

Automatic Software Requirements Classification: A Systematic Literature Review

Zayed, Mostafa Adel

Mostafa.Zayed@fci.helwan.edu.eg

Information Systems Dept., Faculty of Computers and Artificial Intelligence, Helwan University, Egypt

Abstract__ Context: Software requirements must be classified for later use of requirements in the design and implementation phases. This classification can be done manually which takes a lot of time and effort or recently it can be done automatically. Automatic requirement classification is an important and promising area in both industry and research.

Objective: The objective of this paper is to investigate the existing automatic classification techniques in order to help researchers and software engineers to choose the appropriate requirement classification technique.

Method: This research was conducted with three research questions and followed Systematic Literature Review (SLR) guidelines to analyze, evaluate, interpret the relevant researches.

Results: 17 primary studies were conducted from the search processes. 2 out of them were published in Elsevier, 3 were published in Springer, 2 were published in IEEE, and the rest of them were published in different conferences and journals

Conclusions: automatic classification of SW requirements is an important task; however, it has a little work in this field. This SLR summarize all existing techniques available in this topic. It needs more effort and research papers to enhance and increase the performance as well as the quality of proposed tools.

Keywords — Requirements Classification, Requirement Engineering, Automatic Classification, Functional Requirements, Non-Functional Requirements, Framework.

I. INTRODUCTION

SOFTWARE Development Life Cycle (SDLC) possess five stages (Analysis, Design, Implementation, Testing, and Maintenance) [1]. The analysis phase is concerned with requirement analysis also called requirement

engineering. Those Requirements determine user expectations for the new software. Requirement engineering is the cornerstone and the first crucial phase in SDLC [2]. It consists of four tasks (requirement election, requirement analysis, requirement specification, and requirement management) [3].

In requirement election, customer's needs and specifications are well understood. In the next requirement analysis task, those needs are investigated and redefined. Then requirement specification task which documents all user specification requirements in a clear and correct form. Finally, requirement management task that schedules and prioritizes the requirements. After requirement engineering phase, all customer needs and specifications are written in a Software Requirements Specification (SRS) document.

Requirements are classified into many types (e.g. UI requirements, market requirements), but the most important two types of requirements are functional and non-functional [4]. As, functional requirements are translated into functions or features to be developed in the software [5]. Where, non-functional requirements are the constrains that describe how the system work (e.g. performance, security) [4].

Usually, the task of requirements classification is done manually by the requirement engineers which is not an easy task and time consuming. The SRS document is written in natural language without formal structure or even formal language. The difficulty of automatically processing natural language is encompassed in many reasons [6]. The main reason is its ambiguity. The reason behind this ambiguity has many causes such as the diversity of human dialects, idioms, and the different ways of sentences formation. Also, expressing an idea or a concept is not done the

same way or in the same form for everyone which adds another difficulty. As well as, the stakeholders and engineers use different expressions that adds another difficulty to the classification task [7]. These characteristics of natural language give a high level of inconsistency and incompatibility for the SRS document. For these reasons, the automation of requirements classification is helpful and save a lot of time as well as money.

This paper introduces a detailed review of different scientific researches that handle the automatic classification of software requirements. The objective of this paper was achieved through SLR to introduce a balanced and useful summary of research studies.

This paper is organized as follows: Section II represents the relevant published SLRs. Section III introduces the review planning and methodology of the Systematic Literature Review (SLR). Section IV represents the results from the SLR. Finally, section V concludes the SLR.

II. REVIEW OF RELEVANT SLRS

In this section, three of relevant published SLRs are introduced. These three SLRs are in the same area of research but not concerned with the automation of software requirements classification. For the best of our knowledge, there is no SLRs published in the topic of automatic requirements classification. These SLRs were published from 2009 to 2014. Summarization of relevant SLRs is illustrated in Table I, Table II, Table III.

III. PLANNING AND DESIGNING THE SLR

In this section, the SLR protocol is described. Firstly, the need for this SLR is introduced. Secondly, the Research Questions (RQs) that this SLR follows are presented.

The Need for SLR

In this section, the need for the SLR is accurately defined and introduced. The main objective of this SLR is to discuss the different existing techniques for automatic requirement classification, defining the criteria used in comparing the different techniques, and to present the different used tools

for automatic classification, and finally analyzing the results from different techniques for automatic classification for requirements.

Research Questions

In this section, the RQs that will be answered in this SLR are presented as follows:

RQ1: what are the existing techniques used for the automatic requirement classification?

RQ2: what are the criteria used in comparing requirement classification techniques?

