

# RE Flow With Prototype & Its Impact with End- user in System Development

Akshara Dave<sup>1</sup>, Dr. S.M. Shah<sup>2</sup>

<sup>1</sup>CMPICA, CHARUSAT Campus,  
Off Nadiad Petlad Highway 139, Changa, Gujarat, India  
akshara.dave@gmail.com

<sup>2</sup>NICSM, Kadi Sarva Vishwavidhyalaya,  
Kadi, Gujarat, India

**Abstract:** This paper generally focuses on different aspects of techniques related to requirement gathering as well as why and how these techniques may help stakeholder and the system for development. The main entity in SDLC is a customer/ stake holder. As the customer is the one for whom any project is built up or any IT company works, it is mandatory to keep them satisfied by providing them accurate solution of their problem. The paper is about the explanation of different majorly used requirement gathering techniques and their importance. From this paper, one can get idea which technique is suitable in various environments.

**Keywords:** requirement engineering, Software engineering, Requirements gathering techniques, End user

## 1. Introduction

Requirement gathering is a technique for having the user needs to sort out different issues and achieve a particular goal.

In the SDLC process, requirement gathering is the most basic and most important step. If the first step is taken wrongly, whole process afterwards lead to improper directions based on that first step taken incorrectly.

Thus, requirement gathering must be done following a particular systematic approach. Difficulties are generally based on social issues not with the systematic approach issues.

Requirement management generally follows this sequence:

- Requirement Gathering
- Requirement Elicitation
- Requirement Management Plan
- Requirement Traceability
- Change control.

## 2. Reasons for Project failures

A Standish group of research report says that 31% of all the projects are canceled before they get completed and 53% of projects almost cost double than their estimated cost [1]. The reasons are:

- Inability of user / stakeholder to understand their own priorities
- Lack of communication skills among stakeholders.
- Developers may face difficulty mediating user requirements into technical, machine understandable language.
- Continuous changing in both the business world and technology world.
- Changing in requirements too often from stakeholders.

## 3. Challenges

### 3.1 Unclear scope and vision

SRS (Software Requirement Specification) is made at the very beginning before development. Generally, the scope of the system is decided when it been thought to get developed. Vision is decided once the modulation work starts. If the modulation of project is done by 10 % ( on a very beginning level), and the teams decide to overdue time limit, finance or human efforts etc. That would lead to passive change in SDLC of particular project. Similar situation occurs if suddenly team is forced to lower the time limit. It may be possible that project might not get delivered completely. In such situation, feasible solution is to cut down some of the unnecessary functionality of the system. This is called to enlarge the scope of system. As time limit decreased, scope get widen due to inability of development and deliver on time. Main point is: If the scope is not clear, vision cannot be built.

### 3.2 Requirements are defined without priorities

In general, stakeholder simply fills up the form given to him/her to fulfill his requirements for development of the system. If he/she needs something extra on the project that is not mentioned in form given to him, they won't keep it in knowledge of the team. In some special cases, stakeholders won't mention the important and special task they want and will note down simple things they want. For example, In general , the customer want to get a brand new car customized, he/she will mention - "Car must have four wheels, one steering, a well working break, clutch and accelerator along with seats" instead of mentioning special features he/she is looking for like "power window, air bags, parking sensors, automatic gear system etc".

### 3.3 Updating is done after the project has been started modeled.

After the meeting of development team and stakeholders, project has been started to build up. Team is constantly in touch with stakeholders showing them proper inputs and keeping them aware with output generated. As stakeholders are generally interested in output, it is not necessary to show them process that developer team has taken up. Stakeholders are not satisfied with the output they get. It can be the outer designing layer, data, back-end process or simply they want some other functionality to get performed. That lead to conclude stakeholders change their minds and want something new in the page or in the module which already has been developed. Now, making everything may affect the time, human effort, estimated cost etc

Many updates requires more time and more knowledge, whereas many not. It depends upon the feedback from stakeholders.

### 3.4 Lack of communication from stakeholder's side

Once stakeholders place their requirements, they think its turn of team. We will get what we want. It is necessary to be in constant touch with stakeholders. The best way to avoid any hazard while development is to aware stakeholders about product's development on regular intervals and to take feedback from them. Suppose, if a group of stakeholders wants to create a website on crime record management system, he/she needs to provide full details to team. Once, a module is designed, its team's responsibility to take approval of stakeholders for the same and they are also responsible to let them know what they are going to do next on particular module. If time to time work is shown and is discussed, development process goes on smoothly. If Stakeholder is not available for approving, team has to decide their future plan accordingly.

### 3.5 Functionally developed, but rarely used

If the solution towards customers asked modification is done as per rules and regulations, customer may get fail to understand what is presented towards him/her. He gets confused if the solution given to him was this or not if there is a rare difference still. Suppose, if customer ask for long margin, it should not cover half of the page. Though customer's needs are not precised, its team's responsibility is to ask them.

