



# Stakeholder-Centric Clustering Methods for Conflict Resolution in the Requirements Engineering Process

Ishaya Gambo<sup>1,2</sup>  and Kuldar Taveter<sup>1</sup> 

<sup>1</sup> Institute of Computer Science, University of Tartu, Tartu, Estonia  
{ishaya.gambo, kuldar.taveter}@ut.ee

<sup>2</sup> Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria

**Abstract.** This paper describes the development and evaluation of a framework that combines expert-based and clustering methods for resolving conflicts in requirements elicited from stakeholders. The purpose of the framework was to identify and resolve conflicts among expectations by multiple stakeholders that often arise during the requirements elicitation phase. By means of qualitative and quantitative research approaches, face-to-face oral interviews, quantitative surveys, brainstorming sessions, and focus groups, scenarios were generated with stakeholders of a given problem domain. Our approach was implemented within an interactive system that empirically supports the adequacy of our framework with the involvements of experts and other stakeholders of the chosen problem domain. In addition, we presented a dataset of requirements with their weight scales that formed the basis for resolving conflicting views by stakeholders by applying scientific criteria. The framework was validated in a real-life case study. The results demonstrated 85.71% of correctly clustered instances of requirements, based on which the experts agreed that the interactive system was good enough for resolving conflicting subjective views in requirements analysis. The research performed has the two-fold threat to validity, which suggests (i) the need to adequately capture and harmoniously represent the views by different stakeholders in a multicultural and multidisciplinary domain and (ii) the need to validate the framework in other real life case studies in different domains. The research performed has a high potential for reducing software development costs and saving time at the early stage of the development of software products.

**Keywords:** Requirements engineering · Stakeholders · Conflict resolution · Clustering algorithm · Delphi method · Software engineering

## 1 Introduction

In requirements engineering (RE), the involvement of stakeholders is crucial for determining the requirements necessary to build a software-intensive system. As has been observed in [50] and [6], the quality of the system requirements improves when the stakeholders collaboratively interact in defining the overall goals of the system. However, managing different views, perspectives, desires, and goals by stakeholders in such

a collaborative setting requires dealing with conflicting interests [33]. On one hand, conflicts between stakeholders' views harness positive aspects of the problem domain at hand and should therefore be identified and resolved rather than suppressed [20, 61, 72]. In this context, identifying and resolving the conflicts can positively impact the application domain and helps to increase the economic value of the system to be designed for stakeholders' satisfaction [32].

On the other hand, unresolved conflicts make stakeholders lose confidence in accepting the technology that has been developed. Failing to care for the needs of stakeholders, especially in a conflicting situation, often leads to inconsistencies in system specifications and the rejection of software and technology [30, 55]. This can lead to dissatisfaction by stakeholders, waste of invested resources, and ultimately to system failures. Therefore, to achieve mutual understanding between stakeholders, satisfy their needs, and avoid conflicts, it is necessary to devise a framework that would help to resolve issues of misunderstanding between the stakeholders.

In the context of this paper, we strongly agree that eliciting conflicting requirements for software product development is always expected [74]. Conflicts between requirements enable to analyze and discuss further various perspectives of the stakeholders, which have been revealed by the conflicts [33]. As has been described in [32] and [3], conflicting requirements reflect the disagreement between two or more viewpoints by different stakeholders on some decisions or values wished for a software product to be designed. For example, a stakeholder's interest in a development project is a potential candidate for conflict when such interest is frustrated by another interest [7]. Conflicts of this nature can go beyond the normal limit, which might eventually be too big to be reasonable or acceptable. Also, conflicts of this kind can lead to a strong and painful bitterness one feels in a conflicting situation, which potentially can hinder mutual satisfaction and understanding among stakeholders. When this happens, there are bound to be divisions and separation, which will eventually make negotiations difficult [67]. This can lead to poor specification of requirements and, subsequently, system failures.

Moreover, the problem of conflict management is further aggravated by the iterative nature of some popular development methodologies used in the software industry. For instance, agile software engineering (SE) methodologies [58] require changes and elaboration of a requirements specification repeatedly along with the iterations of an agile SE process [21]. As an extension of what was reported by Gambo and Taveter in [32] and [33], in this paper we further articulate the views described on identifying and resolving conflicts in RE by establishing a conflict resolution scheme. An important contribution of this paper is putting forward an analytic process model that supports conflict resolution through requirements negotiation. The analytic process model is required to determine the outcome of the decisions to be made about the requirements to be implemented during the SE process.

The research work reported in this paper aims to improve the resolution of conflicts in requirements elicited from different stakeholders. We have developed a framework for resolving conflicts and evaluated its adequacy in a real life case study. We have validated our approach in the healthcare domain by considering the Pharmacy Department at the Obafemi Awolowo University Teaching Hospital Complex (OAUTHC). Our main consideration for choosing this case study was that today's healthcare environment arguably provides a "perfect storm" of opportunities for conflict resolution because of

the multidisciplinary team and multicultural nature of this problem domain. In the case study, we considered for conflict resolution requirements by many diverse stakeholders.

We have organized the remaining part of this paper as follows. In Sect. 2, we provide the background and motivation for the research work. In Sect. 3, we review and identify gaps in the RE literature addressing conflict resolution. In Sect. 4, we discuss the research methodology to be employed. In particular, we discuss the data collection process, work out the process of conflict identification, and formulate the framework for conflict resolution. Section 5 describes the conflict resolution system implemented by us that provides a tool for putting the framework formulated by us into practice. In Sect. 6, we describe the empirical analysis. The validation of the framework is explained in Sect. 7, and the results are presented and discussed in Sects. 8 and 9. In Sect. 10, we analyze the threats to validity. Finally, the conclusions are presented and future work is outlined in Sect. 11.

## 2 Background and Motivation

Conflict resolution is a universal phenomenon with different research strands within software engineering (SE), management science, international relations, psychology and design science. Problems revolving around resolving conflicts in these disciplines have been a subject of investigation over the years [15], focusing on mutual understanding and satisfaction by stakeholders. In this context, stakeholders can either be the individuals, groups, or organizations that affect or are affected by a system [28]. A common approach to resolving conflicts in different disciplines is through negotiation [32], especially in SE during the RE process.

The adoption of the RE process in the SE community contributes to the development of high-quality software systems [5]. However, the RE process deals with humans [51], whose needs are virtually insatiable, making conflicts inevitable. Also, RE is naturally collaborative [17] and involves various stakeholders and activities to produce the correct, complete, and consistent requirements specifications. Such collaborative nature of RE involving many stakeholders often makes conflicts unavoidable. Because of conflicts involving a large number of diverse stakeholders, requirements engineers face several difficulties when deciding about the priorities and order of implementing the requirements [2, 31]. Involvement of the stakeholders in RE is essential for decision-making, as their different preferences are dependent on their levels of understanding, experiences, and knowledge of the problem domain [84].

On the other hand, requirements are the vital objectives that need to be precise and unambiguous for the resulting software to be consistent in behavior. Still, requirements are mostly inclined to troubles of disputes, inconsistencies, the collision of concerns, disparity, and disagreement amongst the stakeholders [2, 42]. These challenges in requirements often lead to system failures [75] and product dissatisfaction if not correctly managed. In particular, inconsistencies often emerge when stakeholders express multiple disagreeing viewpoints during the requirements elicitation and analysis phase of a SE process [64]. Therefore conflicts always emerge as nearly unavoidable because the stakeholders seek to achieve mismatching goals [12]. It is essential to have requirements specification that has resolved conflicts or contradictions amongst the stakeholders to obtain a failure-free system [9].

As a continuation and extension of the work presented in [32] and [33], our concern is five-fold in this paper. First, we seek to understand and establish how to deal with conflicts in requirements by stakeholders in the RE process, especially during the elicitation and analysis phases. Secondly, we want to know how conflicts as a problem affect software development in a multidisciplinary domain development project. Thirdly, we seek to determine how conflict arising from stakeholders' requirements can be resolved to avoid software system failure. Fourthly, we seek to uncover the challenges of existing techniques to resolve conflicting stakeholders' requirements during RE. Finally, we seek to learn what suitable technique(s) exist to help fill the gaps in conflict resolution to ensure an improved RE process and stakeholders' continuous satisfaction. Specifically, we have addressed these concerns in Sects. 3, 4, 5, 6, 7, 8, and 9 by answering the main research question: **How can we resolve conflicts arising from requirements elicited from different stakeholders in a given problem domain to avoid a failure of the resulting software-intensive system?** This question is answered analytically in Sect. 4 and empirically in Sects. 5 and 6 of this paper.

### 3 Related Work

The RE community has studied the issue of conflicts from a different perspective [23], covering different types and the phases in the RE process where conflicts manifest [63]. Many of these studies primarily support the continuous input from the various stakeholders (e.g., [38, 55]). Overall, Bendjenna et al. [9] explained the importance of dealing with conflicts in requirements by stakeholders during the RE process to obtain a consistent system in meeting its objective(s). For that, common objectives by different stakeholders are recognized to be essential [49].

While Aldekhail et al. [3] reported other literature related to conflicts in RE, Grünbacher and Seyff [35] observed the significant part of conflicts in RE that manifest themselves, especially within the current methods used for developing software systems. Also, many researchers have noted that SE is collaborative (e.g., [19, 54, 62]) and requires the usage of different tools and techniques and participation by stakeholders. In particular, conflicts become inevitable whenever stakeholders have different expectations for a system [12]. Studies have shown that conflicts in SE practices are common [63], and the RE process is the most suitable part of a SE process for addressing conflicting viewpoints and expectations [9, 79]. Conflicts left unattended in a real-life development project tend to escalate [52] and negatively affect the system to be developed. Notably, existing methods do not address practical and inclusive ways of resolving conflicts.

Further, the extent of research in RE devoted to conflicting requirements indicates that conflict resolution requires proper decision-making because unresolved conflicts will always have a strong negative effect on software product success and customer satisfaction [34]. Previous work on conflicts in RE focused on identifying and resolving requirements in general terms [47]. For example, Barchiesi et al. [7] observed that conflicts are resolved through negotiations between human participants [66]. Nevertheless, the negotiation approach could not provide the expected satisfaction for stakeholders [57]. In Easterbrook [22], the usage of the computer-supported negotiation system

(Synoptic) was proposed. However, it could not guarantee absolute satisfaction for the stakeholders. Boehm et al. [10] introduced the Win-Win technique to solve risks and reconcile uncertainties through a negotiation approach. Still, the approach suffers from some setbacks in selecting a resolution plan and scalability.

Several other authors have proposed resolving conflicts in requirements using other techniques (e.g., [11, 14, 24, 25, 63, 79, 82, 87]). Still, inconsistencies and contradictions [36] in stakeholders' views of the system to be developed often arise. These problems hinder the mutual satisfaction of stakeholders and result in poor requirements specifications and faulty systems. This points to the need for a more stakeholder-centric analytical approach that supports conflict resolution through requirements negotiation and prioritization. The desired approach would enable requirements engineers to obtain a consensus among stakeholders.

In addition, some work in conflict resolution has focused on conflicts in particular kinds of requirements and systems, such as conflicts among non-functional requirements [11, 25, 41, 59, 68]; conflicts in pervasive computing systems [46]; compliance requirements [53]; requirements classification [86], web application requirements [78]; contextual requirements [4]; requirements in aspect-oriented RE [13, 69]; requirements in goal-oriented RE [33, 38, 43, 55, 79], and so on.

