

Defining and Managing Requirements: A Framework for Project Success and Beyond

Requirements Are Critical as Project Managers
Try to Control Project Failure Rates



11 Canal Center Plaza
Alexandria, VA 22314
HQ 703-548-7006
Fax 703-684-5189
teamRG.com

Conventional wisdom says you will never reach your destination if you don't know where you're going. However, all too often, that's exactly how projects are implemented, most often as a result of the business and technical staff being unable to communicate their needs to each other. This communication gap usually occurs early in a project, and the negative impacts manifest themselves throughout the entire life-cycle.

Extensive re-work, scope creep, missed deadlines, low morale, and required features being dropped from product releases are just some of the issues that are costly and can be traced back to a fundamental lack of understanding of the business needs.

For the project manager, unclear requirements are the cause of mounting frustration throughout the life cycle. Everything may seem fine – goals were set, the team was built, all the necessary tracking and management tools were in place, technology was selected, requirements gathered and signed off, and the development staff was off and coding.

Then it starts to become apparent everything wasn't fine after all. Faced with unclear requirements, a development staff has two options; either go back to the business and gain a better understanding of its needs, or implement best-guess. These "false starts" are time-consuming for both staffs. Requirements may also be uncovered that result in a significant re-evaluation of work effort (usually more, not less). Implementing best-guess will post-pone the issue until later in the life-cycle; at user acceptance testing it will become apparent to the business that they are not being delivered what they paid for. Obviously, these are both situations the project manager will want to avoid.

So what can be done? To prevent risk of failure, the project manager should incorporate practices that:

- Get at the true requirements
- Enforce tracking and management of requirements throughout the life-cycle
- Allow for a structured approach to the review and approval process
- Contribute to future projects so that valuable assets are not lost

Each of these four critical success factors is discussed next.

Critical Success Factor #1: Eliciting the True Requirements

The key to uncovering real requirements (and by that we mean those that meet true needs) is three-fold. You must have an efficient and effective process to guide the way requirements are gathered and documented; your process must include an architectural structure for capturing requirements information so that all parties understand what they are trying to document; the process must emphasize the use of models as a mechanism to gather the information and communicate with all stakeholders – the sponsor, the subject matter experts, and the developers.

The process you follow should provide a solid foundation for gathering, documenting, and maintaining requirements, and should eliminate the communication gap between IT and the business. This can best be achieved, we feel, by using communication tools and techniques that both IT and the business understand, including visual models. These models, while detailing complex business processes and system behavior, are not themselves complex to understand or create; thus the business is as comfortable participating in their creation and review as is the IT staff.

The emphasis on using models to elicit requirements is in contrast to processes and tools that would have you build requirements by extracting text statements from documents. It is true that some requirements – particularly initial

business objectives – lend themselves to text statements. And text statements are also a good way to capture operational requirements such as the number of users that must be supported, or the necessary levels of security.

But process requirements, functional requirements, user interaction requirements, and information requirements can best be understood using models. It might take several pages of text to describe the process of, say, taking an order. Even then, you risk missing some key business rule or forgetting to document the necessity for using a particular piece of data. For these kinds of requirements, a visual approach is preferable to a written one. If you've ever been part of a building project and reviewed architectural drawings, you'll know exactly what we mean when we say that visual models are superior to text descriptions.

Critical Success Factor #2: Tracking and Managing Requirements

Once you've gathered requirements, you need a process for tracking them and managing changes to them as they go through the design and development phases of the life cycle. It's a given that there will be changes over the life of a project, especially for complex or lengthy projects; why not plan for them? The process you follow should incorporate techniques to meet this goal. First, you must be able to track the status of each requirement. Is it approved, or only under review? Second, you must be able to trace the requirements from the origination through to their development. What was the business objective that caused this requirement to be stated? How will this requirement be realized? Third, you must know the impact of a proposed change. If this change is accepted, what other requirements will be affected and do these also need to change? Finally, you must know the history of requirements changes. How was this requirement initially stated, when did it change and why?

Critical Success Factor #3: Structuring the Review Process

Review and approval of requirements can be cumbersome and difficult to control. To be successful, the project manager must be able to make review assignments to users or groups of users and track comments that have been made. It may seem obvious, but performing review cycles via the web clearly has advantages over paper-based approaches. Your approach should include sharing and review of the information via the web. In that way, you can support any number of reviewers in any number of locations.

