WNR Approach: An Extension to Requirements Engineering Lifecycle

Ahmad Abdollahzadeh Barforoush, Abbas Rasoolzadegan, Reza Gorgan Mohammadi
Information Technology and Computer Engineering Faculty
Amirkabir University of Technology (Tehran Polytechnic)
Iran, Tehran
{ahmad, rasoolzadegan, reza_gorgan}@aut.ac.ir

Abstract—***Requirements engineering (RE) plays a vital role in ensuring the overall success of the software engineering process. Requirements elicitation, which describes the expected behavior of the desired software, is very complicated. In this paper, a new approach, named WNR, is proposed which put emphasis on discovering the user wishes and transforming them to a set of needs and subsequently obtaining the set of desired requirements out of the needs. The new proposed approach is an extension to RE lifecycle. Moreover, the definitions of the terms "requirement" and "requirements engineering" are reviewed in this paper. The new definition of RE in the proposed approach is also elaborated.***

Keywords-Requirements engineering; new approach; RE lifecycle

I. INTRODUCTION

To have a successful business and increase company's profit, it is necessary to develop products that satisfy the expectations of the users. Discovering user wishes and transforming them into requirements is a challenging issue.

Requirements engineering (RE) as a key process of software engineering plays a crucial role throughout the software development lifecycle. Requirements engineering, a branch of software engineering, offers models, methods, metrics, and indicators in a specified methodology to systematically define and manage the requirements of users.

Different studies [1,2] have shown that poorly defined software requirements, lack of user involvement, incomplete requirements, and changing requirements are the major reasons why information technology projects do not deliver all of their planned functionality on schedule and within budget. Well-defined requirements will increase the likelihood of the overall success of the software project. RE lies at the heart of system development. In fact, RE bridges the gap between stakeholder goals and constraints, and their realization in systems. However, it will not be possible to develop better high quality requirements without a well-defined RE process.

Since requirements engineering is the starting point of software engineering and later stages of software development rely heavily on the quality of requirements, there is a good reason to pay close attention to the RE process. Over the past decades there has been increasing interest in requirements engineering. This paper presents a new approach called WNR[1] to RE definition which mainly concern and put emphasis on transforming users wishes to software requirements.

In our previous work [3], we have conducted an extensive meta-analysis of the RE literature. We have also analyzed the trends and issues in several research resources using an iterative and evolutionary method. The approach presented in this paper is based on the results of the work. The main idea of WNR is to improve previous approaches to RE.

The rest of the paper is organized as follows: in section 2, several definitions for the terms "requirement" and "requirements engineering" are reviewed and the key phrases of each definitions are gathered together to find the overlaps between definitions. Also, different types of requirements and main activities in the RE process are elaborated. In section 3, the WNR approach is described as an extension to the RE process. The review of related works comes in section 4. Section 5 discuses future work and draws some conclusions.

II. KEY CONCEPTS AND DEFINITIONS

A. Requirement

The concept of requirement is fundamental in requirements engineering. Generally speaking, the concept of requirement, discussed in the RE domain, refers to what a system should do rather than how it should do it [4]. The literature offers many definitions for the term "requirement". Each definition emphasizes on some aspects and properties. Appendix A summarizes several definitions for the term "requirement", sorted in ascending order by year.

Table I shows the topic/reference matrix for the definitions in Appendix A. This table is organized according to the keywords in the definitions of the term "requirement". In the rows, we have the keywords of the definitions denoting the essence of a requirement, and in the columns, we have the references. A "+" symbol indicates that the topic appears in the reference.

In Table II, in the rows, we have phrases denoting the desired goals of a requirement. As the table indicates, "solving a problem" is considered the most common goal for a requirement.

[1] Wish-Need-Requirement

TABLE I. KEYWORD/REFERENCE MATRIX FOR THE TERM "REQUIREMENT"

	Abbott [5]	IEEE [6,8]	Davis [7]	SWEBOK [9]	Lethbridge [10]	RUP [11]	Sommerville [12]
Functionality	+		+				
Constraint	+		+				+
Property	+			+			
Condition		+				+	
Capability		+				+	
Feature			+				
Attribute			+				
Services			+				+
Statement			+		+		
Need	+	+	+				+

TABLE II. PHRASE/REFERENCE MATRIX FOR THE TERM "REQUIREMENT"

	Abbott [5]	IEEE [6,8]	SWEBOK [9]	Lethbridge [10]	RUP [11]	Sommerville [12]
Solving a problem		+	+	+	+	
Satisfying a condition/ constraint	+	+				+
Satisfying a capability /property	+	+				+
Achieving an objective		+				
Satisfying formally imposed documents		+				
Agreement with the Stakeholder				+		

B. Types of Requirements

The requirement definitions generally describe external and internal behaviour of system which must be met or provide user satisfaction on certain constrains and operational environment. The definitions emphasize that a system must satisfy functional requirements which are recognized as a user need and system requirements which depends on non-functional, constrains, and operational environment.