For instance, van Lamsweerde et al. [79] used a formal and heuristical approach to identify conflicts between requirements by multiple stakeholders specified as goals. The method by van Lamsweerde et al. [79] borders with matching these goals with existing domain-specific divergence patterns, which was based on previous experiences in conflict detection. However, this approach includes some bias in the process of conflict identification, which makes the approach inefficient in a situation where mutual agreement among stakeholders is pertinent [39].

Additionally, the technique by van Lamsweerde et al. [79] only uncovers conflicts. It cannot differentiate the contradictions in the different stakeholders' specifications. The argumentation approach by Jureta et al. [43] supports integration of claims into the goal model. For that, the stakeholders decide on acceptable criteria and can convert the requirements to argument-based models. This approach by Jureta et al. [43] intends to document stakeholders' decision-making processes to generate the specifications without resolving conflicts. Notably, other researchers [81] suggested various frameworks that have not been implemented and experimented with in real-life case studies.

In Table 1, we provide a summary of nine other related works and uncover some strengths and weaknesses based on seven criteria: (i) modeling conflicts in elicited requirements, (ii) ability to identify conflict, (iii) ability to generate a resolution scheme, (iv) ability to make a resolution selection, (v) involvement of stakeholders, (vi) ability to evaluate for consensus, and (vii) scalability..

Differently, in our research, we introduce a streamlined approach for describing the views by the stakeholders of a system to be developed and reconciling the conflicting views and use the framework for conflict identification and resolution by engaging the stakeholders extensively in the process. We achieve this by combining expert-based and clustering approaches. As the literature reflects, the clustering approach by Veerappa and Leiter has been used to group the stakeholders' requirements into clusters. The goal is to make relevant decisions on the similarities to gain insight into the stakeholders'

**Table 1.** Strength and limitations of existing approaches to conflict resolution in RE.

Criteria for resolution	[10]	[27]	[79]	[56]	[24]	[85]	[78]	[55]	[38]	ReqCCR (our tool)
Modelling conflicts in elicited requirements	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ability to identify conflict	✓	✓	×	✓	✓	✓	✓	✓	✓	✓
Ability to generate a resolution plan	Partially	×	×	✓	✓	Partially	×	Partially	×	✓
Ability to make a resolution selection	×	×	×	×	×	×	×	Partially	×	✓
Experts and/or other stakeholders involvement	Partially	×	×	×	×	✓	Partially	Partially	✓	✓
Ability to evaluate for consensus	Partially	×	×	×	×	Partially	×	×	×	✓
Scalability	×	×	×	×	×	×	Partially	×	×	✓

preferences while explaining the relationships between different requirements. In RE, clustering has been used to improve the quality of the system’s requirements [48, 89] and requirements reuse [8, 48]. Clustering as an unsupervised machine learning approach is remarkable and can improve the entire RE methodologies, especially with the recent emergence of explainability and freedom from discrimination as new requirements [83].

## 4 Methodological Approach

We addressed the study through qualitative and quantitative approaches in the case study research [88]. The philosophy behind our approach contributes to the theories about the collection, analysis, and further processing of data about the phenomenon being studied [16]. In this regard, we based our research on the positivist (quantitative) and interpretivist (qualitative) philosophies by employing expert-based and clustering techniques for conflict resolution in RE.

The positivist (quantitative) aspect considers the phenomenon that is measurable by using statistical instruments. This is complemented by the interpretivist approach that helps to understand the phenomenon without searching for determinism or universal laws [65] and supports the interpretation of outcomes based on the context, participants, and resources. The interpretivist (qualitative) approach relates to the interpretivist school of thought about resolving conflicts that arise from requirements expressed by different stakeholders. In our research, the qualitative approach allowed for a number of alternative interpretations of reality that jointly accommodate the scientific knowledge behind conflict resolution.

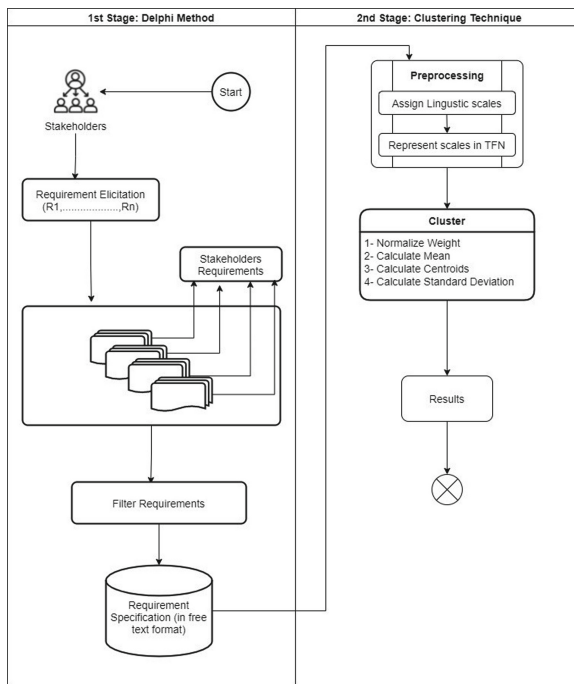
We used the statistical instruments rooted in the Delphi method [44] and the clustering technique for measuring the similarity of requirements. While the Delphi technique supports setting priorities and gaining consensus (agreement) on an issue [44], the clustering approach offers the potential to tackle how to consequently and proficiently coordinate large numbers of requests from stakeholders and organize the resulting requirements into a coherent structure.

We considered requirements elicitation from different stakeholders as a “warfare” process because every stakeholder has his/her views of the same system being designed and is often passionate about her/his understanding of the system. We modified the Delphi technique [44] for filtering and ranking requirements and reducing duplication. Also, we engaged experts to resolve the conflicting requirements. The experts who were

engaged in the modified Delphi process were the pharmacists. They were selected based on their number of years of experience. They had the same background and training, but they had different values that make conflicts between their viewpoints inevitable as humans.

#### 4.1 Framework for Resolution

Our framework is a unified process that recognizes the importance of communication between the stakeholders during requirements elicitation. It comprises streamlined methods for describing the stakeholders' views of a system being designed through an expert-oriented approach - the Delphi method [44]. The framework we have described is shown in Fig. 1. This framework for conflict resolution suggests a process flow that is iterative and incremental. It offers an evolutionary feel crucial in modern SE processes [32].



**Fig. 1.** A description of the framework for resolution (Modified from [32]).

As Fig. 1 reflects, the framework consists of two stages. At the first stage, the modified Delphi method is used in two iterations. In the initial iteration, requirements are elicited from the stakeholders using qualitative interviews, stakeholder analysis, brainstorming sessions, focus group approach, and scenario generation. The next iteration at the first stage involves filtering the lists of the elicited requirements by generating

a master list of requirements. At this point, the selected experts are involved in the process, where their opinions and the expectations extracted from the interviews are captured to produce the master list. With the application of the Delphi method, it was possible to obtain consensus amongst the lists of elicited requirements and produce the final list of requirements for clustering.

After producing the list of user requirements, the second stage of the framework (in Fig. 1) is identifying conflicts and applying the clustering approach. First, we prioritized the requirements based on the linguistic evaluations by the stakeholders. As is shown in Table 2, we used the ranking scales as variables to capture the ranking scales for each stakeholder on each requirement evaluated<sup>1</sup>. Also, since conflicts can be viewed as uncertainties, the fuzzy logic approach reported in [1] and [30] was followed. Thus, we obtained the ranking scale of preference weights for requirements prioritized by different stakeholders on each requirement in the second iteration<sup>2</sup>. We explain in Subsect. 4.2 how our conflict resolution framework (in Fig. 1) was used to identify and resolve conflicts between requirements.

**Table 2.** Ranking and weight scales [32].

Linguistic variables	Weight
Very Low (VL)	1
Low (L)	2
Medium (M)	3
High (H)	4
Very High (VH)	5

## 4.2 Process of Conflict Identification and Resolution

**Conflict Identification Process.** We used Kendall's Coefficient of Concordance (KCoC) [45] for identifying and ascertaining the presence of conflicts based on the weights assigned by each stakeholder. KCoC is a statistical test used within the modified Delphi technique to evaluate consensus and conduct several rankings for N objects or individuals. Given k sets of rankings, KCoC was used to determine the associations among these rankings and served as a measure of agreement among the stakeholders. Thus, KCoC aimed to indicate the levels of disagreement between the stakeholders' views considering the various weights assigned by them to the elicited sets of requirements. We denote KCoC as  $W$  and define it as follows:

**Definition 1:** Let us assume that m stakeholders assign weights to k requirements that are ordered from 1 to k. Let  $r_{ij}$  denote the rating the stakeholder  $j$  gives to the requirement  $i$ . For each requirement  $i$ , let  $R_i = \sum_{j=1}^m r_{ij}$  and let  $\bar{R}$  be the mean of the  $R_i$ , and let  $R$  be the squared deviation [70], that is:

<sup>1</sup> <https://doi.org/10.5281/zenodo.4603841>.

<sup>2</sup> <https://doi.org/10.5281/zenodo.4603824>.



$$R = \sum_{i=1}^k *(R_i - \bar{R})^2 \quad (1)$$

Now  $W$  is defined by:

$$W = \frac{12R}{m^2(k^3 - k)} \quad (2)$$

For each stakeholder  $j$

$$\sum_{i=1}^k r_{ij} = 1 + 2 + \dots + k = \frac{k(k+1)}{2} \quad (3)$$

Therefore, the mean of the  $R_i$  is:

$$\bar{R} = \frac{1}{k} \sum_{i=1}^k R_i = \frac{1}{k} \sum_{i=1}^m \sum_{j=1}^k r_{ij} = \frac{1}{k} \sum_{j=1}^m \sum_{i=1}^k r_{ij} = \frac{1}{k} \sum_{j=1}^m \frac{k(k+1)}{2} = \frac{m(k+1)}{2} \quad (4)$$

On the other hand, if  $W$  is:

$$W = \frac{12S^2}{m^2(k^3 - k)} - \frac{3(k+1)}{k-1} \quad (5)$$

where

$$S^2 = \sum_{i=1}^k *R_i^2 \quad (6)$$

Based on the Wilcoxon Rank Sum Test in [70], if all the stakeholders are in a complete agreement (that is, they give the same rating to each of the requirements), then

$$S^2 = \sum_{i=1}^k *R_i^2 = \sum_{i=1}^k *(m_j)^2 = m^2 * \sum_{i=1}^k *j^2 \quad (7)$$

But

$$\sum_{i=1}^k *j^2 = \frac{k(k+1)(2k+1)}{6} \quad (8)$$

and therefore

$$W = \frac{12S^2}{m^2(k^3 - k)} - \frac{3(k+1)}{k-1} = \frac{12m^2 k(k+1)(2k+1)}{6m^2 k(k-1)(k+1)} - \frac{3(k+1)}{k-1} = \frac{2(2k+1)}{k-1} - \frac{3(k+1)}{k-1} = 1 \quad (9)$$

In the opposite case, if all the values of  $R_i$  are the same (that is, if the stakeholders are in a complete disagreement), then by definition 1,  $W = 0$ . Most often,  $0 \leq W \leq 1$ . We used Algorithm 1 [32] to identify the presence of conflicts. When computing the value of  $W$ , we arranged the dataset into a  $k \times N$  table. Each row represents the weights assigned by a particular stakeholder to  $N$  requirements. After that, each column of the table was summed up ( $R_i$ ) and divided by  $k$  to find the average rank  $R_i$ . The resulting average ranks were then summed and divided by  $k$  to obtain the mean value of the  $R_i$ 's.

