

EXHIBIT 7.2

REQUIREMENTS TRACEABILITY MATRIX

ID	REQUIREMENT	ACCEPTANCE CRITERIA	TYPE	STATUS	STAKE-HOLDER GROUP(S)	PRIORITY	OBJECTIVES
1	The BA must be able to customize the information collected for requirements		Stakeholder	Approved	BA	Must	PO#1
1.1	The system shall allow for renaming of requirement attributes	1. BA can rename an existing field 2. Field displays new name on input forms 3. Field displays new name on reports	Functional	Approved	BA	Must	PO#1
1.2	The system shall allow new requirement fields to be identified	1. BA can add a new field 2. BA can set field attributes 3. BA can indicate field lookup values 4. Custom field available for input 5. Custom field available for reports	Functional	Approved	BA	Should	PO#1
1.3	The system shall allow for lookup of allowable fields for a requirements attribute	1. BA can enter custom list of lookup value 2. Lookup fields can be provided from an external system through data interface	Functional	Approved	BA	Should	PO#1
2	The BA must be able to provide different reports for different audiences		Stakeholder	Approved	BA, Team, Sponsor, Stakeholders	Must	PO#1
2.1	The system shall include a base set of standard reports.	Reports include 1. Requirements Traceability Matrix 2. Business Requirements Documents	Functional	Approved	BA, Team, Sponsor, Stakeholders	Must	PO#1
2.2	The system shall allow a business analyst to filter reports based on various requirement attributes	1. BA can filter report based on a. Type b. Stakeholder c. Status d. Priority e. Objective	Functional	Approved	BA	Must	PO#1
2.3	The system shall provide an option to download data to an Excel supported file so the BA can customize	1. BA can select to extract data to an Excel supported file 2. Extracted data is formatted as a tabular data set with no row breaks	Functional	Proposed	BA	Should	PO#1
2.4	The system shall allow for customization of reports to include filtering and displayed fields.	1. BA can select fields to include or exclude in resulting report 2. BA can filter report (see 2.2.1)	Functional	Approved	BA	Should	PO#1

PO#1 - Project Objective #1 - "record, manage, communicate, and update requirements so that requirements can be captured once and then managed and communicated efficiently"

Priority uses MoSCoW - Must be included in release (mandatory), Should be included in release (highly desired), Could be included in release (nice to have), Won't be included in release (out of scope)

Source: Vicki James, PMP, CBAP, PMI-PBA, CSM.

customers become more fully vested in the project and creates a strong working relationship that is helpful when difficulties arise during project execution.

It is helpful to list requirements and their supporting information in a requirements traceability matrix such as that shown in Exhibit 7.2.

7-2 Collect Requirements

A **requirement** is a condition or capability needed by a user to solve a problem or achieve an objective that satisfies a standard, a specification, or any other formally documented need.

Collect requirements is a systematic effort to understand and analyze stakeholder needs to define and document these needs and requirements with a focus on meeting project objectives. The first step in collecting requirements is to ensure that the project team is clear on the project objectives. This could be accomplished by reviewing the

project charter—particularly the “why” section that justifies the project. The project team members then may describe in more depth what each believes the expected project benefits are and/or what problems the project is attempting to overcome. On simple projects, this may not take a lot of time. On complex projects, a project manager may choose to use idea generation, grouping, and/or cause-and-effect techniques to make sure that everyone on the project team understands why the project is being conducted. Understanding broad project objectives will help in making more-detailed decisions later. This also reinforces the project’s importance and may help motivate team members and other stakeholders during challenging periods in project execution. It is especially useful with multifunctional, virtual, and global project teams. Finally, a clear understanding of the project’s objectives helps if the project plan needs to be revised at some point.



Collecting requirements is the same no matter what type of project approach you undertake; however, in more iterative projects, the documentation of the requirement is normally much less formal. Agile leverages the progressive elaboration mindset that allows for the project to unfold before the implementation team. This works best when the expected outcome is unclear or customers may change their mind once they see the initial product.