RQ3: what are the tools applied in software automatic requirement classification?

Source Selection

Many automatic searches have been done in order to find and choose the most appropriate research papers related to the mentioned RQs. The main online sources are:

- International Requirement Engineering Conference (RE).
- IEEE
- Springer
- Elsevier
- Other journals & conferences

Source Identification Strings of Primary Studies

In order to select search papers related to our SLR topic, search strings are defined as follows:

- (Requirements OR Software Requirements) AND (Classification OR Automatic Classification).
- (Automatic AND Requirements AND Classification) OR (Automatically AND Classify AND Requirements) OR (Automatically AND Classification AND Requirements)
- (Automation AND Requirements AND Classification) OR (Requirements AND Classification AND Automation)
- (Automated AND Requirements AND classification) OR (Automated AND Requirement AND Classify)

Inclusion and Exclusion Criteria

- Papers that have been included and taken into consideration in this research are

TABLE I: SLR1 SUMMARIZATION

Title	A systematic literature review of software requirements prioritization research [8].
Research Period	1996-2013
The Objective	The objective of this research was to identify and examine the status of software requirements prioritization techniques.
Research Questions	<ul style="list-style-type: none"> • RQ1: What are the existing techniques used for prioritizing requirements? • RQ2: What are the descriptions and limitations of existing prioritization techniques? • RQ3: What taxonomy of prioritization scales does each technique exhibit? • RQ4: What are the processes involved in software requirements prioritization?
Inclusion Criteria	<ul style="list-style-type: none"> • All papers published in English language • Papers that focuses on requirements prioritization. • Relevant papers that are published from 1996 to 2013 • All published papers that have the potential of answering at least, one research question.
Exclusion Criteria	<ul style="list-style-type: none"> • Papers that are not published in English language • Papers that do not have any link with the research questions • Gray papers; i.e. papers without bibliographic information such as publication date/type, volume and issue numbers were excluded • Duplicate papers (only the most complete, recent and improved one is included). The rest are excluded.
Quality Criteria	<ul style="list-style-type: none"> • Are the aims of the research clearly articulated? • Is the proposed technique clearly described? • Is the experimental design appropriate? • Is the experiment applied on adequate project data sets or case study? • Does the research add value to the academia or industrial community?
Results	It was discovered that, although a lot of prioritization techniques exist; improvements are still required. Some of these improvement borders on scalability, computational complexities, ease of use, reliability of results, validation of techniques in industrial settings and requirements evolvability and dependency issues.

TABLE II: SLR2 SUMMARIZATION

Title	A systematic literature review to identify and classify software requirement errors [9].
Research Period	No time frame limitation
The Objective	The objective of this research is to identify and classify types of requirement errors into a taxonomy to support the prevention and detection of errors.
Research Questions	What types of requirements errors can be identified from the literature and how can they be classified?
Inclusion Criteria	<ul style="list-style-type: none"> • Papers that focus on analyzing/using the errors (source of faults) for improving software quality • Empirical studies (qualitative or quantitative) of using the error information in software lifecycle • Papers that talk about errors, mistakes or problems in the software development process and requirements in particular • Papers about error, fault, or defect classifications • Empirical studies (qualitative or quantitative) on the cause of software development defects • Papers from human cognition and psychology about human thought process, planning, or problem solving • Empirical studies on human errors • Papers that survey or describe the human error classifications
Exclusion Criteria	<ul style="list-style-type: none"> • Papers that are based only on expert opinion • Short-papers, introductions to special issues, tutorials, and mini-tracks • Studies not related to any of the research questions • Preliminary conference versions of included journal papers • Studies not in English • Studies whose findings are unclear and ambiguous (i.e., results are not supported by any evidence).
Quality Criteria	<p><u>Quality assessment for experimental studies</u></p> <ul style="list-style-type: none"> • Does the evidence support the findings? • Was the analysis appropriate? • Does study identify or try to minimize biases and other threats? • Can this study be replicated? <p><u>Quality assessment for Observational studies</u></p> <ul style="list-style-type: none"> • Do the observations support the conclusions or arguments? • Are comparisons clear and valid? • Does study identify or try to minimize biases and other threats? • Can this study be replicated?
Results	This paper provides a new perspective for the investigation of software quality improvement. It outlines the software quality problem and describes the limitations of the existing practices in ensuring software quality.