We expressed each of the average ranks as a deviation from the grand mean. Thus, we computed  $W$ , as is shown in Eq. (10) [32]:

$$W = \frac{\sum_{i=1}^N (\overline{Ri} - \overline{R})^2}{N(N^2 - 1)/12} \quad (10)$$

where  $K$  is the number of sets of rankings, i.e., the number of stakeholders;  $N$  is the number of requirements ranked;  $Ri$  is the average weight assigned to the  $i$ th requirement; and  $R$  is the average (or grand mean) of the weights assigned across all requirements.

---

**Algorithm 1.** The algorithm for conflict identification [31].

---

**Input:**  $k$  : number of stakeholders (integer);  $D[\text{row}][\text{col}]$  : data set in form of  $k * n$ ;  $n$  : number of requirements (integer);  $Ri$  average of the weight;  $R$  average of all objects

**Output:**  $W$

```

1: for (int i=0, i<k, i++) do
2:   for (int j=0, j<n, j++) do
3:      $r_{ij} = j * (j + 1)/2$ ;
4:     enter  $D[i][j]$ ;
5:   end for
6:    $R_i = r_{ij}$ 
7:    $\overline{R} = R_i$ ;
8:    $R = (R_i - R_i)^2$ 
9:    $W = R/(n(n^2 - 1)/2)$ ;
10:  if  $W=0$  then
11:    There are conflicting expectations
12:  else if  $W=1$  then
13:    No conflicts
14:  end if
15:  Returnmessage
16: end for

```

---

In Eq. (10),  $N(N^2 - 1)/12$  is the maximum possible sum of the squared deviations. That value would hold if there was a seamless understanding among the  $k$  rankings, and the average rankings were 1, 2, . . . ,  $N$ . If  $W = 0$ , it means that there are conflicting expectations based on the subjective weights assigned by each stakeholder, i.e., there is a conflict. If  $W = 1$ , it means that the stakeholders agree about the weights they assigned to each requirement, i.e., there is no conflict. Values between 0 and 1 are approximated to the values 0 and 1 to represent the variability ratio for evaluating consensus [45]. In our case, the KCoC was calculated to be  $0.000115598 \approx 0.00$ , which by approximation is 0.

**Conflict Resolution Process.** In our research work, the K-Means clustering algorithm established the framework for resolving conflicts by grouping the datasets of requirements based on the assigned weights into classes of similar requirements called clusters. The weights assigned to the requirements  $R_1 \dots R_n$  by each stakeholder  $S_i$  represent

the attributes. At the same time, each stakeholder represents an instance of a class (cluster) as specified in the dataset. The K-Means algorithm is described as Algorithm 2.

We used the clustering approach to establish a scheme for conflict resolution. Two major activities of the clustering approach - data preprocessing and data clustering - were used to resolve conflicts. We preprocessed the dataset and used the K-Means clustering algorithm [73] to divide the requirements into clusters. The K-Means algorithm calculates distances between each point of the dataset containing the instances and attributes and the center by utilizing the Euclidean distance measure [40, 73]. Also, the K-Means algorithm automatically normalizes numerical attributes in the process of computing the Euclidean distance [73].

We used the K-Means clustering algorithm to obtain the clusters and identify the most desirable user requirements. The algorithm takes the number of clusters  $K$  as input, generates the initial clusters from the dataset, and computes the average of each individual cluster in the dataset to determine the relative closeness degrees and consistency indexes of the requirements contained by the cluster. Also, the K-Means algorithm assigns each individual record in the dataset to one of the initial clusters using the Euclidean Distance Measure [40]. The individual record is then re-assigned to the most similar cluster in the dataset. After that, the averages of all of the most similar clusters are recomputed. The K-Means algorithm is iterative and ensures the establishment of stable clusters [37]. With the K-Means clustering algorithm, we were able to formulate the framework for conflict resolution. The framework enabled identifying the centers of agreements to support decision-making on the lists of requirements by different stakeholders.

---

**Algorithm 2.** The K-Means algorithm for clustering [31].

---

**Input:**  $k$ : number of clusters

**Output:** the set of clusters

```

1: for (int i=0, i<k, i++) do                                     ▷ iteration until  $ii$  from 0 to  $k$ 
2:   for (int j=0, j<n, j++) do
3:     Enter  $D[i][j]$ 
4:     random_value=rand(1tok + 1)    ▷ calculate the random value of object(s) entered
5:                                     ▷ initial cluster centers
6:     for (int k=0, k<random_value, k++) do                    ▷  $k$  is the no. of clusters
7:       dist = square  $((k - i)^2 + (k - j)^2)$ ;    ▷ determine which  $k$  (clusters) is closer
8:       dist = sq
9:       if dist=j then
10:        sum +=i
11:       end if
12:     end for
13:     sum_cluster+=sum
14:   end for
15:   return sum_cluster
16: end for

```

---

## 5 System Implementation

We implemented an interactive system to empirically prove the adequacy of our framework on a 64-bit Windows 10 Enterprise Operating System (OS) with an Intel(R)

Core™ i5-8250U CPU @ 1.60 GHz processor and 16 GB RAM. The interactive system called Requirement Clustering for Conflict Resolution (ReqCCR) assumed relative weights of requirements provided by the stakeholders as input to generate a list of prioritized requirements. The sample data set used for clustering was based on the “stakeholders’ ranked dataset” available in a spreadsheet CSV format (requirement-datasetN.csv)<sup>3</sup>.

As such, for each element in the dataset, the Euclidean distance between individual requirements and clusters was computed, as was explained in Sect. 4. Table 3 shows the normal distribution of the first 25 requirements with their corresponding minimum and maximum values, the mean, and standard deviation (*stdev*). The distribution presents the data in its normalized form, allowing it to be scaled to fall within a small specified range for clustering. The purpose of normalizing the dataset is to prevent variables with the most significant values from dominating the measure. Again, normalization before clustering was required to determine the Euclidian distance sensitive to differences in the magnitude or scales of the attributes [18]. During the clustering process, we defined the total number of clusters “K” as 5. This number was decided based on the five Ranking Scales used as labels. In general, the K-Means algorithm was quite sensitive to how clusters were initially assigned. Clustering by the K-means algorithm gave us 5 clusters, as shown in Fig. 2, 3, 4, 5 and 6. The results of clustering include 101 requirements in each cluster. The K-Means clustering algorithm split the requirements Ri . . . Rn into k clusters in which each requirement belongs to the cluster with the nearest mean.

**Table 3.** Normal distribution of requirements after preprocessing.

RiD	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25
Min	1	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	3	2	1	2	2	2	1	2	1
Max	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Centroids	4.167	3.667	4.5	3.786	4.405	3.571	4.429	4.5	4.071	3.69	4	3.738	3.452	3.405	3.095	4.024	4.405	4.024	3.905	4.119	4.238	4.5	4.143	4.333	4.167
Stdev	1.208	1.391	0.994	1.138	1.149	1.252	0.77	0.834	1.156	1.316	1.23	1.17	1.4	1.326	1.376	1.137	0.798	0.95	1.165	1.017	1.031	0.773	1.117	0.816	0.853

RiD	Centroids	Stdev	RiD	Centroids	Stdev	RiD	Centroids	Stdev	RiD	Centroids	Stdev	RiD	Centroids	Stdev
R1	4.00	0.89	R22	4.17	0.75	R43	2.67	0.82	R64	3.50	0.84	R85	3.67	0.52
R2	3.50	1.64	R23	3.67	1.51	R44	3.83	0.75	R65	3.17	1.47	R86	4.00	0.89
R3	4.67	0.52	R24	3.83	0.98	R45	3.33	0.52	R66	3.50	0.84	R87	3.33	0.82
R4	3.17	0.98	R25	4.00	0.63	R46	3.50	0.84	R67	3.17	0.98	R88	3.83	0.75
R5	4.50	0.55	R26	3.33	1.03	R47	2.83	0.98	R68	3.50	1.05	R89	4.67	0.52
R6	2.83	1.33	R27	3.83	0.98	R48	3.83	0.75	R69	2.67	1.03	R90	3.83	0.75
R7	4.00	0.63	R28	3.33	0.52	R49	3.17	0.98	R70	3.17	1.47	R91	3.50	0.84
R8	4.33	0.52	R29	3.50	1.05	R50	3.00	0.89	R71	2.83	0.75	R92	3.17	0.75
R9	4.17	0.75	R30	3.33	1.03	R51	2.67	1.63	R72	3.17	1.47	R93	4.17	0.98
R10	3.67	0.82	R31	3.17	1.33	R52	3.00	0.89	R73	3.00	1.41	R94	3.50	0.55
R11	3.83	0.41	R32	3.17	0.75	R53	3.33	1.03	R74	2.83	0.75	R95	3.50	0.55
R12	3.33	1.03	R33	3.50	0.84	R54	2.83	1.17	R75	2.67	1.37	R96	3.83	0.41
R13	3.33	1.03	R34	3.50	1.05	R55	3.17	1.33	R76	3.67	1.37	R97	3.83	0.98
R14	3.17	1.47	R35	3.50	0.55	R56	3.33	1.51	R77	3.50	1.05	R98	3.67	0.82
R15	3.00	0.89	R36	3.17	0.75	R57	3.17	1.33	R78	3.83	0.41	R99	3.67	0.52
R16	4.17	0.75	R37	3.17	1.17	R58	3.50	0.55	R79	3.00	0.89	R100	3.83	0.75
R17	4.17	0.75	R38	3.67	0.52	R59	3.33	1.03	R80	3.67	0.52	R101	4.00	0.89
R18	3.50	1.05	R39	3.67	1.21	R60	3.00	0.89	R81	3.83	1.17	-	-	-
R19	4.00	0.89	R40	3.67	0.82	R61	3.50	1.05	R82	3.67	1.51	-	-	-
R20	3.50	1.05	R41	3.50	1.52	R62	3.50	1.05	R83	3.67	1.37	-	-	-
R21	3.67	1.03	R42	3.67	0.82	R63	3.17	1.17	R84	3.17	1.47	-	-	-

**Fig. 2.** Cluster 1.

<sup>3</sup> <https://doi.org/10.5281/zenodo.4603824>.

