**Requirement Discovery**

*As a requirements engineering Process in software engineering*

**Research Topic by:**

[**Hussam Tarazi**](https://www.researchgate.net/profile/Hussam_Tarazi)

King Saud University

29-5-2016

Table of Contents

|  |  |
| --- | --- |
| Title | Page |
| Introduction | 1 |
| What is Requirements Discovery | 2 |
| Requirements discovery process | 3 |
| Requirements discovery techniques | 4-9 |
| High Level Techniques | 6,7,8 |
| Low Level Techniques | 8,9 |
| Common difficulties in Requirements discovery | 9,10 |
| References | 11 |

**Introduction**

Software engineering requirements gathering is a cyclic or continuous process which may involve many steps and procedures within the business analysis activities, in software engineering, requirement gathering is the process of collecting software requirements from stakeholders, and requirements usually are functional and nonfunctional depending on the stakeholder needs.

Requirements engineering is a system and business analysis mechanisms and functionalities which may include the following continuous steps: Requirements -elicitation, analysis and negotiation, specification, system modeling, validation and requirements management.

Our studied concept lies in the first step of requirements engineering which is requirements elicitation, and discovery is the initial starting part.

Requirement discovery is the first activity in the requirements engineering life cycle [1], which is usually done by the system or/and business analyst roles in the acting company or organization.

According to Sommerville [2] it involves technical staff working with customers to find out about the Application domain, the services that the system should provide and the system’s operational constraints that may involve end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc. These are called stakeholders.

We will discuss more about requirements discovery techniques and processes and the problems and difficulties facing requirements discovery.

1

**What is Requirements Discovery**

Requirement discovery is the act of collecting information about the target or existing systems and getting the user and system requirements from this information.

Sources of information include documentation, system stakeholders and the specifications of similar systems [3].

In other words, we can say that it is the process in which the system/business analysts and stakeholders discover and understand and document the software requirements.

The aim of requirements discovery is to identify as many requirements as possible to prepare solutions for the stated problem.

Early correct requirement discovery and analysis helps building the right requirement specifications which lead to successfully developing the expected software solution and this may incredibly reduce the future costs and efforts for maintaining and re-analyzing the solution.

There are many software engineering practices and procedures relies basically on the requirements discovery process, and requirements discovery can be conducted through different ways and techniques and those techniques depends on the target software solution, not all the techniques success with all software types, therefore, domain experts and previous experiences play a significant role in deciding the problem scope; through cooperation with business and system analysts and also software engineers involved in the project.

Requirements discovery is a recent and important knowledge field and it is being researched and studied increasingly within organizations.

2

**Requirements Discovery Process**

The purpose of discovery of requirements is to get knowledge and general view about the problem and analyze the problem domain [3]; the problem domain is the starting point where a system analyst set system specifications and scenarios from.

The domain knowledge of the target system is an essential part for entering the next requirement engineering processes; therefore, many sources of knowledge can feed the knowledge domain like: Domain experts, literature about the domain, existing software systems and/or applications in the domain, international standards, and other stakeholders in another system or in a larger hosting system.

This process steps can be simply summarized in the following figure below:

**3**

**1**

3

**4**

**2**

Figure 1: knowledge Domain cycle

**Requirements Discovery Techniques**

According to SEI (Software engineering Institute) guides and notes [4]. Requirements discovery techniques can be divided to two categories: high-level, such as frameworks for Discovering general requirements; and low-level, which provides

specific procedures and tactics for discovering details about a specific part of a system or user.

General requirements discovery techniques (from SEI notes):

**Asking**: ask the right person (User or buyer of the software) what the requirements are.

**Observing and inferring:** Observe behavior of system users then infer their needs from their behavior.

**Discussing and formulating**: Discuss users’ needs and form a common understanding of the requirements.

**Negotiating with respect to a standard set**. Beginning with a standard list of requirements, and discuss with users which of those requirements and functionalities will be needed or required or needs modifying.

**Studying and identifying problems**. Perform investigations of problems to identify

requirements for improving a system.

**Discovering through creative processes**. For very complex requirements, there must be a discussion between users and developers.

**Postulating**. Assuming users’ needs and system features when there is no access for the users.

A software engineer has to choose the best set of techniques that match the type of

The system.

4

And in another point of view there are a set of basic good practices summarized from Ian sommerville book “Requirements Engineering: A good Practice Guide” [6] those tips can be used in requirements engineering tasks in general:

* Study system feasibility.
* Follow organizational and political considerations.
* Identify system stakeholders
* Record requirements sources
* Study system’s environment.
* Follow business procedures to conduct requirements discovery.

The Top 10 requirements engineering practices by SEI:

* Prepare a standard document structure.
* Make the document modifiable.
* Separately identify each requirement.
* Set policies for later requirements management.
* prepare standard templates for requirements description.
* Use simple and related language.
  + Make formal requirements reviews.
  + Prepare validation checklists.
  + Make use of checklists for requirements analysis.
* Study expected conflicts.

5

**High-Level Techniques**

High-level requirements discovery techniques are general frameworks for requirements discovery and analysis and other practices in requirements engineering.

1. **Joint Application Design**

Joint Application Design (JAD) is a technique for enhancing cooperation, understanding, and teamwork within users, buyers, and developers. Therefore, it should support defining a united vision of what the system should be.

