Chapter 6

Categorizing Requirements

How many legs does a dog have if you call the tail a leg? Four; calling a tail a leg doesn’t make it a leg.

—Abraham Lincoln

Gathering and documenting requirements can be difficult. Knowing how to create a structure that enhances comprehension and makes it easier to divide and conquer the analysis phase can be useful and is actually critical for success on large and complex undertakings. There are many approaches to categorizing requirements and in this chapter some of the more common ones will be explored. The International Institute of Business Analysis® (IIBA) approach to classification will be reviewed, and there will be a review of other, equally good approaches. It is not the intent in this chapter to promote one specific classification system; rather it is my view that different projects and different business situations need different classification approaches, and the reasoning for this will be explained in this chapter as well.

6.1 Objectives

- Identify the need for categorizing requirements.
- Determine what level of detail a requirements gathering effort should go to.
- Review different categorization schemes.
- Review situations where each scheme of categorization may work.
- Determine roles and responsibilities involved with different categories.
6.2 Overview

Sometimes starting to capture and document requirements seems like a daunting task. It is hard to know where to begin, what level of detail to get into, and it is almost impossible to know when a complete set of requirements has been captured. A well-defined classification system, or taxonomy, can help with these issues. It is similar to going to an automotive dealer to buy a car. It helps to organize the discussion into categories. For buying a car those categories may be interior, exterior, and performance related. For a system it will be different categories, but the purpose is the same: to enhance communication.

A well-defined classification system, or taxonomy, can help...
Categorizing Requirements

Clarity: Are you able to understand the requirements as written? Will different stakeholders interpret them the same way?

Validity: How are the requirements validated? Are there things in the product which the customer does not want?

Feasibility: Are these requirements possible to implement? Have any feasibility studies been done?

Precedent: Are these requirements that have been implemented in this organization before? Have they been implemented anywhere else?

Scale: Is this project size similar to past projects? Does the organization have the capability to implement a project this size?

In this example the hierarchy is intended to evaluate the quality of the requirements gathered and the risk associated with them. The business analyst should use this type of checklist during the requirements checking, when the Business Requirements Document (BRD) is finalized and reviewed.

In its basic form a requirements taxonomy is little more than a checklist which helps the analyst to ask the right questions and ensure that all key areas of the business have been covered. Using the requirements element of the risk taxonomy above, the checklist would simply say:

- Are the requirements stable?
- Are the requirements complete?

and so on.
Determining Project Requirements

In addition to covering different areas of the business, or the technology, it also encourages the business analyst to delve deeper into each part of the taxonomy. For example, a requirements category of “security” may list a requirement such as “The system must limit access to sensitive financial data.” This should prompt the analyst to ask “What is the definition of sensitive financial data?” and “What other types of security requirements are needed?”

Building a classification system for an organization takes a formal process and a repository for lessons learned. Figure 6.2 shows a process for building and improving such a system.

Step 1 is to identify any existing ways of categorizing requirements based on the type of project being developed. Often there is no official standard for categorization within an organization; however, that does not mean that there is no categorization being used. If no formal document exists, such as a BRD template, then check with the people who have experience with projects in the organization. Ask what types of requirements are important here. Are there internal, corporate requirements? Are there any external requirements? What organizations will typically provide requirements? By asking these types of questions, a rudimentary structure for a taxonomy (or classification system) will start to appear.

In Step 2 the actual requirements collected are put into their respective categories. Put any requirements not fitting into the categorization system into an “Other” bucket. At the end of the project these can be used to identify potential new categories for the future. Some requirements may fit into multiple categories. Decide on one and then make a note in the other category or categories referencing the selected category. Why? Often different categories will be assigned to different...
groups or individuals for analysis. Adding the reference without putting the requirement itself in multiple categories will help minimize duplication of effort, while still keeping the lines of communication open.

Step 3 is done during reviews of the BRD, during testing, and ideally in post-project reviews. It identifies missed requirements, which will be used for many types of process improvements; here they will be used to evaluate root causes related to requirements categorization.

In Step 4, the root cause analysis is then performed to try to identify why a requirement was missed. Was it due to poor elicitation techniques? Or, was it due to not asking about certain types of requirements? If so, these requirements will be taken to the next step.

Step 5 then updates the categories of requirements based on the input of the root cause analysis in the previous step as well as the content of the “Other” bucket mentioned in Step 2. This newly created list will act as a repository for lessons learned for future projects.

By the time this process has been used on multiple projects, the classification system will become more helpful and actually assist in standardizing the organization’s approach to requirements gathering.

