

A Systematic Review for the Latest Development in Requirement Engineering

Ishraq A. Al-Fataftah, Ayman A. Issa

Abstract—Requirement engineering has been the subject of large volume of researches due to the significant role it plays in the software development life cycle. However, dynamicity of software industry is much faster than advances in requirements engineering approaches. Therefore, this paper aims to systematically review and evaluate the current research in requirement engineering and identify new research trends and direction in this field. In addition, various research methods associated with the Evaluation-based techniques and empirical study are highlighted for the requirements engineering field. Finally, challenges and recommendations on future directions research are presented based on the research team observations during this study.

Keywords—Current trends, Future trends, Requirement Engineering

I. INTRODUCTION

REQUIREMENT ENGINEERING (RE) is the process of identifying the goal behind a software system by identifying different stakeholders of the system, evaluate their needs with regard to the context to which the system will operate in, state the system services and constraints, and finally communicate the resulted findings in a form of requirements throughout the development life cycle of the software system. According to Cheng and Atlee [3], Requirement engineering activities differs from other software engineering activities as they are concerned with identifying the *problem* and *how* the environment will be affected by the proposed system whereas other activities are concerned with the *solution* of the problem and the *behavior* of the system under these requirements. *Furthermore, different factors contribute in making RE a difficult task:*

1. Requirements are usually abstract and not well-defined at the beginning of the analysis phase. Making it difficult to extract correct, viable and complete requirement statement.
2. Analysts need to gather requirement information from various stakeholders, which raise problems like conflicting requirements, integration and validation problems. The need to document these requirements in order to relate them to their source stakeholders is crucial for subsequent processing and validation.
3. Requirements must be developed and influenced by the environment that will host the system (Organization context), which requires clearly identifying the environment, constraints that are imposed by the environment domain, and the complexity of the environment itself especially when physical aspects of the environment (e.g. Hardware) need to be taken into consideration.

I. A. Al-Fataftah is currently studying Master degree in Princess Sumaya University of Technology, Amman, Jordan (phone: +962 6 535 9949, fax: +962 6 534 7295, e-mail: ishraq.fataftah@students.psut.edu.jo)

Ayman. A. Issa, Department of Computer Science, University of Calgary, Alberta, Canada. (e-mail: issaa@ucalgary.ca).

4. The evolving nature of the environment requires requirements to evolve as well. Maintaining integrity and consistency among already set and new requirements is crucial for the continuity of the project and its ability to conform to various changes in future.
5. It is sometimes the case to be unable to form requirement statements to be understood by both stakeholders (non-technical) and domain experts (technical) people. The use of adequate processes and modeling notations is important to deliver documents that are easily understood by high-level users and detailed enough for the usage of engineers.

In this paper, we will evaluate the development of requirement engineering based on a roadmap designed by Nuseibeh and Easterbrook [1]. We will also identify current trends and revolutions in the main RE activities: elicitation, modeling and analysis, agreeing and communicating requirements, and evolving requirements. Section 2, presents the latest development in RE activities. In section 3, the main research methods and the need to focus on empirical research in RE will be discussed. Finally, new trends in RE will be highlighted and outlined in section 4 followed by some recommendation for the future based on what we observed during this research.

II. ADVANCES IN REQUIREMENT ENGINEERING ACTIVITIES

A. Requirements Elicitation

Requirement elicitation is defined as [4] “*The practice of obtaining the requirements of a system from users, customers and other stakeholders. The practice is also sometimes referred to as requirements gathering*”. So requirement elicitation employs the discovery of requirements that resides in the brain of stakeholders, business in use, flowcharts and business documents to be made after conducting surveys and meetings. This requires a thorough understanding of the system to be developed combined with careful extraction of requirements and business rules and constraints. Usually, in this stage the model to be used in later processes is developed.

Many elicitation techniques resides to help us in elicit requirements from various sources.

These techniques can be grouped as the following:

1. Traditional techniques [1] that intend to identify various sources of requirements and make sure that anyone affected by the system is involved in the elicitation process. Also, an understanding of the existing system in use, if exists, can be obtained using these techniques. These techniques include Interviews, Surveys, Observation and Document analysis. It also includes Group elicitation like JAD/RAD and focus groups.
2. Contextual and personal techniques [3] that intend to analyze requirements in respect to a specific context or

environment and even for a specific user. Also, integration with other environments and systems should be analyzed and integration requirements should be extracted. Such techniques include the use of ethnography [1] and Interface analysis. Sometimes, when it is difficult to obtain requirements from stakeholders and existing systems, reverse engineering can be used as the last option to extract requirements from existing systems.