According to the mentioned definitions, requirements can be categorized into product requirements and process requirements. The former contains functional and non-functional requirements; whereas the latter consists of operational constraints and properties of the operational environment. Fig. 1 illustrates the classification.

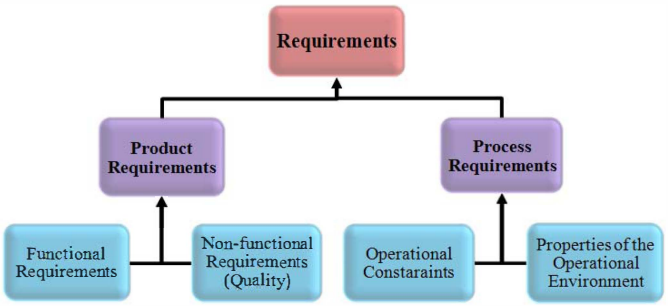

Figure 1. Types of requirements

C. Requirements Engineering

In recent years, various definitions elaborating the term RE from different points of view have been proposed. Appendix B summarizes the significant ones sorted in ascending order by year. We define RE as "A systematic and disciplined approach which transforms user wishes to software requirements based on business needs. Activities are included Eliciting, Analysis, Negotiation, Specification, Modeling, Validation & Verification, Identification, Change Management, and Traceability". In Table III, the key phrases in the definitions of RE are presented, and their existence in each reference are indicated by a "+" symbol.

D. Requirements Engineering Process

Requirements engineering, itself, is a process. Numerous RE process models have been developed in the past two decades [4, 6, 20, 21]. Activities of the RE process generally include requirements elicitation, analysis and modeling, specification, verification and validation (V&V), and requirements management.

TABLE III. PHRASE/REFERENCE MATRIX FOR THE TERM RE

	Ross [13]	Loucopoulos [14]	*Tha* [15]	Zave [16]	Thayer [17]	RUP [11]	Sommerville [18]	Pressman[19]	Our Definition
Understanding needs				+			+	+	+
Discovering / Eliciting requirements						+	+		+
Assessing/ Analyzing needs-Assessing feasibility	+		+	+			+		+
Negotiating a reasonable solution				+			+		+
Specifying / Documenting requirements			+	+	+	+	+		+
Validating the specification			+	+			+		+
Organizing requirements							+		+
Change management				+		+	+	+	+
Systematic approach			+			+	+		+

E. Descriptive and Prescriptive Models

Based on the definitions presented in [24], models in a narrow sense can be classified under various aspects. A model

can mirror an existing original (like a photograph), or it can be used as a specification of something to be created (like a construction plan). In the former case, we call it a descriptive model; in the latter case, we call it prescriptive. This is in fact a property of the relationship between a (particular) model and a (particular) original rather than a property of the model. Therefore, a model may be descriptive with respect to one original, and, at the same time, be prescriptive with respect to another original.

III. WNR APPROACH TO RE

The RE process is inherently complex and critical. The main reason is that the requirements engineering process has the most dominant impact on the capabilities of the resulting product. Furthermore, requirements engineering is the process in which the most diverse set of product demands from the most diverse set of stakeholders is being considered [22].

Activities of the RE process are generally categorized intro "requirements development" and "requirements management" activities. Requirements development activities consist of six main activities including requirements elicitation, negotiation, analysis, specification, modeling, verification, and validation (V&V). Requirements management activities include change management, traceability, and identification.

In the authors' view of requirements engineering, the process is iterative and incremental. RE activities cover the entire system and software development lifecycle and the process will go into more detail in iterations. Therefore, requirements engineering should be viewed as an incremental and iterative process, performed in parallel with other system development activities such as design. Accordingly, RE might not solely be a front end process, but also be part of the later stages of software engineering. Indeed, if a software development process is defined based on RUP, then the proposed approach to engineering requirements, called WNR, can be considered as a systematic approach to the definition of requirements with respect to initial wishes during the requirement discipline throughout the software development process.

