

Shekhovtsov V., Kostanyan A.
Computer Aided Management Systems Department
National Technical University “Kharkiv Polytechnical Institute”
21 Frunze Str., 61002, Kharkiv, Ukraine
E-mail: shekv1@kpi.kharkov.ua, kostanyan@datasvit.net

UNIVERSAL SOFTWARE SYSTEM REQUIREMENTS CLASSIFICATION

© Shekhovtsov V., Kostanyan A., 2006

This paper presents new requirements classification. It's more flexible than traditional functional and non-functional requirements division. Classification model is based on modern faceted approach and used International Standardization Organization software quality model. Join universal model with improvements and usage examples is introduced.

Keywords – system requirements, faceted approach, ISO software quality model.

1. Introduction

The first stage of any software system development according to RUP [1] is requirements analysis. It's the most complicated and important task as errors made on this step cause to project development time increasing and additional cost. System requirements are divided up into two main categories until recently [2, 3, 4]: functional and non-functional requirements. Functional requirements represent the main system goal. Non-functional requirements deal with issues such as performance, reliability, efficiency, usability, portability, testability, understandability and modifiability of the system.

Our paper presents new approach to classification of requirements, which allow to define more precisely requirements categories and thus processing such branches in more adequate and easy way.

The remainder of this work is organized as follows. Section 2 gives brief background information. In Section 3 we define open problems. Sections 4 describe our approach. Conclusions and future work are presented in Section 5.

2. Background

In this subsection, we will briefly describe the approaches that serve as the foundation of our method. After that, we will be able to see the open issues of these approaches and formulate the statement of the problem investigated in this paper.

Faceted classification of requirements

The basic concepts of this classification are facets. There are following facets which describe requirements from different points of view [5]:

1. Kind
2. Representation
3. Role
4. Satisfaction

The kind facet defines the matter of requirements. It could be “function” or “data” representing classical functional requirements. Next possible variant is “performance” of the system using terms like speed, volume, throughput, etc. Requirement could specify a “specific quality” such as reliability or portability. Finally, a requirement may be a “constraint” meaning a design decision or design constraint imposed by stakeholders.

Representation facet deals with verification of requirements. Depending on different verification procedures Glinz [5] defines following forms:

1. Operational form. Requirement can specify actions to perform, data to provide etc. Verification could be done using reviewing, testing or in formal way.
2. Quantitative form. Requirement can be precise, unambiguous. Verification provided by measuring.

3. Qualitative form. Requirement is more abstract and ambiguous. Mostly it defines some business goal. Verification could not be done directly. It's possible only after deployment, when stakeholder using prototype can decide whether or not requirement is satisfied.

4. Declarative form. Requirement just describes some required situation. Verification provided by reviewing.

Satisfaction facet defines different ways of requirement implementation. In first case stakeholder using some criteria can decide whether requirement is satisfied or not. In this case possible only two alternatives – satisfied or not (Boolean logic). Such kind of requirements called “hard”. In other case exists requirements for which it's not possible to define satisfaction exactly. We can only describe the level of satisfaction (fuzzy logic). Such requirements called “soft”.

The role facet defines three possible roles of requirements in system. It could be:

1. Prescriptive. Such requirements concern only system-to-be.
2. Normative. The requirements describe norms in the system environment that the system must be aware of.
3. Assumptive. The requirements describe behaviour of actors that the system-to-be can't control.

ISO/IEC 9126 quality model

The ISO/IEC standard 9126 [6] provides a set of attributes for evaluating the quality of software systems. The ISO 9126 standard is made up of six main attributes, each containing sub-attributes:

1. Functionality. The capability of the software to provide functions which meet stated and implied needs when the software is used under specified conditions.

2. Reliability. The capability of the software to maintain the level of performance of the system when used under specified conditions

3. Usability. The capability of the software to be understood, learned, used and liked by the user, when used under specified conditions.

4. Efficiency. The capability of the software to provide the required performance relative to the amount of resources used, under stated conditions.

5. Maintainability. The capability of the software to be modified.

6. Portability The capability of software to be transferred from one environment to another.

3. Problem statement

We investigated current state of the art and saw the following open problems.

1. Division to functional and non-functional requirements is too abstract. It doesn't allow to exactly define the nature of the requirements and process them in appropriate way. It's also hard to find “red line” between these two categories. The same requirement could be treated as functional and non-functional depending on different points of view. For example requirement “The system grants access to the customer data only to those users that have been authorized by their user name and password” can be classified as both functional and non-functional (security).

2. Faceted classification of requirements is more flexible and powerful. But there are some weak places, which can't allow to use it in original way for universal requirements classification. The main problem in it is kind of facet which subdivisions are quite strange. From authors point of view separation of performance as stand alone variant and using term “specific qualities” for all other variants is not adequate to the requirements nature.

3. There are predefined set of quality positions exist for software systems. They are developed by International Standard Organization and are common for using. So, it's necessary to have possibility consider all these quality items on requirements analysis stage to process them correctly in the future development.