To develop a solution is a task of hard-work, but to have a solution, is a simple work. Many modules are developed just to support other modules. For example, in a loan calculation portal, calculation page is mandatory which is not shown to user. But, if it is deleted by mistake by customer, the whole portal becomes meaningless. Such modules are not used directly, but are well developed.

## 4. Requirement Gathering Techniques

As some challenges are really hard to cop up, Many requirement gathering techniques have been used by various companies. The requirement gathering once again is a big and important task in software project management. Here are some most commonly used requirement gathering techniques with its explanation.

### 4.1 One to one interview

The most common technique for gathering requirement is to meet the clients and note down what they want. It is necessary to start the interview on time and finish on time, covering all the points. There can be many ways of such interviews like to meet at the same place or by video conferencing to hidebound requirements. It is duty of interviewee to fulfill all the criteria about what they actually want.

### 4.2 JAD

Joint Application Development (JAD) is a process where both customer and software development team members' contribution is equal. The customer generally takes part in session till the complete set of requirements is documented. So that, if any extra input or update is required, it can be done on the spot seeking customer/ stakeholder's permission.

### 4.3 Prototyping

The prototyping is an actual model of the application/ System that user wants. It is an underlying working model of a product or system. Usually it is built up as an initial version of a solution - a prototype. It is shown to users/ customers - based on which they provide their opinion for any updating or addition. If customers are satisfied, they will agree with the development team to proceed further. The development team will now actually develop that module which was shown as prototype with given changes from users/stakeholders. The development of actual module will be quiet easy as team can work more easily and fast as they are not supposed to be warned about any change of requirement. This will boost increase in speed of work and their confidence as development strategy has been defined previously.

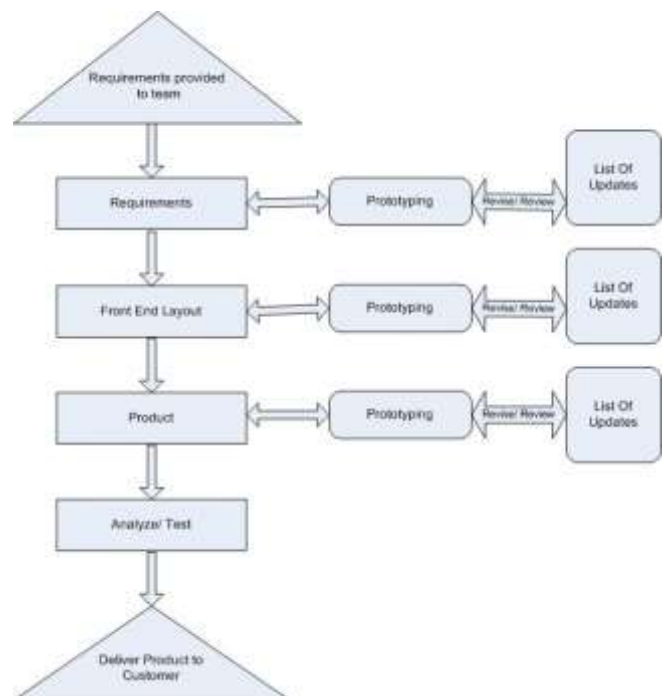


Figure 1: Requirement Engineering flow following prototype

Given in Figure 1, a typical prototype model's flow is generated when requirements are refined. Once SRS is made,

prototype is built up and reviewed to end users. With the help of end users, list of updates is made if any.

As prototype is built with ready updating list, front end layout (Designing) process is carried out. Once again prototype is built and reviewed by end users. This again results on list of updates.

Same follows for every module and at last final product and its prototype is built which is again reviewed by end user for any update.

Later, testing phase is followed by analysis and delivery to the end user. When requirement is given to team to build a project in any form, initially it is formed with the help of team members

Till product meets the near to exact needs, this process keeps on going on for an agreed number of iterations.

#### 4.4 Brainstorming

On some projects, requirements are not “unveil” as much as they are “found”. In other meaning, solution is latest one and it must be developed in such a way in which people agreed. In this type of development, “Brainstorming” can be a fresh new beginning. One of the development team member who is a subject expert get into a private room and start brainstorming what the final product will look like. Once, he/she is done, general priorities are focused which can be the best solution.

#### 4.5 Study Symmetrical System

The start up for many projects is many times a similar system. Sometimes, comparable products and systems contain working versions of good ideas for solving user problems. You can recover the time lost in redeveloping the product by taking help from other available system.

Concentrate on customer’s questions on product’s function which they actually wanted from their product but final product can’t perform, keep a list of these suggestions given by them. Later, use it to start discussions with other users. You should be able to obtain some requirements directly this way. If not, one can save suggestions for future use.

