Component selection in Software Engineering - Which attributes are the most important in the decision process?

Panagiota Chatzipetrou[1], Emil Alégroth[1], Efi Papatheocharous[2], Markus Borg[2], Tony Gorschek[1],Krzysztof Wnuk[1]

[1.] Software Research Engineering Lab (SERL)
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden
e-mail: {panagiota.chatzipetrou, emil.alegroth,
tony.gorschek, krzyszstof.wnuk}@bth.se

[2.] RISE SICS AB,
Scheelevägen 17,
SE-223 70, Lund, Sweden
e-mail: {efi.papatheocharous, markus.borg}@ri.se

***Abstract*— Component-based software engineering is a common approach to develop and evolve contemporary software systems where different component sourcing options are available: 1)*Software developed internally (in-house)*, 2)*Software developed outsourced*, 3)*Commercial of the shelf software,* and 4) *Open Source Software*. However, there is little available research on what attributes of a component are the most important ones when selecting new components. The object of the present study is to investigate what matters the most to industry practitioners during component selection. We conducted a cross-domain anonymous survey with industry practitioners involved in component selection. First, the practitioners selected the most important attributes from a list. Next, they prioritized their selection using the Hundred-Dollar ($100) test. We analyzed the results using Compositional Data Analysis. The descriptive results showed that *Cost* was clearly considered the most important attribute during the component selection. Other important attributes for the practitioners were: *Support of the component*, *Longevity prediction*, and *Level of off-the-shelf fit to product*. Next an exploratory analysis was conducted based on the practitioners' inherent characteristics. Nonparametric tests and biplots were used. It seems that smaller organizations and more immature products focus on different attributes than bigger organizations and mature products which focus more on *Cost*.**

Keywords: Component-based software engineering; Decision making; Compositional Data Analysis; Cumulative voting

I. Introduction

Component-based software engineering (CBSE) is a common approach to develop and evolve contemporary software systems. However, in CBSE it is not always the best option to develop internally (in-house) a new component [1]. Thus, the practitioners are very often asked to choose between different Component Sourcing Options (CSO). But what are the factors that affect the practitioners' decision to choose one CSO against another? In other words, how do the practitioners prioritize the attributes of a component when they have to decide on "buying" or "making" a new component?

Prioritization is a procedure of principal importance in decision making. In Software Engineering it is encountered in cases where multiple attributes have to be considered in order to take a decision. However, the human subjectivity is a source of variation when different people try to prioritize independently a certain number of attributes. These factors led to the adoption of voting schemes where stakeholders express their relative preferences for certain attributes in a systematic and controlled manner.

The Cumulative Voting (CV) or 100-Point Method or Hundred-Dollar ($100) test, described by Leffingwell and Widrig [2], is a simple, straightforward and intuitively appealing voting scheme where each stakeholder is given a constant amount (e.g. 100, 1000 or 10000) of imaginary units (for example monetary) that he or she can use for voting in favor of the most important attributes. In this way, the amount of money assigned to an attribute represents the respondent's relative preference (and therefore prioritization) in relation to the other attributes. The points can be distributed in any way that the stakeholder desires. Each stakeholder is free to put the whole amount given to him or her on only one attribute of dominating importance. It is also possible for a stakeholder to distribute equally the amount to many or even to all of the attributes.

Unfortunately, there is little available research on which attributes of a component are of principal importance when multiple attributes are considered to make a CSO decision. Understanding the source of variation between decision makers between different CSOs in CBSE may optimize the decision process and consolidate opinions with respect to prioritization. In the present work, we focused on the attributes that practitioners typically compare when they are choosing to add or replace a new component for their products. The products concern software-intensive systems and thus entail component complexity. Therefore, an industrial cross-domain anonymous survey regarding the practitioners' decision making in relation to choosing between CSOs was conducted. The questionnaire was web-based and consisted of a number of both open-ended and closed-ended questions. The practitioners were asked to choose between four different Component Sourcing Options (CSO). The CSO decisions can be summarized in the following four alternatives [3], [4]:

978-1-5386-7383-6/18/$31.00 ©2018 IEEE
DOI 10.1109/SEAA.2018.00039

- *Software developed internally (in-house):* This is the case where a company develops a component internally. In addition, development is still considered in-house when the development is distributed in different locations, as long as it takes place within the company. The source code is developed and remains inside the same company.
- *Software developed outsourced*: Another company is developing the component on behalf of the company which wants to obtain the component. Usually the source code is delivered as part of the contract agreed between the two companies.
- *Commercial of the shelf software (COTS):* The company buys an existing component from a software vendor (pre-built). The source code is not available for the buyer.
- *Open Source Software (OSS)*: The company integrates a pre-built, existing component that has been developed by an open source community as an open source software. The source code is publicly accessible.