3. Innovative techniques that tend to define requirements that help making the system more acceptable and attractive to stakeholders. These techniques include brainstorming, and requirement workshops.
4. Feedback techniques that are used to get users and stakeholders opinion regarding the system developed. These can be used when there is no certainty in the requirements in hand and need more feedback from the users of the system. These techniques include prototypes, storyboards, visual models, and use models.

Although requirements elicitation have developed over years and many argues about the complexity and difficulty of eliciting requirements, the situations and context to which elicitation is used is variable and never the same [17]. Each project has its own characteristics and properties. Stakeholders vary and the degree of communication with analysts, their experience and maturity plays a significant role in successful elicitation of requirements. Moreover, technological advances and solutions are evolving making the elicitation process vary by technology used and the degree to which the analyst aware of these advances. The gap between requirement elicitation research and practice still exists due to many factors like missing processes documentation, lack of reports and cases about specific problems, and the difficulty in transferring knowledge in this regard.

B. Modeling and Analyzing Requirements

The requirement modeling activity aims to better represent what the customer wants in terms of more precise and specific details that are easily understood by high-level people and detailed enough to carry out the subsequent development activities [5]. Modeling is carried out to analyze the system to be done. Due to the increased importance of the modeling phase, many recognize it as a separate activity. Its importance increases as the size and complexity of the system increases. [6].

Requirement analysis is usually identified as the process of studying and refining requirements of the system [7]. Requirements usually referred to the functional aspects of the system that need to be presented in a way that is easily understood by the community of stakeholders and engineers of the system.

Requirement Analysis is usually used to refer to the process of identifying the specifications of the system that satisfies the customers' demands and provide enough information to build the system. Usually, these specifications are formal or semi-formal description of the problem to be solved that is

represented in a way that is difficult to be understood by customers. Requirements analysis elaborates on requirements collected at earlier stages of the project and uses models to reflect these requirements, depict user scenarios, problems and their relationships, functional and flow activities and system behavior. Modeling notations are used to specify how the system will be constructed and interpreted. Notations can be any symbols, text, characters and abbreviated expressions that raise the level of abstraction in the requirement description [3], [7]. Notations can be informal, semi-formal or formal that provides a level of details with regard to the data obtained from the system, constraints imposed on them, and responses to various inputs. Scenarios proved their usefulness in eliciting, validating and documenting requirements. Many people spent a considerable amount of time writing description to real world scenario and stories and therefore became the center for research for not only requirement modeling but also the reusing of scenario-based models [8]. Scenario based modeling when performed before the design phase assist in better understanding how the future system will operate, when it will fail to meet demands, and provide description on various user behaviors and interactions.

Requirement models have been extended over years where the concern is no longer bounded only to the environment in which the system will operate in, the organization domain in which the development takes place, and behavior and data modeling [1].

A considerable attention has been given to [12]:

- State-oriented modeling, where the system is modeled around a set of states and transition between these states based on certain criteria. Finite state machines (FSMs) used in control systems, finite state machines with data paths (FSMDs) is an evolution of the FSMs, State Charts and Petri nets that provide a mathematical basis for formal analysis are examples of state oriented models.
- Activity-oriented modeling, where the system is modeled around a set of activities related by data or dependency relation. Data flow diagrams (DFDs) and flowchart are examples of activity oriented models.
- Heterogenous modeling and Multiple view modeling. Heterogenous modeling employs various modeling techniques in one system representation. These models could be any of the above models in addition to data and structured models. In contrast, multiple view modeling uses a different set of models to represent different characteristics of the system. UML is an example of multiple view modeling.
- Structure modeling, where this model is concerned with the physical structure of the system and the characteristics of its components. Component connectivity diagrams (CCDs) is an example of structure modeling.

The resulted requirement model not always can be used directly in the design phase. Mapping stakeholder requirements into design models usually needs several

decisions that are not easily done by a tool or a method. Due to this complication, the need for requirement transformation evolved and many approaches and techniques have been proposed. Use case models can be transformed into object-oriented classes/objects models. Others suggested the use of rationale based on goal identification for the transformation of architecture design issues [12].

As described above, the usage of functional and behavioral notations in requirement modeling is still in use where the progress in the usage of specialized notations of the environment was slow. Major challenges in requirement modeling in Nusaibah's roadmap [1] have been met. The development of new techniques for formal modeling and analysis of non-functional requirements achieved considerable advances [9]. The gap between functional and non-functional requirement modeling has been narrowed by the use of techniques from natural language and semantic notations, formal design analysis and quality attributes taxonomy [10], [11] and [13].

