

Requirement Gathering Methods

Mamoun Eid
November, 2015

Abstract

The purpose of this paper is to examine the different methods in gathering requirements. Requirements are one of the most vital pieces to ensuring the success of a system or project. To ensure the optimal requirements are received, the methods in which those requirements are obtained are equally important. Through this paper, we will look at what requirements are, as well as the different methods in gathering them.

Introduction

The second phase of the systems development life cycle is analysis phase. The analysis phase is the most important stage in SDLC. A study by project management consulting company, PM Solutions identified the top cause of a troubled project was poor requirements. The 2004 CHAOS report from the Standish Group indicated that project success rates were only 34%, the rest were either challenged or failed. They also noted that poor requirements were very common in project failures, as they listed it third on their list common factors of project failures. Interestingly enough, they noted lack of stakeholder involvement as first on their list. Lack of stakeholder involvement, however, can also be tied into poor requirements. When the stakeholders aren't heavily involved, or have poor dialogue between the systems analyst, this directly leads to poor requirements.

Poor requirements have a significant impact on the end results of systems or projects. Requirements are the "blueprints" that everyone involved on the project uses to work from. When there are poor requirements, this can lead to poor designs and tests, which in turn will cause delays in development and testing. The requirements must be revised; hence, all of these delays contribute to a late project. Poor product quality can result from poor requirements in situations where key components are overlooked and left out all together. Testing is also prioritized to focus on what's important. If the requirements are flawed or unclear, testing is not properly executed and will eventually lead to poor quality. To avoid poor requirements, it is highly vital that the analysis phase of the SDLC is thoroughly completed, without being overlooked.

The analysis phase can be broken into to two processes: requirement gathering and analyzing the requirements. Requirements, in turn, are divided into functional requirements and non-functional requirements. Functional requirements are defined as processes, information, and interactions. These are the desired functionality that the client wants built and describe the interaction between the system and its environment. Functional requirements generally describe what the system has to do. Non-functional requirements are non functional characteristics that address operational and technical requirements. They may describe such factors as encryption, security,

hosting, environment, disaster recovery, business continuity. Gathering requirements properly and selecting the appropriate technique can assist in ensuring that you will not have poor requirements.

One-on-One Interviews:

One-on-one interviews are the most common technique for gathering requirements, as well as one of the primary sources of requirements. To help get the most out of an interview, they should be well thought out and prepared before sitting with the interviewee. The analyst should identify stakeholders to be interviewed. These can be users who interact with the current or new system, management, project financiers or anyone else that would be involved in the system. When preparing an interview is it important to ask open-ended questions, as well as closed-ended questions. Open-ended questions generally help in obtaining valuable information, based on various individuals and the way the different way they interact with, or view, the system. These types of questions require the interviewee to explain or describe their thoughts, and cannot be simply answered with a “yes” or “no”. Asking the interviewee what they like about the current system or how they use it would be examples of open-ended questions. These types of questions can provide the consultant to further probe for more detail with follow up questions, in order to get more details. An example open-ended question would be “What are some of the problems you face on a daily basis?” Close-ended questions can also be useful, when the interviewer is looking for a specific answer. They can provide specific answers for the interviewee to choose from, in formats including true or false or multiple choice. Although close-ended questions do not provide as much detail as open-ended, they can be useful to cover more topics in a less amount of time. An example of a close-ended question would be “How many telephone orders are received per day?” Once the questions have been established, it is a good practice to provide the questions to the interviewee prior to the interview, in the event that the interviewee needs to prepare. During the interview, the interviewer should obtain permission from the interviewee that recorders may be used, to ensure that if details are missed while taking notes can easily be retrieved. At the end of the interview, the results should be provided to the interviewee, for confirmation of their responses.

Group Interviews:

Group interviews are similar to one-on-one interview, except there is more than one person being interviewed. Group interviews work well when the interviewees are at the same level or position. A group interview also has an advantage when there is a time constraint. More thoughts and discussion can be generated, as someone in the group may state or suggest an idea that may have been overlooked by others, which in turn can lead to a discussion or provide more information on a particular issue. The interviewer can gauge which issues are more generally agreed upon, and which are which issues differ. A major disadvantage can be scheduling the interview. When more than one person are involved, it may be difficult, or become time consuming, in establishing date and time that works well for all parties.