Practitioners were asked to choose between the above mentioned four CSOs and indicate which information is the most important input for their decision process. They were given 12 attributes (TABLE I. presents the attributes in the same order they appeared in the survey). The practitioners were asked to prioritize the 12 attributes using Cumulative Voting (CV) by distributing 100 imaginary points. The number of the respondents was 157. The complete description and design of the survey is available here [4].

In our study we aimed to investigate the different views of practitioners towards the prioritization of the 12 attributes. The results of the Hundred-Dollar ($100) test are coded as variables and they are statistically analyzed in order to find differences or agreements in views and correlations with other inherent characteristics of the practitioners i.e. role, working experience, level of education, maturity of the product they work with and size of their organization. This information was collected also within our survey [4].

However, since the results from the Hundred-Dollar ($100) test sum up to 1, we cannot treat them as independent variables and since they are restricted in the [0,1] interval normality assumptions are invalid. A methodology which is suitable for the analysis of proportions is Compositional Data Analysis, known as CoDA. This methodology has been widely used in the analysis of materials composition in various scientific fields like chemistry, geology and archaeology, but its principles fit to analyze data obtained by CV.

The paper is structured as follows: Section II provides an outline of the related work. Section III presents the basic principles of CoDA and discusses various challenges related to its application. Section IV presents the results from the application of nonparametric tests and CoDA on the survey data. Finally, in Section V conclusions and future work are provided.

TABLE I. ATTRIBUTES USED FOR PRIORITIZATION

	Attributes	*Description*
1	Size	Size of the component, e.g. lines of code, memory footprint
2	Longevity prediction	Evolution of the component
3	Cost	Development, license and maintenance cost
4	Level of off-the-shelf fit to product	Functional fitness, i.e., how much component customization is needed
5	Complexity	Code complexity
6	API adequacy	Maturity of external APIs
7	Programming language Performance	Computational performance
8	Access to relevant documentation	Access to documentation
9	Code quality	Availability of automated tests, code review practices.
10	Support of the component	Formal support, channels, active development community
11	Adherence to standards	Follow the rules
12	Other	

II. RELATED WORK

The authors in [5] have conducted a systematic literature review about CSO selection. They also investigated decision criteria, methods for decision making, and evaluations of the decision results. The paper highlighted the CSOs compared were mainly focused on In-house vs. COTS and COTS vs. OSS. Generally, no other systematic reviews exist on the topic of CSO selection.

In a recent case survey [3], 22 case studies of how practitioners choose between CSOs are presented. One of the conclusions was that the most frequent trade-offs are carried out between in-house vs. COTS, in-house vs. outsource, and COTS vs. OSS. In-house was the favorable decision option, however, the evaluation of the decision showed that many of the decisions were perceived as suboptimal, indicating the need for optimizing the decision-making process and outcomes.

Several primary studies discussing in-house vs. COTS CSO decisions exist, i.e. [6] and [7]. In [8], a framework was presented to support the decision to buy components or build them in-house. The authors in [9] studied decisions made during integration of COTS vs. OSS and showed significant differences and commonalities.

Cumulative Voting is known as a prioritization technique, used in decision making in various areas. CV has been used also in various areas of Software Engineering, such as requirements engineering, impact analysis or process improvement ([10], [11]). Prioritization is performed by stakeholders (users, developers, consultants, marketing representatives or customers), under different perspectives or positions, who respond in questionnaires appropriately designed for the purpose of prioritization. CV has been proposed as an alternative to the Analytical Hierarchy Process (AHP) and its use is continuously expanding to areas such as requirements prioritization and prioritization of process improvements [2], [12].

In [10], CV is used in an industrial case study where a distributed prioritization process is proposed, observed and evaluated. The stakeholders prioritized 58 requirements with \$100,000 to distribute among the requirements (the large amount of "money" was chosen to cope with the large number of requirements). In [13] the CV was used for an industrial case study on the choice between language customization mechanisms. In [14] CV is one of the four prioritization methods examined, evaluated and recommended for certain stages of a software project. In [15] 18 interviewees were asked to prioritize 25 aspects using CV by distributing 1000 imaginary points to the aspects. Each interviewee prioritized the 25 aspects twice: Under the organizational perspective and under the self-perspective. The data were collected during an empirical study on the role of Impact Analysis (IA) in the change management process at Ericsson AB in Sweden. Compositional Data Analysis (CoDA) has been used also in the software effort phase distribution analysis [16], [17].

In this paper we used the experience from our former studies on CV and on the selection of different CSOs in order to investigate what reasons affect practitioners' decision to choose one CSO against another. Moreover, we aimed to discover if there are any trends among the practitioners based on their inherent characteristics. Therefore, the contribution of the paper is twofold: first we describe how the statistical framework of CoDA can be used on a prioritization study and second, draw conclusions on the reasoning behind decision making in components selection based on the practitioners' characteristics and their opinions. The main contribution is to understand and reason about the intuitive and conditional decision-making process of practitioners in CSO selection. The methodology is applied in a real survey data so as to draw interesting and useful results regarding the practitioners' decision process.