Through JAD, developers help users specify problems and suggest solutions, and the users get a feeling of involvement, ownership, and commitment to the success of the system.

According to software engineering institute, there are four main aspects of JAD:

* group dynamics (using groups to enhance user’s cooperation).
* the use of visual aids to enhance communication and understanding;
* maintaining an organized, rational process;
* “what you see is what you get” documentation philosophy (using standard document forms that are filled in and endorsed by all participants in a session).

JAD has two basic steps, JAD/Plan and JAD/Design. The first step involves

requirements discovery and specification, and the second is about software design.

Each step in JAD consists of three stages: customization, session, and wrap-up [4].

* The customization stage focus on preparation tasks for the session.
* The session phase is mainly about making meetings with the developers and users together, during the meetings the requirements should be developed and documented.

6

* The wrap-up stage is about converting the information collected from the session stage into requirements specification document, like SRS (software requirements sheets).

By making meetings for users and developers together, JAD enhances communication among users and developers. by using standard document forms, JAD also addresses the limitations and conflicts between user roles and solve those conflicts.

1. **Adaptive Loops Framework**

The adaptive loops framework is similar to JAD by applying a process framework that links the users, developers, and system. “It derives its name from the idea that the users’ requirements can be elicited by an adaptive process of learning cycles or loops” [4], according to SEI.

There are three learning cycles:

* Users assess developers in getting new viewpoints about requirements, so that users can learn more about their requirements by re-forming the req.
* The system receives feedback while being evolved, so that users can learn more about the system.
* The system evolves by developer’s actions, so that developers get enhanced understanding of the system.

The requirements discovery process using the adaptive loops framework consists of

addressing, supporting, and facilitating these three learning cycles.

1. **Prototyping**

Users may be able to better show up their requirements when they see and use similar existing system or sample, when there is no similar existing system, prototyping can be used to create a system that provide the discussed features.

7

The prototyping process start with an initial study of user requirements. then

Developers build a prototype and evaluate it with the users in iterations. [4][5]

Each prototype iteration adds an understanding stage for users.

Finally, a set of final requirements can be set and then the prototype will be ended.

1. **Critical Success Factors Analysis**

This technique is about studying the effectiveness of a system by assuming that it depends on a list of critical success factors.

Therefore, developers and analysts focus on enhancing performance of the system depending on these factors.

This process has six major steps:

1. Understand system operations.

2. Identify critical effective and functional factors.

3. Asses weakness and strength points that may come in place when depending on these factors.

4. Identify problems and conflicts if exists.

5. collect enhanced relevant details for empowering the system using those factors.

6. generate requirements using these details.

This technique is commonly used in decision support and information systems.

**Low-Level Techniques**

mainly about providing operational-level procedures, by focusing on specific and detailed aspects of a system.

1. **Brainstorming**

For software requirements discovery, brainstorming can be useful in generating multiple kind of views of the problem. Especially in the early stage of discovery process.

8

1. **Interviewing**

It is usually used for discovering large systems software requirements. Also can be used alone for small systems as requirements discovery technique.

1. **The PIECES Framework**

Commonly uses when requirements engineer or analysts are not well experienced, so they rely on a standard list of basic checklists to extract user needs, it helps in overcoming existing systems.

**Common difficulties in Requirements discovery**

Lots of weaknesses in the process of discovering requirements were found. According to Ronald Kirk Kandt JPL document [6].

We can summarize them in the following list:

* No relevant knowledge of some of the stakeholders about some aspects of technology or being unable to understand some technology fields.
* Managers and Requirements engineers don’t have a thorough view of how to prevent unneeded software requirements and features, so that this may lead to extra costs added for the estimated budget.
* Human behavior conflicts and mistakes and human to human communication problems, like personal relations, it may positively or negatively affect the overall process of requirements discovery if there exists a personal or social relations between employees of the software solution company and even employees of the target company ordering the solution.
* Subsystems requirements maybe unstudied well by requirements analysts.
* System design is being not traced well to the requirements.

9

* Improper selection of the requirements engineering techniques and this may lead to a delayed project development and also may add more costs to the budget.
* Wrongly translate user’s suggestions and opinions from the client side to the developers by requirements engineers.
* Inability to trace and fix bugs in an already existing system which is being used as a sample or prototype for users.
* Lack or weakness of cooperation between business analysts and system analysts of the software solution company.

10

**References**

1. Requirements Engineering Tasks, Donald Firesmith, software engineering institute, USA, Published by ETH Zurich, Chair of Software Engineering ©JOT, 2006 in the JOURNAL OF OBJECT TECHNOLOGY.
2. Software Engineering, 7th Edition by Ian Sommerville.
3. Software Requirements Engineering: Practices and Techniques, Ronald Kirk Kandt, JPL Document D-24994.
4. Lecture Notes on Requirements Elicitation, Software engineering institute, SridharRaghavan Gregory Zelesnik, Gary Ford, Carnegie Mellon University.
5. Software Requirements Engineering: Practices and Techniques, Ronald Kirk Kandt, JPL Document D-24994.
6. Requirements Engineering: A Good Practice Guide, Ian Sommerville.

11