RID	Centroids	Stdev	RID	Centroids	Stdev	RID	Centroids	Stdev	RID	Centroids	Stdev	RID	Centroids	Stdev
R1	5.00	0.00	R22	2.50	0.71	R43	2.00	0.00	R64	2.00	0.00	R85	2.50	0.71
R2	2.00	0.00	R23	2.50	0.71	R44	2.50	0.71	R65	3.00	1.41	R86	3.00	1.41
R3	2.00	1.41	R24	2.50	0.71	R45	2.50	0.71	R66	2.50	0.71	R87	2.50	0.71
R4	2.50	2.12	R25	2.00	1.41	R46	2.00	0.00	R67	2.00	1.41	R88	3.00	1.41
R5	3.50	2.12	R26	2.00	1.41	R47	2.50	0.71	R68	2.50	2.12	R89	1.50	0.71
R6	2.50	2.12	R27	2.00	0.00	R48	2.00	0.00	R69	2.00	1.41	R90	1.50	0.71
R7	2.50	0.71	R28	2.00	1.41	R49	2.00	0.00	R70	2.50	2.12	R91	2.00	0.00
R8	2.00	0.00	R29	2.50	2.12	R50	2.00	1.41	R71	2.50	0.71	R92	2.50	0.71
R9	1.50	0.71	R30	2.50	0.71	R51	2.00	1.41	R72	3.00	1.41	R93	2.50	0.71
R10	1.50	0.71	R31	2.50	0.71	R52	2.00	0.00	R73	2.50	0.71	R94	2.50	0.71
R11	2.00	0.00	R32	2.50	0.71	R53	1.50	0.71	R74	2.50	0.71	R95	2.00	1.41
R12	2.00	0.00	R33	2.00	0.00	R54	2.00	1.41	R75	2.50	0.71	R96	2.00	1.41
R13	2.00	1.41	R34	2.00	0.00	R55	1.50	0.71	R76	2.50	0.71	R97	2.00	1.41
R14	2.00	1.41	R35	2.50	0.71	R56	2.50	0.71	R77	2.50	0.71	R98	2.00	1.41
R15	1.50	0.71	R36	3.00	1.41	R57	1.00	0.00	R78	2.50	2.12	R99	2.50	0.71
R16	1.50	0.71	R37	1.50	0.71	R58	2.50	0.71	R79	2.00	1.41	R100	2.50	0.71
R17	3.00	0.00	R38	2.50	0.71	R59	2.00	0.00	R80	2.00	1.41	R101	3.00	0.00
R18	3.00	0.00	R39	1.50	0.71	R60	2.50	0.71	R81	2.50	0.71	-	-	-
R19	1.50	0.71	R40	2.00	1.41	R61	2.00	0.00	R82	2.50	0.71	-	-	-
R20	2.00	0.00	R41	1.50	0.71	R62	2.00	0.00	R83	3.50	0.71	-	-	-
R21	2.00	0.00	R42	2.00	0.00	R63	2.00	0.00	R84	2.50	0.71	-	-	-

Fig. 3. Cluster 2.

RID	Centroids	Stdev	RID	Centroids	Stdev	RID	Centroids	Stdev	RID	Centroids	Stdev	RID	Centroids	Stdev
R1	4.82	0.40	R22	4.91	0.30	R43	3.91	1.14	R64	4.73	0.47	R85	4.82	0.40
R2	4.55	1.21	R23	4.82	0.40	R44	4.55	0.52	R65	4.82	0.40	R86	4.82	0.40
R3	4.91	0.30	R24	4.82	0.40	R45	4.73	0.47	R66	4.73	0.47	R87	4.64	0.67
R4	4.36	1.21	R25	4.45	0.69	R46	4.82	0.40	R67	4.82	0.40	R88	4.91	0.30
R5	5.00	0.00	R26	4.27	0.90	R47	4.73	0.47	R68	4.91	0.30	R89	5.00	0.00
R6	4.27	0.79	R27	4.09	1.22	R48	4.73	0.47	R69	4.82	0.40	R90	5.00	0.00
R7	4.82	0.60	R28	4.00	1.18	R49	4.64	0.50	R70	4.82	0.60	R91	4.73	0.65
R8	5.00	0.00	R29	4.36	0.92	R50	4.55	0.52	R71	4.82	0.60	R92	5.00	0.00
R9	4.55	1.21	R30	4.18	0.98	R51	4.73	0.47	R72	4.45	1.04	R93	4.91	0.30
R10	4.73	0.47	R31	4.36	0.92	R52	4.91	0.30	R73	4.64	1.21	R94	4.91	0.30
R11	4.82	0.40	R32	4.55	0.93	R53	4.91	0.30	R74	4.91	0.30	R95	4.82	0.40
R12	4.00	1.18	R33	4.36	1.21	R54	4.82	0.40	R75	4.82	0.40	R96	4.82	0.40
R13	3.82	1.33	R34	4.64	0.67	R55	4.82	0.40	R76	5.00	0.00	R97	4.82	0.40
R14	4.27	0.90	R35	4.27	1.10	R56	4.82	0.40	R77	4.91	0.30	R98	4.82	0.40
R15	3.82	1.17	R36	3.73	1.62	R57	4.73	0.65	R78	5.00	0.00	R99	4.82	0.40
R16	4.73	0.65	R37	4.45	0.93	R58	4.91	0.30	R79	4.82	0.40	R100	4.82	0.40
R17	4.91	0.30	R38	4.00	1.61	R59	4.91	0.30	R80	4.91	0.30	R101	4.91	0.30
R18	4.45	0.69	R39	3.82	1.54	R60	4.91	0.30	R81	4.64	0.67	-	-	-
R19	4.64	0.67	R40	3.55	1.69	R61	4.73	0.47	R82	4.91	0.30	-	-	-
R20	4.73	0.47	R41	3.82	1.66	R62	4.82	0.40	R83	4.64	0.67	-	-	-
R21	4.82	0.60	R42	3.82	1.47	R63	4.82	0.40	R84	4.82	0.40	-	-	-

Fig. 4. Cluster 3.

RID	Centroids	Stdev	RID	Centroids	Stdev	RID	Centroids	Stdev	RID	Centroids	Stdev	RID	Centroids	Stdev
R1	3.84	1.39	R22	4.53	0.70	R43	3.63	1.07	R64	4.32	0.58	R85	4.63	0.50
R2	3.42	1.22	R23	3.89	1.10	R44	4.37	0.60	R65	4.32	0.67	R86	4.47	0.61
R3	4.42	1.07	R24	4.42	0.69	R45	4.11	0.99	R66	4.00	0.75	R87	4.32	0.58
R4	3.63	0.96	R25	4.26	0.73	R46	3.74	0.87	R67	4.26	0.81	R88	4.42	0.84
R5	4.21	1.27	R26	3.95	0.97	R47	4.26	0.81	R68	4.21	0.86	R89	4.68	0.75
R6	3.79	1.03	R27	4.32	0.95	R48	4.58	0.51	R69	4.00	1.00	R90	4.58	0.51
R7	4.53	0.61	R28	3.58	1.17	R49	4.11	0.74	R70	4.05	0.78	R91	4.21	0.71
R8	4.53	0.70	R29	3.68	0.82	R50	4.16	1.01	R71	4.16	0.69	R92	4.16	0.76
R9	3.89	0.99	R30	3.74	1.19	R51	4.32	0.67	R72	4.00	0.82	R93	4.63	0.68
R10	3.63	1.26	R31	4.00	0.88	R52	4.58	0.51	R73	4.26	1.10	R94	4.58	0.77
R11	4.05	1.18	R32	3.95	1.08	R53	4.32	0.58	R74	4.21	0.86	R95	4.53	0.70
R12	3.95	1.03	R33	3.95	0.91	R54	4.53	0.61	R75	4.21	0.86	R96	4.32	0.82
R13	3.47	1.50	R34	3.74	1.10	R55	4.16	0.76	R76	4.37	0.68	R97	4.37	0.76
R14	3.26	1.15	R35	3.58	1.07	R56	4.32	0.82	R77	4.26	0.65	R98	4.16	1.17
R15	3.05	1.39	R36	3.63	1.16	R57	4.16	0.76	R78	4.16	0.69	R99	4.21	1.03
R16	4.00	0.94	R37	4.16	1.01	R58	4.21	0.71	R79	4.16	0.76	R100	4.37	1.07
R17	4.42	0.77	R38	4.21	0.79	R59	4.00	0.82	R80	4.21	0.71	R101	4.53	0.77
R18	4.05	0.97	R39	3.89	0.99	R60	4.32	0.82	R81	4.21	0.71	-	-	-
R19	3.89	1.05	R40	3.37	1.26	R61	4.42	0.51	R82	4.63	0.60	-	-	-
R20	4.21	0.86	R41	3.37	1.07	R62	4.42	0.51	R83	4.47	0.84	-	-	-
R21	4.32	0.82	R42	3.26	0.93	R63	4.42	0.51	R84	4.32	0.75	-	-	-

Fig. 5. Cluster 4.

## 6 Empirical Analysis

The research has a mixed approach, as was discussed earlier in Sect. 4. We used these approaches in the case study of RE for the Pharmacy Information Systems to be developed for OAUTHC. Specifically, the case study approach [88] was adopted for elicitation and analysis purposes. Interviews, scenario generation, and document analysis were used for the case study approach.

RID	Centroids	Stdev	RID	Centroids	Stdev	RID	Centroids	Stdev	RID	Centroids	Stdev	RID	Centroids	Stdev
R1	3.75	1.89	R22	4.75	0.50	R43	1.75	0.96	R64	3.75	1.89	R85	2.75	1.71
R2	3.50	1.73	R23	5.00	0.00	R44	4.25	0.50	R65	3.50	1.73	R86	2.75	1.71
R3	4.75	0.50	R24	4.25	0.50	R45	4.00	0.82	R66	4.00	1.41	R87	3.50	1.00
R4	4.50	0.58	R25	4.25	0.50	R46	4.00	0.82	R67	3.50	1.91	R88	3.50	1.91
R5	4.00	2.00	R26	3.75	1.26	R47	2.75	0.96	R68	4.50	1.00	R89	4.75	0.50
R6	2.25	1.50	R27	4.00	1.41	R48	4.50	0.58	R69	3.75	1.89	R90	4.25	0.96
R7	4.50	0.58	R28	4.75	0.50	R49	4.50	1.00	R70	4.75	0.50	R91	2.00	2.00
R8	4.50	1.00	R29	3.50	1.73	R50	4.75	0.50	R71	2.50	1.91	R92	3.50	1.91
R9	4.75	0.50	R30	3.00	1.83	R51	4.75	0.50	R72	3.25	1.71	R93	4.25	1.50
R10	2.25	1.50	R31	3.00	1.41	R52	5.00	0.00	R73	4.00	2.00	R94	3.00	1.83
R11	2.75	2.06	R32	2.50	1.73	R53	4.75	0.50	R74	3.00	1.41	R95	2.25	1.89
R12	3.50	1.73	R33	2.50	1.29	R54	4.75	0.50	R75	2.50	1.91	R96	4.50	1.00
R13	3.25	1.71	R34	3.00	1.83	R55	4.25	1.50	R76	3.00	1.83	R97	4.50	1.00
R14	2.75	2.06	R35	1.75	0.50	R56	3.25	2.06	R77	3.00	1.83	R98	4.00	1.41
R15	2.25	1.89	R36	1.50	0.58	R57	3.25	2.06	R78	3.50	1.73	R99	2.50	1.91
R16	3.25	1.71	R37	2.75	1.50	R58	4.00	1.41	R79	2.75	2.06	R100	2.50	1.91
R17	4.00	1.15	R38	2.25	1.89	R59	2.75	1.71	R80	3.75	1.89	R101	4.00	0.82
R18	4.00	1.15	R39	2.00	0.82	R60	4.25	0.96	R81	1.75	1.50	-	-	-
R19	3.00	1.41	R40	1.25	0.50	R61	3.50	1.73	R82	2.75	2.06	-	-	-
R20	4.00	1.41	R41	1.25	0.50	R62	3.75	1.26	R83	3.00	1.63	-	-	-
R21	4.25	1.50	R42	1.25	0.50	R63	4.50	1.00	R84	3.50	1.00	-	-	-

Fig. 6. Cluster 5.

