Equality in Cumulative Voting: A Systematic Review and Improvement Proposal

K. Riņķevičs[a], R. Torkar[a,b]

[a] *Blekinge Institute of Technology, Sweden*
[b] *Chalmers University of Technology and University of Gothenburg, Sweden*

Abstract

Context. Prioritization is an essential part of requirements engineering, software release planning and many other software engineering disciplines. Cumulative Voting (CV) is known as a relatively simple method for prioritizing requirements on a ratio scale. Historically, CV has been applied in decision-making in government elections, corporate governance, and forestry. However, CV prioritization results are of a special type of data—compositional data.

Objectives. The purpose of this study is to aid decision-making by collecting knowledge on the empirical use of CV and develop a method for detecting prioritization items with equal priority.

Methods. We present a systematic literature review of CV and CV analysis methods. The review is based on searching electronic databases and snowball sampling of the found primary studies. Relevant studies are selected based on titles, abstracts, and full text inspection. Additionally, we propose Equality of Cumulative Votes (ECV)—a CV result analysis method that identifies prioritization items with equal priority.

Results. CV has been used in not only requirements prioritization and release planning but also in e.g. software process improvement, change impact analysis and model driven software development. The review presents a collection of state of the practice studies and CV result analysis methods. In the end, ECV was applied to 27 prioritization cases from 14 studies and identified nine groups of equal items in three studies.

Conclusions. We believe that the analysis of the collected studies and the CV result analysis methods can help in the adoption of CV prioritization method. The evaluation of ECV indicates that it is able to detect prioritization items with equal priority and thus provide the practitioner with a more

Preprint submitted to Information and Software Technology *June 21, 2012*

fine-grained analysis.

Keywords: Cumulative voting, prioritization, requirements engineering

1. Introduction

Software products are becoming larger and more complex. Each product is usually affected by a large number of factors such as functional requirements, quality attributes, or software process improvement issues. Since time, funding, and resources are limited, it is seldom possible or even desirable to fully address all the factors. Therefore, the level of attention to a particular factor should be decided according to its importance (e.g. business value), cost, risk, volatility, dependencies between the factors and other such criteria. These type of decisions are made by product stakeholders: users, clients, managers, sponsors, developers, and other persons associated with the product. In order to make decisions regarding a large number of factors it is highly advisable to prioritize the factors in a systematic way [1].

Prioritization is commonly used in requirements selection and release planning. First, project stakeholders prioritize software requirements. Priority values then can be used to determine the order in which the requirements are going to be implemented. Requirements with higher priority could be implemented early while requirements with lower priority may be postponed for later releases or left out.

Another example could be prioritization of potential security threats. It is done by security professionals, software developers and system administrators to assess the level of risk and to select risk mitigation activities.

One of the prioritization methods used in software engineering is Cumulative Voting (CV) [2]. The main advantage of CV is that it is relatively simple and fast, yet produces priorities in ratio scale [1, 3]. This allows us to not only determine what prioritization items are more important but also how much more important they are. (Ratio scale prioritization is particularly important in software release planning and cost-value analysis [4, 5].)

Prioritization is usually performed by multiple stakeholders where individual priorities are combined into a single priority list. Each stakeholder's preferences may have different weight in the final priority. Such prioritization provides more information than just the priorities of factors. In the end, it may be useful to analyze the results of the prioritization to assess disagreement between stakeholders, measure stakeholder satisfaction with the results or find distinct groups of stakeholders.

The purpose of this study is to help industry practitioners and academia researchers in adopting, using and developing CV, while the importance of prioritization in software engineering and the prospectiveness of CV constitutes a need to do further research in this area.

This study presents a systematic literature review on the empirical use of CV and CV result analysis methods. CV results correspond to special type of data – compositional data. Principles of compositional data analysis are described in this paper. A new method for CV result analysis, called Equality of Cumulative Votes (ECV), is proposed. The method identifies prioritization items with *equal* priority. ECV is evaluated using a considerable amount of data, which was obtained from the primary studies identified by the systematic review (through the kindness of the authors of said studies).

2. Background

This section presents definitions and places this study in a context. In the coming sections we will cover: a description of software requirements prioritization methods; examples of CV result analysis methods; and a description of compositional data analysis and CV.

2.1. Prioritization Methods

Some of the most popular prioritization methods are the analytical hierarchy process (AHP), cumulative voting (CV), ranking, numerical assignment, top-ten, the planning game, minimal spanning tree, bubble sort and binary search tree [1, 6]. Ranking and numerical assignment methods perform prioritization on an ordinal scale. AHP and CV are, on the one hand, considered to be harder to use and also more time consuming compared to other methods but, on the other hand, produce priorities in ratio scale.

Ratio scale priorities have several advantages over ordinal scale priorities. Ratio scale shows not just the order of items but also relative distance between them. This enables the priority of a group of items to be calculated by summing up the priorities of individual items [4]. It is possible to say that one item or set of items has higher priority than another set of items. Supposing stakeholders have to choose between several low priority items and one item with higher priority; with ordinal scale, the item with highest priority will always be selected first. However, if priorities are given on a ratio scale, it is possible that lower priority items will be selected if their cumulative priority is higher.

Finally, the ratio scale allows the combining of multiple priority factors by calculating ratios between them. One example of this is the cost-value ratio that shows which requirements give more value for less money [5].

2.2. Prioritization Result Analysis

Disagreement between stakeholders happens when two or more stakeholders have assigned a different priority to one prioritization item. If the level of disagreement is high it may indicate potential conflicts between stakeholders. Such conflicts may be of technical character, as well as social or cultural.

The satisfaction a stakeholder has with the final prioritization results is determined by the difference between the results and the individual priorities of the stakeholder. A smaller level of difference leads to higher satisfaction. In the end, stakeholder satisfaction is important because it is necessary to achieve stakeholder commitment.

In some cases a part of stakeholders may form a group of some kind and, therefore, prioritize requirements similarly. It may be useful to detect whether a group of stakeholders has different preferences compared to other stakeholders. As an example, in [7], domain experts, technical experts, managers, project managers, testers, and developers use CV to prioritize software process improvement issues and the CV results are analysed using disagreement charts and satisfaction charts. Finally, principal component analysis (PCA) is used to identify distinct groups of stakeholders.

The same items can be prioritized by the same stakeholders multiple times from different perspectives. In this case it is useful to determine correlation between the priorities in different perspectives to assess the differences between the perspectives. As an example, in [8], CV is used by developers, testers and managers to prioritize quality attributes. The same quality attributes are prioritized from two perspectives: the perceived situation today and the perceived ideal situation. Correlation between the two perspectives is evaluated using the Spearman rank correlation matrix. This allows an analysis of how well the company balances the priorities of software quality attributes.

In [9] change impact issues are prioritized by developers, testers, managers, and system architects. The prioritization is done with respect to three perspectives: strategic, tactical, and operative. In order to determine correlation between the perspectives, CV results are analyzed using the Kruskal-

Wallis test. In [10] the results of [9] are further analyzed using PCA, bi-plot, and ternary plot. In this case, PCA is used to find correlated issues, bi-plot shows variance, correlation, difference between the priorities of issues, and the viewpoints of stakeholders, while ternary plots are used to show the relative number of issues that received high, medium, and low priority.

As can be seen above, from the examples above, prioritization has been performed with various stakeholders, using different perspectives and, in the end, also analyzed using various techniques. We will next describe in more detail one of the more common methods to manage prioritization issues—cumulative voting—which has been used in software engineering for some time. (CV has its roots in corporate governance and biology.)

2.3. Cumulative Voting

CV is a prioritization method for prioritizing a list of items [2]. CV has many synonyms in literature: hundred (100) dollar ($) method/test and hundred (100) point method. Before being applied in software engineering CV was used for political elections [11] and corporate governance [12]. CV has also been applied in e.g. decision making in forestry [13], voting in social networks [14] and in computer algorithms for consensus clustering [15] (as a method for combining the results of different clustering algorithms).

In CV a stakeholder is given 100 points, imaginary dollars or units of percentages that can be spent on the prioritization items. In the simplest case, the stakeholder can spend any amount of points on any number of items as long as the total amount adds up to 100. The more points assigned to an item, the higher the priority of the item (and implicitly, the lower priority to the other items). The stakeholder may spend all points on just one item or distribute them among all or some of the items. Once again, this is the simplest case; other variants exist, which we will see next.

Often prioritization is done by more than one stakeholder. The final priority of an item can be calculated by adding up the points each stakeholder has spent on it. Sometimes the vote of some stakeholders may be more important than the votes of others. For example, a manager may be more influential or shareholders may have different amount of shares. In such a case the priorities of each stakeholder may be multiplied by an individual coefficient or a stakeholder may be given a more points to perform the prioritization.

Worth mentioning in this context is that it is advisable to randomize the order of items in a prioritization list. This is necessary in order to minimize the effect of order on the prioritization results, which has shown to have an effect [16].

2.3.1. Benefits and Drawbacks of Cumulative Voting

Compared to analytical hierarchy process (AHP), CV is faster and easier to learn and use [1, 3]. AHP benefits from consistency check, but CV does not require this because all prioritization items are evaluated simultaneously [3].

There are, however, a few problems with CV. First of all, it cannot be repeated for the same stakeholders and prioritization items due to stakeholder bias [2] (c.f. Section 2.3.4). Secondly, CV becomes more difficult to use when the number of prioritization items increases [17].

2.3.2. Example of Cumulative Voting with Several Stakeholders

Let us next give an example of CV with several stakeholders. Suppose Robin, Alice, and John are three friends who want to buy some beverages in a store. They have different preferences but do not want to buy too many drinks. Therefore, they decide to use CV to decide what to buy. Each of the friends distributes 100 points between four items: milk, tea, coffee, and juice (Step 1 in Figure 1). In this case each of them will spend a different amount of money on the purchase, hence, their priorities are multiplied by different coefficients (Step 2 and the stakeholder importance coefficient in Figure 1). The final beverage priorities are calculated by summing up the weighted priorities of stakeholders (Step 3 in Figure 1).

2.3.3. Stakeholder Bias

Prioritization using CV may be biased if a stakeholder knows the preferences of other stakeholders. She may manipulate the results by spending more points on items that are important to her but not to the other stakeholders. On the one hand, stakeholder bias makes it unreasonable to repeat CV with the same prioritization items and stakeholders. On the other hand, this property of CV may be useful in giving more power to important minority stakeholders, such as security experts or software testers. Suppose the same software requirements are prioritized for a second time using CV. A developer might know that all vital functionality is selected by other stakeholders, but his toy feature is left out. In effect, the developer could spend all his points on this feature to put it in the next release.

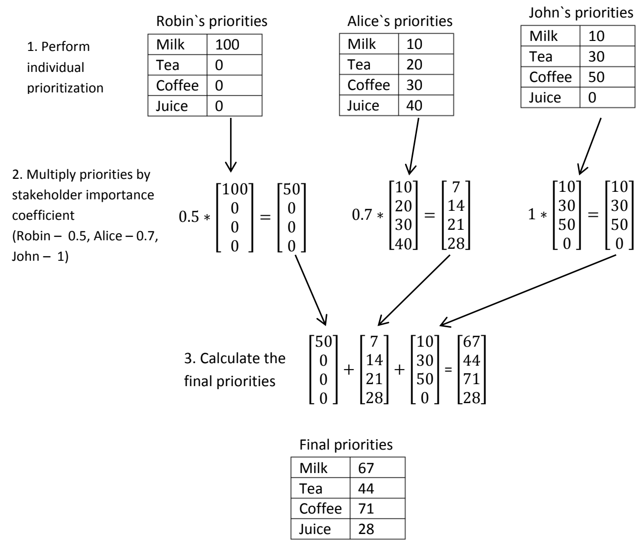


Figure 1: Example of CV with several stakeholders.