You can describe to users the comparison of other products. Explain that the system is designed for another purpose but contains an interesting feature and something similar would help them. Sometimes these systems are described in documents, such as a contract from another organization or a report written for management.

Such process might engage following activities:

Read the document several times to assimilate end user’s needs and what is written in.

Classify all of the types of information in the document. (user, system requirements, design elements, plans, background material, irrelevant detail)

Broaden up the original text to differentiate such requirements.

Build up a good physique for each of the different types of information such as: a scenario for the user requirements, functional breakdown for the system requirements, and architecture for the design.

Formulate the information to show void and over lay. If possible, add missing elements with confirmation from user.

Create trace ability links between these information elements to show the designers exactly what the users want

Convince the customer to accept the new information as the basis for the contract.

#### 4.6 Storyboard

Story board is a vast area to cover each detail. Many of companies prefer storyboard to explain their end users about the output of given project due to its picture driven, visualized explanation including various frames to simplify action. Generally the users who don’t care about the inputs given and technical inside flow of system prefer to get understood by using story board.

Storyboarding is a graphic organizer that provides the viewer with a high-level view of project [2] which explains how a product or different part of a product could work.

The purpose of using story board is:

- Conveying functionality of a proposed solution, product or service
- Convince people value of proposed product in real world domain.
- Collects requirements and generating feedback on how the events and functionalities are depicted in the story map to intended domain.
- Help people understand how they could incorporate a new technology in their own work practice.

Customers, users, or developers start by drawing pictures of the screens, dialogs, toolbars, and other elements they believe the software should provide. The group tries to get these until real requirements and details are worked out and agreed upon.

Storyboards are inexpensive and eliminate risks and higher costs of prototyping [4].

### 5. Importance of Different Requirement Gathering Techniques

People tend to pick proper requirement gathering technique from diversified basis. When it is about to choose the best technique, it varies from person to person or from organization to organization. Many a times, only a simple documentation from client side is enough whereas number of meetings throughout the project also can’t provide proper picture from client side. We can simplify the process requirement engineering process by given figure.

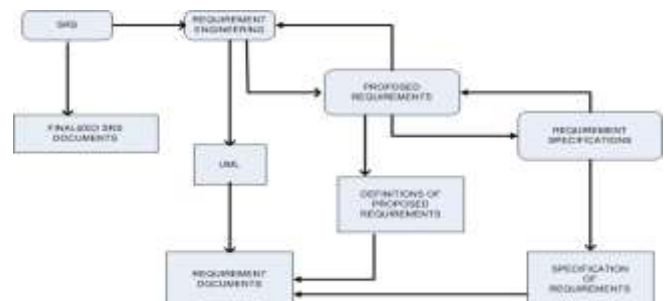


Figure 2: Framework for requirement engineering process

Once, SRS is prepared, it can be considered as final document or requirement gathering methodologies- requirement engineering can be applied. Based on which UML is prepared, functional and non-functional requirements are reframed. Generally it is task to select significant requirements which can be worked out on initial part that in turns defined as “proposed

requirements". On this level, refined, sorted out and detailed requirements are ready to work on. On the other hand, specified requirement is yet to get prepared. Once requirement specification is prepared, proposed requirements are tested again for refurbishing which in turns affect on requirement engineering task. Hence,

Practices, CrossTalk." Journal of defence Software Engineering (2002).

- [6] Davis, A.M. (1990). Software Requirements: Analysis and Specification, Prentice-Hall, Inc.
- [7] Davis, C.G. & Vick, C.R. (1977). The Software Development System, IEEE Transactions on Software Engineering, Vol. 3, No. 1, pp. 69-84.

## References

- [1] NagendraGanuga, <http://www.codeproject.com/Articles/18434/Requirement-Gathering-Techniques>.
- [2] Source: Wikipedia, <https://en.wikipedia.org/wiki/Storyboard>
- [3] Deshpande, Kuldeep, and Bhimappa Desai. "A Critical Study of requirement gathering and testing techniques for datawarehousing." International Journal of Information Technology and Management Information Systems (IJTMIS) 5.1 (2014): 60-71.
- [5] Young, Dr Ralph R. "Northrop Grumman Information Technology, Recommended Requirements Gathering

## Author Profile

**Akshara Dave** received the B.Sc.(I.T). and M.Sc.(I.T) degrees from J.P. Dawer Institute of Information Technology in 2007 and 2009, respectively. During 2009-2013, she served as Assistant Professor at various colleges in Gujarat, India. She taught in BCA(Bachelors in Computer Application), MCA(Masters in Computer Application) and M.Sc.(I.T)- Masters of Science in Information and Technology. at every iteration, requirements get clearer, more separable and more convenient to understand. The result we get is a short yet detailed requirement document.