III. THE STATISTICAL FRAMEWORK

A. Compositional Data Analysis (CoDA)

Compositional Data Analysis (CoDA) is a multivariate statistical analysis framework for vectors of variables having a certain dependence structure: The values in each vector have sum equal to a constant. Usually, for easy reference to the same problem, after division by that constant, the sum of the values of each vector becomes one. Thus, from now on, we can assume that our data set consists of vectors of proportions or percentages in the form:

$$p_i \geq 0 \quad and \quad \sum_{i=1}^{k} p_i = 1.$$

The important point here is to understand that the data are constrained in the [0,1] interval; therefore, the techniques applied to samples from the real Euclidean space are not applied in a straightforward manner.

Various problems are associated with the analysis of those vectors: First, there is a problem of interdependence of the proportions (since their sum is 1) and therefore they cannot be treated as independent variables (the usual assumption of the multivariate methods). Second, their values are restricted in the [0,1] interval, so the normality assumptions are invalid. Third, we are not really interested in absolute values here, but rather for relative values (that is actually the meaning of a proportion). Thus, the whole problem is transferred to the analysis and the interpretation of the ratios of the proportions, i.e. values of the form p_i/p_j. The statistical analysis of these data, using methods based on ratios, tries to provide answers to some research problems which we encounter in any multivariate statistical analysis.

Concerning now the prioritization questionnaires using the 100\$ (or the \$1000) test, the data is essentially representing proportions of the overall importance allocated to each of the aspects examined in a study. The relative importance of the aspects is represented by their ratios, so CoDA seems the appropriate framework for their study. Historically, Karl Pearson in 1897 [18] posed the problem of interpreting correlations of proportions while the milestone for this type of statistical analysis is the pioneer work of John Aitchison [19], [20]. A freeware package for compositional data analysis is the CoDaPack3D, [21] which was used in the present analysis.

B. The problem of the zeros

The variables which form the constrained vectors are the attributes in our context, while their values are the priorities. The data from the CV questionnaires have some special characteristics which cause problems in the analysis.

The problem of zeros is of principal importance. When the number of aspects is large, and the individuals are only few, the data matrix is usually sparse with a large number of zeros. This structure causes problems of interpretation when we consider the relative importance.

Due to the problems of zeros, the various ratios needed for the analysis are impossible to compute. It is therefore essential and necessary to find first a way of dealing with the zeros. In [31] a new simple method is proposed that is most stable regarding the choice of the imputed values. This is called multiplicative replacement strategy and according to it, every vector $p=(p_1,\ldots,p_k)$, having c zeros, can be replaced by a vector $r=(r_1,\ldots,r_k)$ where:

$$r_j = \delta_j \;\; (\text{if } p_j = 0) \qquad \text{or} \qquad r_j = p_j \left(1 - \sum_{l:p_l=0} \delta_l \right) \;\; (\text{if } p_j > 0)$$

where δ_j is a (small) imputed value for p_j. The advantages of multiplicative replacement are discussed extensively in [22], [23]and [24]. This method is implemented in CoDaPack3D.

C. The CLR transformation

In order to treat the data with "common" techniques we need to transform them. Aitchison [20] proposed the centered log ratio (clr) transformation for transforming the raw proportional dataset to the real space and at the same time for retaining their correlation structure. The transformation is simple and achieved after dividing each component of a vector of proportions by their geometric mean. Since in our context the data usually contains zeros, the transformation can be applied only after the replacement of zeros. The formula for the CLR transformation of the compositional vector *r* is:

$$\mathbf{y} = clr(\mathbf{r}) = \left[\log\frac{\mathbf{r}}{g(\mathbf{r})}\right] \text{ where } g(\mathbf{r}) = \left(\prod_{i=1}^{k} r_i\right)^{1/k}$$

D. The biplot

The Biplot [25], [26], [27] is a graphical tool that has been used in various applications. Its compositional data version is a straightforward and useful tool for exploring trends and peculiarities in data.

An exemplary biplot is given in Figure 1. Its basic characteristics are the lines (or rays) and the dots. Rays represent three variables, (labeled with A, B, and C) and dots represent the respondents (the respondents are labeled by 1, 2, 3 and 4). The origin O represents the center of the compositional data set. An important characteristic of the plot is the angle between the rays. The interpretation of a ray is as follows [28]: the length of a ray shows the variance of the corresponding variable. Longer rays depict higher variances. Inferring from Figure 1. the variable corresponding to ray A shows the highest variance among the variables in the biplot, while the variable corresponding to B has the lowest variance. A link is an imaginary line connecting the ends of two rays. It essentially shows the difference between the two variables. Large links show large proportional variation. For example, in Figure 1. the lengths of the links show that the log-ratio having the largest variation is A/C, which is interpreted as a large difference (or dissimilarity or divergence) in the values of these two variables which are proportions. It is essential to emphasize that, in terms of interpretation, links are considered more important than rays since the variables can be examined in a more relative and intuitive manner.