7-2a Gather Stakeholder Input and Needs

The second step is to gather input from the various project stakeholders. Needs assessment begins with a high level of understanding of the client needs during the project inception. A project manager is assigned and more detailed requirements assessment is done after a project’s core team is selected. This core team size would depend on the nature of the project and the number of disciplines required to plan and execute the project.

When a project manager and team listen closely to both internal and external customers, they understand better both what their needs are and what risks and issues may confront them during the project. Successful project managers know that for a project outcome to be useful to the project’s customers, the customers need to be able to use the output to better serve their own customers in turn. In other words, end-users of the project deliverable, the product of the project, and their needs must be integral to the list of requirements.

The methods of developing deep understanding of customers and their needs vary extensively from one industry to another. The traditional methods of obtaining and documenting requirements are many, such as:

- Meetings with key stakeholders
- Interviews
- Focus groups
- Questionnaires
- Surveys
- Observations
- Prototypes
- Industry standards
- Reference documents,
- Market analysis
- Competitive analysis
- Requests from the client
- Standard specifications

For example, in new product development projects, teams often use voice of the customer (VOC) techniques to elicit the benefits and features the customers want out of the

project expressed in the customer's language. Teams using VOC try to understand the customer by not only asking questions but also by placing themselves in the customer's situation. If a project team is designing a new system that is to be used in the field, the team member should get down in the mud with the mechanic and hand the mechanic repair tools to see from the mechanic's point of view how the new system will be used.

Requirements can be classified as functional/technical and nonfunctional. The first category is usually the focus of needs assessment exercises and is centered on performance of the deliverable—such as the mechanic's needs just described. The second category includes requirements such as scalability, reliability, maintainability, and testability.

Once captured, these customer wants and needs are then stated in operational terms that the people performing the project work can use to plan that work. If the customer wants blue food coloring in a food item, the project team developing the item needs to know the precise desired shade of blue, the quality grade, the tolerance for color variation, and how the blue color may interact with other ingredients.

The project manager wants to understand how a project's success will be determined from the customer's perspective. The best way to gain this understanding (and to begin building a strong relationship with customers) is to directly ask customers. The project leaders can ask the customer(s) to specify how they will judge the quality of the project based on both functional and nonfunctional requirements.

On an information systems project, the team may use a joint application design (JAD) session to elicit customer requirements. This is often a facilitated session in which users of the software should articulate their preferences regarding how the software should work. The project manager and the team often send their understanding of the project objectives and deliverables in advance to all the users so that they are better prepared to discuss their needs and provide clarifications. Only one group of users is normally in this meeting at a time, while the project manager and the technical workers are in the session for its duration. Each possible feature of the system should be discussed. If the system is large and complicated, the amount of time that can be spent per item may be scheduled. Users often wish to talk in depth about how they want to use the system, while developers often want a detailed discussion about how they plan to create the feature. To avoid sinking into too much detail, the project manager can ask the users to start with only a high-level description of their reasons for the requested feature and then guide the discussion with the following five questions:

1. What do we not understand about the feature?
2. What is the business reason for the feature?
3. What is the impact of not providing this feature?
4. What action items need to be accomplished if we do this?
5. What impact will this have on other features of the project or elsewhere?

Exhibit 7.2 lists requirement along with other related information such as acceptance criteria for each requirement, which can be either high level or very detailed (using specification in measurable terms). The requirement type suggests whether the requirement is functional, nonfunctional, or needed by a particular stakeholder. The traceability matrix also includes the status of the requirement, its priority, and who is responsible for the requirement.

On some types of projects, the customers can provide their ideas using one of the techniques above, and the project team can be confident that the customers' wants and needs have been captured. On other projects, once the customers' viewpoint is captured, it makes sense to create a model or prototype of some sort so the customers can decide if their wishes have been fully and accurately captured. Often, this extra step helps the

When requirements are complete, each requirement needs to be:

- Traceable back to the business reason for it
- Identified with the stakeholder(s) who need it
- Unambiguous
- Qualified by measurable conditions
- Validated for its value and completion
- Bounded by constraints
- Prioritized according to value, cost, time risk, or mandate so trade-off decisions can be made if needed

Once these requirements are developed, they are translated into specifications, as shown in Exhibit 7.3.