In WNR approach, RE process consists of three main phases namely "Wish", "Need", and "Requirement". Based on the characteristics mentioned about RE process, we choose an idea inspired by the two-dimensional view of RUP[2] [11] to illustrate and depict our RE process as shown in Fig. 2. Like RUP, the horizontal axis represents the dynamic aspect of process, and it is expressed in terms of three phases called Wish, Need, and Requirements. The vertical axis shows the static aspects of the process, and it is described in terms of development and management activities.

The relationship between the three phases, mentioned above, is shown in Fig. 3. The process commences by a primary set of user expectations called wishes. The output of the first phase is presented as a descriptive model since it contains the existing ideas and expectations of the users. The outcome of the Wish phase is called "What$_W$" (What in the stage of Wish). However, some of the wishes are not necessary or even feasible. In the proposed approach, the wishes that are necessary and feasible are called needs.

In the second phase, the list of wishes, called "wishlist", is filtered by asking two questions: "why" and "who". The former determines the motivation for the wishes. The latter determine the related owners of the wishes. In other words, the second phase receives What$_W$, as input, and generates the corresponding needs, as output. This phase presents its output using a prescriptive model since it is the specification of the system to be created. The outcome of the second phase is called "What$_N$" (what in the stage of Need).

Finally, at the Requirement phase, the aim is to form the set of "What$_R$" (what in the stage of Requirement) which are the specification of desired requirements. At this stage, the list of needs are filtered by considering software engineering concerns such as quality, estimated time and cost, and risks of the software. For filtering the list of needs "How", "How much", "When", and "Where" are asked. The requirements are those needs that are cost-effective. At the end of each phase, the existence of reasonable answers for the corresponding questions (such as why and who in the Need phase) is checked as milestones.

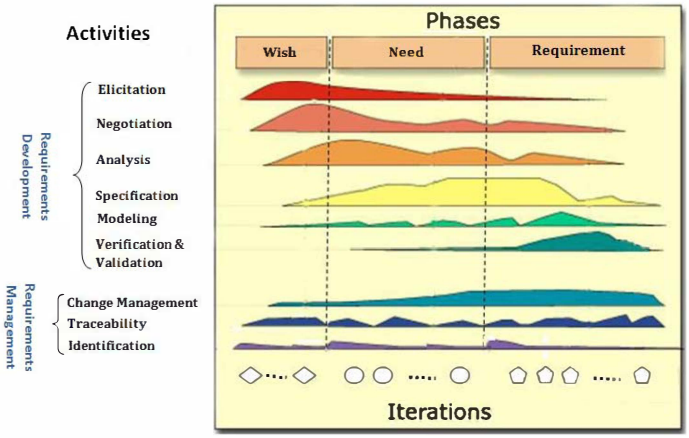

Figure 2. The proposed approach to RE

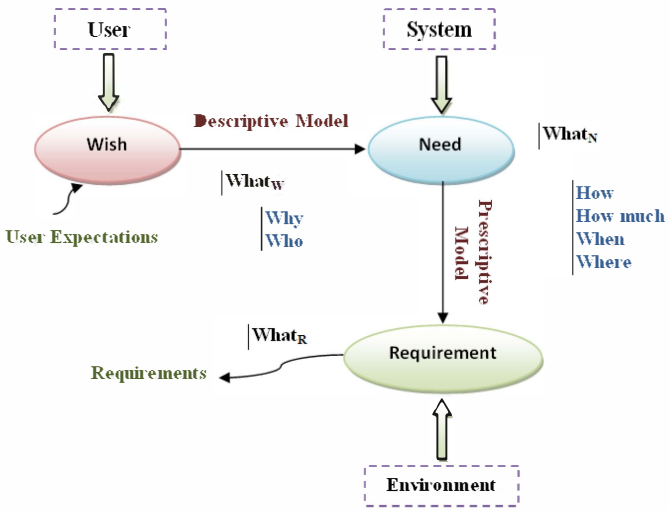

Figure 3. The relation between three phases

[2] Rational Unified Process

Assuming that W, N, and R are the sets of all wishes, needs, and requirements respectively, (1) and (2) hold.

$W = \{\omega_1, \dots, \omega_n\}$ The set of all wishes

$N = \{\eta_1, \dots, \eta_m\}$ The set of all needs

$R = \{\tau_1, \dots, \tau_p\}$ The set of all requirements

$$N = \{\forall \omega \in W \mid Pase(\omega)\} \quad (1)$$

$$N = \{\forall \omega \in W \mid Pase(\omega)\} \quad (2)$$

Relation (1) indicates that the members of the need set are those wishes that have passed the Need phase ($Phase_N$). Similarly, the members of the requirement set are the needs qualified as requirement during the Requirement phase ($Phase_R$).