TABLE III: SLR3 SUMMARIZATION

Title	Classification and Qualitative Analysis of Non-Functional Requirements Approaches [10].
Research Period	No time frame limitation
The Objective	The objective of this research was to document all approaches that deal with NFRs , discuss their scope and characteristics.
Research Questions	<ul style="list-style-type: none"> • RQ1. What are the documented approaches in NFRs engineering process? • RQ2. What are the scopes and characteristics of these NFRs approaches?
Inclusion Criteria	<ul style="list-style-type: none"> • The publications written only in English language were considered. • The abstracts explicitly in the notion of NFRs as a primary focus were considered.
Exclusion Criteria	<ul style="list-style-type: none"> • If there were more publications of the same Research Group on the same approach development then the most complete publication of the approach was considered. • The literatures those were not considered NFRs as the primary contribution.
Quality Criteria	Not clear
Results	The review shows that methods and techniques are available in all NFRs engineering process phases and the approaches are developed within various scopes and characteristics. Elicitation approaches are designed to elicit NFRs from goal-oriented dynamic and pattern-based static sources of requirements with aspect-oriented methods and tools. NFRs taxonomies are developed in pattern-based specification approaches, aspect-oriented specification approaches list and classify elicited NFRs where goal-oriented approaches

following the inclusion criteria as follows:

- All papers published in English language.
- Papers that focuses on our search strings (Requirements automatic classification).
- Relevant papers that are published from 2007 to 2020.
- All published papers that have the potential of answering at least, one research question.

For the exclusion criteria, any search paper irrelevant to search strings and disagree with the inclusion criteria is excluded from our search.

Quality Assessment Criteria

Paper that have been included and taken into consideration in this SLR should also follow the following quality assessment questions:

1. Are the aims of the research clearly defined?
2. Is the proposed technique is clearly described?
3. Is the used tool is mentioned clearly?
4. Are the evaluation criteria of the technique are mentioned?
5. Are the experiments applied on adequate

data sets?

6. Does the research add value to the community, academia, or industry?

Conducting the review

In this stage, different sources for our primary study are stated and established to conduct our SLR. Table IV illustrates the set of relevant papers that have been chosen. Fig.1 shows the distribution of papers over different websites.

TABLE IV
SELECTED PAPERS FROM DIFFERENT SOURCES

Selected Papers	Source
4	International Requirement Engineering (RE) Conference
2	IEEE
3	Springer
2	Elsevier
2	Different Conferences
4	Different Journals

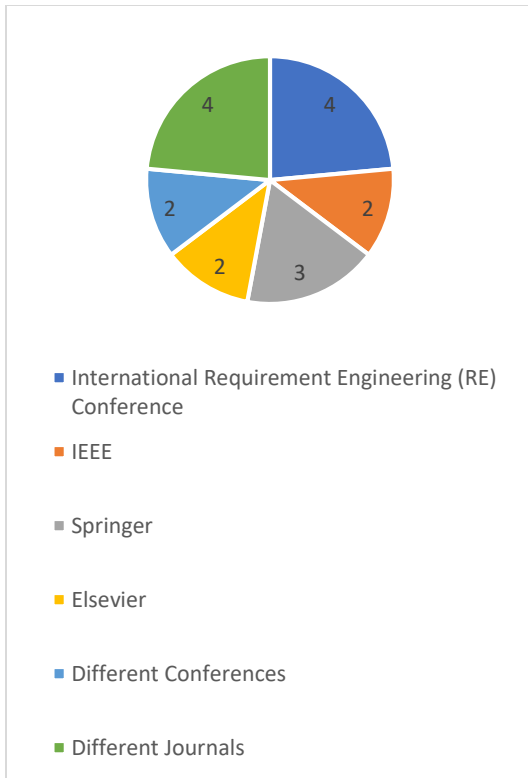


Fig.1 Selected Papers distribution

IV. RESULTS

In the initial stage of this SLR, a set of 17 research papers were selected to complete this review. Results are presented in two dimensions. Firstly, the answers for RQs are presented in details. Secondly, the results are illustrated in table V for summarizing the findings.

The results for RQ1 which is determining the existing techniques for automatic requirements classification resulted in finding two major techniques which are applied in the automation of classifying requirements. The first technique applied one of machine learning algorithms for classification like Support Vector Machine (SVM), Naïve Bays (NB), Multinomial Naïve Bays (MNB), Logist Regression (LR), J48 decision tree or K-Nearest Neighbors (K-NN). The second technique applied one of deep learning techniques convolutional Neural Network (CNN). In both two techniques, some Natural Language Processing (NLP) tasks must be applied to handle the textual nature of requirements.

