Bridging the gap between stakeholders and engineers :
A process with the focus on Rich Media

Andrei Andriesi
s-10-770-261

Institute of Informatics, University of Zurich,
Binzmühlestr. 14, 8050 Zurich,
Switzerland

Abstract. The dialogue between end-user and engineer presents several challenges in requirements development. One issue is the gap between the conceptual models of end-users and formal specification/analysis models of developers [5]. The paper is focusing on three things with a common endpoint: a dynamic **rich media** process for bridging the gap between stakeholders and engineers. The first and second part describe the building blocks. In more particular, different requirements engineering approaches and the factors which influences them. The last part presents how to gather feedback using rich media and how different processes are already implemented.

1. Introduction

Requirements engineering and in general software engineering is a continuous process in a software lifetime. Requirements engineering presumes the continual communication between different parties; engineers and stakeholders. Because of the difference between the two groups, communication problems are doomed to appear. This not because of language restrictions but because of the working domains, background and specialization of each group. As it can be presumed, stakeholders will not understand or fully understand terms and documents formats specialized for engineers and the other way around. To successfully design a system, there is the need for both parties and therefore a common "medium" needs to be defined. This medium should define two things: a common technology (phone, presentation, PC software, …) and a common language. Besides the communication, other problems are focus and feedback. One of the biggest questions being, "How to make a stakeholder focused in meetings and how to trigger him to give good feedbacks?". If communication is the key of understanding requirements by both parties, feedback is the element which defines the shape of the product. By using feedbacks, engineers can validate, change or remove requirements in order to reflect the final, and maybe individual, needs of the customer.

As it has been said, the parties involved in the project usually have different qualifications. What is to be noted is that usually different stakeholders have an interest in the project. Because of this, different “mediums” have to be set and in the end different tactics for gathering requirements need to be implemented. Before describing these approaches, the paper is presenting a set of factors which influence these methods. These factors can be viewed as the reasons why a specific RE approach should be chosen for a specific situation.

Chapter 2 describes in deep the motivation of this paper and what are the current gaps in requirements engineering. Chapter 3 presents the involved parties and their influence in the requirements engineering process. Chapter 4 describes the factors which influence an RE approach and the relationship between the two. Chapter 5 shows different RE approaches. Chapter 6 presents why feedback is important and how can it be gathered. Chapter 7 gives some examples of real projects in which different approaches were implemented. Chapter 8 ends the paper with the conclusion.

2. Engineering the Requirements

If in the mid 70's it was thought that requirements are part of a natural process of software engineering and that they describe natural actions of a system, once with the appearance of the T. E. Bell and T. A. Thayer paper [1], it was made clear that the problem in those days systems was not entirely one of the system design but also of a requirements design. That is why from that moment many papers tried to find and define approaches for requirements engineering.

From a requirement point of view, a system is defined by functional and non-functional requirements. Each type has its own characteristics and types of validation. Many of the papers, which are describing a requirements engineering approach, are focusing on one, and sometimes, both types. An approach usually describes a new way or a new point of view of determining requirements by introducing a new presentation format, a new type of organization and/or new tools. On the other side they omit talking, or better said describing, some really important points:

- the way to use the approach, a guideline from start to finish,
- how to do different validations
- the impact of different kinds of stakeholders and different amounts/types of resources (defined in the paper as factors) in the specified approach.

If we think about the difference between the user and the engineer, more specific at the ability to understand each other field, we can determine that each requirement needs to be expressed in different ways, once for the stakeholder and once for the engineer.

Section 5 describes some of these approaches and their relation with the type of project. They use different types of presentation, each one with his advantages or disadvantages, like text, videos, pictures, animations, simulations and even comic books, to express/describe requirements to stakeholders or engineers.

As stated in [2], software engineering is a evolutionary process which starts from the basic core requirements and evolves to a big and sometimes complex system. Because of this, also the requirements engineering process passes through different stages, stages which may use different types of presentation in different amounts. For example, Figure 1 presents different stages in the software engineering process. Each stage defines different types of requirements for certain people. If in the beginning, only certain information is know, requirements should be presented in a more abstract way, but in the same time they should entertain the stakeholders. Entertainment is the way to keep the stakeholders focused. For example, if in the beginning stakeholders are approached with different UML diagrams and text based reports, their reaction will be much better if a more interactive way would be used, like using animations and videos based prototypes, or better said, something that they can view, feel and touch.