C. Agreeing and Communicating Requirements

According to Palyagar and Richrads [14] "*Communication among all involved in SDLC plays a vital role in providing solution to various problems of the customer. Amongst all SDLC phases, Requirements Engineering (RE) is the most communication rich phase.*" Communicating requirements should be carried out in a structural way in order to enhance the effectiveness of the SE process and overcome the defects of unstructured communication where requirement defects are delayed to later phases.

Communications are crucial for global software development where the team members are not located in one place. As software business nowadays is highly active globally, the need to formalize efforts for communicating requirements among various team members is crucial to the success of these projects. Researchers found that informal communication faces huge challenges when systems are far in distance; tools should be developed to enhance communication among distributed systems [15].

Agreeing requirements is carried out by verifying and validating that requirements analyze and model accurately what the stakeholder need. Verification and validation usually carried out with direct involvement from the stakeholder. Validation is a subjective evaluation task that uses informal specifications or undocumented requirements. When changes in the requirements are verified and validated, it should be ensured that these changes do not employ bad impact on existing requirements [20]. Research in this area focuses on improving the type of information given to stakeholders for feedback such as graphical animation that is used to validate the design model against the informal requirements [21], [22], and simulation in which the modeling language is based on semantics for the simulator to be able to execute the system model [23], [24]. When a formal description of the requirements exists, verification can be made to make sure that system specifications meet these requirements.

Verification can be made by using Model checking in which state models are checked automatically [3], [30].

D. Requirement Management

Requirement management is a continuous process that involves documenting, tracing, prioritizing and agreeing requirements. The identification of requirements, managing of change, and identifying traceability between them can be utilized by using management tools and automated techniques. These tools have been the focus of the market as the need for managing large volume of requirements from various phases and different location are very important [16], [17]. These tools are used also for storing and recalling requirements, annotating requirements, and assigning relationships between them [13]. Indexing is usually used to utilize the search for requirements. Changes in requirements are very common and the need to evaluate these changes, planning for them and managing their effects throughout the entire system should be taken into consideration [18], [25].

Global requirement engineering issues should be addressed when managing requirements. Requirements are distributed in nature where the need for an awareness level regarding workplace implications on RE has always and will still be needed. Researchers have focused on addressing issues related to managing global and distributed requirements [19]. Requirements from large scale systems involve a considerable flow of requirements that need to be carefully managed and evaluated.

Quality of requirements can be improved using different requirement verification and validation techniques. However, it becomes a challenge to select among different techniques that best fit the requirement specifications. To overcome these problems, researcher has suggested a model that divides requirement specifications into executable and non-executable specifications. Non-executable specifications are written in natural language and requirements can be verified using Experimental Requirement Management (ERM) tool. Executable specifications are written in declarative languages such as java modeling languages and requirements can be verified through developing prototypes [29].

E. Evolving Requirements

Systems evolve over time and so do requirements. The dynamic nature of the environment makes it uncertain that requirements developed and modeled at early stages of the project will remain the same toward the end of the project. As a consequence, more changes are needed to be applied to requirements in which complexity of these requirements will increase overtime. Taking into consideration globalization, scalability and outsourcing, adaption to change should be structured and comprehensive to cover all aspects of RE and lead toward success of the project.

Researchers focus when managing requirements evolution on various interactions with the stakeholders, consistent adaption of change and the use of tools for presenting information and enhancing traceability and monitoring of

requirements [26]. Moreover, Empirical analysis of requirement evolution provides practical experience measures on drivers of evolution [27]. The study of requirement evolution is still in its empirical form and there is no huge advances in this regard [28].

III. RESEARCH IN REQUIREMENT ENGINEERING

The gap between research and practice in requirements engineering is huge. To understand practice and the reasons for this gap, empirical evidence about requirements engineering practice is needed. Most of discussion made so far is built on top of solution-based RE that usually includes proofs and evidences to the applicability of these solutions. On the other hand, Evaluation-based research that intends to evaluate current practices and advances is less common to be the subject of RE researches. In this section, we will demonstrate various research strategies and methodologies that are usually used in research in the software engineering field. By evaluating these methodologies, we shall have an idea on how these methodologies can be applied to RE. The aim of these methodologies is to understand current technologies in order to better benefit from them, evaluate and enhance these technologies to anticipate new changes and problems and the introduction of new methodologies that either better utilizes existing approaches or add new knowledge to the state of the art.