Stakeholder bias may be mitigated by setting a maximum priority that can be assigned to an item. This way each stakeholder is forced to distribute the money between several prioritization items [4].

Another bias is that people in general tend to assign round priority values. This is likely caused by lack of objective judgement criteria. Either way it seems to be a problem not acknowledged by many since all prioritization is largely based on expert opinion.

2.3.4. Scalability of Cumulative Voting—Hierarchical Cumulative Voting

The standard CV approach has a low scalability. If the number of prioritization items is high, stakeholders may lose sight of the bigger picture and assign priorities to a limited number of items. One, unsophisticated, solution to the problem is to provide more points for prioritization (1,000 or 10,000 instead of 100); however, one could take another approach.

When the number of prioritization items is high they can usually be grouped hierarchically by forming a tree structure (Figure 2) and, thus, parent-child dependencies will exist between many items.

In [4] the authors propose a method for prioritizing hierarchically structured items called Hierarchical Cumulative Voting (HCV). It may be seen as combination of the hierarchical part of the Analytical Hierarchy Process

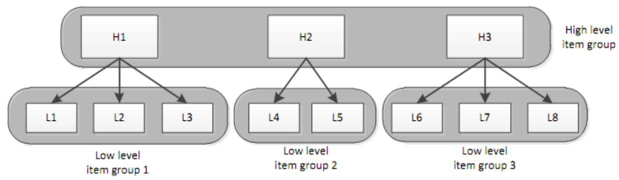


Figure 2: Example of prioritization item hierarchy.

(AHP) [1, 18] and the CV prioritization method. Since items are prioritized in smaller sets, stakeholders do not lose sight of the bigger picture during prioritization, and the prioritization of a large number of requirements is considered easier.

2.3.5. Compensation Factors

HCV deals with the problem of prioritization scalability but it comes at a cost. Low level item groups may consist of different numbers of items, but the number of points spent on each group is the same, i.e. in a small-sized group, the same amount of points is distributed among fewer items. Hence, items in smaller groups are statistically more likely to have a higher priority, on average, compared to items in larger groups. To balance this difference each low level prioritization item can be multiplied by a compensation factor [4].

As an example, suppose an item (A) in a group of 10 items is assigned 60 points. Hence, A will receive 600 compensated points. In this case it is impossible for any item in a group smaller than 6 items to compete with A. Even if item (B) in a group of 5 is assigned the maximum number of points (100), the maximum compensated priority value B can receive is 500.

In [17] the authors suggest that compensated prioritization is more favorable compared to uncompensated. But neither compensated nor uncompensated prioritization is perfect and, as a general rule, it is better to keep the size of prioritization item groups similar.

2.3.6. HCV Execution

According to [4], HCV is conducted with the following steps (Steps 4–5 are optional):

1. Construct hierarchy. Prioritization items need to be divided into one high and several low level item groups. Each low level item group is

child to exactly one high level item. And each high level item has one low level item group. One low level item may belong to several item groups. Even if parts of the items are not logically connected they can be grouped separately and assigned a fake parent item, e.g. 'misc. items'. HCV does not, as far as we know, provide any instructions for creating a requirements hierarchy.

2. Each high and low level item group is prioritized separately using CV. The stakeholder may prioritize all item groups at once or one by one. But it should be possible to prioritize groups in any order and repeatedly, because the stakeholder might learn more about the items while performing the prioritization.

   In particular the stakeholder is likely to learn more about a high level item when prioritizing its low level item group [17]. Some stakeholders may prioritize only part of the groups and each group may be prioritized by different stakeholders.

3. The priority of each low level item is normalized by dividing it with the sum of all low level priorities of each item in all groups.

4. The final priority of each low level item is calculated by multiplying it with the priority of its parent high level item.

5. Then one applies the compensation factor to all low level requirements as described in Section 2.3.5.

6. Finally, when multiple stakeholders have performed the prioritization, priorities of low level items are combined as in standard CV.

It is possible that one low level item is child of more than one high level requirement and, thus, belongs to two or more low level requirement groups (see Figure 3). Such requirements participate in the standard HCV prioritization process and are prioritized two or more times with each group they belong to. At the end of the prioritization they receive several priority values. These values can be summed together to form the final priority of the item. (This is done because the item adds value to both parts of the hierarchy.)

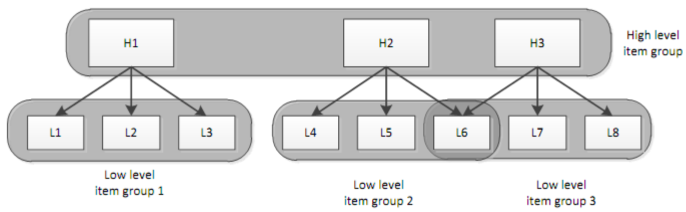


Figure 3: Overlapping prioritization item hierarchy example.

Table 1: Example of hierarchical cumulative voting.

Phone requirements	Compensation factor	Sub-requirements	Priority calculation	Final priority
Calendar	2	Reminder alarm	40*30*2	2400
Calendar	2	Specify repeated event	60*30*2	3600
Phonebook	3	Hide contact	40*20*3	1600
Phonebook	3	Add picture	20*20*3	800
Phonebook & Call	3 & 2	Search contact	40*20*3 + 50*50*2	7400
Call	2	Video call	50 * 50 * 2	2500

2.3.7. Example of Hierarchical Cumulative Voting

Suppose six requirements for a mobile phone operating system need to be prioritized: ‘reminder alarm’, ‘specify repeated event’, ‘hide contact’, ‘add picture to phonebook’, ‘search contact’, ‘make video call’. Three high level requirements can be identified: ‘Calendar’, ‘Phonebook’, ‘Call’. The low level requirements are then grouped as sub-requirements of high level requirements as shown in Figure 4. The ‘Search contact’ requirement is a sub-requirement and has two parent requirements:‘Phonebook’ and ‘Call’. The computation of the final priorities of requirements is shown in Table 1.

After requirements are grouped, and a hierarchy is defined, each group of requirements are then prioritized using CV. The final priority of a low level requirement is computed by multiplying the priority of the requirement with the priority of its parent high level requirement and the compensation factor. The compensation factor in this particular case is the number of elements in a group, two for the ‘calendar’ and ‘call’ sub-requirements and three for the ‘phonebook’ sub-requirement.

2.4. Compositional Data Analysis

CV results can be seen as a special type of data, i.e. compositional data. Compositional data does not contain absolute values. It shows only the

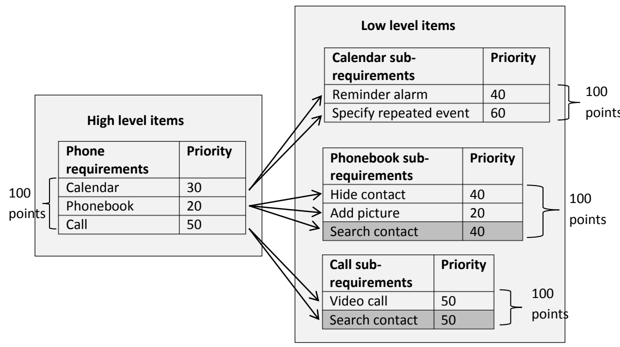


Figure 4: Example of hierarchical cumulative voting with requirement hierarchy.

relative weight of a component compared to the whole. In [10] the authors propose the use of compositional data analysis for the statistical analysis of CV.

A compositional data item is a vector (x) of positive components with a constant sum k:

$$x = (X_1;\, X_2;\, \ldots;\, X_n)\, where\, x_i \geq 0\, and \sum_{j=1}^{n} x_j = k. \tag{1}$$

The property of the sum of the items being restricted is called the constant sum constraint. In CV, priorities assigned by a stakeholder to the items of a prioritization set is a compositional data vector with a constant sum of 100. The value of k (i.e. 100 in this case) is arbitrary and does not affect the analysis of the data because the information is contained in the ratios between the components of the vector. The vector can sum up to any number but still hold the same data, i.e. vectors (1, 2, 7) and (10, 20, 70) are in this case considered equivalent. This principle is called *scale invariance.*

Another property of compositional data items is *subcompositional coherence.* Consider that two compositions are analysed. One composition is a subcomposition of the other. *Subcompositional coherence* means that the results of the analysis are the same for the common parts of the compositions [19]. This property is important for the analysis of HCV results. Statements that are made regarding each smaller group of prioritization items are also true for all items prioritized with HCV.

The priority of an item is relative to the priority of the other items in the set. Hence, the priority of an individual item is meaningless without context, i.e. the complete set of items. The same item may receive different priority when put in two different prioritization sets. If the item is put in a set of items with high priority it will receive a lower relative priority. This also holds true the other way around i.e. if the item is put in a set with low priority items its priority will be higher.

When doing analysis of compositional data one must take into account that compositional data special type of data and should be analysed differently than ordinary data.

When doing analysis of compositional data one must take into account that compositional is a special type of data and should be analysed differently than other data types. Ordinary unconstrained variables are free to take any positive or negative values, whereas, compositional data values can only be positive and have a constrained maximum value. Moreover, components of compositional data vectors are not independent from each other. The fact that an item is assigned 70 priority points means that the next item can take only values between 0 and 30. Hence, there is a negative correlation between the items.

Standard parametric statistical tests require that data vectors have multivariate normal distribution. Vector $X = (X_1, X_2, \ldots, X_n)$ is considered to have multivariate normal distribution if any linear combination of its parts is normally distributed, and linear combination is defined by:

$$Y = a_1 X_1 + a_2 X_2 + \ldots + a X_n, \tag{2}$$

where Y is the product of lineal combination and a_i is any real number. Now, since the sum of priorities assigned in CV must add up to 100, or any other constant number, at least one linear combination of X is not normally distributed because it always adds up to 100:

$$Y = 1 \cdot X_1 + 1 \cdot X_2 + \ldots + 1 \cdot X_n = 100. \tag{3}$$

In our opinion, the above indicates, quite strongly, that CV results do not follow a multivariate normal distribution and, hence, it follows that they should be analyzed using non-parametric statistical tests [20].

2.4.1. Problem of Zeroes

Compositional data analysis requires that log-ratios between any components in a vector can be computed. But computing a log-ratio with a zero value is, in this case, meaningless. This is a problem since CV allows stakeholders to assign zero priorities to some prioritization items (we would even strongly argue that this is very common).

In compositional data there are two types of zeroes: essential and rounded. Essential zeroes mean that a data component is not present. Rounded zeroes mean that the component is present but its value is very low. We, as others have before us, conjecture that zeroes in CV results are rounded because the priority of an item is a completely abstract notion and the instrument for measuring priority is human judgement [10].

Before compositional data analysis can be applied to CV results, we should first remove zeroes in the data. One approach can be to forbid stakeholders to assign zero priorities. This approach is used in e.g. [7]. But this can add some unnecessary complexity to the prioritization process and, explicitly, delimits an expert's freedom. In [10] the authors propose the use of a multiplicative replacement strategy (as defined in [21]) for CV result analysis.

This method replaces rounded zeroes with small values using the expression

$$r_j = \begin{cases} \delta_j, & if\, x_j = 0, \\ (1 - \frac{\sum_{k|x_k=0} \delta_k}{c}) x_j, & if\, x_j > 0, \end{cases} \tag{4}$$

where δ_j is the imputed value and c is the constant sum constraint. In order for the total sum of components to stay constant, the equation subtracts some value from the items with a priority higher than zero. More is subtracted from components with higher values than from components with lower values (and the value of the imputed δ_j is arbitrary).

