Group Decision Support for Requirements Management Processes

R. Samer and **M. Atas** and **A. Felfernig** and **M. Stettinger**
Graz University of Technology, Graz, Austria
email: {rsamer, muesluem.atas, alexander.felfernig, martin.stettinger}@ist.tugraz.at

A. Falkner and **G. Schenner**
Siemens AG, Vienna, Austria
email: {andreas.a.falkner, gottfried.schenner}@siemens.com

Abstract. Requests for proposal (RFP) trigger company-internal requirements management (RM) processes in order to assure that offers comply with a given set of customer requirements. As traditional RM approaches require a deep involvement of the requirements managers of a RM project especially when it comes to assigning suitable stakeholders to requirements, the quality of the decisions and the time effort for making correct decisions mainly depends on these experts. In this paper, we present a novel stakeholder assignment approach that reduces the overall involvement of these experts and also limits the uncertainty of overseeing suitable stakeholders at the same time. The assignment of responsible stakeholders is represented as a *group decision task* expressed in the form of a basic configuration problem. The outcome of such a task is a configuration which is represented in terms of an assignment of responsible stakeholders to corresponding requirements.

1 Introduction

Group-based configuration is an important application area of Artificial Intelligence [3, 4]. It aims to support a group of users in the configuration of complex products or services. In general, when interacting with group-based configurators, group members first articulate their preferences, then adapt inconsistent constraints, and finally, solutions are generated (i.e, reflecting the given configuration). In particular, when interacting with a configurator in the context of a typical requirements engineering task, each group member (i.e., stakeholder) has to evaluate each requirement according to different dimensions such as *priority, effort,* and *taken risk*. However, for the definition and evaluation of these requirements, first, suitable stakeholders have to be identified who are responsible for the development of these requirements. In addition, an early involvement of these stakeholders in the project is essential for the success of a project [5, 6, 13, 18]. This is because a low involvement of stakeholders in a project can lead to project failure. Project failures are often caused by missing or wrong assignments of stakeholders to requirements in early phases of the requirements engineering process [14]. Stakeholder recommendations can help to identify persons who are capable of providing a complete analysis and description of software requirements. Recommended stakeholders also need to bring deep knowledge about the corresponding item domain in order to provide precise evaluations of the requirements.

StakeNet [14] is an application that supports stakeholder identification on the basis of *social network analysis*. This approach builds a social network on the basis of a set of stakeholders. In this social network, stakeholders are represented by *nodes* and recommendations articulated by the stakeholders are represented by *links*. On the basis of such social networks, different *social network measures* are used for the prioritization of the stakeholders. One example of such a measure is *betweenness centrality* which measures the priority of a certain stakeholder s based on the ability of this user to play a role as a *broker* between separate groups of stakeholders. Castro-Herrera et al. [1] and Mobasher et al. [17] introduce a content-based recommendation approach where requirements are grouped by using different clustering techniques. Subsequently, stakeholders are recommended and assigned to these groups on the basis of content-based filtering. In this paper, a novel stakeholder assignment approach is introduced. The presented approach, acting as basic configuration service, lets voters evaluate stakeholders based on different criteria/dimensions and then aggregates their votes to derive possible configurations which are then recommended for the final stakeholder assignment decision to the requirements manager. In contrast to the aforementioned stakeholder recommendation approaches where the generated recommendations are directly suggested to the requirements manager, the content-based recommendation service presented in this paper *only* acts as a single *artificial voter* in addition to some human voters. Hence, the stakeholder recommendations (i.e., possible configurations) shown to the requirements manager are determined based on a combination of votes reflecting opinions of human voters as well as votes reflecting opinions of artificial voters.

The major contributions of this paper are the following. First, we analyze in detail a real-world scenario of a typical bid project. Second, we show an approach to identify relevant stakeholders for specific requirements and thus generate a global assignment of stakeholders to requirements. The remainder of this paper is organized as follows. In Section 2, we describe a typical application scenario of a bid project applied in an industrial context and provide a practical view of a traditional requirements management process commonly used for planning large industry projects. Additionally, a novel sophisticated approach is explained which further improves and extends the traditional approach by considering group decision support techniques. Section 3 discusses some potential issues and several factors this approach depends on. Subsequently, Section 4 explains the implementation of such an approach from a technical viewpoint. Finally, Section 5 concludes with a brief recap of this paper and presents some ideas for future work.