A. Research Methods and Methodologies

A large number of methods and methodologies exist to allow researchers and practitioners to determine the best choice of them. This choice is significantly important and has a great impact on the employment of these methods in requirement-related researches and activities. These methods are evaluated in order to obtain more knowledge about the context in which most of RE evaluation based research methods are built on.

These methods can be classified as follows:

1. Scientific: it aims at observing the world, propose a model or theory, analyze, and validate the model [32]. This method acquires knowledge about the world by either provide assumptions based on ontology (what we believe exists) and epistemology (how beliefs are acquired and how we justify them) [33]. This method usually used when we try to understand software process, product, people and the environment.
2. Engineering: it aims at observing existing solutions, study them and propose new better solutions that are analyzed, modified and tested enough until no further modification can be done on them [32]. Engineering research method usually used when the practice and implementation of technologies introduce a number of research problems that these methods try to simplify and make it easily adapted by practitioners. Heuristics, patterns, visual formalisms are examples on the engineering research method [3].
3. Empirical: it aims to understand, explore and describe natural, social or cognitive phenomena by using evidence

from observation or experience [33]. Empirical researches are evolutionary in nature in which new models are proposed not necessarily based on existing one and an attempt to study this model and its effects is carried out. Once the research methodology is chosen, suitable data collection techniques should be used. Usually, Qualitative and quantitative measures are used when collecting data needed for the research. For an empirical research to contribute to the solution of the problem it should prove its validity and hence can be measurable and interpretable. When validating an empirical research, the right quality of the research should be sought rather than maximizing its quality.

4. Paradigm shift: solutions that impose a radical change in new or existing problem in technology. These solutions are evolutionary and rare and usually carried out when existing technologies cannot make progress any more. Paradigm shift innovations need time before being adapted since a level of maturity should be reached before being widely spread and used. There are two classifications of paradigm shift: one that is driven by the introduction of new technology into community in which they have a great impact on resolving problems and called push paradigm shift. Global requirement engineering considered a push paradigm shift. The other classification is pulled when there is a huge need for solution to a problem when existing solution cannot resolve or even improve such problems. The emergence of object oriented approaches is considered a pull paradigm shift.
5. Generalization: usually, successful organization and domain research methods are generalized to be widely used and applied to different areas of concern. Telecommunication notations for example are used in a more general form using UML tools 2.0 [3].
6. Evolutionary: this field of research is concerned with the continuous improvements of existing technology. Existing requirements techniques, notations, approaches will continue to improve and also can be supported by new tools, patterns and methodologies. Evolutionary research is the basis for the paradigm shift methodology.

B. Evaluation of Empirical Research in RE

Evaluation-based research in RE is concerned with assessing the state of the art in requirement engineering and evaluate the practices developed in this regard. Unlike solution-based research that is concerned with the technological advances in solving requirement engineering problem [3]. Goeken and Patas [31] stated that our knowledge of the appropriateness and desirability of requirement engineering techniques are rare and existing work is not unified as we can see a lot of approaches and techniques that have been developed based on different principles and practices targeting the various phases of requirement engineering and specific tasks within them. This lack of empirical research leads to confusion as to whether their results contribute to the solution of a problem and whether it is

suitable for practical use. A recent evaluation to RE research criteria was referred to as a framework consisting of a set of components [31]. These components were chosen based on whether RE techniques evolved as a result of being published in a book, evolved from standard projects or being established as standards when used for certification. Main components used in this framework were classified into "Technique, Result, Role, Activity, Purpose, Notation, Principle, Tool and Context".

The results of evaluating RE research methods can be summarized as follows:

- Techniques and activities have been studied the most since most RE approaches are process or product driven.
- Notations have been studied less regardless the important role they play in requirement elicitation, documentation and validation of the system.
- Roles that are involved in assigning responsibilities are rarely studied.
- Context and principles also needs more attention since they are less considered in researches. Principles like agility and iterative approaches are less being studied in researches.
- Results as the research shows has a tight relation to both techniques and activities since results are always evaluated in experiments.
- Tools are frequently highlighted in research for their important usage in activities and results.

This shows that only about 10-15 % of the research work in RE is an evaluation based research that developed as a form of practices and state of the art reports. These empirical researches help in highlighting areas that need more attention and focus in requirement engineering research. In addition, they can help in the selection of the best solution based techniques for various RE problems.