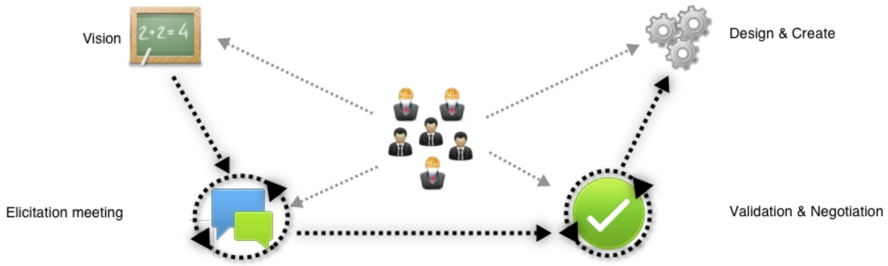


Figure 1. Software engineering stages

3. Involved parties

Requirements Engineering is a process which involves different parties, engineers, which will design and develop the system and stakeholders, the ones which will have an interest in the system.

Table 1 describes their responsibilities but also their interest.

What is it about?	Engineers	Stakeholders
RE Presentation Format	They usually require the requirements to be defined in a more abstract way; as UML diagrams or reports.	Depending on their qualification, requirements can be defined abstractly, but it is preferred in a more interactive way, like videos, pictures, simulations, screenshots, animations ...
Involvement in the project	Engineers are more involved in the projects.	Stakeholders need to be attracted by using different

		presentation formats. As a result from different interviews with different stakeholders it was concluded that the general tendency is that stakeholders tend to rush things and to focus only on certain aspects. Therefore is the job of the engineer to present the requirements in an attractive way.
Relation with other persons.	Engineers are usually organized as a team with a strong communication, with one target and interest.	Stakeholders can be individual persons or groups, with different interests in the project. They have different background and different qualifications.
Responsibility	Their responsibility is to help defining the system and to implement it.	They have to define the system and in the end use it.

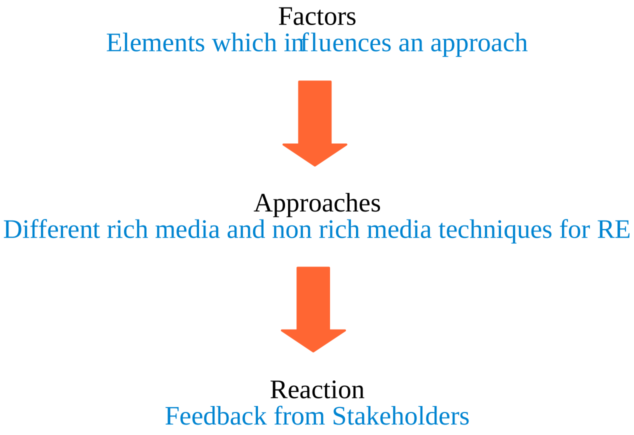
Table 1. Involved parties

Stakeholders are hard to identify because, usually, a new system affects many people. Paper [3] analysis this problem and offers different solutions. Presuming that stakeholders are rightly chosen, the probability of them having the same working field or qualifications is very low. This is why requirements need to be presented in a general way, so that every stakeholder can understand them, but also to be attracted to understanding them and in the end give a feedback. Another way of presenting and defining the requirements is first by identifying the targeted stakeholders and grouping them by requirements conditions. This separation was identified in different ABB projects, during my 2 years as an employee (ABB, short for ASEA Brown Boveri, is a Swiss-Swedish multinational corporation headquartered in Zürich, Switzerland, operating mainly in the power and automation technology areas). Whatever the first or second way is adopted, requirements need to be carefully described. The first technique was used more in the situation when the development, or the engineering team was not so big. In this case it makes sense to use resources as smart as possible. This is why making one presentation in a general way is suitable. A drawback is that the engineers need to find smart ways to capture the attention of stakeholders in order to benefit from their feedback. The second tactic is used when the resources are more generous. In this case, in order to get the maximum feedback from every stakeholder, is useful to split them into different teams based on the information which needs to be discussed (some requirements may address only to certain stakeholders, and some requirements need to be presented in different ways to different interested parties). In the ABB case, some stakeholders were used more with the questionnaire approach, others with power point presentations and videos and others with simulations and animations.