Requirements management activities are similar to supporting disciplines in RUP. In WNR, the supporting activities are requirement change management, traceability, and identification which can be formally defined by using temporal logic as illustrated in (3). The requirement management activities are continuously performed during the lifecycle of requirements as umbrella activities. In temporal logic, the statement □ (φ) means that φ has to hold on the entire subsequent path. In logic, the statement $\neg A$ is true if and only if A is false. For example, the first statement of (3) means that until the development process is not finished, the requirements change management has to done.

□ (¬ Finish (Development Process) ⇨ Do (Requirements Change Management)

□ (¬ Finish (Development Process) ⇨ Do (Requirements Tracing) (3)

□ (¬ Finish (Development Process) ⇨ Do (Requirements Identification)

IV. Conclusion

One of the most important aspects of successful software development is having well defined requirements. The quality of the approach used for requirements engineering affects directly the quality of gathered requirements.

In this paper, we have proposed a new systematic approach, called WNR, to requirements engineering. The proposed approach put emphasis on the elicitation of user wishes and transforming them to needs and then requirements. It is worth mentioning that there has already been no systematic approach to engineering the process of transforming wishes to requirements. In future, we aim to define models and related meta-models for the WNR approach and also apply it to some real case studies.

Acknowledgments

This research has partially used financial support of ITRC (Iran Telecommunication Research Center). We are also grateful to Zoubida Kedad, and Peri Locoupolous for their comments and helps during the sabbatical period of Professor Barfroush at Loughborough University.

References

[1] D. Rubenstein, Standish group report: There's less development chaos today, SD Times, http://www.sdtimes.com/article/story-20070301-01.html, 2007.

[2] R. Charettte, Why software fails, IEEE Spectrum, pp. 42-49, Sept. 2005.

[3] A. Rasoolzadegan, R. Gorgan Mohammadi, A. Abdollahzadeh, Requirements Engineering: Research Trends and Issues, Technical Report, Information Technology and Computer Engineering Faculty, Amirkabir University of Technology (Tehran Polytechnic), CE/TR.C/90/02, Tehran, Iran, November, 2011.

[4] I. Sommerville and P. Sawyer, Requirements Engineering: A Good Practice Guide, New York: Wiley, 1997.

[5] R. J. Abbott, An Integrated Approach to Software Development, New York: John Wiley, 1986.

[6] IEEE Standard Glossary of Software Engineering Terminology (ANSI), [1-55937-067-X] [SH13748-NYF], 1990.

[7] A. M. Davis, Software Requirements: Objects, Functions, and States. Upper Saddle River, NJ: Prentice Hall, 1993.

[8] IEEE Std 830, IEEE Recommended Practice for Software Requirements Specifications, 1998.

[9] SWEBOK, Guide to the Software Engineering Body of Knowledge, A Stone Man Version (Version 0.6), February 2000.

[10] T. Lethbridge and R. Laganiere, Object oriented software engineering: practical software development using UML and Java, England: Mc-Graw Hill Education, 2001.

[11] Rational Unified Process, www.ibm.com/software/awdtools/rup/.

[12] I. Sommerville, Software Engineering, 7th Edition, Addison Wesley, 2007.

[13] D. T. Ross and K. E. Schoman, "Structured Analysis for Requirements Definition", IEEE Transactions on Software Engineering, vol. 3, no. 1, pp. 6-15, 1977.

[14] P. Loucopoulos and E. Kavakli, "Enterprise Modeling and the Teleological Approach to Requirements Engineering", International Journal of Intelligent and Cooperative Information Systems, vol. 4, no. 1 pp. 45-79, 1995.

[15] R. H. Thayer and M. Dorfman, Software Requirements Engineering (2nd Ed), IEEE Computer Society Press, 1997.

[16] P. Zave and M. Jackson, "Four Dark Corners of Requirements Engineering", ACM Transactions on Software Engineering and Methodology, vol. 6, no. 1, pp. 1-30, 1997.

[17] R. H. Thayer and M. C. Thayer, "Software Requirements Engineering Glossary", IEEE Computer Society Press, pp. 489-528, 1997.

[18] I. Sommerville and P. Sawyer, Requirements Engineering: Good Practice Guide. Chichester, England: John Wiley & Sons Ltd., 1997.