Finally, the cosine of the angle between the rays approximates the correlation between the CLR transformations of the variables. The closer the angle is to 90°, or to 270°, the smaller the correlation. An angle near to 0° or 180° reflects strong positive or negative correlation respectively.

The position of the dots with respect to the links indicates the relative values of the variables. For example, in Figure 1. the dot for respondent #2 has a higher value in variable A compared to variable B or C. Finally, the projection of a specific dot to a variable line estimates the value of that project on the variable that the line represents. If the projection falls on the origin, the value of the project is roughly the average of the respective variable. Projections of the dots which intersect the variable line in the same direction indicate high values, while projections which intersect the variable line, which have been extended through the origin, represent low values. Therefore, respondent #2 stands out with the highest value in variable A followed by respondent #4, respondent #1 that seem to have an average value for variable A, and finally respondent #3 seems to have the lowest (or below average) value for variable A.

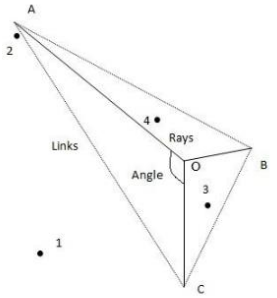

Figure 1. The basic characteristics of a biplot

IV. Results

A. Descriptive Statistics

The results from the descriptive statistics are available at Table II and Figure 2. Table II summarizes the answers of the practitioners in absolute numbers and under two perspectives, the number of the practitioners that choose an attribute and the total points an attribute received from all the practitioners. The practitioners were free to select any number of attributes. The attributes are in descending order based on the practitioners' choices.

The analysis showed that *Cost* is clearly considered the most important attribute between the practitioners when making CSO decisions, selected by 121 out of 157 practitioners (77%). The inputs that are also considered important at the decision process and are mentioned by roughly half of the practitioners are: *Support of the Components* (74 out of 157, 47%), *Longevity prediction* (71 out of 157, 45%) and *Level of off-the-shelf fit to product* (62 out of 157, 40%). An interesting finding is that *Support of a component* is the second most popular choice between the practitioners. However, it is 4th in the total points received and the maximum amount of point it got from a practitioner is 50. In other words, even the practitioners did not assign high values to this choice, still they consider it an important factor that should be taken into account. The same is true for *Longevity prediction* (maximum amount of points: 60, however it is second in the total amount of points received).

On the other hand, *Size* is rarely selected, only by 18 out of 157, (9%). The number of *Other* is also low (18 out of 157, 9%) which reveals that the list of the given attributes covers the most important criteria needed to evaluate a component by the practitioners. Among the *Other* answers, the most frequent responses included licensing issues and long-term strategy such as differentiation on the market and vendor relations. However, these were responses mentioned by just

18 practitioners and thus do not raise any threats to the attributes selected to be included in the survey.

TABLE II. DESCRIPTIVE STATISTICS

Attributes	***Min***	***Max***	***Number of respondents***	***Total points***
Cost	0	100	121	3935
Support of the component	0	50	74	1343
Longevity prediction	0	60	71	1513
Level of off-the-shelf fit to product	0	75	62	1499
API adequacy	0	100	56	1195
Access to relevant documentation	0	100	53	1050
Code quality	0	100	53	1280
Adherence to standards	0	50	42	833
Programming language Performance	0	100	41	1112
Complexity	0	50	33	695
Size	0	60	18	435
Other	0	100	18	810

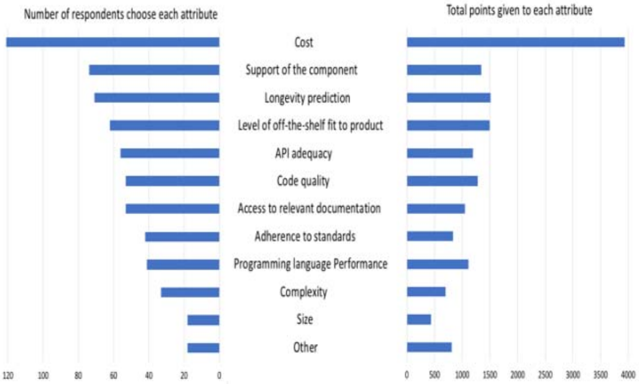

Figure 2. Number of respondents VS Total points choose each attribute

B. Non- Parametric tests

The prioritized data was transformed with the methods of CoDa described in the previous section (replacement of zeros and CLR transformation) and Kruskal – Wallis, a nonparametric test [29], was applied to the obtained data. The test was performed in order to investigate the distribution of each attribute across to all the demographic characteristics. Pairwise comparisons were also performed using Dunn's [30] procedure with a Bonferroni correction [31] for multiple comparisons. However, the results revealed that there are no statistical differences between:

- the different roles of the practitioners,
- the practitioners with different working experience, and
- the education of the practitioners

and the way the practitioners prioritize the 12 aspects.