6.4 How Much Detail Do You Need?

When looking at requirements taxonomies, it is easy to get the impression that the more detail the better. That is not true. There should be as much detail as is needed, not a bit more. Naturally, the difficulty is to judge where that line is. With the standards and processes from PMI®, SEI, and IIBA in their back pockets, many organizations are now developing large, detailed, sophisticated, and utterly confusing requirements documents.

There should be as much detail as is needed, not a bit more.

Karl Wiegers states in his book More about Software Requirements:

The requirements may be vague, but the product will be specific.

Karl Wiegers uses the statement to emphasize the need for precision in requirements definition, which is of course needed, but there is also another thought being triggered by the statement. Who should determine the specifics of the product? Does it have to be the customer? What if the customer doesn’t care? Or, what if the customer is clueless?
Determining Project Requirements

Development organizations often feel that if customers working with developers do not know what they want, or do not understand the business, then the developing organization must push to get different customers involved to ensure that all the requirements come from the customers. It is true that there must be safeguards against making assumptions about the customers’ competency, but it is also true that many times customers do not know what they want, do not want to be involved with the development process, and should not have to be involved. The customer’s expertise is not systems development or process improvements. The business analyst, together with the developers and the customers, must make an assessment up front on what requirements come from the customer and what requirements come from other sources. Compare it to buying a vehicle. Going to a car dealer, most people would not expect to have to tell the dealer that they want four wheels, a steering wheel, carpet in the trunk, seat belts, etc. Although it is true that the customer would notice (and complain) if those items were not there at the time of delivery, the customer has a right to assume that the product developers will do some thinking on their own. Some organizations struggle with taking on that responsibility. They create hundreds of pages of requirements, overwhelming the customers and forcing them to think about things that they should not have to think about.

...many times customers do not know what they want, do not want to be involved with the development process, and should not have to be involved.

A good taxonomy can help with deciding what the customer should or should not be involved with from a requirements-gathering standpoint. If the customer just wants a good sales-reporting system, they should be able to tell the analyst what type of sales reports they want at a fairly high level (business requirements level), and then expect the analyst to work out the details. The analyst can research what packages are available, what other organizations do, and come back with some ideas that the customer can verify as being on the right track.

For the Prescription Interaction Project, the customer should be able to state that they need a system where they can enter all the prescription information, check inventory, look up customer information, but should not necessarily have to state every piece of data, how to interface with other areas, or what the look and feel of the system should be. That can be handled through organization standards and research by the business analyst and the developing organization. On the other hand, if the customer has strong feelings about what the system should look like, then they need to be involved in the detailed definitions of the user interface.

So when trying to define how detailed the requirements-gathering effort should be, keep the following rule in mind: the user has a responsibility to
provide the requirements for the things they care about, but the developer has the responsibility to be able to create a good solution for the areas that the user does not care about.

In general, different approaches will require different levels of details. If the effort will result in buying a package, something available off the shelf, then the requirements should be defined at a higher level, with a focus on what should the system do, rather than what it should look like or how it should flow. Changing look and flow is normally not a good idea when getting a package. If the user is not willing to live with the basic look and feel and flow of the package, it would be better to go to custom development.

Customer development can be broken into two categories for the purpose of this discussion. If the system will be developed in-house with the analyst, developers, and the users co-located, then there is less need for a lot of detailed requirements. Unclear areas can be worked out as the effort progresses. However, if the development will be outsourced, the customer is dispersed globally, or the development team is in a different location, then the requirements document must be much more formal. This is especially true if the development team is offshored, involving time zone, language, and cultural differences. The key here is communication. How easy (and how likely) is communication between the parties? If ongoing communication is rare, then more rigors must be adopted in the development effort.

### 6.5 Stakeholder-Based Classification

One of the best ways to classify requirements is by stakeholders. By documenting whose requirements have been captured, it is easier to see if any stakeholders were overlooked. This helps with the prioritization of requirements, by identifying which stakeholder is the most important (as defined by the sponsor). It also helps with assigning responsibilities for sign-off.