The interview process, which follows the guidelines provided in [88] and [16], entailed speaking directly with the stakeholders involved and asking them questions about their specific needs. These needs are relevant to the pharmacy information services provided at the OAUTHC. The interview started with predefined questions, and in the process, as lots of different critical issues arose, open discussion was granted on the expectations by the participants about the system to be designed. Thirty staff members from ten sub-units of the Pharmacy Department participated in the study<sup>4</sup>. After the interviews, the first author conducted a mini-workshop session with heads of sub-units to synthesize all the lists of requirements into a master list of requirements.

### 6.1 Analysis

The analysis of clusters required the examination of the cluster centroids [26]. These centroids are the clustering factors and the typical values of all the objects in a given cluster [60]. The clustered results (see Fig. 2, 3, 4, 5 and 6) show how the requirements are assigned to the different clusters formed, and the corresponding centroids and *stdev* values. Figure 7 represents the statistics on the number and percentage of instances assigned to different clusters. As Fig. 7 shows, cluster 1, 2, 3, 4 and 5 has 14%, 5%, 26%, 45%, and 10%, of the instances. The cluster centroids are the mean vectors for each cluster that are used to characterize the clusters.

<sup>4</sup> <https://doi.org/10.5281/zenodo.4603848>.

The execution of Algorithm 2 considered the means (averages) of feature values and their *stdev* values. Each cluster is defined by the mean, forming its center and *stdev*, center and perimeter or radius. The *stdev* value for each requirement in a cluster indicates how tightly the given clustered requirement is located around the centroid of the cluster's dataset. We used the "mean of mean" to assess how the values are spread either above or below the mean. We hypothesize that a high *stdev* value implies that the data is not tightly clustered and is therefore less reliable and consistent. In contrast, a low *stdev* value indicates that the data is clustered tightly around the mean, making it more reliable and consistent.

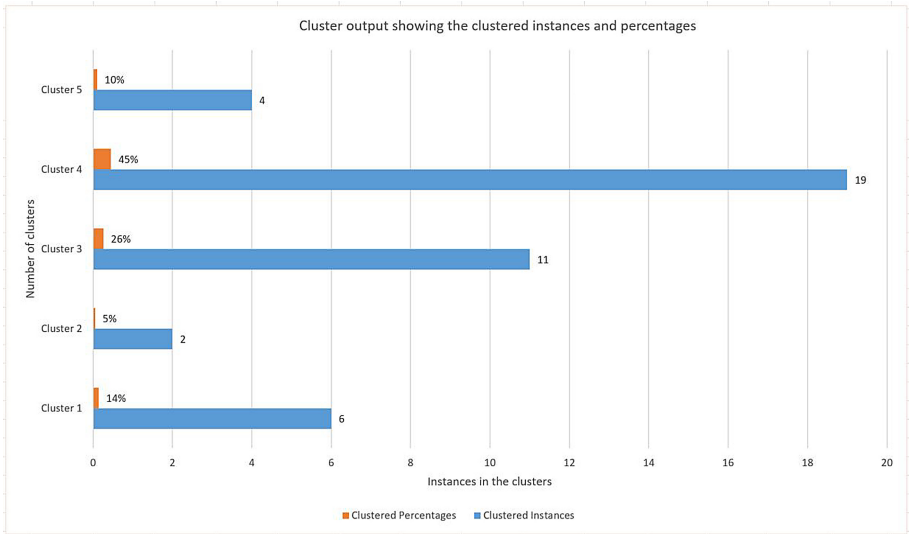


Fig. 7. Cluster output showing the clustered instances and percentages.

## 6.2 The Implication of Empirical Analysis of Cluster Output

Algorithm 2 divided the instances of the elicited requirements into five clusters. Each instance belongs to one and only one cluster. The five clusters (see Fig. 2, 3, 4, 5 and 6) represent the responses by the stakeholders based on the weight they assigned to each requirement. Figure 8 shows the visualized cluster assignments indicating clustered datasets and the classes assigned to each cluster [32]. The x-axis of Fig. 8 denotes the clusters, and the y-axis contains the number of instances in each cluster. Figure 9 shows a scattered chart comparing a selection of centroids of each cluster, indicating centroids of each cluster as separate points. As has been depicted in Fig. 9, clusters 3 and 4 have the highest values of centroids, as is indicated by the green triangle ( $\Delta$ ) and the purple cross shapes ( $\times$ ), respectively. The centroids of cluster 4 are closer to each other, and cluster 4 has the highest number of clustered instances, as Fig. 7 indicates. The centroids of cluster 2 indicated by the red rectangle ( $\square$ ) shapes are far from each

other. As Fig. 9 reflects, on the x-axis is the number of instances while the y-axis represents the clusters. Figure 10 shows the trend of the percentage that centroids of each cluster have contributed over time during the K-Means iteration process to the ordered categorization of the clusters. On the x-axis is the number of instances, while the y-axis represents the percentages of the cluster centroids.

### 6.3 Decision on Resolution Selection of Clusters

We applied the following scientific criteria as conditions to decide on the final results:

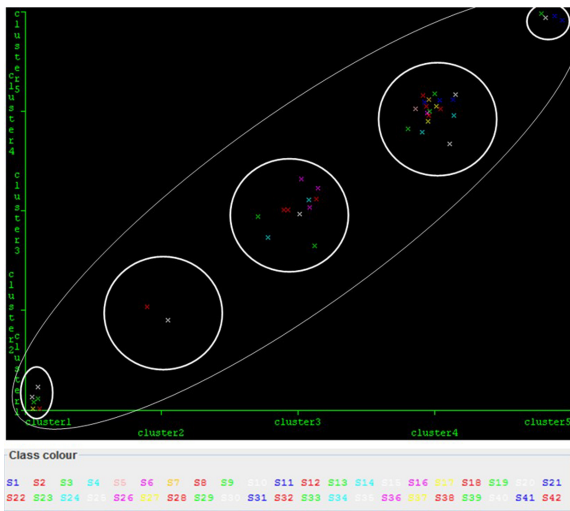


Fig. 8. Visualised cluster assignments [32].

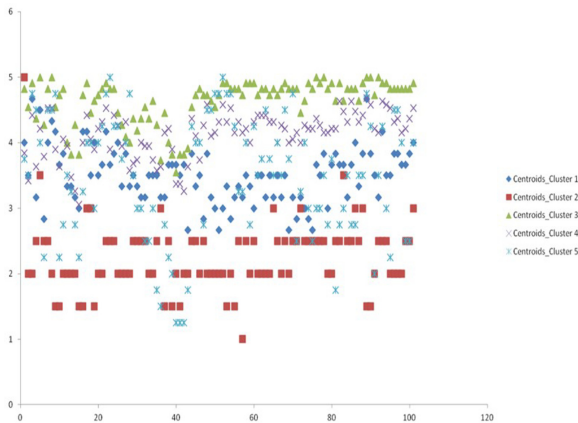
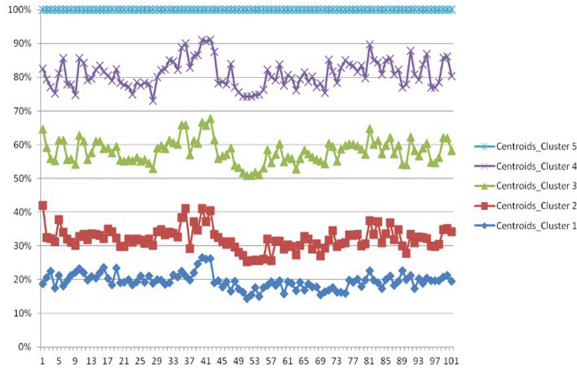


Fig. 9. A scattered chart comparing clustered centroids [32].





**Fig. 10.** Percentages of cluster centroids [32].

- (a) By inspecting the *stdev* value to eliminate clusters with relatively high *stdev* values. In the context of our research, the *stdev* value measures how well the stakeholders agree with each other. The lower the *stdev* value, the stronger the agreement level. A low *stdev* value implies that most of the requirements' instances are exceptionally near to the centroids, while a high *stdev* value implies that the instances are spread out [73]. The *stdev* value for each instance in a cluster determines how dispersed (spread out) the data is from the cluster's centroid. Thus, the *stdev* value establishes the centroid that gives a meaningful representation of the dataset. For example, the *stdev* value 0 would mean that every instance is exactly equal to the centroid. The closer the *stdev* is to 0, the more reliable the centroid is. Also, the *stdev* value close to 0 indicates that there is very little volatility in the sample.
- (b) By computing the average of the *stdev* value for each cluster to determine the cluster with the highest and lowest *stdev* value. As a result, the average *stdev* values for the clusters 1, 2, 3, 4 and 5 are respectively 0.95, 0.78, 0.61, 0.86 and 1.31. Thus, the cluster with the highest *stdev* value is cluster 5, while the cluster with the lowest *stdev* value is cluster 3. Also, by inspection, 81.19% of all the attributes with the lowest *stdev* value are in cluster 3, while 18.81% of all the attributes with the lowest *stdev* value are in the other clusters (i.e., 1, 2, 4, and 5).
- (c) By inspecting the number of instances assigned to each cluster. As shown in Fig. 7, clusters 1, 2, and 4 have a few instances allocated to them, making this cluster inappropriate for any meaningful decision. Clusters 3 and 4 have 11 and 19 instances allocated to them, respectively.

Against the scientific criteria described in the three preceding paragraphs (a), (b), and (c), the decision on the final cluster output was made. As a result, by comparing the average *stdev* value of each cluster with the corresponding average centroid value of the cluster, cluster 3 appeared to be the most reliable one. We observed that 82 out of the 101 requirements in cluster 3 have the lowest *stdev* value within the five clusters. In comparison, the remaining 19 requirements have the lowest *stdev* value within clusters 1, 2, 4, and 5.

Secondly, although cluster 4 has the highest number of instances assigned, this is not the most reliable and suitable criterion for making decisions. Instead of that, a decision on which cluster to use was based on the cluster with the lowest average *stdev* value. As a result of the cluster outputs shown in the respective Fig. 2, 3, 4, 5 and 6, cluster 3 appears to be the most reliable one because, for each requirement instance, the *stdev* value is very low (i.e., between 0.00 to 1.50) compared to the other clusters.

## 7 Model Validation

The software implementation of Algorithm 2 took 0.04 s to build the model with the complete dataset, which implies a good response time. The number of iterations performed was 5, and the sum of squared errors was 223.61. This sum of squared errors is a measure that is specific to K-Means. The squared error is the squared Euclidean distance of the requirement instance from the closest cluster center. The confusion matrix in Fig. 11 summarises our model validation results. The confusion matrix in Fig. 11 contains information about the actual and predicted classifications used to measure the model performance [32].