Therefore, in order to explore trends and peculiarities among our population, we need a powerful descriptive tool designed for this nature of the data: the biplot.

C. CoDA Analysis – Biplot

1) All practitioners

Figure 3. illustrates the biplot for the prioritization of the 12 aspects by all the 157 practitioners. The practitioners are represented by dots while the rays represent the 12 aspects. The results showed that there is a wide spread of the dots in all the axes, and long rays (thus long links too) for most of the aspects. These indicate high dissimilarity between the practitioners, large variability among the aspects and some interesting correlations between variables. The biplot clearly depicts the high level of complexity of the decision process and can illustrate how dissimilarly the practitioners prioritize each choice. More specifically:

- The longest rays correspond to *Level off-the-shelf fit to product*, *Code quality*, *Cost* and *Longevity prediction* indicating the aspects with the highest variance. In other words, practitioners allocated values from 0 to 100 to those attributes.
- The longest links are the ones between *Cost* and *Access to documentation*, between *Longevity prediction* and *Code quality* and between *Level off-the-shelf fit to product* and *Adherence to standards*. Therefore, the largest differences, considering all aspects together, are located between these pairs of variables which also seem to be negatively correlated (due to the nearly 180 angles that they have). For example, if a practitioner chooses to allocate more monetary points to *Cost* then we assume that he/she will assign almost 0 points to *Access to documentation* and the other way round.
- The shortest link connects the ray ends of *Size* and *Complexity* and indicates that the distribution in those two aspects is quite similar and positively correlated (due to the nearly zero angle that they have). *Longevity Prediction* and *Support of a component* seem to be positively correlated too. In other words, the practitioners tend to allocate the same amount of points for the aforementioned aspect. For example, *Size* and *Complexity*.
- The nearly orthogonal pairs of variables (the rays that form with each other right angles), for example the ones corresponding to *Cost* and *Code quality* or *Level off-the-shelf fit to product* indicate correlation of these aspects with *Cost* close to zero, which means that we can claim that the way a practitioner allocates the points for the two above-mentioned attributes is not related either positive or negative.

Regarding the distribution of the practitioners with respect to the attributes prioritized, there are areas of high density as well as areas of low density. This means that groupings of practitioners exist, having differences in their attributes prioritized. For example, there is a group of high density near *Cost* which means that a large number of practitioners have assigned larger proportion of effort to the *Cost*.

At the next step, a deeper analysis on practitioners' decision process was conducted based on their inherent characteristics. The characteristics that were investigated were related to the role a practitioner holds in the company, their general working experience and educational level, the maturity of the product they are working on and the size of the company they belong to. Each group of practitioners was aggregated by computing the mean value of their preferences. In the following figures the groupings of the practitioners' inherent characteristics appear in the dots in italics.

2) Role

Figure 4. illustrates a biplot of the practitioners grouped by their role within the company. It is clear that external and employees working with legal issues are having completely different aspect prioritizations from the rest of the employees. More specifically they are deciding on changing a component based on other issues i.e. licensing issues or vendor relations.

On the other hand, the practitioners who work with management (strategic and operational) but also product developers, are deciding the change of a component based on issues related to the *Level of off-the-shelf fit to product* and *Code quality*.

3) Working Experience

From Figure 5. it is clear that the employees having small working experience are interested more on issues related with *Level of off-the-shelf fit to product*, *Access to the relevant documentation* and *API adequacy*. On the contrary, more experienced employees focus more on the *Code quality and the Support of the component.*

4) Educational level

The practitioners with university education seem to consider similar attributes for their decision process (i.e. *Level of off-the-shelf fit to product* and *Access to the relevant documentation*). *Size* seems also to be considered among the practitioners with academic background. Practitioners attended professional courses and trade school education, when they are choosing a new component, they consider more important as input attributes related with development and the usage of the new component i.e. *Complexity, Programming language performance* and *Code Quality*. (Figure 6.).

5) Maturity of the product

From Figure 7. we can claim that the practitioners who work on more mature products (more than 15 years) giving more emphasis on non-functional attributes (i.e. *Size*). However, *Cost* is still their first priority. On the other hand, the practitioners who work with less matured and newly established products (less than 10 years) seem to be more interested in *Complexity* and *API adequacy*, and they are not focused so much on the *Cost*.

6) Size of the company

Regarding, the size of the company, a practitioner works at, the results are available in Figure 8. It seems that smaller organizations are focusing more on development and maintenance of the component (*Complexity, API Adequacy* and *Access to relevant documentation*). On the other hand, bigger organizations focus on properties associated with *Cost*.