[19] R. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, 7th edition, McGraw-Hill, 2010.

[20] L. Jiang, A Framework for the Requirements Engineering Process Development, Ph.D. Thesis, Department of Electrical and Computer Engineering, University Of Calgary, Canada, Sept. 2005.

[21] T. Gorschek and A. Davis, "Requirements Engineering: In Search of the Dependent Variables, Information and Software Technology", vol. 50, no. 1&2, pp. 65-75, January 2008.

[22] J. L. Mate and A. Silva, Requirements engineering for sociotechnical systems, Idea Group Inc, 2005.

[23] J. Ludewig, "Models in software engineering – an introduction", Journal of Software and Systems Modeling, vol. 2, no. 1, March, Springer-Verlag, 2003.

[24] D. Bjørner, Software Engineering 3: Domains, Requirements, and Software Design, Springer, 2006.

APPENDIX A

Table IV presents several definitions for the term "Requirement".

TABLE IV. DEFINITIONS FOR THE TERM "REQUIREMENT"

Index	Source	Definition	Year
1	Abbott [5]	Any function, constraint, or other property that must be provided, met, or satisfied to fill the needs of the system's intended user(s).	1986
2	IEEE [6]	(1) A condition or capability needed by a user to solve a problem or achieve an objective. (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. (3) A documented representation of a condition or capability as in (1) or (2).	1990
3	Davis [7]	A user need or a necessary feature, function, or attribute of a system that can be sensed from a position external to that system.	1993
4	IEEE [8]	A well-formed requirement is a statement of system functionality (a capability) that must be met or possessed by a system to satisfy a customer's need or to achieve a customer's objective, and that is qualified by measurable conditions and bounded by constraints.	1998
5	SWEBOK [9]	A property that must be exhibited by a system developed or adapted to solve a particular problem.	2000
6	Lethbridge [10]	A statement about the proposed system that all stakeholders agree must be made true in order for the user's problem to be adequately solved.	2001
7	RUP [11]	A requirement is defined as "a condition or capability to which a system must conform.	2003
8	Sommerville [12]	The requirements for a system are the descriptions of the services provided by the system and its operational constraints. These requirements reflect the needs of customers for a system that helps solve some problem such as controlling a device, placing an order, or finding information.	2007

APPENDIX B

Table V summarizes the significant definitions for RE sorted in ascending order by year. The new proposed definition of RE is also given in the last row of this table.

TABLE V. DEFINITIONS FOR THE TERM "REQUIREMENTS ENGINEERING"

Index	Source	Definition	Year
1	Ross [13]	Requirements engineering is a careful assessment of the needs that a system is to fulfill. It must say why a system is needed, based on current or foreseen conditions, which may be internal operations or an external market. It must say what system features will serve and satisfy this context. And it must say how the system is to be constructed.	1977
2	Loucopoulos [14]	Requirements engineering is a systematic process of developing requirements through an iterative and co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained.	1995
3	Thayer & Dorfman [15]	Requirement engineering provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification, and managing the requirements as they are transformed into an operational system.	1997
4	Zave [16]	Requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families.	1997
5	Thayer & Thayer [17]	Requirements engineering covers all of the activities involved in discovering, documenting, and maintaining a set of requirements for a system. The term engineering implies that systematic and repeatable techniques should be used to ensure that system requirements are complete, consistent, relevant, etc.	1997
6	Sommerville & Sawyer [18]	Requirements engineering contains a set of activities for discovering, analyzing, documenting, validating, and maintaining a set of requirements for a system.	1997
7	RUP [19]	A systematic approach to eliciting, organizing, and documenting the requirements of the system, establishing, and maintaining agreement between the customer and the project team on the changing requirements of the system.	2003
8	Sommerville [4]	Appropriate mechanism for understanding what the customer wants, analyzing needs, negotiating a reasonable solution, validating the specification, and managing changes in requirements.	2007
9	Pressman [19]	The broad spectrum of tasks and techniques that lead to an understanding of requirements is called requirements engineering. From a software process perspective, requirements engineering is a major software engineering action that begins during the communication activity and continues into the modeling activity. It must be adapted to the needs of the process, the project, the product, and the people doing the work.	2010
10	Our definition	A systematic and disciplined approach to transform user wishes to software requirements based on business needs. Activities are included Elicitation, Analysis, Negotiation, Specification, Modeling, Validation & Verification, Identification, Change management and Traceability.	2011