		P r e d i c t e d				
		4	0	0	4	0
A c t u a l	4	4	0	0	4	0
	1	1	2	0	0	0
	0	0	0	11	0	0
	0	0	0	0	15	0
	1	1	0	0	0	4

**Fig. 11.** Confusion matrix of K-Means clustering.

We performed sensitivity analysis and determined the confidence level to evaluate the completeness and consistency of the model for the given data in the matrix. In Table 4, the confusion matrix shown in Fig. 11 is summarized by presenting the numbers of predicted and actual requirements instances. Based on the data included by Table 4, we performed the sensitivity analysis (recall), determined the confidence level (precision), and computed the F-score and overall accuracy. The sensitivity analysis, also known as recall, defines the proportion of the actual positive cases correctly identified. The confidence level, also known as the precision, is the proportion of the positive cases that have been correctly identified. The F-score is the degree of the test’s accuracy

**Table 4.** Confusion matrix showing predicted and actual number of instances.

Clusters	Predicted instances	Number of actual instances
Cluster 1	4	8
Cluster 2	2	3
Cluster 3	11	11
Cluster 4	15	15
Cluster 5	4	5

to determine the harmonic mean of the recall and precision for each cluster for which the recall and precision have been calculated. This means that the F-score conveys the balance between the precision  $P$  and recall  $R$ . The accuracy determines the overall correctness of the classifier after prediction [29]. As Gambo and Taveter [32] observed, the accuracy helped to determine whether the resolution resulting from the model reflected the opinions by the stakeholders. The equations that were respectively used for calculating sensitivity, confidence, F-score, and accuracy are presented as the formulae 11, 12, 13, and 14 below:

$$Sensitivity(RecallR) = \frac{TP}{TP + FN} \quad (11)$$

$$Confidence(PrecisionP) = \frac{TP}{TP + FP} \quad (12)$$

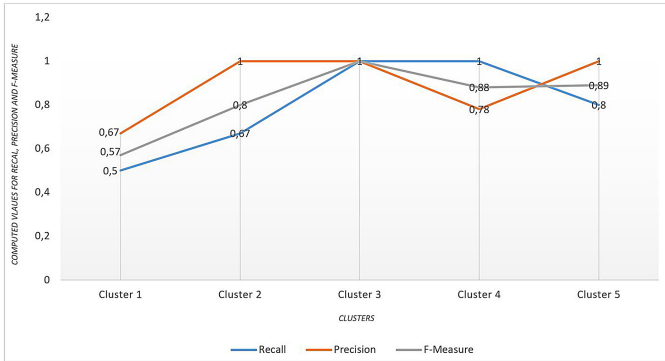
$$F - Score = 2 * \frac{P * R}{P + R} \quad (13)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

As included by Eqs. 11, 12, 13, and 14, TP is the number of true positives; FN is the number of false negatives; FP is the number of false positives; TN is the number of true negatives.

## 8 Results

Figure 12 shows the results of the model evaluation. As revealed by Fig. 12, clusters 3 and 4 had 100% recall, while clusters 1, 2 and 5 had 50%, 66.7%, and 80% recalls, respectively. This means that all of the positive cases correctly identified by the model belong to clusters 3 and 4. Additionally, the model indicated 100% precision for clusters 2, 3, and 5, respectively, while clusters 1 and 4 had 66.7% and 78.95% precision. The F-score showed the harmonic means of precision and recall to be 0.57, 0.80, 1.00, 0.88, and 0.89 for the respective clusters 1, 2, 3, 4, and 5.



**Fig. 12.** Results of the model evaluation.

The F-score of cluster 3 – with the value of 1.00 (100%) – is the most effective and reliable one. In other words, the implementation of Algorithm 2 correctly clustered all the requirement instances that initially belonged to cluster 3 to the same cluster. The result of the F-score value remarkably indicates that all the requirement instances belonging to cluster 3 were correctly clustered. Consequently, inspecting and comparing both the recall and precision proves that cluster 3 has the highest percentage of positive cases correctly identified and the highest predicted number of positive cases that turned out to be correct. This outcome justifies the reason why cluster 3 is the most reliable one for the final conflict resolution.

With the choice of cluster 3 for the final resolution, this result demonstrated that the model is complete and consistent. The total value of false negative (FN) requirement instances defines the number of incorrectly clustered instances, which is 6.0 (14.29%).

Overall, the framework for resolving conflicts we have presented in this paper achieved the accuracy of 85.71%. Consequently, this approach can cater for as many requirements as needed for any SE project. It can be adapted to solve a wide variety of decision-making and selection problems about the order of implementing requirements.

## 9 Discussion

Given the requirements’ dataset with 42 stakeholders as instances and 101 requirements as attributes, the value of Kendall’s Coefficient of Concordance  $W$  calculated by using Eq. 10 was  $0.000115598 \approx 0.00$ . The resulting value of  $W$  indicated some level of disagreement between the stakeholders’ subjective views. This means there are conflicts in the expectations by the stakeholders based on the weights assigned by them to individual requirements.

The framework we presented in Sect. 4 classifies ranked requirements by calculating for each requirement centroids and *stdev* values. This suggests that software engineers can utilize our framework to decide the most and least preferred requirements to support software release planning and avoid breaches of contracts, agreements, and trust [32].

After the clustering analysis, cluster 3 emerged as the final solution based on the scientific criteria outlined in Sect. 6.3 and the statistical evidence shown with the F-score value. Remarkably, the F-score for each cluster captures the properties of recall and precision by combining them as a single measure. The F-score also provided both the recall and precision with the same weight. This means both measures are balanced in importance, resulting in the harmonic mean consisting of two factors. For that, high precision and recall always result in a high F-score, as has been obtained in cluster 3. In addition the F-score takes care of imbalanced classes in the dataset of requirements.

The final solution in Fig. 13 indicates the conflict resolution by the K-Means algorithm presented in the order of priorities assigned to all of the requirements. As Fig. 13 shows, 77 requirements had a “very high” priority, corresponding to 76.24% of the entire set of requirements. On the other hand, 24 requirements had “high” priority, corresponding to 23.76% of the entire set of requirements. The evaluation of the model for completeness and consistency indicated 100% recall and precision of the final solution (cluster 3) and 85.7% accuracy of the resulting model.

Theoretically, our research confirmed that there is no perfect system. However, with 14.29% of incorrectly clustered instances, the experts agreed that the results were good enough for resolving the conflicting subjective views arising during requirements analysis.

Consequently, our research offers an improvement for the requirements engineering stage of SE. Our research results also demonstrate that the framework can accommodate for large sets of requirements by multiple stakeholders by resolving conflicts between these requirements with a very high precision level.

### 10 Threats to Validity

We identified two kinds of threats to validity, as described in [32]:

RID	Cent-roids	Stave	Resolved Weights	RID	Cent-roids	Stave	Resolved Weights	RID	Cent-roids	Stave	Resolved Weights	RID	Cent-roids	Stave	Resolved Weights
R5	5.00	0.00	VH	R11	4.82	0.40	VH	R10	4.73	0.47	VH	R18	4.45	0.69	H
R8	5.00	0.00	VH	R21	4.82	0.60	VH	R16	4.73	0.65	VH	R25	4.45	0.69	H
R76	5.00	0.00	VH	R23	4.82	0.40	VH	R20	4.73	0.47	VH	R37	4.45	0.93	H
R78	5.00	0.00	VH	R24	4.82	0.40	VH	R45	4.73	0.47	VH	R4	4.36	1.21	H
R89	5.00	0.00	VH	R46	4.82	0.40	VH	R47	4.73	0.47	VH	R29	4.36	0.92	H
R90	5.00	0.00	VH	R54	4.82	0.40	VH	R48	4.73	0.47	VH	R31	4.36	0.92	H
R92	5.00	0.00	VH	R55	4.82	0.40	VH	R51	4.73	0.47	VH	R33	4.36	1.21	H
R3	4.91	0.30	VH	R56	4.82	0.40	VH	R57	4.73	0.65	VH	R14	4.27	0.90	H
R17	4.91	0.30	VH	R62	4.82	0.40	VH	R61	4.73	0.47	VH	R6	4.27	0.79	H
R22	4.91	0.30	VH	R63	4.82	0.40	VH	R64	4.73	0.47	VH	R26	4.27	0.90	H
R52	4.91	0.30	VH	R65	4.82	0.40	VH	R66	4.73	0.47	VH	R35	4.27	1.10	H
R53	4.91	0.30	VH	R67	4.82	0.40	VH	R91	4.73	0.65	VH	R30	4.18	0.98	H
R58	4.91	0.30	VH	R69	4.82	0.40	VH	R19	4.64	0.67	VH	R27	4.09	1.22	H
R59	4.91	0.30	VH	R70	4.82	0.60	VH	R34	4.64	0.67	VH	R12	4.00	1.18	H
R60	4.91	0.30	VH	R71	4.82	0.60	VH	R49	4.64	0.50	VH	R28	4.00	1.18	H
R68	4.91	0.30	VH	R75	4.82	0.40	VH	R73	4.64	1.21	VH	R38	4.00	1.61	H
R74	4.91	0.30	VH	R79	4.82	0.40	VH	R81	4.64	0.67	VH	R43	3.91	1.14	H
R77	4.91	0.30	VH	R84	4.82	0.40	VH	R83	4.64	0.67	VH	R13	3.82	1.33	H
R80	4.91	0.30	VH	R85	4.82	0.40	VH	R87	4.64	0.67	VH	R15	3.82	1.17	H
R82	4.91	0.30	VH	R86	4.82	0.40	VH	R2	4.55	1.21	VH	R39	3.82	1.54	H
R88	4.91	0.30	VH	R95	4.82	0.40	VH	R9	4.55	1.21	VH	R41	3.82	1.66	H
R93	4.91	0.30	VH	R96	4.82	0.40	VH	R32	4.55	0.93	VH	R42	3.82	1.47	H
R94	4.91	0.30	VH	R97	4.82	0.40	VH	R44	4.55	0.52	VH	R36	3.73	1.62	H
R101	4.91	0.30	VH	R98	4.82	0.40	VH	R50	4.55	0.52	VH	R40	3.55	1.69	H
R1	4.82	0.40	VH	R99	4.82	0.40	VH	R72	4.45	1.04	VH	-	-	-	-
R7	4.82	0.60	VH	R100	4.82	0.40	VH	-	-	-	-	-	-	-	-

Fig. 13. Prioritized and resolved weights.

The first is internal validity, which involves eliciting, analyzing, and understanding the stakeholders' views while identifying the existence of conflicts. To mitigate this threat, we involved experts - pharmacists - in the process prescribed by the proposed conflict resolution framework. These experts have many years of experience in the problem domain. Remarkably, these experts have common backgrounds and training. They can coherently explain the views by different stakeholders to avoid the exclusion of view(s) and obtain consensus [32].

An additional threat to internal validity is the presentation and acceptance of our results. To mitigate this threat, we demonstrated the interactive system to the experts of the problem domain. We showcased the scientific process inherent in the solution to justify the conflict resolution procedure. Since the experts involved in our case study were scientifically inclined, they agreed with the results.

The second threat to validity is external validity. A threat to external validity is that the developed framework is yet to be validated in other problem domains within and outside the healthcare domain. Also, even while our research was conducted in the healthcare domain, the research was confined to only a subset of this problem domain. However, we anticipate that the overall results of our case study can be repeated to identify and resolve conflicts in a different problem domain where a large number of stakeholders is involved. Besides, we have explained and demonstrated our approach to some software engineers. They have provided positive feedback indicating that the framework proposed by us is required to determine the order of the requirements to be implemented during the software engineering process [30].

## 11 Conclusions and Future Work

The paper describes a framework for conflict identification and resolution spanning from requirements elicitation to requirements analysis. The framework consists of expert-based and clustering techniques for conflict resolution. The approach proposed by us constitutes a significant step that supports making crucial decisions about the software-intensive system to be developed, increasing the value offered by the system to its end users. In this paper, we present technical details of the framework, describe a software tool supporting the ideas, and describe the validation of the framework and the tool in the healthcare domain.