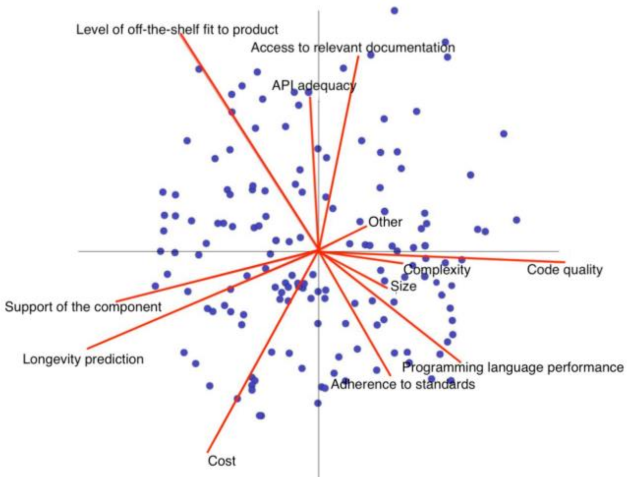

Figure 3. All practitioners prioritize all attributes

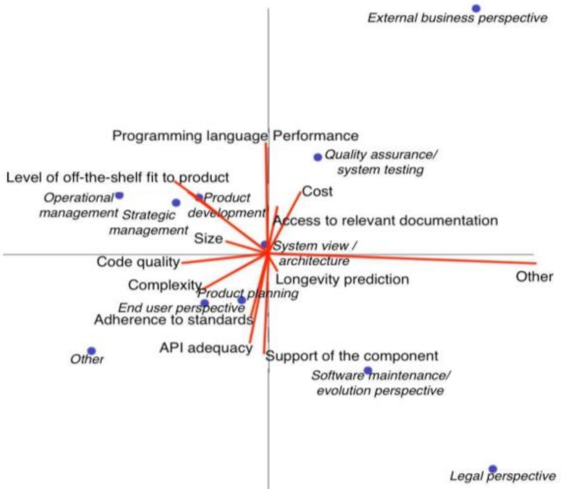

Figure 4. Practitioners grouped by their role

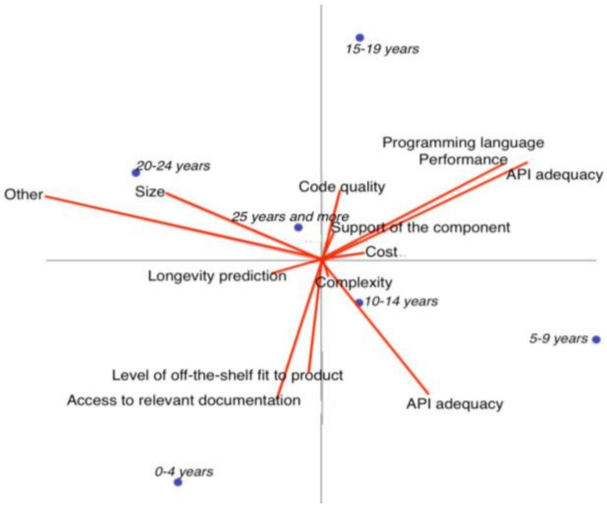

Figure 5. Practitioners grouped by their working experience

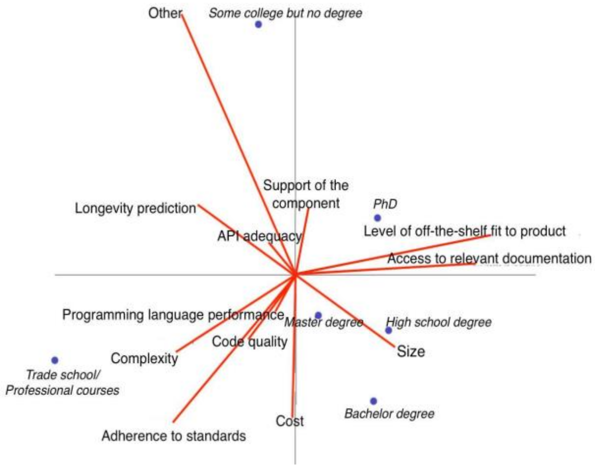

Figure 6. Practitioners grouped by their education

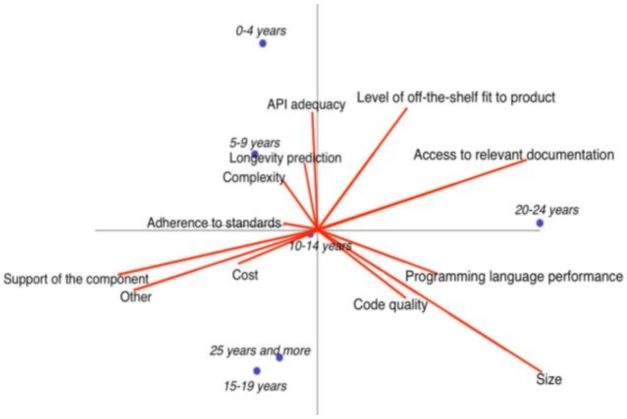

Figure 7. Practitioners grouped by the maturity of the product they are working with