2 Application Scenario

Whenever an organizational unit of a large company (e.g., Siemens) decides to bid for a *Request for Proposal* (RFP), a new bid project for that proposal is initiated and the necessary stakeholders of the bid project are identified. RFPs for technical systems usually consist of a set of PDF or Microsoft Word documents which describe all requirements for the requested system covering technical, financial, legal, etc. aspects. Examples of stakeholders can be project managers, system architects, requirements managers, quality management departments, legal departments, engineering departments relevant for the bid, and potential external suppliers.

Within the context of a bid project, a requirements management (RM) process is initiated at the beginning. The purpose of this process is to assure that no requirement of the RFP has been overlooked. It involves the extraction of all the requirements contained in the RFP documents. The identified requirements must be assessed by the relevant stakeholders. This means that requirements concerning contracts must be assessed by the stakeholder(s) of the legal department, technical requirements must be assessed by the affected engineering department, etc. The assessment may involve statements about various criteria such as compliance, risks, approaches, etc. These statements are interpreted as *evaluation dimensions* in the remainder of this paper. At the end, each requirement of the RFP must have been assessed by at least one appropriate stakeholder.

2.1 Traditional RM Process

The *traditional requirements management* process can be best explained with an example. In the following, we describe a simplified example of a traditional RM process in a rail automation context based on a conventional RM tool such as *IBM DOORS*.

At the beginning, the requirements manager of the bid project creates a new project in the RM tool. After that, the necessary stakeholders for the current bid project are defined. In this context, stakeholders do not necessarily correspond to persons but correspond to roles which are uniquely identified with a unique string (called *Domain*). These string-based identifiers are unique within the organization. Furthermore, the RM tool supports the mapping of existing roles (i.e., domain identifiers) to concrete persons within the bid project. This way, responsible persons are assigned to roles based on their skills and domain knowledge.

Table 1 presents some examples of domain identifiers which occur in the context of rail automation. For such large bid projects usually more than 50 different domains are defined with the RM tool. However, in practice, most projects only use 20 different domains on average.

As a next step, the requirements manager imports all the relevant documents of the RFP into the project by using the RM tool. The RM tool automatically converts each paragraph of the documents into a (potential) requirement whilst the structure of the documents is preserved. The requirement manager then classifies the (potential) requirements in the project as either an actual requirement or as an arbitrary comment (called prose). In general, large infrastructure projects may contain more than 10,000 (potential) requirements.

Domain	Stakeholder
PM	project manager
SA	system architect
RM	requirements manager
RAMS	reliability, availability, maintainability, and safety
S(ignal)	engineering department for railway signals
PS	engineering department for power supply
TVD	department for track vacancy detection
ETCS	department for European Train Control System
Test	quality management department
Supplier1	external supplier, subcontractor

Table 1: Examples of domain identifiers for rail automation

Each (actual) requirement must be assessed by at least one stakeholder. The requirements manager has to figure out which stakeholders are appropriate for which requirements and needs to assign them accordingly. However, other stakeholders may improve such initial assignments later during the assessment phase. The RM tool notifies all assigned stakeholders via e-mail to assess the requirements they are assigned to.

Table 2 shows an example of an initial assignment done by the requirements manager (RM). In this table, each row corresponds to a requirement and each column refers to a stakeholder. Each cell represents a single decision (of a stakeholder) for a stakeholder assignment (to a requirement). At the beginning, only the RM proposes assignments of potential stakeholders to requirements based on the manager's expertise and knowledge. For example, the assignment of $\{S, PM\}$ to the requirement $R5$ in the RM column indicates that $R5$ has been initially assigned to the *signal department* (S) and to the *project management department* (PM) by the requirements manager (RM). As only the RM makes assignments in this initialization phase, the values of all other columns remain empty (i.e., are filled with the "-" label) until the assessment phase.

Req	RM	PM	RAMS	S(ignal)
R1	{PM}	-	-	-
R2	{PM}	-	-	-
R3	{S}	-	-	-
R4	{S}	-	-	-
R5	{S, PM}	-	-	-
...				

Table 2: Initial assignment of stakeholders to requirements done by requirements manager (RM). The dash symbol ("-") indicates that the other stakeholders have not made a decision yet.

Next, in the assessment phase, the affected stakeholders take a look at each of their assigned requirements in the RM tool and can either accept the requirement and assess it or they can veto the proposed assignment. Additionally, they can also propose an alternative stakeholder for the requirement or suggest (although rarely) an additional stakeholder for the requirement. For the remainder of this paper, this process is hereinafter referred to as *assignment feedback*. After that, the requirements manager can either accept the veto and assign the requirement to a different stakeholder or decline the veto and reassign the stakeholder to the requirement.