Questionnaires/Surveys:

Questionnaires, or surveys, allow an analyst to collect information from many people in relatively short amount of time. This is especially helpful when stakeholders are spread out geographically, or there are dozen to hundreds of respondents whose input will be needed to help establish system requirements. When using questionnaires, the questions should be focused and organized by a feature or project objective. Questionnaires should be not be too long, to ensure that users will complete them. When constructing the questionnaire, general guideline to determine the questions would be to ask “how, where, when, who, what, and why.” For how: *“How will you use this feature?” “How might we meet this business need?” “How will we know this is complete?”* For where: *“Where does the process start?” “Where would the user access this feature?” “Where would the results be visible?”* For when: *“When will this feature be used?” “When will the feature fail?” “When will we be ready to start?”* For who: *“Who will use this feature?” “Who will deliver the inputs for the feature?” “Who will deliver the outputs of the feature?”* For what: *“What do I know about this feature?” “What does this feature need to do?” “What is the end result?” “What must happen next?”*

User Observation:

The direct approaches of interviewing and questionnaires provide valuable user feedback based on the questions asked of them; however, there are times when direct observation may be better suited in requirement gathering. To get a better understanding of a user in their in current work environment, the analyst may observe the user themselves. User observation is helpful in assisting the analyst by getting a full grasp of how the user interacts with the system, firsthand. When the objective is to improve a task, the analyst can observe the user and how their surroundings affect their interaction with the system. Sometimes stakeholders may find it difficult in explaining what exactly what their tasks consists of and what their requirements may be, observing the user in cases like these will help provide the requirements. User observation may also be useful in validating data that had been previously collected. Be it in cases where users provide misleading information, or cannot fully recollect all of their tasks in how they use the system.

User observation should be planned to ensure that all elements are constant surrounding the observation. This will assist in uncertainty, and the consultant can focus on the user and assist in knowing what to look for. The analyst will not be distracted and record, or note, irrelevant issues. The more useful information gathered, the less time it will take to the analyst to dissect and evaluate afterwards. Timing of the observation can also prove relevant when planning. For optimal results, the consultant should schedule three different periods of observation: low, normal, and peak times. This may prove helpful in because the user may interact with the system differently during different times. The consultant will take into consideration the differences in times time settings and use it to obtain to construct better requirements to assist in all three times. When observing the user, there are two approaches the consultant can take, passive or active. In passive observation, the consultant does not interact with the user while they are working. They simply observe and take notes. While in active observation, the consultant will ask the user questions during the session. Observations should always be done without bias, as in other requirement gathering techniques, the analyst must simply record what is presented to them, and avoid making comments on whether they believe what is correct or not. A checklist can provide to be useful, with key points already noted and the consultant verifying events on their list. For

example, they can check if a user uses certain features, the frequency of events, triggers that cause different uses. Taking detailed notes is helpful in recording unexpected events. There may be events unknown to the analyst ahead of time, they can be captured by taking notes of the event and why it occurred. Video recorders may be used, but must always be approved with the user and their company.

While there are many advantages to use observation, there are some disadvantages associated as well. As previously mentioned, observations should take place during peak, normal, and low business times. However, it may still be difficult to capture enough information in one of these sessions. There may be the need for multiple sessions to verify that facts collected were constant, rather than isolated incidents. Analysts themselves can sometimes be biased in what they expected to see, and what they actually observed. Again, the focus is to simply collect the facts, not form any biased opinion when seeing the working environment. The users sometimes may not themselves provide an accurate depiction of their every day task and respond differently. Some users may become nervous, and not perform as they normally would. Other users may try and perform more better or worker harder when they know they are being observed. Such observations can inhibit the analyst to decipher what the true requirements actually should be.