The evaluation criteria of existing

techniques which answered RQ2 are recall, precision, f-measure, and accuracy. As recall is the ration between correctly predicted positive classified requirement to the all requirements calculated by this formula $recall = \frac{true\ positive}{true\ positive + false\ negative}$. While precision is the ration between correctly predicted positive classified requirement to the total predicted positive classified requirements calculated using this formula $precision = \frac{true\ positive}{true\ positive + false\ positive}$. F-measure score is the weighted average of precision and recall calculated as $F1\ score = \frac{2 * (recall * precision)}{recall + precision}$. Finally, accuracy is a ratio of correctly predicted requirements to the total requirements calculated using this formula $accuracy = \frac{true\ positive + true\ negative}{true\ positive + false\ positive + true\ negative + false\ negative}$.

The results for RQ3 which is stating the used tools for the automatic classification of requirements. Some researches mentioned the used tools where other researches didn't mention. The mentioned tool which was mentioned more than one time was Weka. While some researches build their own tools for example, LASR (Large Scale Annotation of Software Requirements) tool.

TABLE V
EXISTING TECHNIQUES

Research Papers	Existing technique
[11] , [12], [13]	NB Classifier
[14], [15], [16]	SVM Classifier
[17]	K-NN & SVM & NB Classifiers
[18], [19], [20]	NLP Tool
[21]	SVM & MNB Classifiers
[22], [23]	CNN Classifier
[24], [6]	SVM & NB Classifier
[25]	J48 decision tree Classifier
[26]	K-NN & SVM & NB & LR Classifiers

V. CONCLUSION

In this paper a SLR concerning automatic classification of software requirements is presented. The main contribution of this paper is to collect all the techniques which were applied for the requirements classification and the summarization of these techniques which help the other researchers as well as software engineers in both research and practical work. Additionally, the SLR summarizes all evaluation criteria of applied techniques in automatically classifying requirements. It also showed that some researchers built their own NLP tools to perform the requirements classification. On the other hand, some researchers used built-in tools (e.g. Weka). The number of available research papers in this research topic is still not enough and need more contributions.

Fig. 2 shows the importance of selected research papers by showing the number of every paper citations. Every paper is mentioned with its citation number in the SLR. Paper [24] has no citations, as it was recently published.

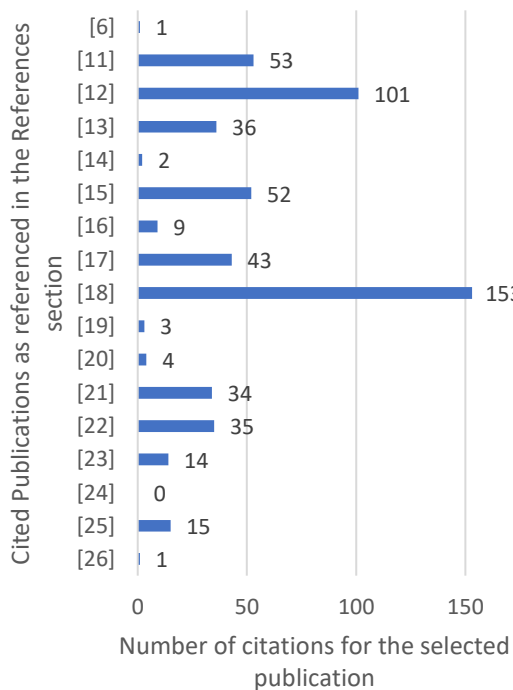


Fig. 2 Number citation for the selected publications

REFERENCES

- [1] Y. Bassil, "A Simulation Model for the Waterfall Software Development Life Cycle," *International Journal of Engineering & Technology (iJET)*, vol. 2, no. 5, pp. 2049-3444, 2012.
- [2] V. Rastogi, "Software Development Life Cycle Models-Comparison, Consequences," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 6, no. 1, pp. 168-172, 2015.
- [3] R. Saranya, "Survey on Security Measures of Software Requirement Engineering," *International Journal of Computer Applications*, vol. 90, no. 17, p. 0975 – 8887, 2014.
- [4] M. Glinz, "On Non-Functional Requirements," in *15th IEEE International Requirements Engineering Conference (RE'07)*, Delhi, India, 2007.
- [5] I. S. Boeard, "IEEE Standard Glossary of Software Engineering Terminology," in *The Institute of Electrical and Electronics Engineers*, New York, USA, 1990.
- [6] H. H. Mustafa and N. R. Darwish, "Automatic Requirement Classification Technique: Using Different Stemming Algorithms," *CiiT International Journal of Data Mining and Knowledge Engineering*, vol. 10, no. 6, p. 122:127, 2018.
- [7] Z. S. H. A. a. G. Ruhe, "Using real options to manage Technical Debt in Requirements Engineering," in *Requirements Engineering Conference (RE), IEEE 23rd International*, Ottawa, ON, Canada, 2015.
- [8] A. S. R. I. M. N. M. Philip Achimugu †, "A systematic literature review of software requirements prioritization research," *Information and Software Technology*, vol. 56, no. 6, p. 568–585, 2014.