Table 3 shows an intermediate state during the assignment phase which demonstrates examples of *assignment feedback* given by the stakeholders PM and $S(ignal)$:

Req	RM	PM	RAMS	S(ignal)
R1	{PM}	{PM}	-	-
R2	{PM}	{RAMS}	-	-
R3	{S}	-	-	{S, RAMS}
R4	{S}	-	-	{}
R5	{S, PM}	{S, PM}	-	{S, PM}
...				

Table 3: State of assignment during assessment phase

- Requirement $R1$ has been accepted by PM
- Requirement $R2$ has been vetoed by PM and $RAMS$ has been proposed by PM as alternative stakeholder
- Requirement $R3$ has been accepted by $S(ignal)$, but $RAMS$ has been proposed by $S(ignal)$ as an additional stakeholder
- Requirement $R4$ has been vetoed by $S(ignal)$
- Requirement $R5$ has been accepted by all proposed stakeholders

It is important to point out the fact that in the traditional scenario, it is always the main responsibility of the requirements manager to resolve potential conflicts. Typically, this usually involves some personal discussions with the involved stakeholders and some final decisions made by the requirement manager. These final decisions then assure a consistent assignment of all requirements to responsible stakeholders. Table 4 presents such a final state where all conflicts have been resolved.

Req	RM	PM	RAMS	S(ignal)
R1	{PM}	{PM}	-	-
R2	{RAMS}	{RAMS}	{RAMS}	-
R3	{S, RAMS}	-	{S,RAMS}	{S, RAMS}
R4	{S}	{S}	-	{S}
R5	{S, PM}	{S, PM}	-	{S, PM}
...				

Table 4: Final state after assessment phase. Consistent assignment of stakeholders to requirements.

The requirements manager periodically reminds the assigned stakeholders about their unassessed requirements. This process is repeated until all requirements have been assessed and the assessment phase is finished. Thus, the assignment of stakeholders can be considered as a manual configuration process. The outcome of this process is a configuration in terms of a consistent assignment of stakeholders to requirements they are responsible for. In our current implementation, the overall goal is to achieve *consensus* regarding the stakeholder assignment. Future versions of our system include further constraints that have to be taken into account in task allocation tasks as discussed in this paper.

2.2 RM Process with Group Decision Support

The main idea of our novel requirements management approach is to introduce additional stakeholder votes made by *artificial stakeholders* (called *bots*). Additionally, the bots automatically propose stakeholders in the initial phase of the RM process. Furthermore, an intelligent *group decision service* is included in the RM tool to automatically aggregate all votes given by human stakeholders as well as artificial stakeholders. On a technical level, such a *group decision service* represents a *group recommender system* which generates recommendations based on aggregated votes given by group members of a group (i.e., the stakeholders) [2]. Basically, there exist different strategies on how to aggregate votes of group members [8] such as *majority*, *average*, *least-misery*, etc. In addition, more sophisticated aggregation functions exist - for further information regarding preference aggregation functions we refer to [2, 15]. To limit the scope of this paper, we assume that the group decision service is a simple group recommender using basic aggregation strategies.

The votes of the artificial stakeholders (i.e., bots) are generated by using appropriate content-based recommendation algorithms (see Section 4). This way, the group decision service allows to replace the traditional mainly manual stakeholder assignment process (see Section 2.1) with a semi-automatic process. As a key difference to the traditional approach, the group decision service automatically aggregates the decisions of all voters and thereby allows the smart incorporation of additional (automatic) voters, i.e., intelligent recommendation services for stakeholder assignments. From an abstract point of view, the process can be interpreted as a basic configuration process. Like in the traditional RM process (see Section 2.1), the outcome of this process represents a consistent assignment of stakeholders to requirements they are responsible for.

Table 5 illustrates a possible initial state in the presence of a group decision service (GDS) and a stakeholder assignment recommendation service (denoted as $RS1$). In sharp contrast to the assignments made by other stakeholders, the recommendation service does not provide a binary decision for every stakeholder but a confidence value which lies in the range between 1 and 10, whereby a higher number corresponds to more confidence and a lower number corresponds to a lower level of confidence.