Analyzing Existing Documents:

Analyzing existing documents can prove to be a useful technique in requirement gathering, on its own as well using it to supplement other techniques. Reviewing the current process and documentation can help the analyst understand the business, or system, and its current situation. Existing documentation will provide the analyst the titles and names of stakeholders who are involved with the system. This will help the analyst formulate questions for interviews or questionnaires to ask of stakeholders, in order to gain additional requirements. If an analyst is uncertain why certain procedures are in place, this can also help the analyst in asking these questions during interviews. When studying the requirements, the analysts may find problems that they may distinguish on their own. The analyst may find there was missing information in old documents. They may also find redundancy, in which steps are unnecessarily repeated. The consultant may look at old requirement documents and reuse of the requirements that may still be relevant, while discarding others that may be out of date. The reason why the current system is designed the way it is, which can suggest why certain features were left out. Principles and rules for the organization itself can be discovered by analyzing documents. Analyzing documents can be used as a supplement to information obtained from interviews, questionnaires, and observations. For example, if some of the interview answers are unclear, organizational documents may help in making sense of some of the interviewee's answers. Reviewing existing documents may also assist in understanding why a user performs certain tasks while observing them. A drawback to analyzing documents is that documents may be outdated, the analyst must confirm whether the documents are current or not. Another drawback to reviewing documents is it can be very time consuming, depending on the organization and the system. A system that interacts with many different facets of the business will have large amounts of documentation to review.

Joint Application Design/JAD:

The fore mentioned techniques have been examples of traditional requirement gathering, whereas JAD is an example of a more contemporary method for gathering requirements. Joint Application Development (JAD) was introduced in the late 1970s so solve some of the problems users experienced in the conventional methods used to gather requirements. Organizations that have adopted JAD report savings of 40% or more in project design time. JAD is successful in reducing time because the user and other key members are heavily involved in the process. The goal is to get the design right the first time, thereby reducing different iterations. JAD is usually conducted in at a location other than the place where the people involved work. This helps to reduce distractions. The participants of a JAD include: JAD session leader (also known as the facilitator), users, managers, sponsors, systems analysts, scribe, and other IS staff members. The facilitator, whom is usually trained in project management as well as system s analysis, manages the entire process. They are responsible for keeping the group on the agenda and resolving conflicts. Facilitators do not contribute ideas; they solicit the ideas of the group. They also make the rules for the sessions, and can dismiss anyone who is troublesome to the process. Users involvement helps eliminate their frustration over the systems development process. The user is involved throughout a JAD session, and can make great contributions, as they will be the ones using the new system. Managers can provide more of insight into the organizational aspects of the system. The sponsor doesn't attend all JAD sessions, but usually just the beginning or end, as they do not contribute to ideas, they merely sponsor the process. Systems analysts are there to learn from users and managers. As in other requirements gathering techniques, they should not show bias. The scribe takes detailed notes during the session. The other IS staff members, such as programmers or database analysts attend to contribute technical aspects to the proposed ideas.

Some of the benefits of JAD include:

- Consolidation of months of work into a structured workshop
- Clarifies specification requirements in an environment of consensus
- Identifies open issues and parties responsible for their resolution
- Ties together the steps of the design process in a concise document
- Increases user satisfaction by directly involving users in the design process
- Builds commitment through the use of the executive sponsor
- Builds a sense of belonging and helps create a cohesive team through the physical and social setting

AJAD:

Although JAD is seen as a newer method, there are some instances where JAD has even developed further into AJAD (Automated JAD). A traditional JAD setting will be in a conference room of some type, with a "U" shaped table. There will be documents on the table, whit boards used to write ideas, and agenda posted on the wall and a screen for projecting. An in AJAD session, users be still be seated in a "U" shaped table, but will have computers in front of them, instead of documents. Some of these settings will have desks that conceal their computer screens and keyboards beneath glass surfaces. The purpose is to promote team interaction and ensure that technology remains an enabler, not the focal point of decision making. AJAD uses groupware packages during their sessions. Groupware capabilities fall into three categories:

information generation, information analysis/decision making, and information documentation. The Information generation capability supports simultaneous generation of information by a large group of people, which can prove to be less time consuming. This can also provide anonymity for participants who otherwise might not have wanted to share their ideas aloud. This feature also reduces the role of the facilitator as the intermediary, as fewer confrontations arise than when contributors are all debating ideas. Just as the information generation capability provided simultaneous generation of ideas, the information analysis/decision portion helps in simultaneous analysis of information. As users identify business activity and data, the matrix capability can be used to translate these lists into the two axes that the participants can use analyze the activity/data relationships. The groupware can use voting tools to generate a vote on for different system features. Once voting is complete, the facilitator can display the results for everyone. Groupware's final category of information documentation is a self-documenting process. An instant record of all the information that the participants generate, all the analysis conducted, and all the decisions reached. Some tools may have the ability to show the different iterations.