This paper describes our approach to solve all mentioned above problems.

4. Universal requirements classification

Our approach is based on joining of ISO/IEC 9126-quality model and faceted presentation of requirements. Also some improvements were made in original faceted model to make it more adequate to requirements nature. Our model also consists of four facets: kind, representation, satisfaction, role. Full classification is listed below.

Kind facet including:

1. Functionality: Suitability, Accuracy, Interoperability, Security.
2. Reliability: Maturity, Fault tolerance, Recoverability.

- 3. Usability: Understandability, Learnability, Operability, Attractiveness.
- 4. Efficiency: Time behaviour, Resource utilization.
- 5. Maintainability: Analyzability, Changeability, Stability, Testability.
- 6. Portability: Adaptability, Installability, Co-existence, Replaceability.

Representation facet including:

- 1. Operational form.
- 2. Quantitative form.
- 3. Qualitative form.

Satisfaction facet including:

- 1. Hard.
- 2. Soft.

Role facet including:

- 1. Prescriptive.
- 2. Assumptive.

As we can see it's close to original faceted model but with some important points:

1. Kind facet is reached with all ISO quality model points. This allows to separate and handle such criteria beginning from requirements analysis stage. And also makes classification compatible with common standard.

2. Prescriptive form of role facet includes inside normative form and constraints (in Kind facet of Glinz). It could be done as there is no semantic difference between stakeholders constraints, "classic" requirements and norms in system for example formulas.

3. Qualitative form of representation facet includes inside declarative form of Glinz as there is no difference between them from authors point of view.

Using this classification we could separate exactly functional and non-functional requirements. Main criteria in this process is kind facet. According to it first three points of functionality are functional requirements (Suitability, Accuracy Interoperability), other seems to be non-functional.

Other facets define more precisely the nature of requirements with important from seen of system design, but can't change the global type of them.

A couple of examples in the table below illustrate using of our classification on sample requirements expressed in natural language for banking system.

TABLE 1

REQUIREMENTS CLASSIFICATION SAMPLES

Requirement	Classification
"The banking system deals with accounts and customers."	Kind: Suitability Representation: Qualitative Satisfaction: Hard Role: Prescriptive
"All the operations with account must be logged into the file."	Kind: Fault tolerance Representation: Quantitative Satisfaction: Hard Role: Prescriptive
"When the customers attempt to withdraw the money from their accounts, it is necessary for them to supply a password."	Kind: Security Representation: Operational Satisfaction: Hard Role: Prescriptive
"System shall run on Linux platform also."	Kind: Adaptability Representation: Qualitative Satisfaction: Soft Role: Prescriptive

Requirement	Classification
““The system shall run on PCs featuring at least a 1.5GHz CPU and 256MB main memory.”	Kind: Resource utilization Representation: Quantitative Satisfaction: Soft Role: Prescriptive
“Will be good, if bank’s staff easy understand principles of system working.”	Kind: Understandability Representation: Qualitative Satisfaction: Soft Role: Assumptive
“If security manager will see in system money withdraws greater than 10000\$ from client account, he should phone the client.”	Kind: Security Representation: Operational Satisfaction: Hard Role: Assumptive

5. Conclusions and future work

Presented classification allows to regulate requirements in flexible, informative and mature based form on the early stages of software development process. This fact makes handling of such requirements more correct and adequate. As the result target software system will satisfy all customer requirements in more preferred way.

Our approach is the next step in research started in [7]. In it we described our “Aspectual Predesign” technique to process requirements. One of the main goals of this research is to separate concerns based on natural language requirements specification. To make this we need universal and powerful requirement classification, presented in this paper. Future work will be concentrated on creating NLP technique capable to extract and classify requirements from customer documentation expressed on natural language.

References

- [1] Booch, G.,; Rumbaugh, J.; Jacobson, I.: The Unified Modeling Language User Guide. Addison Wesley Publ. Comp. 1999.
- [2] IEEE (1993). IEEE Recommended Practice for Software Requirements Specifications. IEEE Standard 830-1993.
- [3] Kotonya, G., I. Sommerville (1998). Requirements Engineering: Processes and Techniques. John Wiley & Sons.
- [4] Van Lamsweerde, A. (2001). Goal-Oriented Requirements Engineering: A Guided Tour. Proceedings of the 5th International Symposium on Requirements Engineering (RE'01), Toronto. 249-261.
- [5] Glinz, M. (2005). Rethinking the Notion of Non-Functional Requirements. Proceedings of the Third World Congress for Software Quality (3WCSQ 2005), Munich, Germany, Vol. II, 55-64.
- [6] ISO/IEC 9126-1 (2001). Software engineering – Product quality – Part 1: Quality model. International Organization for Standardization.
- [7] Shekhovtsov V.A., Kostanyan A.V.: Aspectual Predesign. In: Kashek, R.; Mayr, H.C.; Liddle S. (eds.): Information Systems Technology and its Applications – ISTA'2005. Lecture Notes in Informatics P-63, GI-Edition, 2005, pp. 216-226.