2.4.2. Isometric log-ratio transformation

In order to apply standard statistical methods to compositional data it should be transformed to remove the inherent correlation of the values. Compositional data analysis proposes special transformations that change the compositional data values to unconstrained real values. One such transformation is isometric log-ratio (*ilr*) transformation (as proposed by [20, 22]).

After compositional data vectors are transformed using zero replacement and *ilr*, any standard statistical tests can be applied.

3. Related Work

Cumulative voting has been studied and applied in various fields. In forestry it is used to take into account opinions of different parts of society while planning forest harvesting [23]. CV is used to vote in government elections [24] and aid decision making in corporate governance [12]. CV is a part of various software algorithms. For instance, in [25] it is used as part of pattern detection algorithm that is used to locate optic nerve in a retinal image.

In software engineering CV has been applied not only requirements engineering and software release planning [26] but also in software security [27], software quality [8], software metrics [28], software process improvement [7], and software verification and validation [29].

Many studies use CV as a research method. For instance, in [30] software impact analysis issues are elicited in structured interviews. Afterwards the importance of each issue is determined with the help of CV.

CV results are compositional data. Principles of compositional data analysis were first defined by J. Aitchison in [31]. Two aspects of compositional data particularly important for the present study are the need to replace zeros in compositional data and transform the data in order to perform statistical analysis. Paper [21] proposes a method to replace zeros and missing values in compositional data. And an important method for compositional data analysis, *ilr*, is proposed in [32].

A systematic review of requirements prioritization methods is presented in [33]. The study focuses on prioritization method comparison and selects eight relevant studies. Two of the studies use CV. These studies are also revealed by the systematic literature review conducted as part of this study. In [33] the author concludes that there is little research on requirements prioritization and studies usually deal with a small number of requirements.

The systematic literature review presented in this paper does not reveal any CV result analysis methods that allows to identify prioritization items with equal priority. Thus, this problem is not addressed in any way.

4. Methodology

This section covers the research questions of this study and the methods used to answer them.

4.1. Selection of Research Methods

The main purpose of this study is to collect knowledge on the use of CV in order to help software engineers and researchers in adopting it.

One way of collecting this knowledge is to conduct an empirical study. A survey in a large number of software companies can be used to quantify the level of adoption of CV in industry (similarly to the study by [34]), while a case study can be used to receive qualitative feedback on the use of CV [35].

Knowledge on the empirical use of CV can also be obtained from existing studies. This may be done by means of a systematic literature review. Several studies have used CV in industry as well as in academic settings. Nevertheless, there are no studies that provide an overview of the current state of the practice in this field (as reported by research studies). Therefore, before continuing with the refinement of CV and conducting new empirical studies (i.e. case study or experiment), a systematic literature review would be required.

This paper proposes a new method for CV result analysis, called Equality of Cumulative Votes (ECV). (ECV groups prioritization items into groups of items with similar priority.) As will be presented later, the systematic review did not reveal any methods that solve this problem; however, ECV needs to be evaluated and, hence, applied to CV results.

There are two options to obtain CV results in order to test ECV. One is to conduct a new empirical study. The second option is to collect CV results from existing studies. The latter approach also has the added benefit of trying to replicate the results from previous studies and, if data from several other studies are used, a larger amount of data can be obtained. Moreover, the generalizability of the evaluation increases when prioritization results from different sources and domains are used. On the other hand, the main benefit of conducting a separate empirical study is the possibility to control the conditions of CV.

In our study we evaluated ECV by obtaining data from previously conducted studies as found by the systematic literature review. In order to obtain the data, authors of relevant primary studies were contacted.

In short, this study consists of two parts: a systematic literature review (SLR) of CV and an evaluation of ECV based on the data from the primary studies found in the SLR.

4.2. Research Questions

The systematic review should focus on catching studies that empirically use CV. Information about place, time, scale, and domain of the studies should be collected and the results of the review will hopefully aid academic researchers by identifying paths for further investigation of CV. Hence, the first research question is:

RQ 1. What is the state of practice in empirical studies that use CV?

The level of trust in research results considering CV is determined by the quality of the studies that use CV, hence this study includes an evaluation of the quality of primary studies identified by the systematic review.

Next, a valuable aspect of decision-making is the analysis of prioritization results. Thus, the second research question is:

RQ 2. What CV result analysis methods have been presented in papers as identified by RQ 1?

Finally, the evaluation of ECV answers the third research question:

RQ 3. Is ECV capable of identifying prioritization items with equal priority?

5. Systematic Literature Review

This section presents the design of the systematic literature review. For the results of the execution please see Section 7.1 and 7.2.

Table 2 presents an overview of activities performed during the systematic literature review. The review protocol was developed by one researcher and evaluated by another researcher. Studies were searched for in two iterations. The first search was performed using databases. The second search was performed using snowball sampling [36] (snowball sampling examines the references of primary studies revealed by the first search). References that are relevant to the review, i.e. they pass the selection criteria, are then added to the set of primary studies.

The search for papers was performed by a single researcher. Study selection, on the other hand, was performed by two researchers. First, one researcher examined all found studies. Next, another researcher re-examined all studies classified as primary studies in addition to 20 randomly selected excluded studies to ensure the quality of the selection.

To ensure the quality of the review, the quality evaluation and data extraction was performed independently by two researchers. Inter-rater analysis was performed using Krippendorf's Alpha statistics [37, 38].

Table 2: Review activities.

Review phase		Researchers involved
Trial search in databases		A
Develop review protocol		A
Evaluate review protocol		B
Paper search and selection from databases	Search in databases	A
	Search string validation	A
	Selection based on metadata	A and B
	Selection based on full text	A and B
Pilot data extraction (3 papers)		A
Paper selection from the reference lists	Selection based on metadata	A and B
	Selection based on full text	A and B
Data extraction		A and B
Data synthesis		A

5.1. Data Sources and Search Strategy

The SLR was designed based on the guidelines by Kitchenham [39]. First a trial search in electronic databases was conducted. In order to scale the review to a manageable, yet sufficient size, databases were searched with different search strings. Relevant papers that were found during the trial search were used to extract additional search strings. The trial search revealed that the number of studies that use CV is not very large. Therefore, we decided to include not only software engineering studies but also studies in other research areas, such as forestry or corporate governance, since one key aspect we intended to investigate was analysis methods for CV.

Since CV is frequently used in studies without mentioning this in the abstract, full text search in databases is preferable. Unfortunately not all databases support full text search. Full text search was performed in the IEEE Xplore and Springer Link databases. In ACM Digital Library, Inspec/Compendex, ISI Web of Knowledge, and SCOPUS only metadata was searched. The search strings used, consisting of a Boolean expression (A or B or C or D or E or F or G), where:

(A) Cumulative voting
(B) 100 dollar method
(C) 100 dollar test
(D) 100 point method
(E) hundred dollar method
(F) hundred dollar test
(G) hundred point method

Search strings contained only synonyms of CV and they did not limit the research area to software engineering. The search was performed independently using each of the search strings in each database. All search results were combined and documented using reference management software. The quality of the search strings and the selection of electronic databases were validated against a previously known core set of papers—[3, 10, 28, 40]—checking that all papers from the core set were found by the search.

5.2. Study Selection

To select relevant papers a set of criteria were designed. The criteria for paper selection are presented in Tables 3 and 4.

Papers were selected in two phases: based on metadata and based on full text.

Table 3: Paper search and selection in the databases.

Selection phase	Inclusion criteria	Number of papers selected
Search in databases	published 2001–2011 (databases last accessed Feb. 20, 2011)	256
	contains search strings	
Selection based on metadata	exclude duplicates and tables of contents	177
	written in English	
Selection based on full text	full text is available	127
	study involves empirical use of CV or presents analysis of empirical use of CV	58
	CV is done by humans and not software	25

Table 4: Paper selection from the reference lists of the selected papers.

Selection phase	Inclusion criteria	Number of papers selected
Selection from references	papers included in the reference lists of relevant papers found in databases	467
Selection based on metadata	written in English	462
	reference is already revealed by search in databases	450
Selection based on full text	full text is available	329
	study involves empirical use of CV or presents analysis of empirical use of CV	15
	CV is done by humans and not software	

Obviously, the main criterion for inclusion of a paper is that it must present empirical use of CV or present an analysis of the results of using CV. However, there are papers that pass this criterion but are not relevant for this review. CV is frequently used in computer algorithms. There is a significant difference between the way humans and computers make decisions. Since this review in concerned with human decisions we excluded papers that present CV that is not performed by humans. In addition, only papers that were written in English were selected and duplicate studies were automatically excluded by the citation management software used in this review. We searched for papers between 2001–2011. By then performing a snowball sampling of these papers we are convinced that we have a representative sample and, futhermore, that the bulk of the studies are relevant from a software engineering perspective.

5.3. Quality Evaluation

The goal of quality evaluation is to determine the best primary studies according to some measure of quality. Since the number of studies that use CV is not large, quality evaluation was not used as an exclusion criterion.

The quality of a study obviously depends on the correctness of the study process including planning, operation, analysis and interpretation of the results (is the study right?) The correctness of the process can be measured by evaluating the description of the study or replicating the study. Thus, to gain the trust of industry practitioners and other researchers, the process of the study should be rigorously described. In short, the description has to facilitate the replication of the study as well as the presentation of limitations and validity threats.

Even the most correct and rigorously described study is useless if it does not contribute to the industry or research community (is it the right study?) The topic of the research ought to address important goals and issues. The findings of the study should also be significant, i.e. there is a high probability of the results of the study are true. The significance of the findings depends on how realistic the study is, the correctness of the process and the results of the study, as well as the statistical significance of the findings.

Realism of a study depends on the context, scale, and subjects of the study. The study should be conducted in a **setting** that is similar or equal to the setting in which the findings of the study are intended to be used. Hence, studies that are conducted in an industrial setting are in many cases valuable. The **subjects** of a study should be similar to the people who are supposed to use the findings of the study. The subjects ought to have appropriate work experience, role in the organization, skills, cultural background, motivation, and so forth. The **scale** of a study refers to the size of the study objects. In the case of this systematic review the scale of a study is measured as the number of prioritization items. Study in academia may have a large number of prioritization items. At the same time, an industrial study, with professionals as subjects, may involve a smaller number of prioritization items.

Each study may have a different level of realism. Some studies involve industry practitioners in an academic setting to simulate real word practice in a laboratory environment. Other studies may involve academic researchers that execute a project. For example, researchers may be developing open source software. On the reality scale these studies are somewhere in between the purely academic and industrial studies.

The **type** of the research study can be considered as a criterion for the evaluation of study realism. Reference [41] suggest that study designs that are more rigorous (e.g. experiments) are more realistic than observational studies (e.g. case study) due to a higher level of control. On the other hand [42] rate study designs based on other criteria, i.e. how frequently each type of study design is used in an industrial or academic setting. If a study design is used more in an industrial setting, then it is considered more realistic. For instance, in software engineering, case studies are frequently used in industrial settings, whereas, experiments are usually performed in academia using students as subjects. Therefore, [42] argue that case studies are more realistic than formal experiments. Obviously the effect of study design on the study realism may be interpreted in different ways. Therefore, we will not use this parameter in our quality evaluation.

The statistical significance of the results of a study can be used to evaluate the significance of the study findings. This measure will not be used, because the studies that are evaluated belong to very different research areas, i.e. the significance levels of the findings of the studies are not directly comparable for meta-analysis. Additionally, sometimes no result is more interesting than a significant result, i.e. it may reveal important gaps in existing knowledge.