- [9] J. C. C. Gursimran Singh Walia a, "A systematic literature review to identify and classify software requirement errors," *Information and Software Technology*, vol. 51, no. 7, p. 1087–1109, 2009.
- [10] M. M. Hasan, P. Loucopoulos and M. Nikolaidou, "Classification and Qualitative Analysis of Non-Functional Requirements Approaches," in *International Conference on Exploring Modeling Methods for Systems Analysis and Design*, Springer, Berlin, Heidelberg, 2014.
- [11] S. P. ., J. S. ., S. C. Youngjoong Ko, "Using classification techniques for informal requirements in the requirements analysis-supporting system," *Information and Software Technology - ELSEVIER*, vol. 49, pp. 1128-1140, 2007.
- [12] "Identification of non-functional requirements in textual specifications: A semi-supervised learning approach," *Information and Software Technology*, vol. 52, no. 4, pp. 436-445, 2010.
- [13] D. D. G. P.-C. a. J. C.-H. Eric Knauss, "Detecting and Classifying Patterns of Requirements Clarifications," in *20th IEEE International Requirements Engineering Conference (RE)*, Chicago, IL, USA, 2012.
- [14] D. O. a. F. Houdek, "Automatic Requirement Classification: Tackling Inconsistencies Between Requirements and Regulations," *International Journal of Semantic Computing*, vol. 8, no. 1, p. 47–65, 2014.
- [15] Z. Kurtanović and W. Maalej, "Automatically Classifying Functional and Non-functional Requirements Using Supervised Machine Learning," in *IEEE 25th International Requirements Engineering Conference (RE)*, Lisbon, Portugal, 2017.
- [16] E. Knauss and D. Ott, "(Semi-) automatic Categorization of Natural Language Requirements," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*, Springer, Cham, 2014.
- [17] M. Riaz, J. King, J. Slankas and L. Williams, "Hidden in plain sight: Automatically identifying security requirements from natural language artifacts," in *IEEE 22nd International Requirements Engineering Conference (RE)*, Karlskrona, Sweden, 2014.
- [18] J. S. R. Z. X. e. a. Cleland-Huang, "Automated classification of non-functional requirements," *Requirements Engineering*, vol. 12, no. 2, pp. 103-120, 2007.
- [19] "A Suggested Framework for Software Requirements Classification," in *17th UKSIM-AMSS International Conference on Modelling and Simulation*, Cambridge UK, 2015.
- [20] O. O. K. Ishrar Hussain, "LASR: A tool for large scale annotation of software requirements," in *Empirical Requirements Engineering (EmpiRE), 2012 IEEE Second International Workshop*, 2012.
- [21] D. Ott, "Automatic Requirement Categorization of Large Natural Language Specifications at Mercedes-Benz for Review Improvements," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*, Springer, Berlin, Heidelberg, 2013.
- [22] J. Winkler and A. Vogelsang, "Automatic Classification of Requirements Based on Convolutional Neural Networks," in *IEEE 24th International Requirements Engineering Conference Workshops (REW)*, Berlin, 2016.
- [23] R. I. N. Almanza, R. Juárez-Ramírez and G. Licea, "Towards Supporting Software Engineering Using Deep Learning: A Case of Software Requirements Classification," in *5th International Conference in Software Engineering Research and*

Innovation (CONISOFT), Mérida,
Yucatán, México, 2017.

- [24] N. D. R. Haidy H. Mustafa, "Automatic Requirement Classification: A Comparative Study," *CiiT International Journal of Data Mining and Knowledge Engineering*, vol. 11, no. 1, pp. 1-6, 2019.
- [25] R. M. A. J. Rajni Jindal, "Automated Classification of Security Requirements," in *Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Jaipur, India, 2016.
- [26] E. D. Canedo and B. C. Mendes, "Software Requirements Classification Using Machine Learning Algorithms," *Entropy*, vol. 22, no. 9, pp. 1-20, 2020.