The column for the GDS shows the result of the group decision service for each requirement, i.e., the aggregated decision of all voters (including humans and bots/algorithms). Note that a clear benefit of the group decision service is that some requirements can already be assessed by the assigned stakeholders, even though they have not yet been proposed/assigned by the requirements manager. In other words, stakeholders are automatically proposed by the bots/algorithms based on their skills in the *initial phase* and can already evaluate their assignment to the requirements. Hence, much assignment effort is taken away in the initial phase from the time-pressured requirements managers and the initial phase can be significantly speeded up. Moreover, it is necessary to point out that the stakeholders GDS (perform aggregation) and RM (perform final decision) can be considered to have a special role in this evaluation process, whereas all other stakeholders only occur as voters in the process. Consequently, the major responsibility/task of a RM in this process is to review the decision suggested by the GDS and to perform the final decision about the assignment of the stakeholders to the requirements.

3 Potential Issues of Group Decision Support

The exact behavior of the new system presented in Section 2.2 will depend on various factors. Examples of such factors include the *aggregation strategy* used by the group decision service to aggregate the votes (e.g., majority, average, etc.), the individual weight of the voters (e.g., "deciders"/experts count higher than normal stakeholders), and the confidence/trust users have in different recommendation algorithms.

Furthermore, the question arises how conflicting decisions (for example, stakeholder A assigns stakeholder B and B assigns A) can be resolved or supportive advice to manually resolve such conflicts can be given to the voters by the system. Also, inconsistencies and contradictions may occur in the evaluation of stakeholders between the voters. These voters can be other stakeholders and *artificial stakeholders*. In particular, for artificial stakeholders textual explanations

Req	GDS	RS1	RM	PM	RAMS	S(ignal)
R1	{PM}	{PM:9}	{PM}	-	-	-
R2	{RAMS}	{RAMS:8, PM:5}	-	-	-	-
R3	{S}	{S:8, RAMS:6}	-	-	-	-
R4	{S}	{S:5}	-	-	-	-
R5	{S}	{S:6}	{S,PM}	-	-	-
...						

Table 5: State of assignment with group decision service (GDS) and stakeholder recommendation service (RS1). The recommendation service provides a confidence value which lies in the range between 1 and 10.

can be presented to the group of voters being in conflict. Such textual explanations can then express the concrete reason and arguments for the votes provided by the artificial stakeholders.

Moreover, the prediction quality (i.e., performance) of the *artificial stakeholders* (i.e., the recommender systems) plays a major role in the process. In particular, the generated recommendations should be evaluated and examined with respect to completeness. In terms of common information retrieval measures (such as precision and recall), this would, for example, mean that more emphasis should be given to the recall of the results rather than the precision achieved by the recommender. In addition to that, an appropriate recommendation algorithm should also be capable of giving negative indication by telling the RM which stakeholders are definitely not suitable to be assigned to a requirement at all. Such a negative indication can be shown as, e.g., RAMS:0. Finally, another important aspect would be to take the availability of stakeholders into account before they get finally assigned to a requirement. This adds another complexity dimension to the underlying basic configuration problem.

4 Group Decision Support for Bidding Processes

In this section, a slightly modified version of the aforementioned RM process based on *Group Decision Support* (see Section 2.2) is described. The description explains the technical implementation of this process provided by the requirements engineering platform OPENREQ MVP[1] which is developed within the scope of the OpenReq EU Horizon 2020 research project. At the current stage, the implementation is already in use, however, still ongoing and ready to be further enriched with additional features. The remainder of this section describes the current status of the existing implementation.

In the initial phase, the requirements manager (RM) is asked by the system to propose suitable stakeholders for each requirement. As already described in Section 2.2, a content-based recommender system (RS1) helps the RM to find stakeholders based on keywords extracted from former requirements those stakeholders have solved. Thereby, on an abstract level, the automated *stakeholder-recommendation* algorithm (of RS1) can be interpreted as a *text classification* task [7] where the recommendation algorithm exploits several *Natural Language Processing* [19, 20] techniques in order to correctly classify stakeholders suitable for a given requirement.

The algorithm automatically extracts relevant keywords from the title and description text of all former requirements which a stakeholder was assigned to, in order to build a user profile for the respective stakeholder. First, the title and description text is cleaned by removing special characters (such as ".", ",", ";", "#", etc.). Next, the text is split into tokens (which, basically, represent the words in the text) and *stop words* such as prepositions (e.g., "in", "on", "at", etc.) or articles (e.g., "the", "a", "an") are removed. After applying *Part-of-speech tagging*, tokens/words of classes (such as verbs, adjectives, or numbers) that are most probably irrelevant to be used as keywords are removed. Finally, the remaining tokens of each former requirement (which was assigned to the stakeholder) are merged together into a single user profile.