The ultimate goal of research, at least in software engineering, is in many cases industry impact. However, most of the time ideas need to be developed and validated in academia before industry professionals will risk to adopt them. Therefore, academic impact is important as well. Academic impact is usually measured by the number of citations. Academic impact is also measured for particular researchers, using the number of papers she has published and the number of times her papers have been cited. This measure will not be used in our quality evaluation because it is somewhat biased. The number of citations is likely to be lower for newer papers and the number of papers that a researcher has published gives little information about the actual quality or impact of her research.

5.3.1. Rating of the Studies

The quality evaluation in our review is based on the evaluation of: (*i*) Study realism. (*ii*) Study scale. (*iii*) Availability of raw results of CV. (*iv*) Quality of the research methodology.

Realism of the studies is rated in three aspects: subjects, setting, and scale. The subjects and setting is rated according to Table 5. The total rating of study realism is determined by summing up the ratings of the two

Table 5: Rating of study reality level.

Aspect	Contribute to relevance (rating 1)	Do not contribute to relevance (rating 0)
Subjects	Industry professionals	Academia students or teachers, or other
Context	Industrial	Academia

aspects. For instance, if a study is conducted with industry professionals as subjects in an academic context the study will receive rating 1 (out of 2 maximal points).

In order to rate the scale of a study the number of prioritization items was counted. If a paper presents several prioritization cases only the prioritization with the largest number of the prioritization items is considered. If HCV is used all of the prioritization items on different levels are counted together. However, if an item is present in several groups in the hierarchy it is counted only once.

The availability of raw results from the application of CV is rated separately because it is especially important for our purposes (and for most other researchers in order to replicate a study). The data availability rating criteria is given in Table 6. If the data of a study is not available it is not possible to validate the results of the study and, hence, the credibility of the findings is lower. Ideally the data collected in the study should be presented directly in the paper. An alternative may be to make the data freely available online and reference the online source.

The quality of the research methodology of a paper is rated according to a checklist presented in Appendix C. The checklist is based on guidelines for presenting research studies (as presented in [43, 44]) and the guidelines for quality evaluation of research studies as presented in [39, 42]. Evaluation is done with regard to the rigor of the description and correctness of the research process and reasoning. Checklist items represent issues that research studies should implement and present in a research paper. The checklist also contains item descriptions or questions that are used to evaluate the quality. Each item in the checklist is rated according to criteria presented in Table 7. The final rating of correctness of the research process of a study is computed by summing up the ratings assigned to all items in the checklist.

Study rating criteria was validated during a trial data extraction. Two researchers each rated three randomly selected papers. Afterwards, differences in ratings were discussed and study rating criteria were updated to avoid differences in interpretation.

Table 6: Research data availability rating.

Rating	Study rating criteria
0	CV results was not provided in the paper and we was unable to obtain the results from the authors.
1	CV results are not provided in the paper but the data was obtained from the authors. Part of the data is lost or corrupted.
2	CV results are not provided in the paper but all the data was obtained from the authors.
3	All CV results are included in the paper or reference is given to online source where all the data can be accessed.

Table 7: Rating of correctness of research process.

Rating	Study rating criteria
0	No description provided.
1	Only basic information is provided about the checklist item. Or significant validity threats exist with regard to this item.
2	Description is sufficient. Some minor questions are left unanswered. Validity threats may exist but they are not likely to affect the results of the study.
3	Description is rigorous and clear. Questions presented in quality evaluation checklist in Appendix C are answered. Decisions of the study are well justified, alternatives are discussed. No unhandled validity threats can be identified.

As a result of the rating each study was assigned four rating values on an ordinal scale. In order to perform a more advanced analysis of the quality evaluation results these ratings were then converted into ratio scale ranks. For each study, the number of studies that had received lower ratings were counted. The resulting number is the rank of the study; thereby, the quality of a study is expressed as four rank values.

An example of rating values is shown in Table 8. Table 9 shows ranking values computed for the studies in Table 8. We can observe that study realism level rating for ST3 is 0. There are no studies that have a lower study realism. Therefore, realism ranking for ST3 is 0. ST1 on the other hand has the highest realism rating. Since ST1 has higher reality level than both ST2 and ST3 it is assigned reality level rank 2.

Table 8: Example of rating values.

Study	Realism	Research data availability	Correctness of research process	Number of prioritization items
ST1	2	0	15	6
ST2	1	3	20	69
ST3	0	3	10	6

Table 9: Example of ranking values.

Study	Reality level	Research data availability	Correctness of research process	Number of prioritization items
ST1	2	0	1	0
ST2	1	1	2	2
ST3	0	1	0	0

5.4. Data Extraction

The goal of data extraction is to understand how and why CV is used and how CV results are analysed in research studies. Ultimately, this will allow us to answer the first and second research questions in our study.

Data extraction was documented with the help of spreadsheet software. Extracted data items are available from [45].

6. Equality of Cumulative Votes

In the previous section we described the execution of the systematic literature review. In order to perform a more thorough analysis later we here present the design of ECV before presenting the results of the systematic literature review. For the results of the evaluation of ECV please see Section 7.3 (ECV is implemented in the *R* programming language [46] and the code can be found at [47].)

In CV stakeholders may assign similar or equal values to several prioritization items. As a result the difference between the items is small. The variation in priorities is caused not only by the difference between prioritization items but also by human error and lack of information. For instance, people tend to simplify the task of prioritization by assigning rounded values to items or giving equal values to several items [48].

During prioritization it may be beneficial to know which items are equal. A common example is software release planning where requirements are distributed among several product releases. If two or more requirements are considered equal they can be interchanged between the releases regardless of their priority. That allows other criteria, such as cost or effort, to be used as sole indicators for planning that particular release.

6.1. Testing Equality of Two Items

There are two ways to determine which prioritization items have similar priority. One approach is to find items that are different and consider other items as equal. Another approach is to find items that are equal.

The first approach uses statistical tests to evaluate differences between e.g. two sample means, in order to determine that two items are different. Samples in this case consist of priorities assigned by all stakeholders to a particular prioritization item. The number of stakeholders that perform the prioritization is frequently small. Hence, the size of the sample is very often too small for statistical tests to detect a significant difference in the tests, thus, identify too many equal items to make any useful conclusions.

ECV, in contrast, uses the second approach. It finds items that are similar and the rest of the items are considered different. This method tests the probability of the difference between the means of two items being smaller than the given value. In short, ECV tests the probability of the means of two prioritization items differing by less than 25%. If the probability is higher than 70% the items are considered equal.

The input to ECV is an $n \times p$ matrix A that contains the raw results of the prioritization. The columns of the matrix represent prioritization items while rows represent stakeholders. ECV performs the following operations for the priorities of each of the two prioritization items:

1. Replace zeroes in CV results.

2. Transform the data using *ilr* transformation.

3. Determine distribution function using kernel density estimation.

4. Use the distribution function to find the probability that the difference between two prioritization items is smaller than 25%.

5. Form groups of equal prioritization items.

Since CV results are compositional data, zeroes in A are replaced with other values. This is done using the multiplicative replacement strategy which is described in Section 2.4.1. Next, two columns are extracted from matrix A to create the new matrix B:

$$B = [a_{*,k} a_{*,l}] \,, \tag{5}$$

where a is an element of matrix A, and k and l are the columns that represent items that are tested for equality, "$*$" denotes all rows of corresponding column.

The ilr transformation is then applied to each row of the matrix B and the new vector C is obtained. The equation for calculating elements of C using ilr transformation is:

$$c_i = ilr(b_{i1}, b_{i2}) = \sqrt{0.5}\log(b_{i1}/b_{i2}), \tag{6}$$

where c_i is the i^{th} element of C and b_{i1} and b_{i2} are the first and second elements in the i^{th} row of B. Each value c_i represents a log-ratio between values of columns k and l. The mean of the values of C can be interpreted as an average log-ratio between the items that expresses the difference between the items.

After the data is transformed into log-ratios statistical test can be applied. The purpose of the test is to determine what the probability is of the relative difference between two prioritization items k and l being less than 25%. Or in terms of log-ratios it means determining the probability of c_i (obtained from priorities assigned to k and l) as being in the range of $\frac{3}{4}$ to $\frac{4}{3}$. Hence, the objective of the test is to determine the probability of the sample mean (i.e. mean value of the items of C) laying between the two values.

The probability that the mean takes a particular value can be expressed in the form of a cumulative distribution function. The probability of the mean being between two values a and b (where a is smaller than b) can be determined by subtracting the probability of the mean being smaller than a from probability of the mean being smaller than b.

However, CV result data may or may not have multivariate normal distribution. If the data is normally distributed a Student's t-test can be used; otherwise, a non-parametric estimation of the distribution function is needed.

Otherwise a non-parametric estimation of the distribution function could be performed. In our case, the CV result data obtained from the primary studies identified by the systematic review, were tested for normality using the Anderson-Darling test. Before applying the test the data was transformed using methods of compositional data analysis. To compute the test we used method $adtestWrapper$ from R language library $robCompositions$.

The tests we performed indicated, quite strongly, that in most of the prioritization cases the data is not normally distributed. Hence, our recommendation is that, in general, a non-parametric approach should be used to determine the probability density function, and one such, common, approach would be to use the kernel density estimation. (In our implementation of ECV in the R programming language, kernel density estimation is per-

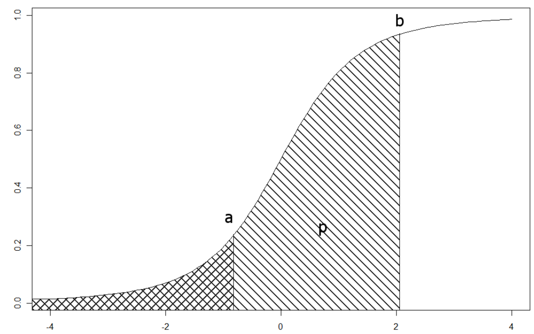


Figure 5: Cumulative distribution function of the log-ratio c_i between the items k and l (area p denotes probability that c_i is between $\frac{3}{4}$ and $\frac{4}{3}$.)

Table 10: Example of an equality table.

prioritization items	i1	i2	i3	i4
i1	equal	equal	-	equal
i2	equal	equal	-	-
i3	-	-	equal	-
i4	equal	-	-	equal

formed using the package ks.)

To determine the probability of $\bar{x}$ being between a and b the following equation is used:

$$p = P(b) - P(a), \tag{7}$$

where P is the cumulative distribution function obtained by applying kernel density estimation on ilr-transformed priority values denoted by vector C. Variable a is equal to $ilr(3, 4)$ and b is equal to $ilr(4, 3)$. (A graphical interpretation of Equation (7) is presented in Figure 5.) The area that is denoted by letter p represents the probability computed by the equation.

After both prioritization items are tested for equality it may be convenient to display the equality of different items in the form of a table. Please see Table 10 for an example.

6.2. Grouping Prioritization Items

When equal items are determined they can be divided into groups of equal items. Division is performed in such a way that each two items in a group are equal. The test for equality of the items described in Section 6.1 is not transitive. Hence, if prioritization item A is equal to B and B is equal to C

then it does not automatically imply that A is equal to C. Therefore, there may be several ways to group the equal items. The two possible division criteria that we have considered in this study are:

1. Maximize the number of items that have a group.
2. Maximize the number of items in each group.

Current implementation of ECV (available from [47]) does not include the division of items into groups. In this study the division is done manually, so that each two items in a group are equal.

