

Requirement Engineering: A Review

Bhawna¹, Sanjay Kumar², Amita Dhankar³

¹Student M.Tech., U.I.E.T., M.D.U., Rohtak

²Asstt. Prof., VCE, Rohtak

³Asstt. Prof., U.I.E.T., M.D.U., Rohtak

Abstract

Requirement Engineering acts as foundation for any software and is one of the most important tasks. Entire software is supported by four pillars of requirement engineering processes.

Goal-oriented requirements engineering is concerned with the use of goals for eliciting, elaborating, structuring, specifying, analyzing, negotiating, documenting, and modifying requirements. This area has received increasing attention over the past few years.

The paper has highlighted requirement engineering challenges encountered in development of software applications. Comprehending stakeholder's needs; incomplete and inconsistent process description; verification and validation of requirements; classification and modeling of extensive data; are foremost challenges faced during requirement engineering. Recent attempts by the RE research community to address this issue include cumulative knowledge studies to describe the current state of RE research and the establishment of a new conference dedicated to the comparative evaluation of RE. During the period 2001-2005 there was only an incremental development of RE research without any radical theoretical contributions to its body of knowledge. The paper also poses a challenge for the RE research community to respond to the dramatic changes in the social and business world.

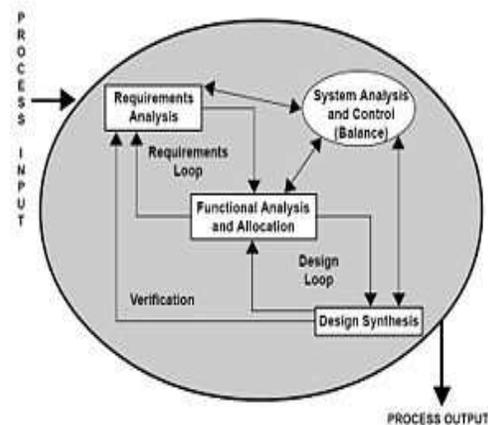
Keywords: Requirement Engineering, Tools of Requirement engineering

1. Introduction

Requirements engineering (RE) is a systems and software engineering process which covers all of the activities involved in discovering, documenting and maintaining a set of requirements for a computer-based system. While there are differing definitions of the term, common factors are that requirements engineering is a sub discipline of systems and software engineering and is concerned with establishing the goals, functions and constraints of hardware and software systems.

The first use of the term 'requirements engineering' was probably in 1979 in a TRW technical report but did not come into general use until the 1990s with the publication of an IEEE Computer Society tutorial and the establishment of a conference series on requirements engineering.

In the traditional waterfall model of the systems or software engineering process, requirements engineering is presented as the first stage of the development process, with the outcome being a requirements document or Software requirements specification. In fact, requirements engineering is a process that continues through the lifetime of a system as the requirements are subject to change and new requirements must be elicited and documented and existing requirements managed over the lifetime of the system.

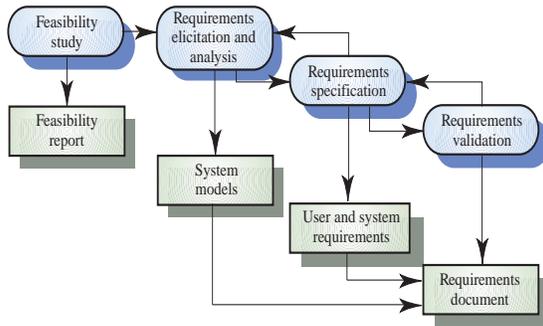


2. The requirements analysis process

2.1 Requirement Engineering Process

Processes used to discover, analyses and validate system requirements. The processes used for RE

vary widely depending on the application domain, the people involved and the organization developing the requirements. However, there are a number of generic activities common to all processes - Requirements elicitation, Requirements analysis, Requirements validation and Requirements management.



2.3 Requirement Engineering Issues

The general difficulties involved with requirements

Analysis is increasingly well known:

- The right people with adequate experience, technical expertise, and language skills may not be available to lead the requirements engineering activities.
- The initial ideas about what is needed are often incomplete, wildly optimistic, and firmly entrenched in the minds of the people leading the acquisition process.
- The difficulty of using the complex tools and diverse methods associated with requirements gathering may negate the hoped for benefits of a complete and detailed approach.

Stakeholder Issues

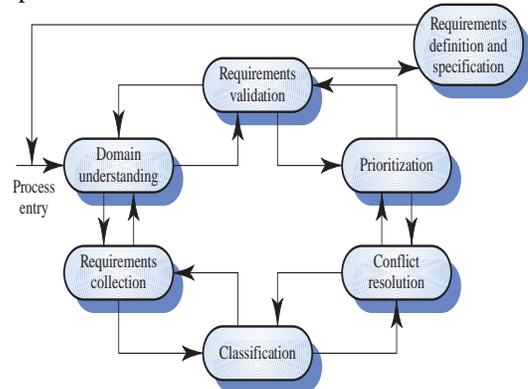
- Users don't understand what they want.
- Users won't commit to a set of written requirements.
- Users insist on new requirements after the cost and schedule have been fixed.
- Communication with users is slow.
- Users often do not participate in reviews or are incapable of doing so.
- Users are technically unsophisticated.
- Users don't understand the development process.