There are two general categories of stakeholders:

- **External**: Government, customers, vendors
- **Internal**: Management, users, other departments

Within each high-level category, add the people and organizations whose requirements may be pertinent for the project. As an example for external, sub-categories may include:

- Local government for building codes
- IRS for tax laws
- FDA for drug laws
- Major customers who have requested services
- Vendors who will provide part of the product

© 2008 by Taylor & Francis Group, LLC
Determining Project Requirements

For internal, it may look like:

- Project sponsor
- Product manager
- Purchasing Department
- Order entry clerks
- Pharmacist

Once the initial list of stakeholders has been captured, a review of that list should be done with the sponsor, project manager, and other key decision makers. There are three main goals to be accomplished by this review. First, look for completeness. Are there any other key stakeholders who should be added to the list? Second, evaluate the priority of the stakeholders. Not all stakeholders are of the same importance and when prioritization of requirements is needed, it helps to know whose requirements matter the most. The third reason for the review is to identify representatives from each stakeholder group. The earlier they are identified and notified about their participation in the requirements effort, the more likely it is that they will be available for the requirements gathering sessions. Because these representatives will actually be the people from whom the requirements will be gathered, the process of buy-in and commitment must start as early as possible.

6.6 Sequence-Oriented Classification

Sequence-based classification recognizes that different levels of the organization have different requirements of the product being developed. It also recognizes that these requirements should be captured in a certain sequence, from high level to detailed, and from the upper levels of the organizations to the end users. For top management, these may be captured as goals and objectives; for middle management, it may be operational improvements; and for the users, it may be what they need to get their job done. The following is a description of a possible sequenced-based structure, followed by an example of the corresponding organizations or people who will provide the different categories of requirements for the Prescription Interaction Project. The categories are:

- Regulations, Industry Standards, and Corporate Policies: These may be mandatory or they may be recommendations. They are often documented as constraints on the solution. IIBA does not currently list this one as a category. It is probably assumed that it is included in the Assumptions and Constraints category mentioned below. Examples: FDA for drug information, FCC for how data can be transferred, local government for reporting requirements.
Categorizing Requirements

- Business Requirements: This category identifies what management is expecting out of a project or product. When looking at business requirements, they are gathered at all different levels. It is a good idea to sub-categorize into:
  - Strategic: Where is the business heading? This may have been documented in the product vision or within business goals and objectives. Understanding the strategic requirements will tell a lot about executive expectations and can also drive the type of questions the analyst should ask. If the strategy is global expansion, it is likely to create a different set of interviews than if the goal is cost reduction.
  - Tactical: How will the strategic goals be reached? Often the project is part of a corporate initiative which should be tied back to the strategic vision of the organization. The analyst needs to understand what those initiatives are and how this project fits in with them. This is often focused within segments of the business in support of the strategic goals. So at this level the customer could be the director of marketing, the operations manager, or the person in charge of data entry. The focus for the business analyst is to find out what these people need to be able to support and operate within the strategic requirements. The requirements being given here must be traceable back to the strategic level.
  - Operational: How should the business operate on a daily basis? This will include requirements dealing with productivity, training, process performance, security, and other requirements important to the first-line manager in the organization who will be using the system. It will also include information about what reporting and outputs that manager is looking for. This is targeting the first-level manager, the supervisory level. It could be a supervisor over the group whose people will be using the system, or someone who needs to be informed about impacts on the business by the system.

- User Requirements: These are requirements needed by the person who will actually be using the system being developed, or who needs to perform some of the needed tasks manually if that is what the eventual solution calls for. For most users the system is a tool to help them perform their job. Different users are looking for different things. Some need higher productivity, some need more security, and others just need a repository of information. When capturing user requirements, do not be too narrow in the requirements elicitation. The focus should be on what the users need to do in their job, not what they need the system to do (that will be captured in the next bullet). This means that some of the user requirements that are captured will never be implemented within the system. Depending on the solution selected during the design phase, some of these requirements may be automated, some may be handled manually. For example, the requirement of “User must be able take an order” will likely be automated, and “User must verify customer’s home phone number” will likely be a manual job requirement, dealt with outside the boundaries of the system.
Determining Project Requirements