7. Results

This section presents the results of this study including the systematic literature review and the application of ECV on industry and academic data collected from the primary studies. Data extracted from primary studies and the results of the quality evaluation are available in [45].

7.1. State of Practice in Empirical Studies that use CV or Analyze the Results of CV (RQ 1)

The study search resulted in 634 unique studies. The search in databases revealed 180 papers, while an additional 454 papers were discovered using snowball sampling. The study selection resulted in 40 primary studies. Hence, 94% of the studies were excluded by the selection criteria. Snowball sampling revealed 15 (36%) out of all primary studies. The study selection criteria and the number of papers excluded by each criterion are shown in Tables 3 and 4. In total 163 of 634 studies were excluded because full text was not available.

All results of the study selection are available online and can be obtained by contacting the authors of this paper. For each study we specify keywords and databases that were used to find the study. If a study has been excluded, the exclusion criteria are provided.

The number of papers revealed by each search string and database is presented in Table 11. It should be noted that several papers were found by more than one search string or in more than one database. Table 11 shows that the search string ‘cumulative voting’ was the most frequently used in the research community to denote CV. Therefore, researchers should use or reference this term when discussing CV.

To perform snowball sampling we examined the references of primary studies that were found during the database search. References were used to search for the papers in the Google and Google Scholar search engines. Studies that were found in the search and passed the study selection criteria were added to the set of primary studies.

After the primary studies were selected, data extraction and quality evaluation was performed by two researchers. One researcher examined all studies while the second researcher did quality evaluation and data extraction for 10% of the studies. The studies were randomly selected. Inter-rater agreement were calculated by means of Krippendorff's alpha coefficient. Agreement for data extraction results was 0.86 and agreement for the quality evaluation was 0.73. According to [38] it is common to require agreement above 0.8 and the lowest acceptable agreement is 0.667. Therefore, we conclude that the agreement calculated for this study is sufficient. Ratings of the study setting, correctness, research data availability, and number of prioritization items are presented in Figure 6.

Table 12 shows the studies with the highest quality according to our criteria. These studies show a high level of rigor in a realistic setting. Moreover, authors of the studies manifest confidence by providing raw data for further use and evaluation.

Figure 7 shows a bubble chart of the distribution of studies over research areas and time. The figure shows that CV was, as far as we know, first applied some time ago in research of government elections. Nowadays, though, CV has been adopted in a wide range of software engineering areas, most frequently in requirements engineering and software release planning. Eight studies use CV in academia while the remaining 32 studies report on using CV in industry.

7.2. CV Result Analysis Methods Identified by RQ 1 (RQ 2)

The papers identified in the review use various CV result analysis methods. The main goals for CV result analysis are presented in Table 13 and a summary of methods used in the primary studies can be found in Section Appendix B.

In order to present prioritization results many studies use charts or tables. These charts and tables show the average priority of each prioritization item that is computed from priorities assigned by all stakeholders. In [54] a table of five items with highest total priority is presented. [55] shows tables with min, max, $\tilde{x}$, $\bar{x}$ and σ of priorities assigned by different stakeholders to a

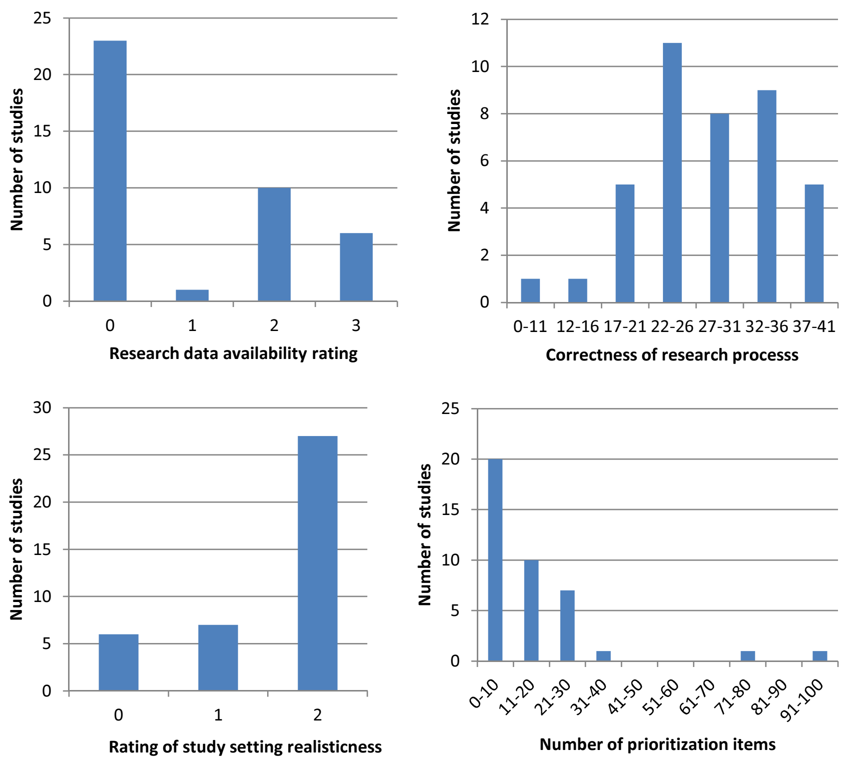
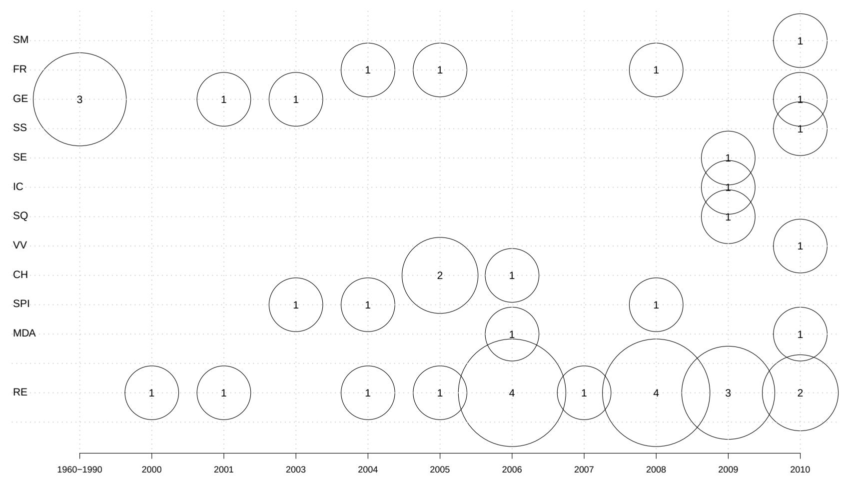


Figure 6: Study quality ratings.

Table 11: Number of papers found in the databases.

database	search strings: "100 point method"	"100 dollar method"	"100 dollar test"	"hundred point method"	"hundred dollar method"	"hundred dollar test"	"cumulative voting"	unique papers found	primary studies selected
ACM	2	0	0	1	2	3	31	34	7
IEEE	3	2	0	1	2	6	38	46	11
Inspec/Compendex	1	0	0	1	1	1	22	14	7
ISI web of science	0	0	0	0	1	1	15	16	6
SCOPUS	2	0	0	0	1	2	24	25	9
Springer	2	0	2	0	2	2	89	95	6
unique papers found	6	2	2	1	4	11	165	180	
primary studies selected	1	2	1	1	2	4	18		25



MDA - model driven software development

CH - change impact analysis in software engineering

RE - requirements engineering and software release planning

IC - intellectual capital in software company

SPI - software process improvement

V&V - software verification and validation

FR - forestry

GE - government elections

SS - software security

SQ - software quality

SM - software metrics

SE - software engineering in general

Figure 7: Distribution of studies over time.

Table 12: Top ranked studies.

	Correctness of research process	Research data availability	Study setting	Number of prioritization items
Barney 2009 [49]	36	2	2	17
Berander 2009 [17]	41	2	0	29
Barney 2009 [50]	40	2	2	5
Barney 2009 [8]	31	2	2	27
Barney 2008 [51]	34	2	2	14
Laukkanen 2005 [52]	22	3	2	30
Hu 2006 [53]	34	2	1	14
Feldt 2010 [29]	24	3	2	8
Regnell 2001 [40]	21	3	2	91
Svahnberg 2008 [30]	34	1	1	7

particular prioritization item. Finally, in [55, 56] error bars are added to the chart of final priorities (denoting σ of priorities).

In a few cases final priorities are presented in the form of ranks and CV results are degraded from ratio to ordinal scale. This is done when the interest lies only in the order of final priorities.

Several papers are interested in the difference between priorities from different prioritization perspectives (e.g. current and ideal situation) or stakeholder groups (e.g. software developers and management). Pearson or Spearman correlation coefficients are commonly used to determine what the level of similarity is between all priorities from two perspectives. Whereas, Wilcoxon, Kruskal-Wallis, Nemenyi-Damico-Wolfe-Dunn tests and the χ^2 statistic are used to detect if there is a significant difference in the value of one prioritization item from two or more perspectives. In addition, PCA is used to detect if there are distinct groups of stakeholders with common priorities [7, 10, 57].

In some cases, a stakeholder may assign equal priority to several prioritization items or leave several items unrated, e.g. the stakeholder may not have carefully considered all prioritization items. Hence, the difference between the items may have been unnoticed.

In [4] the scalability of prioritization is measured using two charts. The first chart shows the average percentages of items given a non-zero value. The second chart shows average percentages of divergence of values. If a stakeholder assigns equal priorities to many prioritization items the divergence of values is low. Unfortunately it is unclear from [4] how the average percentage of divergence is calculated.

In [58] distribution, disagreement, and satisfaction charts are presented. The distribution chart shows how the final value of a prioritization item is constructed from priorities assigned by different stakeholders. This chart shows how much each stakeholder has contributed to the final value of a prioritization item. The disagreement chart shows the level of agreement between different stakeholders on the value of a particular prioritization item. The satisfaction chart shows stakeholder satisfaction with prioritization results by calculating the correlation between final priorities and priorities assigned by a stakeholder.

The use of bi-plots and ternary plots are proposed in [10]. A bi-plot shows final priorities and stakeholder viewpoints in a two dimensional plane while a ternary plot shows prioritization items inside a triangle. Ternary plots show how many low, medium or high priorities are assigned to a prioritization item. The corners of the triangle represent high, medium, and low priority,

Table 13: Goals for CV result analysis.

Purpose of the method	Name
Show the final priority of each prioritization item. Stakeholder priorities are combined into one value.	Chart or table of final priorities
Difference between priorities assigned by different perspectives (status quo, ideal situation) or different stakeholder groups (developers, management) [10]	Bi-plot
detect stakeholder groups with similar priorities [10]	Bi-plot
show the relative number of issues that have received high, medium, or low priority [10]	Ternary plot
detect stakeholder groups with common priorities [10]	PCA
how the final value of prioritization item is constructed from priorities assigned by different stakeholder. This chart shows how much each stakeholder has contributed to the final value of prioritization item [58]	Distribution chart
the level of agreement between different stakeholders on value of particular prioritization item [58]	Disagreement chart
satisfaction of a stakeholder with the prioritization results by the calculating correlation between the final priorities and priorities assigned by a stakeholder [58]	Satisfaction chart
percentage of the divergence of the priorities assigned by a stakeholder [4]	average percentage of divergence
average percentage of items given a non-zero value [4]	
detect equal prioritization items (presented in this paper)	ECV

e.g. if a prioritization item has received mostly high priority values then it is shown closer to the high priority corner.

7.2.1. Problems with Data Analysis in Primary Studies

A few primary studies, as revealed by the systematic review, have problems with the data analysis. These studies disregard the compositional nature of CV results.