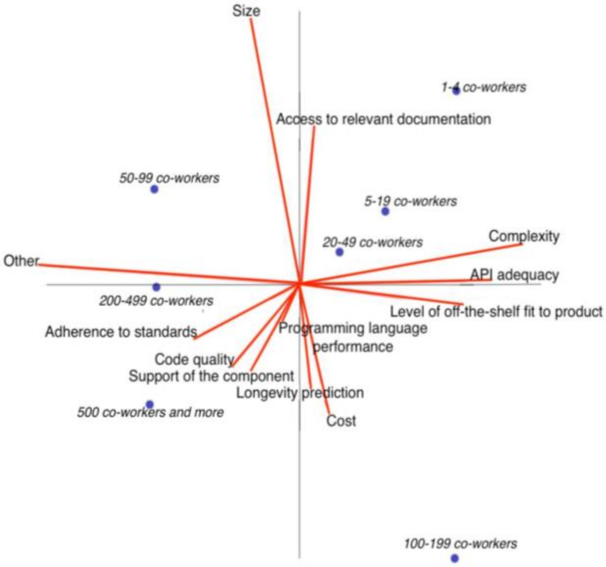

Figure 8. Practitioners grouped by the size of their company

V. Conclusions

The present study focuses on the investigation of what matters the most to industry practitioners during component selection. The descriptive results showed that *Cost* is clearly considered the most important attribute during the component selection. Other important attributes for the practitioners were: *Support of the component*, *Longevity prediction*, and *Level of off-the-shelf fit to product*. However, after more detailed analysis based on practitioners' inherent characteristics, it seems that smaller organizations and more immature products focus on properties associated with ease of use, development and maintenance of the component. On the other hand, bigger organizations and more mature products focus more on properties associated with cost. Therefore, smaller companies need support in order to identify components more easily that have more ease of use regarding the development. On the other side of the spectrum, bigger organizations with mature products need and are looking for less costly components.

The data gathered in such studies are affected by various sources of variation and are therefore subject to large variability. The statistical analysis of such data can reveal significant differences, trends, disagreements and groupings between the practitioners and can constitute a valuable aid for understanding the attitudes and opinions of the interviewed persons and therefore a tool for decision making.

Regarding the validity threats, and more specifically construction validity, since the practitioners work for different organizations and products, their component selection may differ; however, the deeper analysis on practitioners' decision process based on their inherent characteristics was valuable. Overall, the study is clearly exploratory and by no means can the findings be generalized to an isolated situation or company. The practitioners do not constitute a random sample; however, they were approached for their experience and expertise, so their responses are considered valid.

As a future work, we will investigate how do the preferences on specific CSOs affect the prioritization of the aspects investigated in this work, which are seen in isolation. Our focus is in providing support to companies in improving their component selection process. Within our work we plan to continue research and efforts towards efficient and effective decision making in component-based software engineering.

VI. Acknowledgments

The work is partially supported by a research grant for the ORION project (reference number 20140218) from The Knowledge Foundation in Sweden. The authors have no competing interests to declare.

VII. References

[1] Wohlin, C., Wnuk, K., Smite, D., Franke, U., Badampudi, D., & Cicchetti, A. (2016, June). Supporting strategic decision-making for

selection of software assets. In International Conference of Software Business (pp. 1-15). Springer, Cham.

[2] Leffingwell, D. and D. Widrig, Managing software requirements: A Use Case Approach, 2nd ed. Addison-Wesley, Boston, 2003.

[3] Petersen, K., Badampudi, D., Shah, S., Wnuk, K., Gorschek, T., Papatheocharous, E., ... & Cicchetti, A. (2017). Choosing Component Origins for Software Intensive Systems: In-house, COTS, OSS or Outsourcing? -A Case Survey. IEEE Transactions on Software Engineering.

[4] Borg, M., Chatzipetrou, P., Wnuk, K., Alégroth, E., Gorschek, T., Papatheocharous, E., Shah, SMA. & Jakob Axelsson, J., (2018). Selecting Component Sourcing Options: A Survey of Software Engineering's Broader Make-or-Buy Decisions. Journal of Systems and Software (under revision).

[5] Badampudi, D., Wohlin, C., & Petersen, K. (2016). Software component decision-making: In-house, OSS, COTS or outsourcing-A systematic literature review. *Journal of Systems and Software*, *121*, 105-124.

[6] Brownsworth at al. (2000), Brownsword, L., Oberndorf, T., & Sledge, C. A. (2000). Developing new processes for COTS-based systems. *IEEE software*, *17*(4), 48-55.

[7] Li, J., Bjørnson, F. O., Conradi, R., & Kampenes, V. B. (2006). An empirical study of variations in COTS-based software development processes in the Norwegian IT industry. *Empirical Software Engineering*, *11*(3), 433-461.