There are several differences in gathering stakeholder input in Agile. In an Agile project, the Product Owner is the interface to the product stakeholders and is responsible for aligning stakeholders to priorities and capabilities. Agile focuses on delivering value to the customer quickly, so feedback can get to the development team quickly. This eliminates waste. Agile further assumes that the people doing the work know how to do the work, and requirement writers are not qualified to tell them how to do that work. At every iteration, the delivered product should be ready for use, if the customer would choose to do so. As you can see, this would not work for a building that does not have windows, so the type of project in which you engage will start to take shape as you see the outcome being requested. In Agile, we are more likely to produce a color of blue that seems to make sense to the team and get feedback. The goal is not to be right: it is to get feedback. In a more Agile environment, there is only the judgment of how well the product works. The customer ideally would not get involved in how the product is created. Creating a model or prototype described above is analogous to Agile delivering working software every few weeks to get feedback. If all is well, we keep going; if not, we pivot and deliver more. In Agile, the requirements are captured in a product backlog. The product manager prioritizes them on an ongoing basis. They are delivered in short iterations and reviewed with the stakeholders on a normal cadence.

EXHIBIT 7.3

REQUIREMENTS TRANSLATED INTO SPECIFICATIONS	
REQUIREMENTS	SPECIFICATIONS
• <u>Unambiguous</u>—not subject to interpretation • <u>Complete</u>—nothing left out • <u>Consistent</u>—no conflicts, which also means no duplication • <u>Modifiable</u>—amenable to change • <u>Traceable</u>—to a customer need • <u>Verifiable</u>—means provided to verify the requirement	• <u>Unique set</u>—each stated only once • <u>Normalized</u>—should not overlap • <u>Linked set</u>—shows relationships • <u>Complete</u>—nothing left out • <u>Consistent</u>—no conflicts • <u>Bounded</u>—specifies nonnegotiable constraints • <u>Modifiable</u>—amenable to change • <u>Configurable</u>—traceable changes • <u>Granular</u>—right level of abstraction
Adopted from: IEEE 1233	

7-3 Define Scope

Define scope is the process of translating stakeholder needs and requirements into detailed specifications of the project outcomes and products. Essentially, the project scope statement includes three things regarding the total scope. First, the team needs to determine both what they will deliver to the project stakeholders at the end of the project and what they need to deliver along the way to ensure they will be successful in the end. These are the deliverables—the *product scope*. For example, if a final project deliverable is a new computer program, intermediate deliverables may include an outline of what will be included and a prototype. Second, the team should decide what work needs to be accomplished to create the deliverables. This is the project work statement—the *project scope*. Third, the team needs to determine what will limit or influence the project work—such as exclusions, constraints, and assumptions.

7-3a Reasons to Define Scope

Scope definition is an important part of project planning because all other planning is based on the project scope. While the requirements collected represent the customers' statement of what they need, the defined scope is the project team's response—asking the customer, "If we provide this, will it solve your problem?" It is impossible to estimate how much a project will cost, how many (and what type of) workers will be needed, how long a project will take, what risks are involved, or what quality standards will be invoked without first understanding what work is included in the project.

Scope definition also is vital in preventing scope creep. Scope creep happens for two common reasons. First, if the scope is not clearly defined and agreed upon, it is easy to add additional work (scope creep) to the project with or without realizing that more time and resources (additional cost) will be required. Second, sometimes when a project is going as planned, a customer is so excited that he or she asks an innocent-sounding question: "Can the project output also do ... ?" The person performing the project work is often flattered and agrees without understanding the implications of making this change. In contemporary business, pleasing the customer is desirable. However, the best time to gain customer understanding is when the project team is defining the scope—not while executing the project scope work.

7-3b How to Define Scope

Scope definition can vary greatly from one project to another. For a small, routine construction project, it may be quite simple to determine what project outputs will be created and what work is involved in creating them. On other projects, such as one large company acquiring another, it may be very difficult to determine the total amount of work that needs to be accomplished. Regardless of how easy or difficult it may be to define scope and despite industry-specific methods that may be helpful in doing so, all project teams need to complete each part of this process.