By applying the same procedure to new requirements, keywords for new requirements are extracted as well. Given the keywords of a new requirement and the user profiles of the individual stakeholders, a similarity between a new requirement and a stakeholder is calculated for every stakeholder provided that the stakeholder has been assigned to an (already completed) requirement in the past. Formula 1 shows the Dice coefficient formula [9] which is a variation of the Jaccard coefficient and used to compute the similarity between a stakeholder and a requirement. The similarity is measured by comparing the overlap of the keywords of the stakeholder's user profile (denoted as U_a) and the relevant keywords of the respective requirement (denoted as r_x) with the total number of keywords appearing in U_a as well as r_x.

$$sim(U_a, r_x) = \frac{2 * |keywords(U_a) \cap keywords(r_x)|}{|keywords(U_a)| + |keywords(r_x)|} \quad (1)$$

Stakeholders who are most similar to a given requirement are suggested by the content-based recommender to the RM. This way, the initial phase can be speeded up and the chance of overseeing suitable stakeholders for requirements at this early stage of the process, is decreased. In the next step, the OPENREQ MVP system shows a list of the initially assigned stakeholders for each requirement. Stakeholders who are assigned to a requirement can either accept or reject their assignment. In addition, the assignments of the stakeholders for the requirement can be evaluated by all stakeholders.

This evaluation of a stakeholder-assignment is done based on the criteria *Appropriateness* and *Availability* (see Figure 1). Both criteria are interpreted as *evaluation dimensions* and stakeholders are evaluated based on both dimensions. Furthermore, an assigned stakeholder can also propose the assignment of further stakeholders to the requirement. These newly assigned stakeholders can then be evaluated again. After a new vote has been given, the *group decision service* (GDS) is triggered to compute a utility value for the rated stakeholder. Formula 2 shows the calculation of the utility value of an evaluated stakeholder s, whereas D is the set containing both dimensions, i.e., $D = \{Appropriateness, Availability\}$.

$$utility(s, r) = \frac{\sum_{t \in T} \frac{\sum_{d \in D} eval(s,r,d,t) \cdot weight(d)}{\sum_{d \in D} weight(d)}}{|T|} \quad (2)$$

The formula describes the stakeholder s to be voted by other stakeholders, whereby T represents the set of stakeholders $t \in T$ who evaluated s. More formally expressed, T is a set which contains the

[1] OpenReq MVP: `http://openreq.ist.tugraz.at`

Figure 1: Evaluation of stakeholders in OPENREQ MVP. Each stakeholder-assignment is evaluated by two evaluation dimensions (appropriateness and availability). The utility value of an evaluated stakeholder is calculated by using Formula 2.

stakeholders (including s) who evaluated stakeholder s, i.e., $T \subseteq S$. Furthermore, the OPENREQ MVP platform allows the requirements manager to define different importance levels for both dimensions. In Formula 2, the importance of a dimension $d \in D$ is expressed by the function $weight(d)$. Moreover, $eval(s, r, d, t)$ refers to the dimension-specific rating given by stakeholder t for stakeholder s for the requirement r. Finally, the result of $utility(s, r)$ represents the aggregated utility of a stakeholder s for requirement r.

Once all assignments have been evaluated by a sufficient number of stakeholders, a stable state of the assignment utilities is achieved. The utility values are then used as main feedback source for the requirements manager to make the final decision about which stakeholder(s) should be assigned to the requirement.

5 Conclusion and Future Work

Conclusion. In this paper, we discussed common application scenarios of requirements engineering in the context of industry projects. These scenarios range from traditional requirements management processes where the assignment process of stakeholders is solely controlled by the requirements manager, to more sophisticated automated approaches where the involvement of the requirements manager is reduced to a minimum. The latter represents a basic configuration service which includes *artificial stakeholders* as additional voters and a *group decision support system* as a vote aggregation component in the evaluation of stakeholder assignments to requirements. On the basis of this scenario we showed how these two components can be applied in order to improve the requirements management process such that the overall effort and the chance of overseeing stakeholders suitable for requirements can be reduced for the time-pressured requirements managers.

Future Work. As bidding processes can be seen as repetitive processes, mechanisms which are capable of learning stakeholder weights and taking individual expertise levels of stakeholders into account can be considered as potential ideas regarding future work. Moreover, the set of existing evaluation dimensions can be further extended such that more fine-grained control is given to the evaluation process as well as to the *group decision service*. Additionally, the concept of *liquid democracy* can be integrated into the evaluation process [10]. This way, stakeholders who do not have sufficient knowledge concerning the details of a requirement can easily delegate their votes to more well-informed and experienced experts.