IV. EMERGING TRENDS AND CHALLENGES IN RE

The rapid revolution of technology nowadays and its impact on the software industry is very tremendous. The need to adapt and live with these advances requires the software industry to evolve to meet these emerging needs. Existing trends may change or enhanced, new trends may develop and in a way or another will affect the software engineering process and hence have a direct and great impact on RE. The field of research in software engineering is very vast and covers a great deal of what we are talking about now. More attention should be given to the field of evaluation-based research as its importance is tremendous to the field of software engineering, and hence, RE. In this section, we will have an overview on some of the emerging trends in software engineering in the past ten years and its impact on RE, the subject of this paper.

A. Globalization

Globalization has quickly become a significant practice among various industries, the technological revolution in telecommunication and infrastructure make it easier for these

industries to expand their work worldwide. Software industry is affected by this and global software development emerged as a paradigm shift in the software industry. Different needs of stakeholders had a great affect toward globalization of software development as team members located in separate geographical workspace, mainly with different culture and skills. The need to develop new roles and organizational models to accommodate this diversity is huge. The decision of outsourcing specific tasks or in-house is crucial to the success of these projects.

Globalization affects RE in different ways. The distributed nature of requirements needs more effort to be gathered and managed [19]. Communicating these requirements among various distributed parties is challenging. Downstream activities such as design, implementation and testing requires new RE methodologies to support them.

Many challenges are facing RE in this distributed environment. First, since mostly analysts and development teams are located in different locations and geographical areas, requirements should be communicated in an efficient way. Different cultures, timing, and language should be taken into consideration. Poor defined requirements have a great impact on the final system as it will not match what stakeholders want and the cost of re-work will increase. Second, tools needed for eliciting, supporting and modeling distributed requirements, managing and communicating them, negotiating and quality should be developed and gain more attention and effort to comply with this huge shift. The introduction of interactive digital surfaces has offered some solutions to the communication problems, but much needs to be done in employing such technology in the RE context. Other challenges are in the area of distributed coordination and collaboration. Conflicts can arise due to distance and poor collaboration among various team members. Engineers may be informally involved in change and communicating this change across multiple sites may be absent. Finally, with this huge pool of people working, not only new roles need to be created, but also assigning responsibilities to right people could be of a great challenge [34].

B. Scale

Large scale systems traditionally characterized by a large number of variables mainly uncertainty and nonlinearity. Dividing these systems into smaller controllable parts used to be in a form of hierarchy. Control systems for example that are large in nature are supported with current advances in technology and communications. Software industry is growing as most of companies are growing in size. Large scale software systems are not only characterized by number of line of codes but extended to include number of people employing the system for different purposes, amount of data stored, accessed, manipulated, and refined, number of connections and interdependencies among software components [2].

RE in large scale systems is not an easy task. Many challenges are facing requirement engineering due to the large number of people and distributed teams, assigned roles and

responsibilities, and communication with stakeholders. Scalability of huge number of requirements unfortunately is not supported by current tools, techniques and models since they are difficult to be applied and do not support the complexity, variability and uncertainty of requirements. In addition, existing requirement methodologies for requirement specifications are costly and unlikely to be adapted to large scale requirements. Analyzing, designing, communicating and managing requirements in large scale systems should be supported in new emerged methodologies and trends. Large scale systems increases the reliance on the environment, people, hardware and software are tightly coupled that led to the emergent of new integrated systems and techniques.

A substantial research is carried out in the area of RE, but the amount of research regarding RE in large scale systems is not satisfying, practical advices and guidelines on how we can overcome these challenges should be studied and documented. Continuous requirement engineering and process improvement, training and documentation, cultural change is essential for improving our experience in RE for large scale systems.

C. Dynamicity

The main work of RE is concerned with static elicitation, representation, and analysis of requirements. Even handling issues related to change is made under the context of static requirements. Advances in technologies and systems that are highly adaptive and dynamic with high rate of change incorporate requirements to adapt and change dynamically as well [35]. Research in the area of dynamic requirement engineering is few and more attention of research evaluation should be adapted to this newly emerging trend as most systems are developed toward this trend.

One of the remarkable works on this field was carried out by Berry et al. [36]. They suggested four levels of RE activities needed to support adaptive systems:

1. Level one, associated with human and related to activities associated with elicitation and analysis of requirements.
2. Level two, associated with the system when it is running and its ability to adapt according to various inputs and behaviors.
3. Level three, associated with human and related to the previous level by determining the adaption elements of the adaptive system.
4. Level four, also done by human and associated with the general understanding of adaptive systems and events.

