

# Application of Natural Language Processing with GQM and AHP approaches for requirements quality assessment \*

Evgenii Timoshchuk  
Innopolis University  
Innopolis, Russia  
e.timoshchuk@innopolis.university

Sergey Kuznetsov  
Innopolis University  
Innopolis, Russia  
ser.kuznetsov@innopolis.university

ZiyoMukhammad Usmonov  
Innopolis University  
Innopolis, Russia  
z.usmonov@innopolis.university

Amartiwi Utih  
Innopolis University  
Innopolis, Russia  
u.amartiwi@innopolis.university

Atif Farah  
Innopolis University  
Innopolis, Russia  
f.atif@innopolis.university

Harrif Saliu  
Innopolis University  
Innopolis, Russia  
h.saliu@innopolis.university

## Abstract

The quality of requirements is difficult to measure in an automated way because of need in reviews and subjective opinion of stakeholders. Plenty of attributes can be used to evaluate requirements quality, but most of them have vague meaning and no concrete metrics for measurement. We proposed a model based on a goal-question-metric approach to identify the most important quality attributes and its metrics, which can be calculated in an automated way. Text of requirements can be analyzed by natural language processing techniques to reveal weak words and phrases, which make sentence subjective and ambiguous. We proposed metrics for such quality attributes as unambiguity, subjectivity, singularity, completeness, and calculated indexes based on the number of words and sentences for the read-ability attribute. Analytic hierarchy process for complex decisions was applied to convert calculated metrics of every requirement into overall quality evaluation of requirement document according to customer's priorities. Model was implemented in a prototype with focusing on adopting NLP techniques for Russian language and supporting external API.

## 1 Introduction

This work aims to combine the efforts of NLP [1], GQM [2] and AHP [3] approaches for assessing overall quality of requirement documents in automated way. Techniques to process the words from the document were applied that enable the system to carry out further analysis on the syntactic and semantic structure of the text. After processing, each requirement statement and the overall requirement are assigned to numeric values based on calculations carried out by the system to determine what areas of the requirement document need modification. The ultimate goal of this work was encapsulating the best of these techniques and methods for measurement requirement quality into a single model and provide a prototype of a tool for automated validation of real-world requirements against it.

---

\* Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

## 2 Quality assessment model

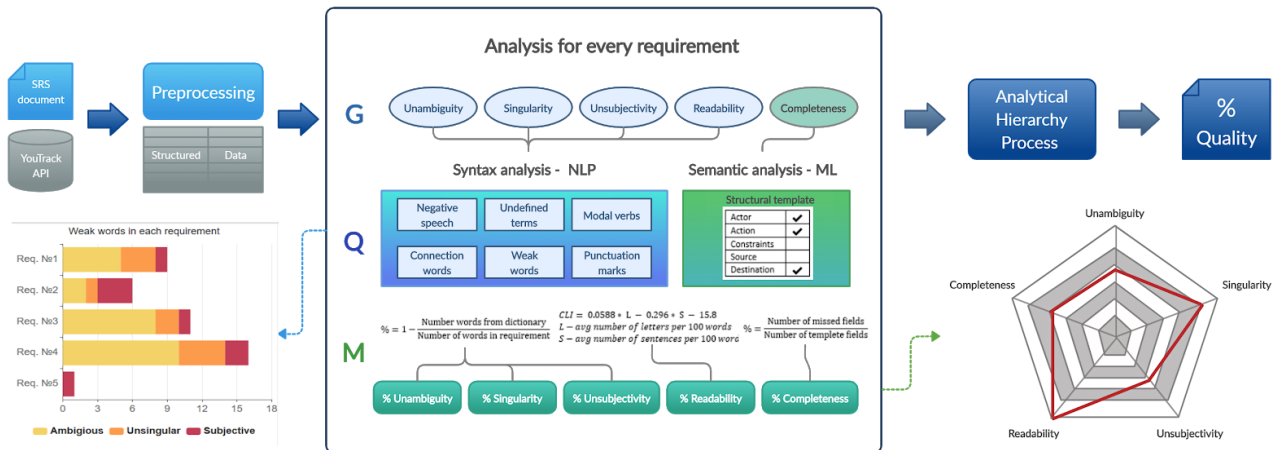


Figure 1:

### 2.1 Goal/Question/Metric approach

The Goal-Question-Metric method based on a system of questions and straightforward answers about properties evaluation [4]. This approach consists of three main steps: specifying goals, pointing relevant attributes, and providing measurements. GQM framework helped to define appropriate metrics and estimate the quality of requirements in our case. The goal should be defined for an object, with a purpose, from a perspective, in an environment. The overall goal of current the project is to measure quality of requirements and can be formulated by following template:

*Analyze requirement quality  
 for the purpose of improving  
 with respect to quality attributes  
 from the viewpoint of project managers  
 in the context of product development.*

In addition, we identified several sub-goals, which should be fulfilled to achieve the primary goal. For instance:

*Sub-goal:* Analyze requirement unambiguity for the purpose of improving with respect to quality attributes from the viewpoint of project managers in the context of product development.