With respect to conflicting decisions (see Section 3), future work should also include mechanisms to automatically resolve such conflicts or mechanisms which provide supportive advice to the voters, showing how they can manually resolve such conflicts. Furthermore, the configuration approach can be enriched with further constraints taking resource management aspects of stakeholders into consideration, in order to optimize the overall allocation of human resources in release planning.

Finally, there is also still plenty of room for improvement regarding the extraction of keywords used by the discussed content-based recommender system (i.e., *artificial stakeholder*). For example, a more descriptive and characteristic representation of the keywords can be obtained by using more sophisticated content-based approaches such as *Latent Semantic Analysis* (LSA) [11, 16] or *word2vec* algorithms [12, 16].

Acknowledgment

The work presented in this paper has been conducted within the scope of the Horizon 2020 project OPENREQ (732463).

REFERENCES

[1] Carlos Castro-Herrera, Chuan Duan, Jane Cleland-Huang, and Bamshad Mobasher, 'Using data mining and recommender systems to facilitate large-scale, open, and inclusive requirements elicitation processes', 165–168, (09 2008).

[2] A. Felfernig, L. Boratto, M. Stettinger, and M. Tkalcic, *Group Recommender Systems – An Introduction*, Springer, 2018.

[3] Alexander Felfernig, M Atas, T Tran, and Martin Stettinger, 'Towards group-based configuration', in *International Workshop on Configuration 2016 (ConfWS16)*, pp. 69–72, (2016).

[4] Alexander Felfernig, Lothar Hotz, Claire Bagley, and Juha Tiihonen, *Knowledge-based Configuration: From Research to Business Cases*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1 edn., 2014.

[5] Alexander Felfernig, Martin Stettinger, Andreas Falkner, Muesluem Atas, Xavier Franch, and Christina Palomares, 'Openreq: Recommender systems in requirements engineering', pp. 1–4, (10 2017).

[6] Hubert F. Hofmann and Franz Lehner, 'Requirements engineering as a success factor in software projects', *IEEE software*, **18**(4), 58, (2001).

[7] Emmanouil Ikonomakis, Sotiris Kotsiantis, and V Tampakas, 'Text classification using machine learning techniques', **4**, 966–974, (08 2005).

[8] Anthony Jameson and Barry Smyth, 'The adaptive web', chapter Recommendation to Groups, 596–627, Springer-Verlag, Berlin, Heidelberg, (2007).

[9] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich, *Recommender Systems: An Introduction*, Cambridge University Press, New York, NY, USA, 1st edn., 2010.

[10] Anson Kahng, Simon Mackenzie, and Ariel D. Procaccia, 'Liquid democracy: An algorithmic perspective', in *AAAI*, (2018).

[11] Thomas K. Landauer, Peter W. Foltz, and Darrell Laham, 'An introduction to latent semantic analysis', (1998).

[12] Jey Han Lau and Timothy Baldwin, 'An empirical evaluation of doc2vec with practical insights into document embedding generation', *CoRR*, **abs/1607.05368**, (2016).

[13] Dean Leffingwell, 'Calculating your return on investment from more effective requirements management', *American Programmer*, **10**(4), 13–16, (1997).

[14] S.L. Lim, D. Quercia, and A. Finkelstein, 'Stakenet: Using social networks to analyse the stakeholders of large-scale software projects', in *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 1*, ICSE '10, pp. 295–304, New York, NY, USA, (2010). ACM.

[15] J. Masthoff, 'Group recommender systems', *Recommender Systems Handbook*, 677–702, (2011).

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean, 'Distributed representations of words and phrases and their compositionality', in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pp. 3111–3119, USA, (2013). Curran Associates Inc.

[17] B. Mobasher and J.Clehand-Huang, 'Recommender systems in requirement engineering', 81–89, (2011).

[18] Bamshad Mobasher and Jane Cleland-Huang, 'Recommender systems in requirements engineering', *AI magazine*, **32**(3), 81–89, (2011).

[19] Kevin Ryan, 'The role of natural language in requirements engineering', in *[1993] Proceedings of the IEEE International Symposium on Requirements Engineering*, pp. 240–242, (Jan 1993).

[20] J. Winkler and A. Vogelsang, 'Automatic classification of requirements based on convolutional neural networks', in *2016 IEEE 24th International Requirements Engineering Conference Workshops(REW)*, volume 00, pp. 39–45, (Sept. 2016).