In [7, 57] standard PCA is performed without applying log-ratio transformations to compositional data. According to [59], this is likely to be inadequate and in [60], a more appropriate method for performing PCA on compositional data is presented.

The normality of compositional data is defined in [61]. It is stated that it is convenient to transform compositional data using isometric log-ratio transformation before the tests for normality can be applied. [54] violates this requirement by applying the Shapiro-Wilk test for normality to untransformed compositional data.

The Kruskal-Wallis test is used in [54] to analyze compositional data. The test is used to evaluate the difference between three organization levels. The Kruskal-Wallis test assumes that variables within each sample are independent [62]. However, values within compositional data vectors are not

Table 14: Identified groups of equal items.

Paper identifier & Description	Type of CV	Pairs of equal items	Groups of equal items
Barney 2009 [50] Perceived priorities of software product investments in an ideal situation	comp. HCV	(A2, B4) (B4, B5) (B4, C1) (B5, B15) (B6, B7) (B7, B8) (B14, B15) (B14, B18) (B17, B18)	(A2, B4) (B4, C1) (B5, B15) (B6, B7) (B14, B15) (B17, B18)
	uncomp. HCV	(B4, B5) (B4, B8) (B5, B15) (B6, B7) (B7, B12) (B14, B15) (B14, B18) (B16, B17) (B12, B13)	(B4, B5) (B5, B15) (B6, B7) (B14, B15) (B16, B17) (B12, B13)
Berander 2009 [17] Software requirements for course management system	uncomp. & comp. HCV	(3:2, 3:3)	(3:2, 3:3)
Svahnberg 2008 [30] The view of academia researchers on the requirements understandability criteria	CV	(Development, Verification & Validation) (Development, Product Planning 1)	(Development, Product Planning 1)

independent (as described in Section 2.4). Hence, we claim the Kruskal-Wallis test to be somewhat misused in [54].

7.3. Identifying Prioritization Items with Equal Priority Using ECV (RQ 3)

This section presents the results of applying ECV to the industrial and academic CV data as found through the systematic literature review. Six primary studies included the raw prioritization results in the paper itself or referenced online sources where the data was available. To collect the data from the remaining 34 papers, the authors of all papers were contacted.

First, the email addresses provided in the papers were used. If no answer was received authors were searched for using Google, Facebook and LinkedIn. Authors from 11 papers provided us with data to be used in the evaluation of ECV. However, due to confidentiality reasons we can not publish this data directly.

In short, ECV was applied to 27 CV prioritization cases from 14 studies. In the cases of HCV, ECV was applied two times to the same data to test both compensated and uncompensated priorities. Equal items were detected in three prioritization cases. A summary of the results is presented in Table 14 and below follows a summary of each relevant study.

In [30] a prioritization of requirement understandability criteria is presented. One of the main findings of the paper is that two criteria - "Development" and "Verification & Validation" - are most important from an academic viewpoint. ECV adds new knowledge to these results. It shows that "Development" and "Verification & Validation" are equally important, i.e. it is not true that either one of the criteria is more important.

A prioritization of software requirements for an academic course management system is presented in [17]. ECV detected that two features—Assignment Submission and Assignment Feedback—have the same priority. If the system is developed in several releases Assignment Submission and Assignment Feedback features can be freely interchanged between the releases and, hence, in this way ECV simplifies release planning.

In [50] software product investments are prioritized with HCV. The results of ECV was different for uncompensated and compensated HCV. When compensated HCV was used ECV detected equal items that belonged to different high level prioritization groups (A, B and C) indicating that ECV provided a more fine-grained view. In the case of uncompensated HCV, on the other hand, all equal items belonged to one high level prioritization group (group B).

8. Discussion and Conclusions

This section discusses the results of the systematic review and evaluation of ECV conducted as part of this study.

CV has been applied in various areas, but most frequently in requirements prioritization and release planning, and quite often also as part of research methodologies. A large part of the studies have been conducted in Sweden, at Ericsson AB.One can see a slight increase in the interest in CV. During the last five years there have been more studies that use CV than between, say, 2000–2005.

Overall, studies that use CV or analyze the results of CV have a high quality in terms of correctness of research process and study realism. However, very few studies present prioritization of more than 30 items and the availability of research data is somewhat limited. In our particular case we were able to obtain data from 43% of the primary studies.

8.1. Implications for Practitioners

The results of this study provide decision support for industry practitioners. We believe that a collection of state of the practice studies help the adoption of CV prioritization method. (The top studies are summarized in Table 12.) In addition, a set of CV analysis methods enables comprehensive understanding of the prioritization results. (The analysis methods are presented in Table 13.) One of the most common goals of CV analysis is to display the prioritization results and, thus, to show the difference between several prioritization perspectives.

Additionally, we present ECV—a novel method for CV analysis. Prioritization often results in the assignment of similar priorities to several prioritization items. CV results contain both 'real priorities' and random errors. Due to random errors, equal prioritization items may receive different priorities. ECV identifies such items. It allows stakeholders to disregard the random part of the CV results. Thus, ECV simplifies the understanding of the prioritization results.

ECV identifies prioritization items with similar priority and tests whether these items can be considered equal. In this case, ECV can be used in software release planning. For example, let us suppose that a set of software requirements are prioritized with regard to the implementation costs. First of all, ECV can then detect items with equal cost. Second, the equal items can be freely interchanged between the releases. Finally, the decision to allocate a requirement to a particular release can be made based on another criteria, such as risk or business value.

ECV has been successfully applied on a considerable amount of CV data and, additionally, has also detected equal items in different groups of HCV hierarchies.

8.2. Implications for Academia

In the systematic review 36% of papers were revealed by the snowball sampling. That is a considerable amount. Several studies do not mention the name of the prioritization method (i.e. cumulative voting or hundred dollar test). Others are not available through selected databases because they are conference publications or theses. It shows, in our opinion, that snowball sampling ought to be used in all systematic literature reviews.

CV results are a special type of data—compositional data. Standard statistical analysis methods that assume the independence of the samples cannot be applied to CV results. In [63] methods for the analysis of compositional data have been presented. The systematic review conducted as a part of this study revealed that 22 studies analyze CV results; yet, only one study uses compositional data analysis methods, i.e. [10]. None of the studies, including [10], present methods for detecting items with equal priority in CV results. Hence, ECV is, in this respect, a unique method.

The small use of compositional data analysis is really not surprising, since literature describing CV does not state that the results are compositional data. Standard statistical analysis methods may produce useful results for compositional data. However, there are cases when they are misleading or even faulty. Section 7.2.1 contains evidence of inappropriate use of statistical methods by several papers.

This study has collected a set of compositional data analysis methods for CV analysis (see Table 13). We believe that this could help researchers to improve the analysis of CV results with appropriate methods.

Since CV is associated with compositional data, it might be tempting to choose another requirements prioritization method. However, it would not solve the problem *per se*, because any ratio scale prioritization, for instance AHP, contains compositional data.

The principal implications for the academia are mainly the following:

1. All systematic literature reviews should include snowball sampling.

2. Researchers can improve their statistical analysis of CV results using compositional data analysis methods collected and developed by this study.

3. When CV or any other ratio scale prioritization method is taught, compositional data analysis should also be presented as part of the solution.

8.3. Validity Threats

The validity of the systematic review is mainly limited by the chosen databases, the design of the review, and human judgement in study selection and data extraction.

To mitigate the threats we use the most popular databases in the field of software engineering. In the beginning of the systematic review a review protocol was developed, peer-reviewed, and revised. Search strategy was validated against a set of previously known papers obtained from other researchers.

One of many terms used to name cumulative voting is '$100 method'. We were not able to search for this term because non of the chosen databases support search for special characters like '$' and the search string '100 method' yields too many hits. To increase the likelihood of discovering relevant studies snowball sampling was extensively used.

To increase the validity of study selection, all included studies and 20 randomly selected excluded studies were examined by two researchers. There were no disagreement on the inclusion/exclusion of the studies.

The large number of studies identified by snowball sampling (15 out of 40 studies) may be caused by faulty design or by faulty execution of the search in the databases. There are several reasons why the studies revealed by snowball sampling are not revealed by the search in databases. (Reason for each study is given in Table Appendix A.2.) Based on these reasons we argue that snowball sampling does not indicate any problems with the design of the search in the databases.

Four studies were not found because they were not available through databases used in this systematic review. Out of them one is a master thesis, two are conference publications and one is a publication in the area of forestry. Seven studies do not mention the name of the prioritization method (i.e. hundred dollar method or cumulative voting). Only phrases like "distribution of a predefined amount of fictitious money ($100,000) over the items to be prioritized" or "1,000 points" allowed us to identify that CV was indeed used. One paper used a previously unknown name for CV, i.e. the 100-point technique.

The quality of the data extraction and quality evaluation was validated using inter-rater agreement analysis. In our case, 10% of the studies were rated by two researchers and Krippendorff's alpha was calculated. The agreement for the data extraction results was 0.86 and the agreement for the quality evaluation was 0.73 (indicating a credible level of quality).

There are two main validity threats with ECV itself. First, ECV may not detect prioritization items with equal priority. Second, ECV may produce a false positive result, i.e. there may be a real difference between items that ECV claims as being equal.

To mitigate the first threat ECV was applied on artificially created test data with and without items with similar priority. ECV worked correctly in both cases.

To mitigate the second threat we visually inspected the results of the application of ECV on the real world data from the primary studies. We concluded that items identified by ECV can be considered equal.

CV results used in the evaluation of ECV were tested for normality. The tests indicated that CV results do not have multivariate normal distribution. Therefore, the design of ECV was based on a non-parametric statistical test.

8.4. Future Work

With respect to future work one can distiguish two interesting paths: Scalability and improvements to ECV.

First, there are very few studies that apply CV on prioritization sets of more than 30 items. However, in requirements engineering, industry practitioners need to prioritize much larger numbers of software requirements. Therefore, the state of art could benefit from the application of CV and HCV to large prioritization sets.

The proposed method, ECV, has now been evaluated on existing research data. To further evaluate ECV, it would be appropriate to apply it in direct industry practice and in prioritization cases with a larger number of prioritization items (>30). Additionally, compositional data analysis methods, as the ones identified by this paper, should be tried with other prioritization methods that produce ratio scale results.

Second, ECV may be improved to find groups of equal items not just pairs. Equality of a pair (or a group) of items to another item can be tested with the help of compositional balances.

The CV process itself can also be improved with the help of compositional data analysis. Weighting of stakeholder priorities could be done using compositional powering, which could be presumed as better compared to using a multiplication that is removed in a log-ratio tranformation.

Additionally, compensation of priority values in HCV is not *subcompositionally coherent*; thus, sequential binary partition could quite possibly be used to improve the compensation.

8.5. Conclusions

CV prioritization results are special type of data – compositional data. Any analysis of CV results must take into account the compositional nature of the CV results.

This study presents a systematic literature review of the empirical use of CV. CV has been applied in various areas, but most frequently in requirements prioritization and release planning. The review has resulted in a collection of state of the practice studies and CV result analysis methods. We believe that it can help the adoption of CV prioritization method.

In our case, snowball sampling was performed as a part of the review. Since it revealed 36% out of all primary studies, we believe that in future snowball sampling should be used in all systematic reviews.

Additionally, we present ECV—a novel method for CV analysis. As suggested by our evaluation, ECV is able to detect prioritization items with equal priority (i.e. items that have insignificant difference in priority). The evaluation of ECV was based on the data obtained from the authors of the primary studies.

References

[1] P. Berander, A. Andrews, Requirements prioritization, in: A. Aurum, C. Wohlin (Eds.), Engineering and managing software requirements, Springer-Verlag, Berlin/Heidelberg, 2005, pp. 69–94.