*Question:* How many vague words and weak phrases make requirement ambiguous?

*Metric:* Number of ambiguous words in 1 requirement divided by an average number of words in 1 requirement.

### 2.2 Quality attributes and their metrics

Our model adopted the five core quality attributes to give final quality measurement for the whole requirement set evaluating by syntax and semantic analysis.

**Unambiguity.** It requires that only one semantic interpretation of the requirement exists. To evaluate the ambiguity of each requirement, we propose to use dictionaries with a set of words, which indicates ambiguity in the requirement [6][7]. As the metric for assessing ambiguity, we used the following formula:

$$Unambiguity \% = \left(1 - \frac{N_{ambig}}{N_{total}}\right) \times 100$$

Where  $N_{ambig}$  – the number of words in the requirement,  $N_{total}$  – the number of ambiguous words in the requirement.

**Singularity.** Statement of the requirement must relate to only one unique requirement that does not overlap with others. The presence of several modal words tells us that the requirement contains several meanings and that the statement does not have the characteristic of singularity. These words may include could, may, might, can, should, will, shall, must, would, etc. The number of connective words may also indicate the presence of several requirements within one (mentioned above). As the metric for assessing singularity, we used the following formula:

$$Singularity \% = \left(1 - \frac{(N_{modal} - 1) + N_{connective}}{N_{total}}\right) \times 100$$

where  $N_{total}$  – the number of words in the requirement,  $N_{modal}$  – the number of modal verbs which are not zero,  $N_{connective}$  – the number of connective words in the requirement.

**Readability.** This attribute indicates how easily requirement text can be read and understood, it can be based on the number of syllables per word and number of words per sentence. It can be calculated by Flesch-Kincaid Grade Level [8], Coleman-Liau Grade Level [9], and Smog Grade [10]. We chose the second one:

$$Redabiliti\ CLI = 0.0588 L - 0.296S - 15.8$$

where L – average number of letters per 100 words, S – average number of sentences per 100 words. If CLI is around 10, text is easy to read, but if  $CLI > 15$  text is too difficult for understanding. We made a mapping into percentage interpretation (if CLI index is more than 17.5, than readability is 0%) by following formula:

$$Readability \% = \left(1 - \frac{|CLI - 12.5|}{5}\right) \times 100$$

**Completeness.** It requires that the requirement contain all necessary elements, including constraints and conditions, to enable the requirement to be implemented [18]. We calculated completeness quality attribute by this formula:

$$Completeness \% = \frac{N_{filled}}{N_{total}} \times 100$$

where  $N_{total}$  – the number of elements in the structural template,  $N_{filled}$  – the number of elements from templated that were identified in requirement sentence.

### 2.3 Natural Language Processing

NLP is considered a branch of Artificial Intelligence that is concerned with the analysis and interpretation of natural language or human language via several techniques such as Parsing, Part of Speech Tagging, Named Entity Recognition, Tokenization, Sentiment Analysis, etc. NLP system is asked to make unambiguous decisions about word meaning, category, syntactic structure, and semantic scope [5]. In software engineering, requirements can be seen as a set of sentences written in a specific language, and as any text data requirements may suffer from ambiguity. That's why NLP is handy to extract meaning and insight from requirements and, in our case, to get know how good requirements to a set of quality attributes.

## 2.4 Analytical Hierarchy Process

One of approaches that can help us in analyzing the priority of quality attributes is Analytical Hierarchy Process (AHP). In this case, there are 5 attributes used to analyze the requirement. Then we ask our customer to fill this questionnaire about their priority:

Table 1: Customer priority

Feature	Importance scale									Feature
Unambiguity	5	4	3	2	1	2	3	4	5	Singularity
Unambiguity	5	4	3	2	1	2	3	4	5	Readability
Unambiguity	5	4	3	2	1	2	3	4	5	Unsubjectivity
Unambiguity	5	4	3	2	1	2	3	4	5	Completeness
Singularity	5	4	3	2	1	2	3	4	5	Readability
Singularity	5	4	3	2	1	2	3	4	5	Unsubjectivity
Singularity	5	4	3	2	1	2	3	4	5	Completeness
Readability	5	4	3	2	1	2	3	4	5	Unsubjectivity
Readability	5	4	3	2	1	2	3	4	5	Completeness
Unsubjectivity	5	4	3	2	1	2	3	4	5	Completeness

From this table, for example, in the third row we got that unambiguity is 3 levels more important than unsubjectivity and Unambiguity and completeness are in same level of impotancy.

