2014) 49–59.
- [9] K. J. Stewart, A. P. Ammeter, L. M. Maruping, A preliminary analysis of the influences of licensing and organizational sponsorship on success in open source projects, in: *System Sciences, 2005. HICSS’05. Proceedings of the 38th Annual Hawaii International Conference on*, IEEE, 2005, pp. 197c–197c.
- [10] J. Li, F. O. Bjørnson, R. Conradi, V. B. Kampenes, An empirical study of variations in cots-based software development processes in the norwegian it industry, *Empirical Software Engineering* 11 (3) (2006) 433–461.
- [11] R. Torres, Developer-led adoption of open source software libraries: A conceptual model.
- [12] W. Chen, J. Li, J. Ma, R. Conradi, J. Ji, C. Liu, An empirical study on software development with open source components in the chinese software industry, *Software Process: Improvement and Practice* 13 (1) (2008) 89–100.
- [13] J. Rudzki, K. Kiviluoma, T. Poikonen, I. Hammouda, Evaluating quality of open source components for reuse-intensive commercial solutions, in: *Software Engineering and Advanced Applications, 2009. SEAA’09. 35th Euromicro Conference on*, IEEE, 2009, pp. 11–19.
- [14] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, *Empirical Software Engineering* 14 (2) (2009) 131–164.
- [15] R. Larsson, Case survey methodology: Quantitative analysis of patterns across case studies, *Academy of Management Journal* 36 (6) (1993) 1515–1546.
- [16] V. Cortellessa, F. Marinelli, P. Potena, An optimization framework for build-or-buy decisions in software architecture, *Computers & Operations Research* 35 (10) (2008) 3090–3106.
- [17] T. Kramer, M. Eschweiler, Outsourcing location selection with soda: a requirements based decision support methodology and tool, in: *Advanced Information Systems Engineering*, Springer, 2013, pp. 530–545.
- [18] T. Kramer, A. Heinzl, K. Spohrer, Should this software component be developed inside or outside our firm?-a design science perspective on the sourcing of application systems, in: *New Studies in Global IT and Business Service Outsourcing*, Springer, 2011, pp. 115–132.
- [19] J. Li, R. Conradi, O. P. N. Slyngstad, C. Bunse, U. Khan, M. Torchiano, M. Morisio, Validation of new theses on off-the-shelf component based development, in: *Software Metrics, 2005. 11th IEEE International Symposium*, IEEE, 2005, pp. 26–26.
- [20] T. Helokunnas, The dimensions of embedded cots and oss software component integration, in: *Product Focused Software Process Improvement*, Springer, 2002, pp. 509–518.
- [21] L. Brownsword, T. Oberndorf, C. A. Sledge, Developing new processes for cots-based systems, *IEEE Software* 17 (4) (2000) 48–55.
- [22] S. A. Hissam, C. B. Weinstock, Open source software: The other commercial software, in: *1st Workshop on Open Source Software at ICSE*, 2001.
- [23] S. A. Hissam, D. Plakosh, C. Weinstock, Trust and vulnerability in open source software, *IEEE Proceedings-Software* 149 (1) (2002) 47–51.
- [24] B. Mishra, A. Prasad, S. Raghunathan, Quality and profits under open

- source versus closed source, ICIS 2002 Proceedings (2002) 32.
- [25] S. U. Khan, M. Niazi, R. Ahmad, Factors influencing clients in the selection of offshore software outsourcing vendors: An exploratory study using a systematic literature review, *Journal of systems and software* 84 (4) (2011) 686–699.
 - [26] T. Helokunnas, M. Nyby, Collaboration between a cots integrator and vendors, in: *Software QualityECSQ 2002*, Springer, 2002, pp. 267–273.
 - [27] J. Xu, Y. Gao, S. Christley, G. Madey, A topological analysis of the open souce software development community, in: *System Sciences*, 2005. HICSS’05. Proceedings of the 38th Annual Hawaii International Conference on, IEEE, 2005, pp. 198a–198a.
 - [28] R. Land, L. Blankers, M. Chaudron, I. Crnković, Cots selection best practices in literature and in industry, in: *High Confidence Software Reuse in Large Systems*, Springer, 2008, pp. 100–111.
 - [29] T. Wanyama, B. Far, An empirical study to compare three methods for selecting cots software components, *International Journal of Computing and ICT Research* 2 (1) (2008) 34–46.
 - [30] M. Morandini, A. Siena, A. Susi, Risk awareness in open source component selection, in: *Business Information Systems*, Springer, 2014, pp. 241–252.
 - [31] E. Papatheocharous, K. Petersen, A. Cicchetti, S. Sentilles, S. M. A. Shah, T. Gorschek, Decision support for choosing architectural assets in the development of software-intensive systems: The GRADE taxonomy, in: *Proceedings of the 2015 European Conference on Software Architecture Workshops*, Dubrovnik/Cavtat, Croatia, September 7-11, 2015, 2015, pp. 48:1–48:7.
 - [32] R. K. Yin, K. A. Heald, Using the case survey method to analyze policy studies, *Administrative science quarterly* (1975) 371–381.
 - [33] J. Strydom, *Introduction to marketing*, Juta and Company Ltd, 2005.
 - [34] A. Rubin, *Statistics for evidence-based practice and evaluation*, Cengage Learning, 2012.
 - [35] K. Wnuk, J. Kabbedijk, S. Brinkkemper, B. Regnell, D. Callele, Exploring factors affecting decision outcome and lead time in large-scale requirements engineering, *Journal of Software: Evolution and Process* 27 (9) (2015) 647–673.
 - [36] A. M. Davis, The art of requirements triage, *Computer* 36 (3) (2003) 42–49.
 - [37] P. Berander, A. Andrews, Requirements prioritization, in: *Engineering and managing software requirements*, Springer, 2005, pp. 69–94.
 - [38] S. Barney, K. Petersen, M. Svahnberg, A. Aurnum, H. T. Barney, Software quality trade-offs: A systematic map, *Information & Software Technology* 54 (7) (2012) 651–662.
 - [39] C. Wohlin, K. Wnuk, D. Smite, U. Franke, D. Badampudi, A. Cicchetti, Supporting strategic decision-making for selection of software assets, in: *Proceedings of the 7th International Conference on Software Business (ICSOB 2016)*, accepted, 2016.
 - [40] C. Gencel, K. Petersen, Worldviews, research methods, and their relationship to validity in empirical software engineering research, in: *Proceedings of the International Workshop on Software Measurement (Mensura 2013)*, 2013.
 - [41] B. Kitchenham, Procedures for performing systematic reviews, *Keele, UK, Keele University* 33 (2004) (2004) 1–26.