[2] D. Leffingwell, D. Widrig, Managing software requirements: A unified approach, Addison-Wesley Professional, 1999.

[3] V. Ahl, An experimental comparison of five prioritization methods, Master's Thesis, School of Engineering, Blekinge Institute of Technology, Sweden.

[4] P. Berander, P. Jönsson, Hierarchical cumulative voting (HCV) - Prioritization of requirements in hierarchies, International Journal of Software Engineering and Knowledge Engineering 16 (6) (2006) 819–850.

[5] J. Karlsson, K. Ryan, A cost-value approach for prioritizing requirements, IEEE Software 14 (5) (1997) 67–74.

[6] J. Karlsson, An evaluation of methods for prioritizing software requirements, Information and Software Technology 39 (14-15) (1998) 939–947.

[7] F. Pettersson, M. Ivarsson, T. Gorschek, P. Öhman, A practitioner's guide to light weight software process assessment and improvement planning, Journal of Systems and Software 81 (2008) 972–995.

[8] S. Barney, C. Wohlin, Software product quality: Ensuring a common goal, in: Q. Wang, V. Garousi, R. Madachy, D. Pfahl (Eds.), Trustworthy Software Development Processes, Vol. 5543 of Lecture Notes in

Computer Science, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 256–267.

[9] P. Jönsson, C. Wohlin, A study on prioritisation of impact analysis issues: A comparison between perspectives, Software Engineering Research and Practice in Sweden.

[10] P. Chatzipetrou, L. Angelis, P. Rovegard, C. Wohlin, Prioritization of issues and requirements by cumulative voting: A compositional data analysis framework, in: Proceedings of the 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications, IEEE Computer Society, Washington, DC, USA, 2010, pp. 361–370.

[11] R. L. Engstrom, D. A. Taebel, R. L. Cole, Cumulative voting as a remedy for minority vote dilution: The case of Alamogordo, New Mexico, Journal of Law & Politics 5 (1988) 469–497.

[12] S. Bhagat, J. A. Brickley, Cumulative voting: The value of minority shareholder voting rights, Journal of Law and Economics 27 (2) (1984) 339–365.

[13] V. Hiltunen, J. Kangas, J. Pykalainen, Voting methods in strategic forest planning - Experiences from Metsahallitus, Forest Policy and Economics 10 (3) (2008) 117–127.

[14] P. Boldi, F. Bonchi, C. Castillo, S. Vigna, Voting in social networks, in: Proceedings of the 18th ACM conference on Information and knowledge management, ACM, New York, NY, USA, 2009, pp. 777–786.

[15] H. Ayad, M. Kamel, Cumulative voting consensus method for partitions with variable number of clusters, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (1) (2008) 160–173.

[16] M. Svahnberg, A. Karasira, A study on the importance of order in requirements prioritisation, in: Proceedings of the Third International Workshop on Software Product Management, IEEE Computer Society, Washington, DC, USA, 2009, pp. 35–41.

[17] P. Berander, M. Svahnberg, Evaluating two ways of calculating priorities in requirements hierarchies - An experiment on hierarchical cumulative voting, Journal of Systems and Software 82 (2009) 836–850.

[18] T. L. Saaty, The analytic hierarchy process, McGraw-Hill, New York, 1980.

[19] J. Aitchison, J. J. Egozcue, Compositional Data Analysis: Where Are We and Where Should We Be Heading?, Mathematical Geology 37 (7) (2005) 829–850.

[20] V. Pawlowsky-Glahn, J. J. Egozcue, Compositional data and their analysis: An introduction, Geological Society, London, Special Publications 264 (1) (2006) 1–10.

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

[22] P. Filzmoser, K. Hron, Outlier detection for compositional data using robust methods, Mathematical Geosciences 40 (2008) 233–248.

[23] S. Laukkanen, T. Palander, J. Kangas, Applying voting theory in participatory decision support for sustainable timber harvesting, Canadian Journal of Forest Research 34 (7) (2004) 1511–1524.

[24] D. Cooper, A. Zillante, A comparison of cumulative voting and generalized plurality voting, Public Choice 150 (1-2) (2010) 363–383.

[25] A. Hoover, M. Goldbaum, Locating the optic nerve in a retinal image using the fuzzy convergence of the blood vessels, Medical Imaging, IEEE Transactions on 22 (8) (2003) 951–958.

[26] V. Heikkilä, A. Jadallah, K. Rautiainen, G. Ruhe, Rigorous support for flexible planning of product releases - A stakeholder-centric approach and its initial evaluation, in: 2010 43rd Hawaii International Conference on System Sciences, IEEE Computer Society, 2010, pp. 1–10.

[27] D. Baca, K. Petersen, Prioritizing countermeasures through the countermeasure method for software security (CM-Sec), in: M. Ali Babar, M. Vierimaa, M. Oivo (Eds.), Product-Focused Software Process Improvement, Vol. 6156 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 176–190.

[28] P. Berander, P. Jönsson, A goal question metric based approach for efficient measurement framework definition, in: Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering, ACM, New York, NY, USA, 2006, pp. 316–325.

[29] R. Feldt, R. Torkar, E. Ahmad, B. Raza, Challenges with software verification and validation activities in the space industry, in: Proceedings of the 2010 Third International Conference on Software Testing, Verification and Validation, IEEE Computer Society, Washington, DC, USA, 2010, pp. 225–234.

[30] M. Svahnberg, T. Gorschek, M. Eriksson, A. Borg, K. Sandahl, J. Börster, A. Loconsole, Perspectives on requirements understandability – For whom does the teacher's bell toll?, in: Proceedings of the 2008 Requirements Engineering Education and Training, IEEE Computer Society, Washington, DC, USA, 2008, pp. 22–29.

[31] J. Aitchison, The statistical analysis of compositional data, Journal of the Royal Statistical Society. Series B (Methodological) 44 (2) (1982) 139–177.

[32] J. Egozcue, V. Pawlowsky-Glahn, G. Mateu-Figueras, C. Barceló-Vidal, Isometric logratio transformations for compositional data analysis, Mathematical Geology 35 (3) (2003) 279–300.

[33] K. Khan, A systematic review of software requirements prioritization, Master's thesis, Blekinge Institute of Technology, Ronneby, Sweden.

[34] F. Zahedi, The analytic hierarchy process: A survey of the method and its applications, Interfaces 16 (4) (1986) 96–108.

[35] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical Software Engineering 14 (2) (2008) 131–164.

[36] L. Goodman, Snowball sampling, The Annals of Mathematical Statistics 32 (1) (1961) 148–170.

[37] K. Krippendorff, Bivariate agreement coefficients for reliability of data, Sociological Methodology 2 (1970) 139–150.

[38] K. Krippendorff, Content analysis: An introduction to its methodology, 2nd Edition, Sage Publications, 2003.

[39] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Tech. Rep. EBSE 2007-001, Keele University (2007).

[40] B. Regnell, M. Höst, J. Natt och Dag, P. Beremark, T. Hjelm, An industrial case study on distributed prioritisation in market-driven requirements engineering for packaged software, Requirements Engineering 6 (1) (2001) 51–62.

[41] B. Kitchenham, Procedures for performing systematic reviews, Tech. Rep. TR/SE-0401, Keele University (2004).

[42] M. Ivarsson, T. Gorschek, A method for evaluating rigor and industrial relevance of technology evaluations, Empirical Software Engineering 16 (2011) 365–395.

[43] C. Wohlin, P. Runeson, M. Höst, Experimentation in software engineering: An introduction, Springer Netherlands, 2000.

[44] A. Jedlitschka, D. Pfahl, Reporting guidelines for controlled experiments in software engineering, in: Proceedings of the 2005 International Symposium on Empirical Software Engineering, IEEE Computer Society, 2005, pp. 95–104.

[45] K. Rinkevics, R. Torkar, Data extraction and quality evaluation results (2011).
URL `http://rinkevic.wordpress.com/2011/11/26/data-extraction-and-quality-evaluation-results/`

[46] R. Ihaka, R. Gentleman, R: A language for data analysis and graphics, Journal of computational and graphical statistics 5 (3) (1996) 299–314.

[47] K. Rinkevics, R. Torkar, ECV implementation source code in R (2011).
URL `http://rinkevic.wordpress.com/2011/08/14/ecv-implementation-in-r/`

[48] R. M. Groves, F. J. Fowler, M. P. Couper, J. M. Lepkowski, E. Singer, Survey methodology, John Wiley and Sons, 2009.

[49] S. Barney, A. Aurum, C. Wohlin, The relative importance of aspects of intellectual capital for software companies, in: Proceedings of the 2009 35th Euromicro Conference on Software Engineering and Advanced Applications, IEEE Computer Society, 2009, pp. 313–320.

[50] S. Barney, C. Wohlin, Software product quality: Ensuring a common goal, in: Proceedings of the International Conference on Software Process: Trustworthy Software Development Processes, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 256–267.

[51] S. Barney, A. Aurum, C. Wohlin, A product management challenge: Creating software product value through requirements selection, Journal of Systems Architecture 54 (6) (2008) 576–593.

[52] S. Laukkanen, T. Palander, J. Kangas, A. Kangas, Evaluation of the multicriteria approval method for timber-harvesting group decision support, Silva Fennica 39 (2) (2005) 249–264.

[53] G. Hu, A. Aurum, C. Wohlin, Adding value to software requirements: An empirical study in the Chinese software industry, in: Proceedings of the Seventeenth Australasian Conference on Information Systems, 2006.

[54] P. Jönsson, C. Wohlin, Understanding impact analysis: An empirical study to capture knowledge on different organisational levels, in: Proceedings of International Conference on Software Engineering and Knowledge Engineering, IEEE Computer Society, 2005, pp. 707–712.

[55] L. Kuzniarz, L. Angelis, Empirical extension of a classification framework for addressing consistency in model based development, Information and Software Technology 53 (2011) 214–229.

[56] P. Rovegard, L. Angelis, C. Wohlin, An empirical study on views of importance of change impact analysis issues, IEEE Transactions on Software Engineering 34 (4) (2008) 516–530.

[57] C. Wohlin, A. Aurum, Criteria for selecting software requirements to create product value: An industrial empirical study, in: S. Biffl, A. Aurum, B. Boehm, H. Erdogan, P. Grünbacher (Eds.), Value-based software engineering, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 179–200.

[58] B. Regnell, M. Höst, J. Natt och Dag, Visualization of agreement and satisfaction in distributed prioritization of market requirements, in: Proceedings of REFSQ2000, 6th Int. Workshop on Requirements Engineering: Foundation for Software Quality, 2000, pp. 1–12.

[59] J. Aitchison, Principal component analysis of compositional data, Biometrika 70 (1) (1983) 57–65.

[60] P. Filzmoser, K. Hron, C. Reimann, Principal component analysis for compositional data with outliers, Environmetrics 20 (6) (2009) 621–632.

[61] V. Pawlowsky Glahn, J. Egozcue, R. Tolosana Delgado, Lecture notes on compositional data analysis, Tech. rep., Universitat de Girona, Spain (July 2007).

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

[63] J. Aitchison, The statistical analysis of compositional data, Chapman & Hall, London, 1986.

[64] S. Bowler, D. Brockington, T. Donovan, Election systems and voter turnout: Experiments in the United States, The Journal of Politics 63 (3) (2001) 902–915.

[65] D. Brockington, A low information theory of ballot position effect, Political Behavior 25 (1) (2003) 1–27.