Critical Success Factor #4: Moving Beyond the Project – Reuse

Once a project is complete – that is the requirements have been realized – then what? Typically, there's some kind of documentation, but that's it. But if your process has reuse as one of its embedded disciplines, you can reap many advantages beyond the project close date. In this way, your process can provide support for continuous business improvement – not just one project at a time.

Reuse of project artifacts can serve multiple goals. The most obvious is that the requirements artifacts serve as documentation of the delivered functionality. If an enhancement is requested, the requirements artifacts can be used to perform impact analysis and to judge the appropriateness, cost, and complexity of implementing the request. The second goal is to provide a jump-start to subsequent projects. For example, the deliverables generated from understanding and modeling the claims process can apply to other projects that touch the claims area. Finally, the models developed for a project can become input to developing an architecture of information that can be shared enterprise-wide.

To fulfill these four important critical success factors, you need two things: first, a defined

process for the requirements lifecycle; second, a requirements definition and management tool (or set of tools) to make the process practical.

Let's talk about the process first.

A Defined Process for the Requirements Life Cycle

There are many methodologies out there, yet most are missing some crucial component: they aren't model-based, or they don't address the critical task of managing requirements throughout the life cycle, or they don't include a mechanism for the review and approval process. Look for a comprehensive, complete, repeatable, and teachable approach that can be used for many types of projects, including ones focused on:

- Building and managing enterprise architectures
- Understanding and improving business processes
- Changing organizations
- Developing new systems or enhancing existing ones
- Exploring new technologies
- Developing new product offerings

There are four basic principles of a good approach that can be summarized by the acronym "L-I-N-K". Following these principles leads to requirements that link the business side of the house with the IT side of the house.

(L) Lead with Business Understanding.

Never launch a requirements effort without first understanding the business process that the solution will be supporting. The first goal is to prevent the wrong solution from being developed and to ensure the solution meets real needs. Even in what seems at first to be a system development project, it's possible that what really needs to be changed is the process itself, or that a simple technology change such as a scanning capability can bring significant improvement.

Understanding the business process provides a context for all other project work. It can help discover business policy and business rules that may be appropriate for a rules engine. It also uncovers "process rules", that is, those rules that outline when things should occur and in what order. Finally, business process understanding helps you identify the pain points and the opportunity areas that a technical solution could address.

(I) Identify Critical Requirements.

Follow a systematic approach to identifying the requirements that the business deems important. Use repository-based tools to capture the requirements and develop the architecture (sometimes called the "meta model") so that users of the process know exactly what needs to be documented, where it needs to be documented, and how it needs to be documented. The meta model provides a template structure that makes it easier to gather information, to build relationships among the information, and to produce project documentation based on the information.

(N) Navigate the Deliverables.

Architect the business analysis and requirements deliverables so that they can be used in many ways through the life of a project and beyond. For example, use cases and data models can be transferred to an object-oriented development tool to jump-start the design effort, or they can be transformed into testing scripts. The business policy and business rules identified can be separated from other deliverables and used in development of a rules engine. The process flows can be exported to the business process management language (BPML) and used for business process management efforts, or they can help drive workflow rules. System requirements can generate user acceptance test scripts.

(K) Keep and Manage a Requirements Architecture Repository System.

Store all the deliverables created during business process analysis and requirements analysis in a repository. Having requirements artifacts in a repository means those artifacts can:

- Be shared with all project participants and published on the web.
- Be tracked – what they trace from and to, their change history, their status.
- Jump-start subsequent work. The next time a system enhancement is contemplated, you already have the requirements that represent the current system to start your work.
- Provide information to an Enterprise Architecture Repository. Each project produces deliverables – for example, business process maps – that are company assets and therefore are candidates for updating your company’s Business Enterprise Architecture. Your process should include steps to support and manage this critical aspect of the work.

The process should include a standard set of work products and deliverables for each life-cycle phase. This ensures that each project is analyzed to the appropriate level, and that no important component is overlooked. As requirements are gathered, they are modeled and captured in a repository that stores the visual models and all the project business and system rules. As modifications to the project occur during development, the requirements in the repository can be updated. Therefore, at the end of the project, the repository serves as an accurate representation of the delivered functionality.

Finding the Right Process

The processes and methodologies that are outlined in books and on web sites contain important and useful information. Yet they all have a critical drawback: they cannot take into account what is happening in your organization or the problems you are trying to solve.