After that we calculated pairwise matrix, where the score from questionnaire is provided and  $a_{ij} = 1/a_{ji}$  and  $a_{ii} = 1$ .

Table 2: Pairwise comparison matrix

	Unambiguity	Singularity	Readability	Unsubjectivity	Completeness
Unambiguity	1	4	1	3	1
Singularity	1/4	1	1/4	1/4	1/5
Readability	1	4	1	4	1/2
Unsubjectivity	1/3	4	1/4	1	1/4
Completeness	1	5	2	4	1
Sum	3.58	18	4.5	12.25	2.95

Then we normalized matrix by formula:  $a_{ij} = a_{ij}/sum_j$

Table 3: Normalized matrix

	Normalized Matrix					Average
Unambiguity	0.279	0.222	0.222	0.245	0.339	0.261
Singularity	0.070	0.056	0.056	0.020	0.068	0.054
Readability	0.279	0.222	0.222	0.327	0.169	0.244
Unsubjectivity	0.093	0.222	0.056	0.082	0.085	0.107
Completeness	0.279	0.278	0.444	0.327	0.339	0.333

From the average above, we got the weight of each attribute. In this case, the prioritization order is completeness, unambiguity, readability, unsubjectivity, and singularity.

Table 5: Sample table

Attributes	Weight
Completeness	0.333
Unambiguity	0.261
Readability	0.244
Unsubjectivity	0.107
Singularity	0.054

The final quality of requirements can be calculated by this formula:

$$Q = \sum_{Attributes} (Score_A \times Weight_A)$$

### 3 Prototype

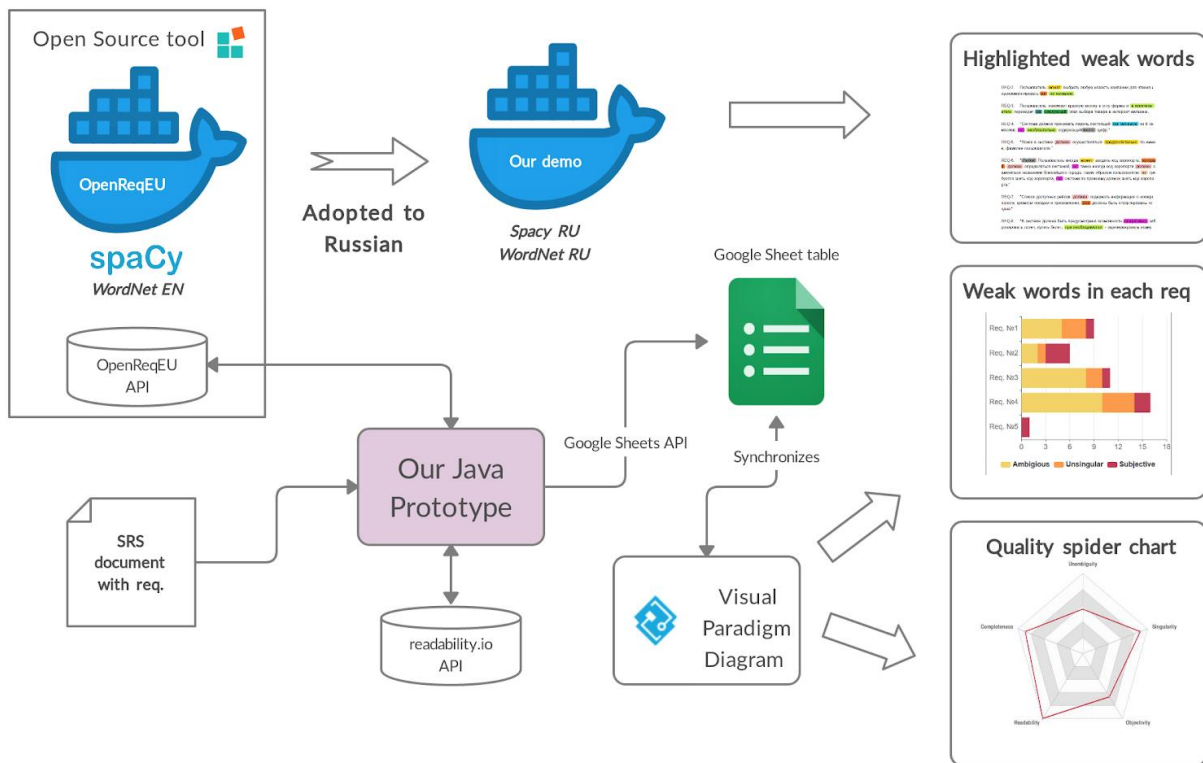


Figure 2: Prototype architecture

To fully support the extraction of metrics for all before-mentioned quality attributes, the prototype should have several features. The prototype is a software tool, which main goal is to perform requirements quality measurement.