What is to be concluded is that before documenting requirements, it is best to first identify the target, their background and work environment and second to define the resources for the project (time, money, people). Based on that we can choose the right approaches for our requirements documentation and presentation.

4. Requirements Engineering Factors

The **purpose** of the paper is to give a guideline of why, how and when to use these methods and how to combine them to suit different needs. During the reading of the current and next chapters is it recommended that the fallowing relationship to be kept in mind. This relationship defines the connection between factors, RE approaches and reaction. In other words, the factors will always define which approach to use and each approach will define the amount and quality of feedback.



Factors
Elements which influences an approach

Approaches
Different rich media and non rich media techniques for RE

Reaction
Feedback from Stakeholders

4.1 Stakeholders

As explained in the previous chapter stakeholders can belong to different fields with different backgrounds and qualifications. Because of this, the best a person can interact and understand the system, the better the feedback will be. During their work, stakeholders are used with specific practices. If the requirement presentation is not according with their practices or at least with an understanding practice, stakeholders will tend to rush things and not focus.

The type of presentation and feedback gathering is very important in this case because it determines the amount of requirements and their correctness.

4.2 Project type

Different kinds of projects can take advantage of different approaches, others may compose methods for resolving different problems and others may use different tactics for solving individual issues.

Determining the type of project will help to plan in advance how different requirements will be handled, presented and defined.

As a small example, imagine that an engineer needs to create a software for simulating different hardware components. In order to successfully design each component one idea would be that the engineer video records the workflow of the user with the actual hardware. Another approach can be to create basic animations, pictures and sketches. Because of this, the engineer will have two advantages: he will be able to write the main scenarios and have a basic knowledge about how the component looks and operates. As a second step, in order to get detailed information about how the component works, the engineer will start reading different documents. As a last step, the engineer will spend time with different stakeholders and design the GUI in a way that is intuitive, by using normal whiteboards.

As it could be seen, because of the nature of the project the engineer had turned to different RE approaches, from videos, pictures and animations to more classical approaches like whiteboards and documents.

4.3 Design and development team

Different requirements engineering approaches may require specific knowledge. Having specialized people will help preparing, managing and presenting/describing requirements to stakeholders giving them better results and more confidence. Specialized people can also be used in the process of documenting the requirements. A person with more knowledge in using specific software for mounting videos or for designing animations and 3D models will give a better result and stimulate a better feedback.

4.4 Resources

In any project in general, resources represent an important factor for any decision. Number of people, time and money are only some of the decision making factors for a project. These decisions can influence the requirements engineering approach more than other factors, because they dictate the hardware and software equipment used to develop the project, the time needed to engineer requirements and most important the types of persons in the design and development team. Compromising on resources can sometimes help with the financial problem but will definitely have a big impact in the overall project quality.

4.5 Location

The location of the design and development team comparing with the one of the users can sometimes contribute to the project requirements definition mostly because of the necessity of direct contact between the team and the users' working environment. Requirements engineering is about defining users' requirements, users' interaction with the system, use cases and other principles. Most of the time the working environment hides extra requirements. Being able to interact and observe the environment, the team's ability to identify requirements increases.

Going back to the example with the hardware components, the engineer can gather more requirements in the laboratory where the actual devices are operated and tested then at his own desk by reading different specifications or user manuals. This is mainly because of the direct contact with the working environment.

As it can be seen, these factors can have a big impact on the software engineering and on the requirements engineering process. Because of this, there is no perfect guideline of how to choose an RE approach and how to organize them. Yet, because of the elements, which can be viewed as filters and organization methods, taking the decision can be much easier. Taking a look at all of these factors, it can be observed that the key in requirements identification and later on validation, is the ability to present as best as possible a use case or scenario to the user. Because of this, different approaches have been studied [4], [5], [6], [7] (these approaches are explained in Chapter 7). They introduce different ways to handle requirements engineering. If the papers mentioned above are going into the direction of creating or defining a model which can make everyone understand the system, this paper will try a different approach. Why not use all the power of these methods, which were proved to be effective in certain situations, but also profit from the knowledge and experience of stakeholders. In other words, why create a complex system which will try to satisfy everyone, but do the job only in a certain percent, when we can define a dynamic system which can be adapted according to the stakeholder. Such an intelligent system will not only use the latest advancements in technology but what is more important, will introduce the stakeholder in a familiar environment which guarantees a better productivity.