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

[9] Li, J., Conradi, R., Slyngstad, O. P. N., Bunse, C., Torchiano, M., & Morisio, M. (2006, May). An empirical study on decision making in off-the-shelf component-based development. In *Proceedings of the 28th international conference on Software engineering* (pp. 897-900). ACM.

[10] B. Regnell, M. Host, J. Natt och Dag, P. Beremark, and T. Hjelm, "An industrial case study on distributed prioritisation in market-driven requirements engineering for packaged software", Requirements Eng, 2001, 6:51–62.

[11] P. Berander, and C. Wohlin, "Difference in views between development roles in software process improvement – A quantitative comparison", Empirical Assessment in Software Engineering (EASE 2004).

[12] D. Firesmith, "Prioritizing requirements", Journal Of Object Technology, Vol. 3, No.8, September-October 2004.

[13] M. Staron, and C. Wohlin, "An industrial case study on the choice between language customization mechanisms", J. Münch, and M. Vierimaa (Eds.): PROFES 2006, LNCS 4034, pp. 177 – 191, 2006. Springer-Verlag Berlin Heidelberg, 2006.

[14] S. Hatton, "Choosing the "right" prioritisation method", 19th Australian Conference on Software Engineering.

[15] Chatzipetrou, P., Angelis, L., Rovegard, P., & Wohlin, C. (2010, September). Prioritization of issues and requirements by cumulative voting: A compositional data analysis framework. In Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference on (pp. 361-370). IEEE.

[16] Chatzipetrou, P., Papatheocharous, E., Angelis, L., & Andreou, A. S. (2015). A multivariate statistical framework for the analysis of software effort phase distribution. Information and Software Technology, 59, 149-169.

[17] Chatzipetrou, P., Papatheocharous, E., Angelis, L., & Andreou, A. S. (2012, September). An investigation of software effort phase distribution using compositional data analysis. In *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on* (pp. 367-375). IEEE.

[18] K. Pearson, "Mathematical contributions to the theory of evolution: on a form of spurious correlation which may arise when indices are used in the measurements of organs", Proc. Roy. Soc. 60, 1897, pp.489-98.

[19] J. Aitchison, "The statistical analysis of compositional data (with discussion)", J. R. Statist. Soc. B, v.44,1982, pp. 139-177.

[20] Aitchison, J. "The statistical analysis of compositional data", London, 2003, The Blackburn Press.

[21] Comas-Cufí, M., & Thió i Fernández de Henestrosa, S. (2011). CoDaPack 2.0: a stand-alone, multi-platform compositional software.

[22] J. A. Martín-Fernández, C. Barceló-Vidal, and V. Pawlowsky-Glahn, "Zero replacement in compositional data sets", In H. Kiers, J. Rasson, P. Groenen and M. Shader (Eds.), Studies in Classification, Data Analysis, and Knowledge Organization, Proceedings of the 7th Conference of the International Federation of Classification Societies (IFCS'2000), pp. 155–160. Berlin, Springer-Verlag.

[23] J. A. Martín-Fernández, C. Barceló-Vidal, and V. Pawlowsky-Glahn, "Dealing with zeros and missing values in compositional data sets using non-parametric imputation", Mathematical Geology, 35(3), 2003, pp.253–278.

[24] J.A. Martín-Fernández, J. Palarea-Albaladejo, and J. Gómez-García, "Markov chain Monte Carlo method applied to rounding zeros of compositional data: first approach", In S. Thió-Henestrosa and J.A. Martín-Fernández (Eds.), Proceedings of CODAWORK'03 - Compositional Data Analysis Workshop,2003,ISBN 84-8458-111-X. Girona.

[25] K. R., Gabriel, "The biplot-graphic display of matrices with application to principal component analysis", Biometrika 58,1971, pp. 453-467.

[26] K. R., Gabriel, "Biplot display of multivariate matrices for inspection of data and diagnosis", In: V. Barnett, Ed., Interpreting Multivariate Data, Wiley, New York, 1981, 147-173.

[27] Aitchison, J., & Greenacre, M. (2002). Biplots of compositional data. Journal of the Royal Statistical Society: Series C (Applied Statistics), 51(4), 375-392.

[28] J. Aitchison, and K.W. Ng., "Conditional compositional biplots: theory and application", 2nd Compositional Data Analysis Workshop CoDaWork'05, 2005, http://ima.udg.edu/Activitats/CoDaWork05/CD/Session1/Aitchison-Ng.pdf

[29] Kruskal, W. H., & Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. Journal of the American Statistical Association, 47(260), 583-621.

[30] Dunn, O. J. (1964). Multiple comparisons using rank sums. Technometrics, 6(3), 241-252.

[31] Dunn, O. J. (1959). Estimation of the medians for dependent variables. The Annals of Mathematical Statistics, 192-197.