Requirements can be of any type expressed in the text form: functional, non-functional, use-cases. The prototype is able to perform several functions:

- Integration with project management system to gather textual requirements from it (via API)
- Perform syntax and semantic analysis of said requirements (supporting Russian language [11][12])

The core of the prototype is the Requirement Quality Model, which contains a consistent set of requirements quality metrics and is expressed in algorithms on how to measure these metrics and how to draw conclusions (average quality of a requirement/set of requirements). The prototype provides a requirement engineer with a graphical user interface or command-line interface to obtain the results of requirements measurement. For NLP were used custom alternative Python libraries Wordnet [13] and Spacy [14] with Russian language support.

## 4 Conclusions

We proposed the model for process of quality assessment was based on NLP tools. Different quality attributes were analyzed and adopted. We developed a prototype that capable of reducing the challenges development team face with interpreting requirements due to ambiguity, subjectivity, poor readability or incompleteness. Suggested approach was tested on sample of requirements text. Quality metrics for different attributes were calculated according to customer's priorities for every require-ment and for overall document. This prototype can be further improved by exploring other NLP techniques to furnish users with a detailed explanation of why requirements lack quality attributes.

## 5 Acknowledgements

We thank the organizers of CASE in Tools International Hackathon: Andrey Sadovykh, Alexandr Naumchev, and every single person who contributed to the success of this event. We are immensely grateful to Giancarlo Succi for comments on an earlier version of the proposed model. This research would not have been conducted without efforts of Konstantin Valeev, the challenge owner, who shed more light on grey areas of this project and provided us with enough resources. We are also indebted to appreciate Rostelecom IT company for the opportunity to work on industry-related challenge. se a third level heading for the acknowledgements

## References

1. Khurana, Diksha & Koli, Aditya & Khatter, Kiran & Singh, Sukhdev. (2017). Natural Language Processing: State of The Art, Current Trends and Challenges.
2. Solingen, Rini & Berghout, Egon. (1999). The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development.
3. Ayalew, Yirsaw & Masizana, Audrey. (2009). Requirements Elicitation Techniques Selection Using AHP.. I. J. Comput. Appl.. 16. 180-190.
4. Basili, Victor; Gianluigi Caldiera; H. Dieter Rombach, The Goal Question Metric Approach, Basili,Victor;GianluigiCaldiera,1994
5. Daniel Jurafsky & James H. Martin. (2006). Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition.
6. Chantree, F. & Nuseibeh, B. & De Roeck, Anne & Willis, Alistair. (2006). Identifying Nocuous Ambiguities in Natural Language Requirements. Proceedings of 14th IEEE International Requirements Engineering Conference (RE'06). 59 - 68. 10.1109/RE.2006.31.
7. Massey, Aaron & Rutledge, Richard & Antón, Annie & Swire, Peter. (2014). Identifying and classifying ambiguity for regulatory requirements. 2014 IEEE 22nd International Requirements Engineering Conference, RE 2014 - Proceedings. 83-92. 10.1109/RE.2014.6912250.

8. Kincaid, J.P., Fishburne, R.P., Rogers, R.L., & Chissom, B.S. (1975). Derivation of new readability formulas (automated readability index, fog count, and flesch reading ease formula) for Navy enlisted personnel. Research Branch Report 8-75. Chief of Naval Technical Training: Naval Air Station Memphis.
9. Coleman, Meri; and Liao, T. L. (1975); A computer readability formula designed for machine scoring, *Journal of Applied Psychology*, Vol. 60, pp. 283-284
10. McLaughlin, G. Harry (May 1969). "SMOG Grading — a New Readability Formula" (PDF). *Journal of Reading*. 12 (8): 639-646
11. Kirill Igorevich Gaydamaka, "Characteristics and quality indicators of requirements for the Russian-speaking engineering environment," in Conference "Technologies for the Development of Information Systems" (Federal State Autonomous Educational Establishment of Higher Education "Southern Federal University", 2017)
12. Victor Konstantinovich Batovrin and Kirill Igorevich Gaydamaka, "Some features of the assessment of the characteristics of the requirements for systems," *Informatization and communication*, no. 4 (2017): 191-196.
13. (2017). ru-wordnet. GitHub repository. Retrieved from <https://github.com/jamsic/ru-wor>
14. Baburov, Y. (2018). spacy-ru. GitHub repository. Retrieved from <https://github.com/buriy/spacy-ru>