We at RG have developed a methodology too, which we call LINKProcess™. At its core, LINKProcess™ encompasses best practices and approaches that take advantage of our twenty-years of experience and use of tools. Even so, we’re advocating that each organization needs its own methodology which, while based on the best thinking and best practices of experienced practitioners, is tailored to fit your organization’s style and needs.

Steps to Get There:

Initiation: The first step is for someone in the organization to recognize the need for change. Perhaps there’s been a reorganization of responsibilities; perhaps the last project did not succeed as well as had been hoped; perhaps there is a desire for new tools. Whatever the precipitating event, the organization has made a decision that the process for understanding the requirements for a new system must be improved and documented, so that it can be repeated, measured, and taught.

Assessment: With the basic goal of the organization in mind, you assess where things currently stand. This involves understanding how the processes are currently carried out, since we want to build on what already works. You also need to understand the roles that are played, as well as those roles that aren’t currently being performed and should be added, and you need to evaluate the types of projects the organization does.

Planning: This step involves forming the core team that will work on tailoring the process and determining the details (for example, meeting times and stakeholders).

Developing a Meta Model: A requirements life cycle process has two main purposes. The first is to provide guidelines for accomplishing work. The second is to outline the artifacts (definitions and models) that must be understood and captured and to structure how those artifacts should be organized into deliverables. A meta model provides the necessary structure for classifying and organizing what is significant.

Documenting the Approach: This step may take the most time in developing your process, as it involves core team meetings to hash out what roles should participate in what activities and how those activities should be sequenced. The deliverable is a set of business process models.

Trying the Approach: In the ideal world, the newly documented approach would be tested out on a real project. The knowledge earned by performing pilot project work and by having the opportunity to challenge or validate the process and its guidelines provides real-world practical ideas for improvement.

Reviewing and Revising the Approach: At the completion of the pilot project, the team should take a few days for review and debriefing. What worked well? What should be discarded? What was not clear? Was the order of tasks the best it could have been? Any lessons learned need to be documented and incorporated into the written methodology.

Incorporating Business Analysis Tools to Support the Process

In addition to having defined requirements processes, you need to identify mechanisms to improve the efficiency and effectiveness of those processes. An important piece of the solution is to

incorporate tools that provide support for three areas: communication and collaboration; structure, rigor, and completeness; and traceability and relationships.

Communication and Collaboration:

An effective requirements approach requires clear and consistent communication among business analysts, users, developers, testers and other stakeholders. This sharing of information enables meaningful collaboration and improves the quality and accuracy of requirements.

Structure, rigor, and completeness:

Requirements definition is not as simple as pulling statements out of thin air or blindly repeating the wishes of users. Developing meaningful requirements necessitates a sophisticated level of analysis, which requires the use of models, diagrams, and other business analysis artifacts to gain an in-depth understanding of the users' and stakeholders' needs.

Traceability and relationships:

Few requirements or business analysis artifacts offer sufficient detail alone. It is the combination of artifacts and their relationships to one another that provide the complete view of a system that is required for the later stages of the software development lifecycle.

To support these three needs, organizations should introduce software tools that support the full requirements lifecycle while catering to the specialized needs of the business analyst. The implementation of a specialized business analysis tool can decrease the overall time needed for requirements development, increase requirements quality, and minimize rework.

The most popular requirements definition tool today is actually Microsoft Word, although it was never intended to be one. For years, business analysts have depended on standard office

productivity software--Microsoft Word, Excel, and PowerPoint--as tools of their trade. Overextending and forcing these tools to perform tasks for which they were never designed limit our productivity and generally do more harm than good.

There is a better way, but navigating the myriad business analysis tools to find the right solution can be difficult. By understanding the types of tools that exist, identifying your organization's constraints and needs, and knowing what traits to look for, you can make an informed choice for your organization.

Business Analysis Tools 101

The majority of software products for the business analyst can be broken down into several distinct categories based on functionality and the products' places within the requirements lifecycle.

Requirements definition tools:

Software products designed to aid the business analyst in the elicitation and documentation of requirements are considered requirements definition tools. This category includes computer-aided software engineering tools, visualization products, and business process modeling software. These tools are used during the planning and requirements analysis phases of a project. Requirements definition tools provide a means to link IT and the business through standard languages, such as UML, or through the development of visualizations. These tools are adept at documenting several views of the candidate system, and some support the standard architecture frameworks, such as Zachman, DODAF, and TOGAF.