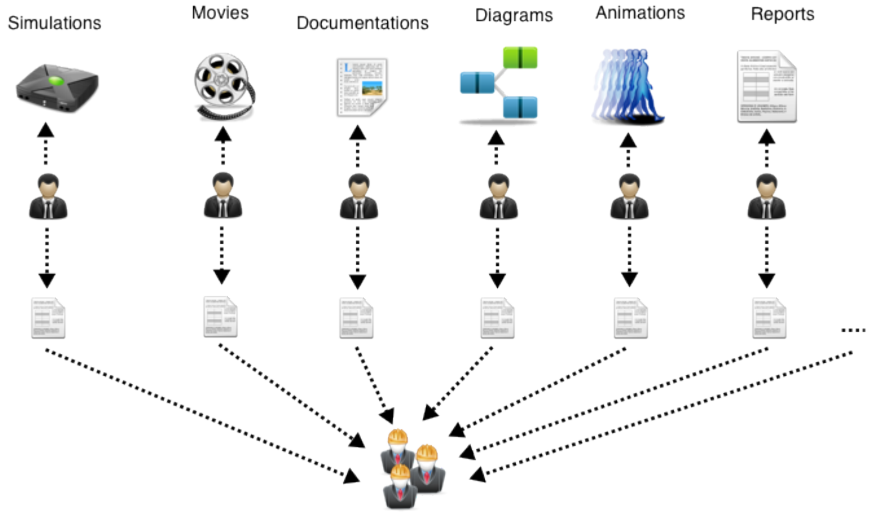


Figure 2. Dynamic RE approach

To build such a system, engineers need to first identify all the RE approaches which can be used in the system. As it was stated before, these approaches depend on different factors, like stakeholders knowledge and their ability to operate specific hardware or software equipment, project type (do we have something to simulate, instead of giving dozens of pages to read), engineers (does the project have experts which will support stakeholders operate equipment), resources, location etc.

5. Requirements Engineering Approaches

As it was stated before, defining, documenting and presenting requirements depends on certain factors. This chapter will introduce some of the RE approaches and will explain the relation between them and those factors.

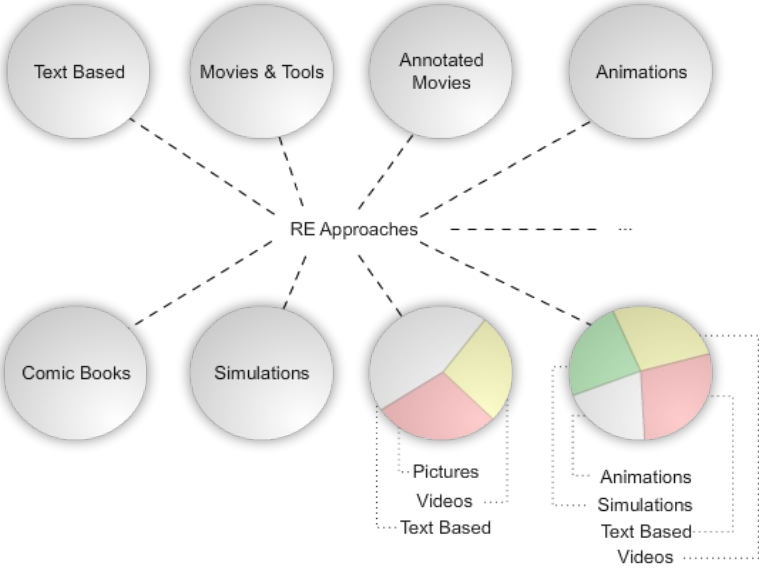


Figure 3. Requirements Engineering Approaches with different compositions

As it can be seen from Figure 3, different approaches can be composed in order to benefit to the maximum in the RE process. The composition is defined by the factors explained in chapter 4 (e.g. as it was described in the example from *4.2 Project Type* the composition was formed from videos (or animations, pictures and sketches), Text-based documentation and whiteboard sketches). Because not all RE approaches have the same level of requirements, from a resources point of view, and because resources represent an important factor in a project development, the next RE approaches will be described from the most costly ones to the less ones (costly from the point of view of resources).