D. Aspect-Oriented RE

Software engineering accommodated aspect-oriented advances and requirement engineering architecture design and design are affected with this topic. Aspect oriented techniques provide a systematic means for identification, modularization, representation, and composition of crosscutting concerns [37]. Chitchyan and Rashid [38] conducted a survey on aspect-oriented analysis and design approaches. Although a set of criteria has been identified, a precise definition of some of the

criteria used to evaluate the approach is missing. Also, a set of models and use cases were developed such as AOV-graphs and aspect-oriented uses cases.

E. Scenario-Based RE

Scenarios and use cases are the most useful tools used in requirement elicitation, validation and documentation activities. A scenario is a series of interactions between the user and the system whereas use cases are an abstraction that describes these scenarios [40]. According to its increased importance in modeling, design, validation, obstacle analysis, scenario-based requirement engineering has emerged to employ the effectiveness of scenarios in bridging the gap in requirement resulting from model and goal oriented design. Many researches have conducted in an effort to define scenario-based requirement engineering, but no agreed formal definition has been introduced. Scenario-based requirement engineering can be defined as an approach that represents requirements based on a real world example or story. A lot of methodologies and tools have emerged to formalize scenario-based requirement engineering, Natural language, pictures, use cases, behavior and sequence diagrams are all examples on these models.

F. Service-Oriented RE

As advances in technology nowadays lead the direction of business investment, more attention is given nowadays to service-oriented technology. Service-Oriented architecture is a collection of services that communicate with each other delivering service or value to subscribers over the internet. As more and more service-oriented projects are invested in, the need to clearly understand and identify how requirements of such services can be developed. Although a lot of research has been made in the area of service-oriented models and techniques, a little attention was given to related requirement engineering activities [41].

Having service-oriented in mind, requirement activities had to put business point of view before considering the functionality aspect of the system and has to lead the development efforts according to Flores et al. [42]. Researchers outlined that requirements are affected by both service providers and customers. On one hand, providers will try to design services that can suit multiple customers but their effort to standardize services will not suit all customers. On the other hand, customers are looking for services that accommodate their needs and based on service providers, they will have to change their processes to suit these services and therefore requirement engineering must be capable to comply with all these issues. The research of service-oriented requirement engineering is promising and can open the door for a lot of both solution-based and evaluation-based research.

G. Requirement Reuse

The reuse of existing and past requirements artifacts for future usage is a common trend nowadays. The tendency to reuse artifacts from past software development efforts such as

portion of code, test cases design and alike is common and so do requirements. The most reused elements nowadays are frameworks and components. In order to reuse requirements, they should be developed in the most abstract form. Most of the reuse is done by engineers who developed similar products before in an informal way. Their experience allows them to reuse these requirements informally. Frameworks usually are domain specific, providing common functionalities to a group of applications. A number of different frameworks usually used when building complex applications, each with its own requirements. In order for a successful reuse of requirements across these frameworks, requirements should be well defined and managed and interaction among reused and new requirements should be defined well.

Most of the research in the area of reuse requirements engineering was in the product lining where group of products are treated as one component and requirements can be derived from all products for an individual product. Although it is argued that requirements reuse can introduce even more reusability at later stages in the product life cycle, it was not until recently that requirements reuse has received greater attention by researchers [39]. Researchers [44] outlined that for an artifact to be reused it should have a standard pattern such as context, domain, problem, properties and consequences. Patterns are usually used as a reusable artifacts but more attention should be given from researchers and practitioners on how we can adapt and employ these patterns.

H. Commercial of the Shelf

Commercial of the Shelf (COTS) products are software available in the market that can be reused and integrated throughout the system. These products or components are developed by vendors to target various customers and therefore does not necessarily accommodate all the needs of a single customer. Requirement engineering in this regard is challenged due to huge modifications and changes made on them over the time that are not in the hand of the customer and therefore the need by analyst to understand these components and changes made on them in order to determine the degree to which this component will still satisfy the needs of the client [43]. Moreover, risk analysis should be made in order to assess the degree of risk associated with these changes.

Many researches have been made in the area of requirement engineering and COTS, and initial results showed that there is no need to start with requirement gathering and then selecting the appropriate COTS to meet these requirement since no product will satisfy all the needs of a customer. In fact, requirements should be abstract and capable of determining which market to search for the COTS without specifying internal details and quality measurements. Once a market is identified, differences between these components will be identified and used to choose the best product that will maximize customer satisfaction. More research should be taken in this regard for the challenges that analyst faces when COTS are selected and integrated in the system.