- System Requirements: What will the system do to assist users in their job? It can be sub-categorized into:
  - Functional Requirements: This identifies the core reason for the system. These requirements concentrate on what the system does and tend to be what first comes to mind for the customer. Examples: “System must allow for entry of orders,” “System must print out a receipt.” Functional requirements are things that the business would need to do even if the system was not there and the process was manual.
  - Quality-of-Service Requirements: This is an IIBA term. In other standards this is often referred to as non-functional requirements or supplementary requirements. These requirements are really the characteristics of the system. They are sometimes hard for the customer to determine because the requirements relate to the system more so than the business problem that the customer is trying to solve. In general, if there were no system most of the quality-of-service requirements would not apply. Quality-of-service requirements come in all different flavors and can be sub-divided in many ways. Some common sub-categories include:
    - Environmental: Where will the system be located? Are there weather-related concerns? Are there concerns about infrastructure? Are there legal issues? This can deal with accessibility outdoors, or legal restrictions on where a certain piece of hardware can be placed.
    - Interface: Most systems today interface with other systems. What are those systems and what type of interface is needed? The interfaces may be internal or external, new or existing, depending on how the current effort will change business processes.
    - Operational: This can include number of operators, operational environment, and access issues. Does the system need to be accessible without using a keyboard? Does it need to have voice recognition?
    - Performance: At this level the requirements deal with system performance. Process performance is included under Business Requirements, but here the focus is on expected systems responses. How much time is acceptable for the system to look for potential drug interactions? This will potentially drive the design later on. If the requirement is for sub-second response time, then most of the information may need to be stored locally.
    - Privacy: Is there sensitive information which must be kept private? Who should have access to what information? How to prevent unauthorized use of the information? Are there legal issues with using the data in a way that may not be visualized by the customer?
    - Safety requirements: In what environment will the system be used and are there any safety concerns related to that? As an example, if
a navigational system is developed, how do you ensure that it is not impeding the safety of a driver using it?

- Security: Who should have access to the system? Who can see data? Who can update data? This may be overlapping with privacy requirements which bring up a point worth remembering. Quality-of-service categories are often overlapping. It is not terribly important to put a requirement in the right category; the focus should be on looking at a project from different angles to get a complete picture. Whether the requirement has been classified as a safety versus a security requirement is less interesting than the fact that it has been discovered.

- Training: What training is needed for users, support personnel, and other stakeholders? This includes formal training but also “train-the-trainer” approaches and in self-directed learning.

- Assumptions and Constraints: These are predetermined characteristics of the solution. Assumptions are educated guesses. The right answer is not known, but in order to make progress an assumption is made and work is done based on that. For requirements an assumption may be that “the current infrastructure will be able to handle the new workload.” As the requirements gathering progresses this may prove to be false and, if so, this will lead to a change control item. All assumptions carry a risk, but it is impossible to make progress capturing requirements unless some are made. It is important to document them and get agreement from the key stakeholders. Constraints are limitations. A constraint may be that “the new system must use the existing infrastructure.” This is a requirement and it must be captured and communicated to the developers.

- Implementation Requirements: This deals with the capabilities that the solution must have for the transition into the solution. How much downtime can the business afford? Is there a need to run two systems in parallel? Does data need to be converted? Example: “All historical data must be converted and available for the new system when it is installed,” and “A system back-out plan must be developed that allows the business to revert to the old system with a maximum downtime of 15 minutes.”

All of these requirements should be traceable to and from each other as seen in Figure 6.3. This is a key concept of scope control. If a requirement is found at the system level and cannot be traced to the levels above, there should be an evaluation of why the requirement exists. Is it scope creep? Did a higher level requirement get missed? Quality-of-service requirements are often traceable to regulations, industry standards, and corporate policy type requirements, and functional requirements tend to trace back to user and business requirements.

The following is a review of an example of this traceability for the Prescription Interaction Project. For each requirement level it identifies who the requirements provider is and gives an example of a requirement that belongs at that level.
Determining Project Requirements

Regulations, Industry Standards, and Corporate Policies
Source: FDA
Requirement: All prescriptions filled must be checked against the FDA recall database for any open recalls. Any prescriptions with a recall issued against them should result in a rejection of the prescription, a notification to the doctor, and a notification of the attempt to FDA.

Strategic
Source: Executive Steering Committee
Requirement: The Prescription Interaction Project must reduce lawsuits based on known drug interactions by 75 percent. Currently C.V. Green is receiving 100 lawsuits per year and is paying $10,000,000 in compensatory damages. Both of these measurements must be reduced by 75 percent.

Tactical
Source: Sales and Marketing
Requirement: The Prescription Interaction Project must integrate information from all C.V. Green’s locations to discover potential prescription drug interactions with medications purchased at a different C.V. Green location.

Operational
Source: Pharmacy Manager
Requirement: All attempted overrides of drug interaction warnings must be approved by a senior pharmacist. The senior pharmacist can, after consultation with the doctor or the FDA, override a potentially harmful interaction when a decision has been made that the risk of taking the drug is less than the risk of not taking the drug.
Categorizing Requirements

User
Source: Pharmacist
Requirement: Must be able to enter a customer prescription drugs purchased from other pharmacies in order to search for drug interactions.

System — Functional
Source: Pharmacist
Requirement: The system must be able to print out a complete set of drug interactions found for a specific prescription. This includes drugs available locally and for all applicable government agencies.