2.2 Engineer/ Developer Issues

- Technical personnel and end users often have different vocabularies. Consequently, they can believe they are in perfect agreement until the finished product is supplied. In business systems domain, the duty to bridge that gap is often assigned to Business Analysts, who analyze and document the business processes of business units affected by the proposed business solution, and Business Systems. Analysts, who analyze and document the proposed business solution from a systems perspective.
- Engineers and developers often try to make the requirements fit an existing system or model, rather than develop a system specific to the needs of the client.
- Analysis is often carried out by engineers or programmers, rather than personnel with the people skills and the domain knowledge to understand a client's needs properly.

They include the following.

- Tasks that must be completed: elicitation of information from clients, validation, specification



- Problems that must be solved: barriers to communication, incompleteness, inconsistency
- Solutions to problems: formal languages and analysis algorithms, prototyping, metrics, traceability
- Ways of contributing to knowledge: descriptions of current practice, case studies, controlled experiments; and
- Types of system: embedded systems, safety-critical systems, distributed systems.

Requirement Elicitation Techniques:-

Several techniques may be employed in order to gain such an understanding, for example, we may use

- **Interviewing**
- **Questionnaires**
- **Focus Groups**
- **Direct Observation**
- **Apprenticing**
- **Brainstorming**
- **Joint Applications Development**
- **Prototyping**
- **Use Case Modeling.**

3. Requirement Engineering Tools:-

The RE-Tools is a model-based modeling tool. All supported notations can be used together in an integrated manner, which means functional and non-functional requirements, agents, goals, soft goals, formal goals, and objects can be related as needed.

1. **CASE Spec:-** CASE Spec can be used for any software, systems or product development projects, and with any process, including Agile, Incremental, Waterfall, Spiral, etc. CASE Spec is a flexible, problem-solving tool that enables you to solve your simple or complex problems effectively.
2. **DOORS :-** DOORS is a requirements management tool for systems and advanced IT applications
3. **IBM Rational Rose Tools:-** Rose Data Modeler Accelerates database design by providing a sophisticated visual modeling environment Rose Developer for Java Model-Driven Development for Java and J2EE Rose Developer for UNIX Provides the industry-leading model-driven development tool
4. **Optimal Trace:-** Compuware Optimal Trace is a requirements management tool (noted on their website in 2006)that enables IT and the business to collaborate more effectively and improve delivery outcomes

using a requirements specification approach that it calls "structured requirements." This approach captures software requirements from the perspective of the user, complete with visual storyboards and traceable relationships to business needs. Using structured requirements, IT organizations ensure that they are capturing the right requirements, communicating them effectively and dramatically improving their ability to deliver on the expectations of the business.

5. **RTIME Tools:-** RTIME Features Include: Requirements Management - Document and track requirements - Visual Capture, Elaborate and Collaboration with Mind Manager Integration - Attach images, documents and URLs to Requirements - Allow external contributors to submit requirements with RTIME Community - Prioritization, release and iteration planning and trade-off analysis tools.

Conclusion:-

"A Problem well stated is half resolved"

This research will give a conceptual view of the requirement engineering and its types. It is important to verify that software applications implement their specifications correctly. However, do these specifications meet the software requirements? Do these requirements meet the system's goals, and under realistic assumptions? Are these goals, requirements, and assumptions complete, adequate, and consistent? Requirements engineering can be decomposed into the activities of requirements elicitation, specification, and validation. Most of the requirements techniques and tools today focus on specification, i.e., the representation of the requirements. A multi-view model integrating system goals, objects, agents, operations, and behaviors is a key artifact for articulating requirements elicitation, evaluation, specification, consolidation, and evolution. It will also give the industrial perspective toward requirement engineering process.

References:-

- [1] Bashar Nuseibeh Steve Easterbro
Department of Computing Imperial College
University of Toronto 180 Queen's Gate
London SW7 2BZ, U.K.

- [2] The Slow Wheels of Requirements Engineering Research: Responding to the Challenge of Societal Change.
- [3] Requirement Engineering Challenges in Development of Software Applications and Selection of Customer-off-the-Shelf (COTS) Components
- [4] Goal-Oriented Requirements Engineering: A Guided Tour
- [5] Critical Requirements Engineering Issues & their Solution
- [6] Classification of Research Efforts in Requirements Engineering PAMELA ZAVE
- [7] Requirements Engineering by E. Kazmierczak
<http://ww2.cs.mu.oz.au/~dmwilm/downloads/641.pdf>