We emphasize that an important part of our approach consists of engaging stakeholders extensively in the conflict resolution process. Moreover, the whole Delphi process employed by the framework is stakeholder-centric because of the involvement by experts. Considering this, experts and other stakeholders participated fully in devising a unified conflict resolution scheme for the framework. In addition, the Delphi process ensured the reliability of various ideas explored and the generation of suitable information for decision-making. During this process, we introduced the ranking scale for prioritizing the specification of requirements. According to the ranking scale, the level of acceptability of requirements specified by each stakeholder was determined with the help of linguistic variables. The linguistic variables were used to capture the priority of each stakeholder on each requirement.

In addition, the clustering approach consisting of an unsupervised machine learning algorithm established the basis for achieving the resolution plan. The clustering technique organized various requirements into clusters so that the requirements inside a given cluster are more similar to each other than to the requirements belonging to the other clusters. In particular, it was possible with the clustering approach to characterize the dataset of requirements according to their weight scales suitable for analysis and prediction. For that purpose, we followed fuzzy logic [1, 30] to establish pairwise comparisons of criteria and alternatives by means of ranking scales. This resulted in a data matrix that could be applied for clustering. The research performed by us showed that the clustering approach in RE can improve the quality of the resulting requirements and contributes to requirements reuse [8, 89].

From the point of view of conflict management in RE, our research revealed two crucial features [32]. The first feature is the ability to cater for a large volume of requirements in a multi-stakeholder and multicultural setting. The second feature is the ability to successfully make clearcut decisions based on acceptable scientific criteria to minimize conflicts between prioritized sets of requirements expressed by the stakeholders. Importantly, these two features can considerably reduce software development costs and save time at the early stage of developing software products.

The results of this study suggest that the algorithm we implemented resolved the issues of scalability and computational complexity by providing a reliable conflict resolution. It can be implied that the stakeholders had some level of understanding of the choice of requirements needed for the Pharmacy Information System at the OAUTHC. The research results indicated that requirements engineers were able to resolve conflicts by means of our framework. Notably, the stakeholders participating in the case study agreed with the framework proposed by us. The research work also pointed out that there is no perfect system for resolving conflicts in requirements.

With respect to the future work, we suggest combining our strategy and tool with other methods and tools for data mining and analysis, such as the one proposed in [90], especially for dealing with large sets of requirements and many stakeholders in real life projects. By doing this, the scalability of our framework can be evaluated in different problem domains.

Additionally, we consider the need to conduct further research work in conflict management in the RE community to address conflicts in goal-oriented RE (GORE). For example, a pragmatic view about the need for better conflict management in GORE for sociotechnical systems (STS) is presented in [32]. This view is the first step that requires further investigation, especially considering the goals presented by the stakeholders, representing their needs and intentions. Interestingly, it is recommended to investigate conflicts in the agile agent-oriented modeling (AAOM) methodology for designing STS [71, 76, 77]. Within the AAOM methodology, it is also crucial to consider addressing some of the psychological problems stakeholders have in dealing with their collective and individual goals. In particular, we plan to continue the work reported in [74] about discovering and reconciling conflicts between emotional or affective requirements, which have been rendered as emotional goals attached to functional goals in a goal tree.

**Acknowledgment.** The authors would like to thank all the experts at the Pharmacy Department of the OAUTHC who provided invaluable input to the framework for resolution. The research work reported in this paper has received funding from the European Social Fund via the IT Academy programme and from the Mobilias Pluss Postdoctoral Research grant MOBJD343 by the Estonian Research Agency awarded to the first author.

## References

1. Achimugu, P., Selamat, A., Ibrahim, R.: A clustering based technique for large scale prioritization during requirements elicitation. In: Herawan, T., Ghazali, R., Deris, M.M. (eds.) *Recent Advances on Soft Computing and Data Mining*. AISC, vol. 287, pp. 623–632. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-07692-8\\_59](https://doi.org/10.1007/978-3-319-07692-8_59)
2. Ahmad, S.: Negotiation in the requirements elicitation and analysis process. In: 19th Australian Conference on Software Engineering (ASWEC 2008), pp. 683–689. IEEE (2008)
3. Aldekhail, M., Chikh, A., Ziani, D.: Software requirements conflict identification: review and recommendations. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* **7**(10), 326 (2016)
4. Ali, R., Dalpiaz, F., Giorgini, P.: Reasoning with contextual requirements: detecting inconsistency and conflicts. *Inf. Softw. Technol.* **55**(1), 35–57 (2013)
5. Alshazly, A.A., Elfatry, A.M., Abougabal, M.S.: Detecting defects in software requirements specification. *Alex. Eng. J.* **53**(3), 513–527 (2014)
6. Azadegan, A., Cheng, X., Niederman, F., Yin, G.: Collaborative requirements elicitation in facilitated collaboration: report from a case study. In: 2013 46th Hawaii International Conference on System Sciences, pp. 569–578. IEEE (2013)
7. Barchiesi, M.A., Costa, R., Greco, M.: Enhancing conflict resolution through an AHP-based methodology. *Int. J. Manag. Decis. Mak.* **13**(1), 17–41 (2014)
8. Benavides, D., Cortés, A.R., Trinidad, P., Segura, S.: A survey on the automated analyses of feature models. In: *JISBD*, pp. 367–376 (2006)
9. Bendjenna, H., Charrel, P., Zarour, N.: Using AHP method to resolve conflicts between non-functional concerns. In: *International Conference on Education, Applied Sciences and Management (ICEASM 2012)*, pp. 26–27 (2012)
10. Boehm, B., Bose, P., Horowitz, E., Lee, M.J.: Software requirements negotiation and renegotiation aids: a theory-w based spiral approach. In: 1995 17th International Conference on Software Engineering, p. 243. IEEE (1995)
11. Boehm, B., In, H.: Identifying quality-requirement conflicts. *IEEE Softw.* **13**(2), 25–35 (1996)
12. Boehm, B., Port, D., Al-Said, M.: Avoiding the software model-clash spiderweb. *Computer* **33**(11), 120–122 (2000)
13. Brito, I.S., Moreira, A., Ribeiro, R.A., Araújo, J.: Handling conflicts in aspect-oriented requirements engineering. In: Moreira, A., Chitchyan, R., Araújo, J., Rashid, A. (eds.) *Aspect-Oriented Requirements Engineering*, pp. 225–241. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-38640-4\\_12](https://doi.org/10.1007/978-3-642-38640-4_12)
14. Butt, W.H., Amjad, S., Azam, F.: Requirement conflicts resolution: using requirement filtering and analysis. In: Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B.O. (eds.) *ICCSA 2011*. LNCS, vol. 6786, pp. 383–397. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-21934-4\\_31](https://doi.org/10.1007/978-3-642-21934-4_31)
15. Capra, L., Emmerich, W., Mascolo, C.: A micro-economic approach to conflict resolution in mobile computing. *ACM SIGSOFT Softw. Eng. Notes* **27**(6), 31–40 (2002)
16. Coolican, H.: *Research Methods and Statistics in Psychology*. Psychology Press (2017)



17. Damian, D., Izquierdo, L., Singer, J., Kwan, I.: Awareness in the wild: why communication breakdowns occur. In: International Conference on Global Software Engineering (ICGSE 2007), pp. 81–90. IEEE (2007)
18. De Souto, M.C., De Araujo, D.S., Costa, I.G., Soares, R.G., Ludermir, T.B., Schliep, A.: Comparative study on normalization procedures for cluster analysis of gene expression datasets. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 2792–2798. IEEE (2008)
19. Derntl, M., Renzel, D., Nicolaescu, P., Koren, I., Klamma, R.: Distributed software engineering in collaborative research projects. In: 2015 IEEE 10th International Conference on Global Software Engineering, pp. 105–109. IEEE (2015)
20. Deutsch, M.: *The Resolution of Conflict: Constructive and Destructive Processes*. Yale University Press (1973)
21. van Dijk, R.: Determining the suitability of agile methods for a software project. In: 15th Twente Student Conference on IT (2011)
22. Easterbrook, S.: Handling conflict between domain descriptions with computer-supported negotiation. *Knowl. Acquis.* **3**(3), 255–289 (1991)
23. Easterbrook, S.: Resolving requirements conflicts with computer-supported negotiation. *Requirements Eng. Soc. Tech. Issues* **1**, 41–65 (1994)
24. Egyed, A.: Fixing inconsistencies in UML design models. In: 29th International Conference on Software Engineering (ICSE 2007), pp. 292–301. IEEE (2007)
25. Egyed, A., Grunbacher, P.: Identifying requirements conflicts and cooperation: how quality attributes and automated traceability can help. *IEEE Softw.* **21**(6), 50–58 (2004)
26. Faber, V.: Clustering and the continuous k-means algorithm. *Los Alamos Sci.* **22**(138144.21), 67 (1994)
27. Finkelstein, A., Kramer, J., Nuseibeh, B., Finkelstein, L., Goedicke, M.: Viewpoints: a framework for integrating multiple perspectives in system development. *Int. J. Softw. Eng. Knowl. Eng.* **2**(01), 31–57 (1992)
28. Freeman, R.E.: *Strategic Management: A Stakeholder Approach*. Cambridge University Press (2010)
29. Gambo, I., Adjicheboutou, A., Ikono, R., Iroju, O., Yange, S.: An investigative process model for predicting information diffusion on social media: Information system perspective. *Ife J. Technol.* **27**(1), 47–59 (2020)
30. Gambo, I., Ikono, R., Achimugu, P., Soriyan, A.: An integrated framework for prioritizing software specifications in requirements engineering. *Requir. Eng.* **12**(1), 33–46 (2018)
31. Gambo, I.P., Ikono, R., Iroju, O.G., Omodunbi, T.O., Zohoun, O.K.: Hybridized ranking model for prioritizing functional software requirements. *Int. J. Softw. Innov.* **9**(4), 1–31 (2021). <https://doi.org/10.4018/IJSI.289167>
32. Gambo, I., Taveter, K.: Identifying and resolving conflicts in requirements by stakeholders: a clustering approach. In: ENASE, pp. 158–169 (2021)
33. Gambo, I., Taveter, K.: A pragmatic view on resolving conflicts in goal-oriented requirements engineering for socio-technical systems. In: Proceedings of the 16th International Conference on Software Technologies, pp. 333–341 (2021). <https://doi.org/10.5220/0010605703330341>
34. Gobeli, D.H., Koenig, H.F., Bechinger, I.: Managing conflict in software development teams: a multilevel analysis. *J. Product Innov. Manag.* **15**(5), 423–435 (1998)
35. Grünbacher, P., Seyff, N.: Requirements negotiation. In: Aurum, A., Wohlin, C. (eds.) *Engineering and Managing Software Requirements*, pp. 143–162. Springer, Heidelberg (2005). [https://doi.org/10.1007/3-540-28244-0\\_7](https://doi.org/10.1007/3-540-28244-0_7)
36. Hadar, I., Zamansky, A., Berry, D.M.: The inconsistency between theory and practice in managing inconsistency in requirements engineering. *Empir. Softw. Eng.* **24**(6), 3972–4005 (2019). <https://doi.org/10.1007/s10664-019-09718-5>