System — Quality of Service Requirement
Source: User and developer
Requirement: The pharmacist must be able to enter a prescription for one drug in less than 60 seconds. This assumes that there are no drug interaction warnings and that the customer is an existing customer, already in the system.

System — Assumptions and Constraints
Source: Organizational standards
Requirement: All data must be stored in the corporate Oracle™ database, accessible from all locations.

System — Implementation Requirement
Source: Customer
Requirement: Past prescriptions must be converted and available on the day of implementation. Prescriptions from the 24 hours prior to implementation can be converted and added to the system within four hours of implementation.

Similar to the stakeholder classification system discussed earlier, the sequence-oriented classification is likely to need different people involved with giving the requirements for each level. Start with regulations, industry standards, and corporate policies. Then move down through strategic, tactical, and operational requirements, and continue with the user, and finally the system. It is likely to be an iterative process though. While progressing down the list, it is likely that it will be necessary to back up and go through the process multiple times.

In this structure, the requirements should be traceable from top to bottom (and vice versa). All system requirements must come from somewhere. It could be based on a user requirement, or a business requirement, or maybe even an external requirement, but it cannot just appear at the system level. If it does, then there is scope creep or a missed requirement at a higher level. There are two ways of solving this issue. The customer can either decide that this is a valid business requirement, which means it should be added to that category, or it is not a valid requirement, which means that it should be eliminated as a system requirement.
Determined Project Requirements

6.7 Purpose-Based Classification

In this type of classification the project is classifying requirements based on the type of functionality desired. Is it behavior related? Does it relate to security? Does it define the features needed? Is special reporting needed?

Tying purpose-based categories back to what was reviewed earlier for functional requirements, one way to identify the purposed-based categories is to ask different stakeholders what they want to accomplish with the system. Another way is to think of it as functions that the business needs to perform, regardless of whether or not they are to be automated. Examples include:

- Enter order.
- Pay bill.
- Collect customer information.

From a non-functional requirements view, the purpose-based categories deal with the things that are needed because of the system. Categories such as training requirements, security requirements, and performance requirements fit in here. They are really the same as the quality-of-service requirement category mentioned above.

Purpose-based requirements are often tied in with specific job functions. What does the order clerk need? What does the Purchasing Department want? Does finance have any requirements related to this project? The advantage of this approach is that it is easy to set up an interview with the representative for that job function and document the result in its own category. However, there are often conflicts between different job functions about what they would like to have the system do, and those conflicts can be difficult to find in this type of approach.

Another purpose-based approach is to focus on key modules of the product. What are the reporting requirements? What are the data entry requirements? What are the query requirements? This is a common and effective approach when the system being developed is large and complex and contains natural modules.

6.8 Combining Structures

The business analyst must always keep in mind the purpose of using a taxonomy. It is used as a communication tool to help identify a complete set of requirements and to group related requirements. It organizes the thoughts of the analyst and of the customer. Identify as many structures as possible and review them to find applicability to the current business. There is probably more than one classification approach that will work well within the business area, but do not assume that any classification system will work. The classification system selected should reflect the customers’ view of their business in a manner which is easy for them to follow.
Customers can be frustrated if they are forced to fit a requirement into a category with which they do not agree. Don’t argue with the customer; just add a category and customize the classification system for the project. Then, as a part of lessons learned for the project, evaluate if it is a value-added category that should be kept, or if it is just a one-time event. Look at the taxonomy as any other tool that must be customized for each project and realize that the main drive behind having a strong taxonomy is to improve communication.

There is probably more than one classification approach that will work well within the business area, but do not assume that any classification system will work.

6.9 Summary

The best approach for classification is to start with a simple base, maybe a combination of stakeholder and purpose-based classification. Then evaluate and add on to the structures for each project the organization completes. This eventually builds a taxonomy that the organization is comfortable with and that reflects the uniqueness of the organization’s business. When discussing it with the customer, review the taxonomy and also have examples available for each category.

6.10 Activity

Review the case study in Chapter 11. Document a requirement for each of the following categories:

- External
- Business — Strategic
- Business — Tactical
- Business — Operational
- User
- Quality of service
- Assumption
- Constraint
- Implementation

When documenting these requirements, focus on making them specific, unambiguous, and verifiable. There will be a further discussion of how to write good requirements in Chapter 8. Write the requirements so that they will be meaningful to someone reading them three months from now without having been involved with the development effort. If you do that, many misunderstandings will be avoided in the future.