5.1 Simulations / Emulations / Animations [6]

The simulation approach can be used in different situations, when we want to model an equipment, to exemplify an use case or scenario in a specific environment or when we want to emulate specific devices. Of course building a simulation/emulation is not easy. It requires time and money, but in the same time a simulation can help identify additional requirements and validate the existing ones. A simulation is easy to observe and manipulate, and does not require specific equipment (if we consider large scale simulations in virtual reality, specific equipment may be needed).

Simulations and emulations are two of the most used techniques in requirements engineering. More and more companies are adopting these techniques to test and improve their products. For example aircraft builders, power plants, cars manufacturers, tunnel builders, in trains and almost everywhere, simulations test the product and environment before giving them to the user. It is cheaper and safer to build a simulation, find requirements and verify them, than testing on real products with real people.

Using simulations we can not only predict the behavior of a system, but also the interaction and reaction of users.

Besides this, simulations are part from the dynamic methods, something that the stakeholders can interact with. Because of this, users are more focused and the probability of receiving better feedbacks is higher.

5.2 Movies / Clips / Pictures

As simulations, movies are also part of the same category, dynamic methods. Building movies is much easier than building simulations. Because of the latest technological advancements, it is enough to use a cheap web-camera to film different clips. Filming does not require specific training and can basically be done by anyone. Attention has to be paid to the object and environment which is filmed. Clips can on one side give important information about the environment but also can hide details which may be crucial for the product development. Once enough clips are gathered to describe the system, they have to be mounted in movies representing different use cases or scenarios. There are different techniques of doing this, and there are different approaches describing variations of this technique. For example paper [4] describes the usage of videos as part of the written documentation, where others go even further [5], where special tools are used to annotate and describe specific objects and users, interactions and in the end derive sequence diagrams. Building movies, annotating and applying other elements is time consuming, therefore it is recommended (by [5]) to use videos in cases where is really needed. Because of the evolutionary nature of the software engineering, videos can be used efficiently in specific software engineering iterations.

5.3 Comic books

Paper [5] also suggests the possibility of creating comic books style documents, if the hardware, software or other resources are not available. Drawing them may be time consuming, but combining them with pictures may sometimes give better results than movies. If specialized persons are available (drawing artists) then even the time factor can be considered a minor problem.

5.4 Documentations, Reports and/or Diagrams

This is the classical approach where requirements are described using different UML notations, like class diagrams and sequence diagrams. The advantage in this case is for the developer (in the software field), because of the similarities between software and diagrams. Because of this, it would be hard for the stakeholders to understand these notations. In the end users may tend to lose focus or to give unsatisfying feedback.

Other techniques include: live whiteboard sketches and presentations.

As it was stated, depending on the stakeholders background, they can be structured in groups. Meetings with each group can be made, where different RE approaches can be used. Feedbacks can also be gathered using different ways, again depending on the stakeholder's preference, emails, reports, questionnaires, meetings, recordings … .

[6] Text-based representations are very popular because they focus on use and change, and are thus often crisp, easy to maintain and manage. As they are also rough, they tend to suspend commitment and are thus quite suitable for difficult discussions with a lot of rapid change. However, text scenarios also have their limitations. First, there can be settings where pictures or sound are much more informative than textual descriptions, e.g. in many technical applications; the use of multimedia scenarios, mock-ups, and the like is well known especially in human-computer interaction. Second, the interplay of components in distributed systems is hard to capture in a small number of scenarios – and too many, or too complex scenarios destroy the advantages of short textual scenarios. Finally, any model, but also any scenario, is an abstraction of reality, which may or may not turn out to be adequate to its task. In the short term, this is not a problem and indeed one of the big advantages of scenarios. However, once the creation context of the scenario is lost, it becomes no longer criticisable; therefore, richer capture of reality scenes in multimedia, which allows scenario or model revision even after relatively long times, has become popular in documenting requirements meetings [9] and is increasingly proposed [10], [11], [12].

Why, how and when: The approaches presented above can be used in order to increase the performance of the project by identifying and verifying more requirements earlier in the project. These approaches provide a dynamic way of recognizing conditions, stimulating stakeholders and analyze their reactions (**why**). To successfully apply these methods different factors need to be first considered. These factors define which RE approach is suitable for which stakeholder and situation (**how**). The chosen approaches can be implemented during the any of the stages: elicitation, verification and negotiation, design and creation (**when**).