37. Haraty, R.A., Dimishkieh, M., Masud, M.: An enhanced k-means clustering algorithm for pattern discovery in healthcare data. *Int. J. Distrib. Sens. Netw.* **11**(6), 615740 (2015)
38. Hassan, T., Hussain, M.Z., Hasan, M.Z., Ullah, Z., Qamar, N.: Quantitative based mechanism for resolving goals conflicts in goal oriented requirement engineering. In: Bajwa, I.S., Kamareddine, F., Costa, A. (eds.) *INTAP 2018. CCIS*, vol. 932, pp. 822–831. Springer, Singapore (2019). [https://doi.org/10.1007/978-981-13-6052-7\\_71](https://doi.org/10.1007/978-981-13-6052-7_71)
39. Hassine, J., Amyot, D.: An empirical approach toward the resolution of conflicts in goal-oriented models. *Softw. Syst. Model.* **16**(1), 279–306 (2015). <https://doi.org/10.1007/s10270-015-0460-6>
40. Hennig, C., Meila, M., Murtagh, F., Rocci, R.: *Handbook of Cluster Analysis*. CRC Press (2015)
41. In, H., Olson, D., Rodgers, T.: A requirements negotiation model based on multi-criteria analysis. In: *Proceedings Fifth IEEE International Symposium on Requirements Engineering*, pp. 312–313. IEEE (2001)
42. In, H., Roy, S.: Visualization issues for software requirements negotiation. In: *25th Annual International Computer Software and Applications Conference. COMPSAC 2001*, pp. 10–15. IEEE (2001)
43. Jureta, I.J., Faulkner, S., Schobbens, P.Y.: Clear justification of modeling decisions for goal-oriented requirements engineering. *Requirements Eng.* **13**(2), 87 (2008)
44. Keeney, S., McKenna, H., Hasson, F.: *The Delphi Technique in Nursing and Health Research*. Wiley, Hoboken (2011)
45. Kendall, M.G., Smith, B.B.: The problem of m rankings. *Ann. Math. Stat.* **10**(3), 275–287 (1939)
46. Khaled, O.M., Hosny, H.M., Shalan, M.: Exploiting requirements engineering to resolve conflicts in pervasive computing systems. In: Damiani, E., Spanoudakis, G., Maciaszek, L. (eds.) *ENASE 2017. CCIS*, vol. 866, pp. 93–115. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-94135-6\\_5](https://doi.org/10.1007/978-3-319-94135-6_5)
47. Kim, M., Park, S., Sugumaran, V., Yang, H.: Managing requirements conflicts in software product lines: a goal and scenario based approach. *Data Knowl. Eng.* **61**(3), 417–432 (2007)
48. Lim, W.C.: Effects of reuse on quality, productivity, and economics. *IEEE Softw.* **11**(5), 23–30 (1994)
49. Litvak, C., Antonelli, L., Rossi, G., Gigante, N.: Improving the identification of conflicts in collaborative requirements engineering. In: *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 872–877. IEEE (2018)
50. Lutz, R., Schäfer, S., Diehl, S.: Using mobile devices for collaborative requirements engineering. In: *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*, pp. 298–301 (2012)
51. Maalej, W., Thurimella, A.K.: Towards a research agenda for recommendation systems in requirements engineering. In: *2009 Second International Workshop on Managing Requirements Knowledge*, pp. 32–39. IEEE (2009)
52. Matthesen, S.B., Aasen, E., Holst, G., Wie, K., Einarsen, S.: The escalation of conflict: a case study of bullying at work. *Int. J. Manag. Decis. Mak.* **4**(1), 96–112 (2003)
53. Maxwell, J.C., Antón, A.I., Swire, P.: A legal cross-references taxonomy for identifying conflicting software requirements. In: *2011 IEEE 19th International Requirements Engineering Conference*, pp. 197–206. IEEE (2011)
54. Mistrik, I., Grundy, J., Van der Hoek, A., Whitehead, J.: Collaborative software engineering: challenges and prospects. In: Mistrik, I., Grundy, J., Hoek, A., Whitehead, J. (eds.) *Collaborative Software Engineering*, pp. 389–403. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-10294-3\\_19](https://doi.org/10.1007/978-3-642-10294-3_19)

55. Murukannaiah, P.K., Kalia, A.K., Telangy, P.R., Singh, M.P.: Resolving goal conflicts via argumentation-based analysis of competing hypotheses. In: 2015 IEEE 23rd International Requirements Engineering Conference (RE), pp. 156–165. IEEE (2015)
56. Nentwich, C., Emmerich, W., Finkelstein, A., Ellmer, E.: Flexible consistency checking. *ACM Trans. Softw. Eng. Methodol. (TOSEM)* **12**(1), 28–63 (2003)
57. Nuseibeh, B., Easterbrook, S.: Requirements engineering: a roadmap. In: Proceedings of the Conference on the Future of Software Engineering, pp. 35–46 (2000)
58. Version One: 9th annual state of agile survey. Survey (2015). Accessed Online 15
59. Poort, E.R., de With, P.: Resolving requirement conflicts through non-functional decomposition. In: Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), pp. 145–154. IEEE (2004)
60. Punj, G., Stewart, D.W.: Cluster analysis in marketing research: review and suggestions for application. *J. Mark. Res.* **20**(2), 134–148 (1983)
61. Robbins, S.: *Organisational Behaviour, Concepts, Controversies and Applications*, p. 599. Prentice Hall Inc., New York (1989)
62. Robillard, P.N., Robillard, M.P.: Types of collaborative work in software engineering. *J. Syst. Softw.* **53**(3), 219–224 (2000)
63. Robinson, W.N., Pawlowski, S.D., Volkov, V.: Requirements interaction management. *ACM Comput. Surv. (CSUR)* **35**(2), 132–190 (2003)
64. Rodrigues, O., Garcez, A.A., Russo, A.: Reasoning about requirements evolution using clustered belief revision. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI), vol. 3171, pp. 41–51. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-28645-5\\_5](https://doi.org/10.1007/978-3-540-28645-5_5)
65. Rombach, H.D., Basili, V.R., Selby, R.W.: *Experimental Software Engineering Issues: Critical Assessment and Future Directions*. International Workshop, Dagstuhl Castle, Germany, 14–18 September 1992. Proceedings, vol. 706. Springer, Heidelberg (1993)
66. Ross, S., Fang, L., Hipel, K.W.: A case-based reasoning system for conflict resolution: design and implementation. *Eng. Appl. Artif. Intell.* **15**(3–4), 369–383 (2002)
67. Saaty, T.L.: The analytic hierarchy process in conflict management. *Int. J. Conflict Manag.* (1990)
68. Sadana, V., Liu, X.F.: Analysis of conflicts among non-functional requirements using integrated analysis of functional and non-functional requirements. In: 31st Annual International Computer Software and Applications Conference (COMPSAC 2007), vol. 1, pp. 215–218. IEEE (2007)
69. Sardinha, A., Chitchyan, R., Weston, N., Greenwood, P., Rashid, A.: EA-analyzer: automating conflict detection in aspect-oriented requirements. In: 2009 IEEE/ACM International Conference on Automated Software Engineering, pp. 530–534. IEEE (2009)
70. Siegel, S.: *Nonparametric statistics for the behavioral sciences* (1956)
71. Sterling, L., Taveter, K.: *The Art of Agent-Oriented Modeling*. MIT Press, Cambridge (2009)
72. Strauss, A.L.: *Negotiations: Varieties, Contexts, Processes, and Social Order*, vol. 1. Jossey-Bass San Francisco (1978)
73. Tan, P.N., Steinbach, M., Kumar, V.: *Data mining cluster analysis: basic concepts and algorithms*. In: *Introduction to Data Mining*, pp. 487–533 (2013)
74. Taveter, K., Sterling, L., Pedell, S., Burrows, R., Taveter, E.M.: A method for eliciting and representing emotional requirements: two case studies in e-healthcare. In: 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), pp. 100–105. IEEE (2019)
75. Taylor, P.S., Greer, D., Coleman, G., McDaid, K., Keenan, F.: Preparing small software companies for tailored agile method adoption: minimally intrusive risk assessment. *Softw. Process Improv. Pract.* **13**(5), 421–437 (2008)
76. Tenso, T., Norta, A., Vorontsova, I.: Evaluating a novel agile requirements engineering method: a case study. In: ENASE, pp. 156–163 (2016)

77. Tenso, T., Taveter, K.: Requirements engineering with agent-oriented models. In: ENASE, pp. 254–259 (2013)
78. Urbietta, M., Escalona, M.J., Robles Luna, E., Rossi, G.: Detecting conflicts and inconsistencies in web application requirements. In: Harth, A., Koch, N. (eds.) ICWE 2011. LNCS, vol. 7059, pp. 278–288. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-27997-3\\_27](https://doi.org/10.1007/978-3-642-27997-3_27)
79. Van Lamsweerde, A., Darimont, R., Letier, E.: Managing conflicts in goal-driven requirements engineering. *IEEE Trans. Softw.Eng.* **24**(11), 908–926 (1998)
80. Veerappa, V., Letier, E.: Clustering stakeholders for requirements decision making. In: Berry, D., Franch, X. (eds.) REFSQ 2011. LNCS, vol. 6606, pp. 202–208. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-19858-8\\_20](https://doi.org/10.1007/978-3-642-19858-8_20)
81. Viana, T., Zisman, A., Bandara, A.K.: Identifying conflicting requirements in systems of systems. In: 2017 IEEE 25th International Requirements Engineering Conference (RE), pp. 436–441. IEEE (2017)
82. Vieira, F., Brito, I., Moreira, A.: Using multi-criteria analysis to handle conflicts during composition. In: Workshop on Early Aspects, 5th International Conference on Aspect-Oriented Software Development (AOSD 2006) (2006)
83. Vogelsang, A., Borg, M.: Requirements engineering for machine learning: perspectives from data scientists. In: 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), pp. 245–251. IEEE (2019)
84. Wang, X.T., Xiong, W.: An integrated linguistic-based group decision-making approach for quality function deployment. *Expert Syst. Appl.* **38**(12), 14428–14438 (2011)
85. Yang, D., Wu, D., Koolmanojwong, S., Brown, A.W., Boehm, B.W.: WikiWinWin: a wiki based system for collaborative requirements negotiation. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), p. 24. IEEE (2008)
86. Yang, H., Kim, M., Park, S., Sugumaran, V.: A process and tool support for managing activity and resource conflicts based on requirements classification. In: Montoyo, A., Muñoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 114–125. Springer, Heidelberg (2005). [https://doi.org/10.1007/11428817\\_11](https://doi.org/10.1007/11428817_11)
87. Yen, J., Tiao, W.A.: A systematic tradeoff analysis for conflicting imprecise requirements. In: Proceedings of ISRE 1997: 3rd IEEE International Symposium on Requirements Engineering, pp. 87–96. IEEE (1997)
88. Yin, R.K., et al.: Case study research and applications: design and methods (2018)
89. Zhang, W., Mei, H., Zhao, H.: Feature-driven requirement dependency analysis and high-level software design. *Requirements Eng.* **11**(3), 205–220 (2006)
90. Iqbal, T., Khan, M., Taveter, K., Seyff, N.: Mining reddit as a new source for software requirements. In: 2021 IEEE 29th International Requirements Engineering Conference (RE), pp. 128–138. IEEE, 20 September 2021