Prototyping:

Prototyping is another form a contemporary requirement gathering method. Prototyping is iterative process that heavily involves the users to complete. The user provides the requirements, in which the analyst can plug in directly and show the user the outcome. Prototyping is dependent on user interaction and cannot be utilized as its own method of gathering requirements. The analyst must interview or perform some other form of requirement gathering to perform before they begin prototyping. However, prototyping is very effective in specifying requirements, because of how heavily involved the user is. The user will still be sitting side by side with the analyst, providing them requirements as the analysts enters them into a working system. This will allow the user to instantly see the outcome of their requirements. At this point the user may change some of their requirements. They may see that what they provided was not what they had in mind. A form may appear cluttered with information; at this point the user can go back and adjust their information. This may also be the case in when the user forgets important information; they may not realize it until they actually see a working version of the system. The user and analyst will continue to go through different iterations, until all specifications are complete. The last prototype will be used as a model to build the actual system. Some of the disadvantages of prototyping is the user will pay too much attention to details on the screens, rather than what the prototype is meant to communicate. Executives can grow impatient as they see a complete prototype, but will not understand why the finished system takes so long to complete.

Conclusion:

There are many different methods in requirement; all will have advantages and disadvantages. Cost and time proved to be the two most important factors when determining which method was to be used. Many methods were used in supplement of each other. User observation was followed by interviewing, questionnaires, or analyzing documents. As was prototyping, they cannot be used as a method on their own. Prototyping can be useful in that it heavily involves the

user, but at the same the developing quick prototypes can cause the user to be misled in expectations of when the real system may be completed. JAD was very efficient in it involved everyone at one time; however, sometimes the cost of JAD may be too much. The method in gathering the requirements may vary depending on the situation and constraints, but using the various methods to supplement each other will help in achieving complete requirements

Information gathering techniques

By [Manick](#) On [Saturday, February 18, 2012](#)

Information gathering includes various techniques like brainstorming, Delphi technique, interviews and root cause analysis. The ultimate aim of all these techniques is to identify and prepare a comprehensive list of risks in the project.



Brainstorming

It is one of the most widely used techniques to identify risks in a project. Project team usually performs brainstorming, often with subject matter experts, risk management experts and other important stakeholders who can contribute to the risk identification. It allows people to come up with risks. During brainstorming sessions there should be no criticism of ideas. The main focus is to open up possibilities of risk. Judgments and analysis at this stage inhibit idea generation. Ideas should only be evaluated at the end of the brainstorming session. Brainstorming sessions always have a facilitator to lead the team and help turn their ideas into a list of risks

Delphi technique

This technique is used to build consensus of experts who participate anonymously. A facilitator uses a questionnaire to solicit ideas about important project risks. The questionnaire is often designed with forced choices that require the experts to select between various options. The responses are summarized and re-circulated to the experts for further review until consensus on the final list of risks is reached. Delphi technique helps reduce bias in the data and keeps any one person from having undue influence on the outcome.

Interviewing

Interviewing is generally a face-to-face meeting that includes question and answer sessions. The interviews are conducted with project manager, project team, stakeholders, subject-matter experts, and individuals who may have participated in similar, past projects. Interviews help us to get first-hand information about others' experience and knowledge.

Root cause analysis

Root cause identification is a technique for identifying essential causes of risk. Reorganizing the identified risks by their root causes will help to identify more risks. This technique enables you to understand the risk more clearly so that responses can be planned to prevent recurrences.

10 interviewing rules

Stay focused on these 10 basic tenets to prepare for your next interview and to drive off any potential jitters.

Carole Martin, Monster contributor