V. CONCLUSION AND FUTURE WORK

In this paper, an overview on requirement engineering evolution over the past ten years is presented. RE challenges, research directions and evolving trends were also evaluated. Many of the research work and practical activities evolved over time due to changes and paradigm shifts in technology and practice in software engineering in general, and RE, in particular.

Huge effort is needed toward a collaborative work between researchers and practitioners in order to take RE into a further step to aid in confronting these changes in the software industry.

Below, are some suggestions to take into consideration for future work in RE:

1. Requirement engineering field is very huge, the need to have some sort of centralized repository or index to the work in this field can help researchers utilize research time in evaluating work and find solutions.
2. The gap between researchers and practitioners should be minimized. The need for collaborative efforts between them will help researchers obtain a better understanding of problems and challenges. Real data and cases provided by customers can help researchers in clarifying various data impacts on problems to be solved.
3. Lots of solutions and models have been developed addressing specific contexts and scenarios, sharing this knowledge through standardized channels and repositories will be beneficiary.
4. More attention should be given to evaluation-based and empirical research.
5. Technological advances and new trends are leading the change now. Requirement engineering activities and methodologies should evolve too to comply with these changes.

In conclusion, requirements engineering facing successive changes and advances that are affecting it and making it susceptible to change all the time. An attempt to sum up most of these changes has been accomplished in this paper but still lots of work need to be done in this regard. The research opportunities in this field are drivers and huge and more work can be done since the reliance on technology and software will remain for a quite long time.

REFERENCES

- [1] B.Nuseibeh and S.Easterbrook, "Requirements Engineering: A Roadmap", ACM, New York, USA, 2000.
- [2] M.Qadir, M.Asghar and S.Ghayyur, "Scaling of Critical success factors for Requirements engineering in the development of Large Scale Systems", International Journal of Reviews in Computing, Pakistan, 2009.
- [3] B.Cheng and J.Atle, "Research Directions in Requirements Engineering", IEEE Computer Society, Washington, DC, USA, 2007.
- [4] Requirement Elicitation definition, Wikipedia, URL: http://en.wikipedia.org/wiki/Requirements_elicitation.
- [5] Beaver consulting, "White Paper: A Primer on Requirements Engineering Introducing Beavers Requirements Engineering Services Capability", Beaver Computer Consultants Ltd, UK, 2003, Vol. 0.4.