6. Stakeholder reaction

As stated before stakeholder's reaction in very important for the project. This is because it provides the necessary data for applying modifications or for validating implemented requirements.

Besides the official classical feedback (text based feedback), another solution would be to capture the reaction of the stakeholder in the moment of testing. This reaction can be captured physically, by using a web camera and verbally, by using a microphone. In this way, the engineers can determine which elements need to be improved or changed. For example, let us suppose that the stakeholder needs to test different functionalities of a software application. During the tested period he is filmed and audio recorded by the engineers. In the course of testing, some GUI elements are not intuitive or placed in wrong positions. Because of his physical reaction (gestures, how much he has to move the mouse, is it more intuitive in some situations to have keyboard shortcuts instead of using the mouse ...) the engineers can determine new requirements or verify the ones implemented. Using a camera and microphone triggers the discovery of more requirements, mainly because during tests stakeholders do not focus on taking notes about each inconvenient. Of course the success of this method is tied to the focus of the stakeholder in the testing process. For example if the stakeholder knows that he is filmed he may filter his emotions and reactions. The best case is if the stakeholder is not aware of being filmed. This last remark can have a legal implications, therefore a middle way needs to be determined.

7. Experiments, examples and results

This chapter will introduce two examples/experiments in which rich media and other techniques are used to give a better understanding of the system and provide more information on how users interact with it.

7.1 Interactive presentations

The first example is similar to the ones presented in paper [2] and [4], but is also inspired from different experiences gained from the collaboration with ABB Corporate Research Center in two years period. The following observations were made:

1. Talking from a normal presentation point of view, people are much more interested if the presentation contains images, charts, formulas, tables, relationships between objects, animations, videos and so on, comparing with a full text one.
2. People are better stimulated when images are presented because the human brain is used to recognize objects instead of imagining them based on a textual description.
3. The feedback and involvement of stakeholders is higher when dynamic facts are presented instead of textual documentation.

4. From an analysis point of view, a person will prefer testing or viewing how a system should work, instead of reading dozens of pages about it.
5. Time is important, therefore loosing too much time with writing requirements documents is not so much preferred.

A small example. As it is known, a project is built in an iterative way, meaning that there are different stages in which more and more requirements are identified. For this example I will describe two of these iterations (Iteration3, Iteration5), their results and the reaction of the stakeholders. In Iteration3 a scenario was taken, a textual description was made (as a small report) and then in a meeting the interaction with the system was described. The presentation had different strong points, but all of them described in pure text. As a result after only 20 minutes some stakeholders lost their focus and started checking emails and review other documents on their laptops. This was a sign that even if the content of the presentation was very strong, it was not enough to catch the attention of stakeholders. The second iteration (iteration5) was done in a different way. Instead of textual description, different examples of how the system should behave were given. Because the system was a software, different functionalities as small prototypes exemplified how different components would behave. After that, a simple animation showed how they will be connected and how the user interaction will look. As a result, stakeholders were very focused, very excited about what they were seeing, the discussions very rich and feedbacks very useful.

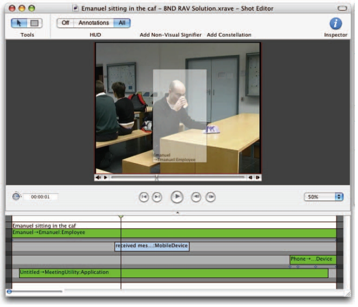
7.2 Videos and tools

If the first example showed how important is to make an interactive presentation, this example is focusing on the ability of stakeholders to identify requirements and objects in movies, and based on them to define different relationships. Later, these relationships would be exported as different UML diagrams, which would help developers in their work.

Videos represent a rich source of information, a source where objects interact, and based on that interaction use cases can be defined. This example is described in paper [5]. There, movie clips describe different interactions with the system. These small movies can be filmed even with low quality web-camera and by anyone. After enough clips were filmed they were composed in movies, using different software applications. After that, different objects were identified and regions and descriptions were associated with them.

[5]. Another tool, the sequence editor shown in figure 5, is used by the analyst to model the flow of the narrative events occurring in a movie. Such flows are represented as Live Sequence Charts. The analyst can create, edit, and delete associations between objects. These relationships are either spatial or temporal in nature.

Using this method stakeholders could interact with the system and observe how different items would interact one with each other. Based on this, other requirements were identified.