[66] N. Dzamashvili Fogelström, M. Svahnberg, T. Gorschek, Investigating impact of business risk on requirements selection decisions, in: Proceedings of the 2009 35th Euromicro Conference on Software Engineering and Advanced Applications, IEEE, 2009, pp. 217–223.

[67] S. Hatton, Choosing the right prioritisation method, in: Proceedings of the 19th Australian Conference on Software Engineering, IEEE Computer Society, Washington, 2008, pp. 517–526.

[68] S. Hatton, Early prioritisation of goals, in: Proceedings of the 2007 Conference on Advances in Conceptual Modeling: Foundations and Applications, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 235–244.

[69] M. Staron, C. Wohlin, An industrial case study on the choice between language customization mechanisms, in: J. Münch, M. Vierimaa (Eds.), Product-Focused Software Process Improvement, Vol. 4034 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 177–191.

[70] T. Touseef, C. Gencel, A structured goal based measurement framework enabling traceability and prioritization, in: Proceedings of the 2010 6th International Conference on Emerging Technologies, 2010, pp. 282–286.

[71] P. Berander, C. Wohlin, Differences in views between development roles in software process improvement - A quantitative comparison, in: Proceedings of the 8th Internation Conference on Empirical Assessment in Software Engineering, 2004.

[72] P. Berander, Using students as subjects in requirements prioritization, in: Proceedings of the 2004 International Symposium on Empirical Software Engineering, IEEE Computer Society, 2004, pp. 167–176.

[73] P. Berander, C. Wohlin, Identification of key factors in software process management - A case study, in: Proceedings of the 2003 International Symposium on Empirical Software Engineering, IEEE Computer Society, Washington, DC, USA, 2003, pp. 316–325.

[74] R. L. Cole, D. A. Taebel, R. L. Engstrom, Cumulative voting in a municipal election: A note on voter reactions and electoral consequences, The Western Political Quarterly 43 (1) (1990) 191–199.

[75] J. Kuklinski, Cumulative and plurality voting: An analysis of Illinois' unique electoral system, The Western Political Quarterly 26 (4) (1973) 726–746.

[76] J. Sawyer, D. MacRae, Game theory and cumulative voting in Illinois: 1902-1954, The American Political Science Review 56 (4) (1962) 936–946.

Appendix A. Primary Studies

Appendix A.1. Primary studies found in databases.

Title	Reference
Prioritizing countermeasures through the countermeasure method for software security (CM-Sec)	Baca 2010 [27]
The relative importance of aspects of intellectual capital for software companies	Barney 2009 [49]
Software product quality: Ensuring a common goal	Barney 2009 [8]
Balancing software product investments	Barney 2009 [50]
Hierarchical cumulative voting (HCV) prioritization of requirements in hierarchies	Berander 2006 [4]
A goal question metric based approach for efficient measurement framework definition	Berander 2006 [28]
Evaluating two ways of calculating priorities in requirements hierarchies: An experiment on hierarchical cumulative voting	Berander 2009 [17]
Election systems and voter turnout: Experiments in the United States	Bowler 2001 [64]
A low information theory of ballot position effect	Brockington 2003 [65]
Prioritization of issues and requirements by cumulative Voting: A compositional data analysis framework	Chatzipetrou 2010 [10]
A comparison of cumulative voting and generalized plurality voting	Cooper 2010 [24]
Challenges with software verification and validation activities in the space industry	Feldt 2010 [29]
Investigating impact of business risk on requirements selection decisions	Fogelstrom 2009 [66]
Choosing the right prioritization method	Hatton 2008 [67]
Early prioritization of goals	Hatton 2007 [68]
Rigorous support for flexible planning of product releases: A stakeholder-centric approach and its initial evaluation	Heikkila 2010 [26]
Voting methods in strategic forest planning: Experiences from Metsähallitus	Hiltunen 2008 [13]
Empirical extension of a classification framework for addressing consistency in model based development	Kuzniarz 2010 [55]
Evaluation of the multi-criteria approval method for timber-harvesting group decision support	Laukkanen 2005 [52]
A practitioner's guide to light weight software process assessment and improvement planning	Pettersson 2008 [7]
An empirical study on views of importance of change impact analysis issues	Rovegard 2008 [56]
An industrial case study on the choice between language customization mechanisms	Staron 2006 [69]
Perspectives on requirements understandability—For whom does the teacher's bell toll?	Svahnberg 2008 [30]
A study on the importance of order in requirements prioritization	Svahnberg 2009 [16]
A structured goal based measurement framework enabling traceability and prioritization	Touseef 2010 [70]

Appendix A.2. Primary studies revealed by snowball sampling.

Reference	Title	Reason why the paper is not revealed by the search in databases
Ahl 2005 [3]	An experimental comparison of five prioritization methods	Selected databases does not contain the paper, master thesis at BTH
Barney 2008 [51]	A product management challenge: Creating software product value through requirements selection	Prioritization method name not mentioned, phrase "1,000 points" used instead.
Berander 2004 [71]	Differences in views between development roles in software process improvement—A quantitative comparison	Prioritization method name not mentioned, phrase "100 points" used instead.
Berander 2004 [72]	Using students as subjects in requirements prioritization	Unknown CV name: 100-point technique
Berander 2003 [73]	Identification of key factors in software process management: A case study	Prioritization method name not mentioned, phrase "100 points" used instead.
Cole 1990 [74]	Cumulative voting in a municipal election: A note on voter reactions and electoral consequences	Study published before year 2001.
Hu 2006 [53]	Adding value to software requirements: An empirical study in the chinese software industry	Prioritization method name not mentioned, phrase "1,000 points" used instead.
Jonsson 2005 [9]	A study on prioritization of impact analysis issues: A comparison between perspectives	Selected databases does not contain the paper.
Jonsson 2005 [54]	Understanding impact analysis: An empirical study to capture knowledge on different organizational levels	Selected databases does not contain the paper.
Kuklinski 1973 [75]	Cumulative and plurality voting: An analysis of Illinois' unique electoral system	Study published before year 2001.
Laukkanen 2004 [23]	Applying voting theory in participatory decision support for sustainable timber harvesting	Selected databases does not contain the paper.
Regnell 2001 [40]	An industrial case study on distributed prioritization in market-driven requirements engineering for packaged software	Prioritization method name not mentioned: "distribution of a predefined amount of fictitious money ($100,000) over the items to be prioritized."
Regnell 2000 [58]	Visualization of agreement and satisfaction in distributed prioritization of market requirements	Prioritization method name not mentioned: "distribution of a predefined amount of fictitious money ($100,000) over the items to be prioritized."
Wohlin 2006 [76]	Game theory and cumulative voting in Illinois: 1902–1954	Study published before year 2001.
Wohlin 2006 [57]	Criteria for selecting software requirements to create product value: An industrial empirical study	Prioritization method name not mentioned: "The subjects had 1,000 points to spend among the 13 criteria."

Appendix B. CV Result Analysis Methods

	Paper																					
Analysis method	Svahnberg2008	Svahnberg2009	Staron2006	Pettersson2008	Wohlin2006	Laukkanen2005a	Hu2006	Jonsson2005a	Kuzniarz2010	Rovegard2008	Berander2006a	Berander2004a	Berander2006	Feldt2010	Barney2009b	Barney2008	Barney2009a	Barney2009	Jonsson2005	Chatzipetrou2010	Regnell2001	Regnell2000
Table that shows final priorities	x			x												x						
Chart that shows final priorities	x			x	x	x	x									x						
Table of top-5 prioritization items								x														
min, max, $\tilde{x}$, $\bar{x}$ and σ of priorities assigned by different stakeholders									x	x												
Bar chart of prioritization results showing $\bar{x}$ priority and σ of priorities									x	x												
Pearson correlation coefficient		x										x										
Nemenyi Damico Wolfe Dunn														x								
Spearmans r															x		x					
Kruskal-Wallis								x														
Wilcoxon							x															
Correlation matrix		x													x		x					
Chart for comparing priorities from two perspectives, priorities are points in two dimensional plane, x- and y-axis represent two different perspectives										x									x			
Difference between priorities assigned by each two stakeholders using χ^2-statistic										x												
Median ranks		x																				
CV results converted to priority ranks		x											x					x				
PCA				x	x															x		
Percentage of divergence of priorities assigned by a stakeholder											x											
Average percentage of items given non-zero value											x											
Distribution chart																					x	x
Disagreement chart				x																	x	x
Satisfaction chart				x																	x	x
Bi-plot																				x		
Ternary plot																				x		

Appendix C. Quality Evaluation Checklist

	Item	Question or Description of the Item	Rating
1.	Background, introduction	Introduce research area	
2.	Problem statement, purpose	What is the problem [44]? Where does it occur [44]? Who has observed it [44]? Why is it important to be solved [44]?	
3.	Context, independent variables (aka. environment, setting)	Study location, time constraints, application domain, organization, tools, market, process (e.g. software development methodology), size of project, product that is being developed	
4.	Related work	Other existing work, alternative technologies, solutions, and studies	
5.	Goals and Hypotheses	Null hypothesis and one or more alternative hypotheses for each goal	
6.	Research questions		
7.	Design, Research methods		
7.1.	Design	Description of each step of the study	
7.2.	Control group	If there is a control group, are participants similar to the treatment group participants in terms of variables that may affect study outcomes[39]?	
7.3.	Randomization	Random selection of participants and objects Random assignment of treatment and objects to participants Random order of treatments in case of paired design. If each participant is assigned two treatments A and B, then part of participants perform A first and the other part start with B	
7.4.	Blocking	Group participants of the study into homogeneous groups called blocks (e.g. students in one course, database developers in one company) and implement the study design within each block independently. The idea is that variability of independent variables (e.g. experience and knowledge of subjects) is smaller within a group. That helps measuring changes in dependent variables [41].	
7.5.	Balancing	Equal number of subjects should be assigned to each treatment [41].	
7.6.	Blinding	Automated assignment of treatments to subjects [41] Automated distribution of study materials to subjects [41] Persons who grade the task results should not know which treatment was used [41] Analyst should not know which treatment group is which [41] Automated data collection from subjects [41]	
8.	Subjects (participants)		
8.1.	Population		
8.2.	Sampling	How sampling is performed? What subjects are included and excluded? [39] What is the type of the sampling (e.g. convenience, random)? Is the sample(selected participants) representative of the population?	
8.3.	"Drop outs" and response rate	Are reasons given for refusal to participate[39]?	
8.4.	Subject motivation	E.g. material benefits, course credits for students, etc.	
9.	Objects	E.g. documents and other artifacts	
10.	Measures, Data collection procedures	Who, when, and how to measure [39]? How is the measurement supported? Is it automated [39]? Are the measures used in the study the most relevant ones for answering the research questions [39]?	
11.	Analysis procedure		
11.1.	Data description	Do the numbers add up across different tables and subgroups [39]?	
11.2.	Data types (continuous, ordinal, categorical)		
11.3.	Scoring systems		
11.4.	Data set reduction, outliers		
11.5.	Statistical methods	Are the assumptions of statistical methods met? What statistical programs are used?	
11.6.	Statistical significance	If statistical tests are used to determine differences, is the actual p-value given [39]? If the study is concerned with differences among groups, are confidence limits given describing the magnitude of any observed differences [39]?	
12.	Validity threats	Threats, implications of the threats, and threat mitigation	
12.1.	Side-effects during study execution	Deviations from the plan, solutions for the deviations	
13.	Most important findings	Are all study questions answered [39]? Are negative findings presented [39]?	
14.	Industry impact, inference, generalization	What implications does the report have for practice [39]? How and where the results can be used? Limitations under which findings are relevant [44]?	
15.	Future work		