LIST DELIVERABLES AND ACCEPTANCE CRITERIA The first step is to list project deliverables. The requirements elicited from the customer often lead to some of the final deliverables. Project teams need to understand that there are often multiple deliverables. For example, if a project entails constructing a house, the homeowners probably want not only the house but also documentation on systems within it, perhaps an explanation (training) on how to use certain items such as an innovative thermostat, and a warranty procedure. The project team also needs to list intermediate deliverables—those things that need to be developed while making progress to complete the project. Some of

these were probably listed in the charter, but others may not yet be identified. Then the project team needs to determine the acceptance criteria for each deliverable.

ESTABLISH PROJECT BOUNDARIES The second step in defining scope is to establish the project boundaries. Think of the project boundaries as the sidelines on an athletic field. By understanding what is in play and what is not, athletes know clearly when to play and when to stop. Likewise, project team members need to know which tasks should be executed and which tasks need not be executed.

The first part of the boundary definition is to decide which features and work elements are included (in scope) and which are excluded (out of scope). Collectively, clients and end users often request far more features and work than a project is originally planning to deliver or can deliver. Therefore, the team needs to know and decide what is included and what is not. Usually, the sponsor makes decisions regarding larger scope decisions, but the project manager and team still have many detailed scope decisions to make.

The second part is to manage expectations regarding any project. The project team members need to understand the constraints imposed on the project. If the work must be delivered by a certain date or if only limited resources are available, the project may be constrained, and the team should be careful to promise only what it can deliver. In planning, people make assumptions about dates, times, and availability of resources; for example, a shipment of required materials will arrive by the date the supplier promised. These assumptions should be stated. If an assumption proves to be false, it frequently increases the project risk and may also limit the project scope.

CREATE A SCOPE DESCRIPTION The final step is to create a **scope description**. This description briefly states the work that needs to be accomplished to create the project deliverables.

A project scope statement guides the project team during subsequent planning and execution. For some very small projects, a well-developed project charter could also serve as a scope statement. On most projects, a scope statement needs to be developed prior to development of the WBS. An example scope statement for the Alternative Breaks project is shown in Exhibit 7.4.

7-3c Defining Scope in Agile Projects



Agile strives to use smaller iterations to get feedback because understanding the desired outcome tends to evolve as the customers see the work being done by the team.

Humans tend to be poor estimators, and the more unique the project, where volumes of reliable data are not available for making estimates, the harder it is to be predictable. In construction, for example, there are software packages that help estimate how long it will take to hang drywall or run electrical wire. However, in more creative endeavors like creating software, there is little documented knowledge of how long a project will take. This is where the adage to underpromise and overdeliver becomes words to the wise.

With Agile projects, the project manager is challenged with conflicting aspirations and actions between finalizing the scope specifications and maintaining flexibility to modify them to meet changing business needs or adding new requirements of stakeholders. Agile scope definition is a complex process as the scope is not clear to either the project team or the client. The project manager and the project team must demonstrate greater adaptability to frequently changing scope and employ iterative or phased planning of scope. Consequently, Agile projects present more flexibility.

On Agile projects, the scope definition starts with large chunks of work; for example, we want to be able to take credit card payments on a website. This large feature, and

there will be many of them for a fully functioning website, will be broken down into stories and prioritized later. The team creates “personas,” which are fictional people who represent user types. These personas provide information about what they will do with the project deliverables and how they will benefit. These user stories define scope and functionality. Acceptance tests will also be agreed upon during the scope definition phase by describing the way project deliverables will be tested and how they should prove workable. At the project outset, the overall scope is only defined at a high level, and a backlog of possible work is identified. The customer representative (sometimes called the owner) prioritizes the scope based upon business need, value, cost, and risk. The team then commits to the amount of work they can perform in the first iteration. As the project progresses, the scope is described more specifically and is documented more closely. The level of documentation is less important and takes a secondary role. The primary measure of success in an Agile project is working software. The Agile method for defining scope is primarily applicable when the project scope is unclear or poorly defined.