- [6] E.Insfran, O.Pastor and R.Wieringa, "Requirements Engineering-Based Conceptual Modeling", Springer-Verlag London Ltd., United Kingdom, 2002, pp. 61-72.
- [7] K.Ryndina, "Improving Requirement Engineering: An Enhanced Requirements Modeling and Analysis Method", 2005.
- [8] J.Ralyte, "Reusing Scenario Based Approaches in Requirement Engineering Methods: CREWS Method Base", IEEE Computer Society, Washington, DC, USA, 10th International Workshop on Database & Expert Systems Applications, 2003.
- [9] A.Matoussi and R.Laleau, "A Survey of Non Functional Requirements in Software Development Process", University of Paris, Paris, France, 2008.
- [10] V.Mayank, N.Kositsyna and M.Austin, "Requirements Engineering and the Semantic Web, Part II. Representaion, Management, and Validation of Requirements and System-Level Architectures", University of Maryland, USA, 2004.
- [11] T.Breaux, A.Anton and J.Doyle, "Semantic Parameterization: A Process for Modeling Domain Descriptions", ACM Transactions on Software Engineering and Methodology, New York, USA, 2008.
- [12] R.Machado, I.Ramos and J.Fernandes, "Specification of Requirements Models", 2004.
- [13] J.Dag and G.Gervasi, "Managing Large Repositories of Natural Language Requirements", 2005.
- [14] B.Palyagar and D.Richrads, "Communication Protocol for Requirements Engineering Processes", Macquarie University, NSW, Australia, 2004.
- [15] S.Cherry and P.Robillard, "Communication Problems in Global Software Development: Spotlight on a New Field of Investigation", Montreal School of Applied Arts, Montreal, Canada, 2004.
- [16] S.Heinonen, "Requirements management tool support for software engineering in collaboration", University of Oulu, Oulu, Finland, 2006.
- [17] D.Zoqghi and C.Coulin, "Requirements elicitation: A survey of techniques, approaches, and tools", 2004.
- [18] K.Weigers and S.McKensey, "Serena: Requirements management, a proven way to accelerate development", Serena Software, Inc., California, USA, 2005.
- [19] D.Damian, J.Chisan, P.Allen and B.Corrie, "Awareness meets requirements management: awareness needs in global software development", 2003.
- [20] B.Palyagar and F.Moisiadis, "Validating Requirements Engineering Process Improvements – A Case Study", IEEE Requirements Engineering Visualization, 2006. REV '06. First International Workshop on Requirements engineering visualization, Washington, DC, USA, 2006.
- [21] J. Magee, N. Pryce, D. Giannakopoulou and J. Kramer, "Graphical Animation of Behavior Models", 22nd International Conference on Software Engineering (ICSE '00), Limerick, Ireland, 2000.
- [22] E. Kazmierczak, M. Winikoff and P. Dart, "Verifying Model Oriented Specifications through Animation", IEEE, Washington, DC, USA, 1998.
- [23] R. Schmid, J. Ryser, S. Berner and M. Glinz, "A Survey of Simulation Tools for Requirements Engineering", 2000.
- [24] G. Torenvliet, A. Hilliard, C. Burns, G. Lintern and JY. Lamarre, "Modelling & Simulation for Requirements Engineering and Options Analysis", Defence R&D Canada – Toronto, Toronto, Canada, 2010.
- [25] P. Jönsson and M. Lindvall, "Impact Analysis", 2004.
- [26] M. Lormans, "Managing requirements evolution using reconstructed traceability and requirements views", 2009.
- [27] S. Anderson and M. Felici, "Quantitative Aspects of Requirements Evolution", IEEE, Computer Software and Applications Conference, 2002 Proceedings 26th Annual International, Washington, DC, USA, 2002.
- [28] M. Felici, "Observational Models of Requirements Evolution", University of Edinburgh, Edinburgh, United Kingdom, 2004.
- [29] S.Asghar and M.Umar, "Requirement Engineering Challenges in Development of Software Applications and Selection of Customer-off-the-Shelf (COTS) Components", CSC Journals, KL, Malaysia, 2010.
- [30] J. Marques-Silva, "Boolean Satisfiability & Model Checking", University of Southampton, Southampton, United Kingdom, 2008.
- [31] M.Goeken and J.Patas, "Evidence-Based Structuring and Evaluation of Empirical Research in Requirements Engineering", Gabler Verlag, Wiesbaden, Germany, 2010.
- [32] V. Basili, "The experimental paradigm in software engineering", Springer, Berlin/Heidelberg, Germany, volume 706, 1993.
- [33] D. Sjoberg, T. Dyba and M. Jorgensen, "The Future of Empirical Methods in Software Engineering Research", IEEE Computer Society, Washington, DC, USA, 2007.
- [34] J. Herbsleb, A. Mockus, T. Finholt and R. Grinter, "An Empirical Study of Global Software Development: Distance and Speed", IEEE Computer Society ICSE '01 Proceedings of the 23rd International Conference on Software Engineering, Washington, USA, 2001.
- [35] J. Herbsleb, A. Mockus, T. Finholt and R. Grinter, "An Empirical Study of Global Software Development: Distance and Speed", ACM, New York, USA, 2001.
- [36] D. Berry, B. Cheng and J. Zhang, "The Four Levels of Requirements engineering for and in dynamic adaptive systems", 2005.
- [37] R. Chitchyan, A. Rashid, P. Sawyer and A. Garcia, "Aspect-Oriented Requirements, Architecture, and Design approaches", 2005.
- [38] R. Chitchyan and A. Rashid, "Survey of Aspect-Oriented Analysis and Design Approaches", Lancaster University, Lancaster, United Kingdom, 2005.
- [39] O. Villegas and M. Laguna, "Requirements Reuse for Software Development", 2001.
- [40] X. Sun, "A Model-Driven Approach to Scenario-Based Requirements Engineering", 2007.
- [41] S. Lichtenstein, L. Nguyen and A. Hunter, "Issues in IT Service-oriented Requirements Engineering", AWRE'04 9th Australian Workshop on Requirements Engineering, 2004.
- [42] F. Flores, M. Mora, F. Alvarez, L. Garza and H. Duran, "Towards a Systematic Service-oriented Requirements Engineering Process (S-SoRE)", Springer, Berlin/Heidelberg, Germany, Volume 109, 2010.
- [43] C. Ebert and R. Wieringa, "Requirements Engineering: Solutions and Trends", Springer, Berlin/Heidelberg, Germany, pp. 453-476, 2005.
- [44] Issa, Ayman, and Al-Ali, Amani, (2011). Automated Requirements Engineering: Use Case Patterns Driven Approach. IET-Software, IET, 5(3), pp.287-303.