Emanuel sitting in the caf – BND RAV Solution.xrave – Shot Editor
Off
Annotations
All
Tools
HUD
Add Non-Visual Signifier
Add Constellation
Inspector
Emanuel
→Emanuel:Employee
00:00:01
50%
Emanuel sitting in the caf
Emanuel→Emanuel:Employee
received mes…:MobileDevice
Phone→…Device
Untitled→MeetingUtility:Application

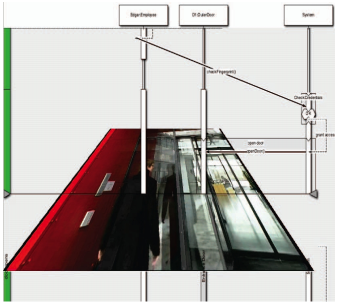


Figure 4. Shot editor: a object is identfiied and a description is associated. [5]

Figure 5. Sequence Editor. [5]

7.3 Results

The RE approaches and the experiments and example discussed in this paper are based on different other paper [4], [5], [6], [7] and different observations made by the author during his period in ABB. Because many of the presented approaches were tested in small projects and with a small number of people it is safe to say that more research is needed. First of all, these approaches need to be tested in different kinds of projects, with different sizes and infrastructure. Then, it is recommended that the engineers and stakeholders to be as diverse as possible, in order to get a realistic result. Different compositions need to be tested and if possible different rankings to be created.

8. Conclusion

Requirements Engineering is a field in which different parties are involved, parties with different background and qualifications. Because of this, we have to recognize the existence of a gap between engineers and stakeholders from the point of view of how requirements are identified, described and validated. The classical approach tells us about different types of documents and diagrams, with which a system can be designed. As it could be noticed this approach is not so suitable anymore.

The scope of the paper was to give a guideline, which shows how a dynamic process of bridging the gap between the stakeholders and engineers can be made. The paper presented different project factors like stakeholders, team, project type, resources and location. After that, different RE approaches were introduced and the relation between them and the factors was discussed. Besides different RE approaches and factors, the paper presented different ideas of using rich media in gathering feedback from stakeholders.

The experiments, examples and results section described different examples in which rich media and other approaches were used. Besides this, the section talked about the reaction of stakeholders and the benefits on the engineers side.

References

[1] T. E. Bell and T. A. Thayer, **“Software Requirements: Are they Really a Problem?”**, San Francisco, USA, 1976.

[2] Olesia Brill, Kurt Schneider, Eric Knauss, **“Videos vs. Use Cases: Can Videos Capture More Requirements Under Time Pressure? ”**, Hannover, Germany, 2010

[3] Helen Sharp, Anthony Finkelstein & Galal Galal, **“Stakeholder Identification in the Requirements Engineering Process”,** London, UK, 1999

[4] Konstantinos Zachos and Neil Maiden, **“Rich-Media Scenarios forDiscovering Requirements”**, London, UK, 2005

[5] Oliver Creighton, Martin Ott, Bernd Bruegge, **“Software Cinema—Video-based Requirements Engineering”**, Munich, Germany, 2006

[6] Peter Haumer*, Patrick Heymans**, Matthias Jarke*, Klaus Pohl*, **“Bridging the Gap Between Past and Future in RE: A Scenario-Based Approach”**, (*) Aachen Germany, (**) Namur Belgium, 1999

[7] Ruzanna Chitchyan, Awais Rashid, Peter Sawyer, **“Comparing Requirements Engineering Approaches for Handling Crosscutting Concerns”**, Lancaster, UK, 2008

[8] F. Brun-Cottan, P. Wall, **“Using Video to Re-present the User”**, Communications of the ACM, Vol. 38, No. 5, Mai 1995, pp. 61-71.

[9] P. Haumer, K. Pohl, K. Weidenhaupt, **“Requirements Elicitation and Validation with Real World Scenes"**, Aachen, Germany, 1998.

[10] K. McGraw and K. Harbison, **“User-Centered Requirements: The Scenario-Based Engineering Process.”**, New Jersey, USA, 1997.

[11] D.P. Wood, M.G. Christel, S.M. Stevens, **“A Multimedia Approach to Requirements Capture And Modelling”,** Proc. 1st Int’l. Conf. on Requirements Eng. (ICRE’94), IEEE Computer Society Press, Colorado Springs, CO, USA, Apr. 1994, pp. 53-56.