Requirements management tools:

Once requirements have been defined, requirements management tools pick up where definition tools left off. They store requirements in a single location, allow you to view relationships between requirements, and track changes.

Requirements management tools are effective when used within organizations with a mature requirements process and skilled business analysts. Unlike requirements definition tools, requirements management tools do not improve the quality of the requirements during elicitation but maintain the requirements during subsequent phases of the SDLC, resulting in fewer overlooked requirements and a tighter integration with other project artifacts.

Requirements collaboration tools:

There are few, if any, tools that focus simply on collaboration (although software such as SharePoint could be said to fulfill some collaboration needs), but the ability for stakeholders and developers to review and comment on requirements deliverables and models is an important component for success. Some requirements definition tools and some requirements management tools are beginning to incorporate a review and commenting capability.

Hybrid tools:

Some business analysis tools contain functionality supporting requirements definition, requirements management, and in the best case, requirements collaboration. Functionality may or may not be as mature in hybrid tools, but these tools provide cost savings and allow users to perform their work in a single environment.

Each type of tool has a different purpose and features suited to it. When selecting a tool, decide where your organization needs to improve and select the tool suited to that area of improvement.

Selecting Your Business Analysis Tool

A robust, repository-based tool solution that supports modeling, collaboration among stakeholders, requirements traceability, and integration with testing and application development is ideal, particularly with respect to

moderate- to high-risk projects. However, organizations and projects vary in maturity and complexity. To evaluate a tool, consider the traits both of the tool and of your organization:

Organizational maturity:

Be honest and evaluate the maturity of the business analysis practice within your organization. Conduct an assessment of your organization using a business analysis maturity model, which will offer insight into the level of complexity that your organization can support. Implementing an overly complex solution will slow tool adoption. Instead, select a tool that provides the functionality that your organization needs and can support now, but one that also will grow with you over time as business analysis maturity increases. For organizations at a starting maturity level, selecting a requirements definition tool will be far more effective than a requirements management tool.

Scalability:

Your organization will take on projects of different sizes, and not every project needs a full business analysis tool solution. Tools range in complexity from solutions for large advanced projects to back-of-the-napkin documentation tools. An ideal solution should scale to meet your project's needs.

Life Cycle Integration:

The majority of the work performed by the business analyst is undertaken during the early phases of a project, but that work influences all other phases of the SDLC. A business analysis tool—particularly requirements management tools—must be capable of supporting activities within the entire SDLC by providing the ability to trace work done during development, testing, and implementation back to requirements. If a tool cannot provide support across the SDLC, its usefulness is diminished.

Simplicity and the user experience:

Business analysts use Word, Excel, and PowerPoint for one reason: They offer a positive user experience. Any new business analysis tool should provide the same. The user interface must be easy to navigate and relatively intuitive. Commonly used functionality should be easily accessible. Look to minimize the number of mouse clicks that a user will need when performing common tasks.

Information accessibility and reporting:

Ease of compatibility and widespread use is a reason the Microsoft Office suite has continued to be used for business analysis work. Consider tools that allow users to access and share information easily either through a shared repository or through reporting capabilities that generate documents in commonly accessible formats such as RTF or HTML.

Cost:

The cost of tools varies greatly. Some are available as free downloads; others can cost up to \$10,000 a license. Evaluate the cost and benefit of each tool, focusing on the features your organization needs and the relative cost of each feature. Beware of hidden costs of tool implementations. Additional features, software add-ons, hardware upgrades, and training and support can add to the upfront cost of purchasing software.

Selecting a business analysis tool is no easy undertaking. However, by understanding the type of tool you require, your organization, and what to look for, it is possible to make an informed decision that will benefit your organization.

The Benefits of a Defined Requirements Life Cycle Process Supported by Requirements Tools

For the business, a well-thought-out and well-defined requirements life cycle process ensures that the business process and business need drives project results. This alone would make it valuable. But when combined with repository tools, the business also reaps benefits by being able to re-use process definitions and business requirements as a way to start the next project, or as part of an enterprise architecture. For IT, a well-defined requirements life cycle process can shorten development time because it prevents false starts (you know what the business wants before you start designing) and minimizes change requests. Even more important, IT can be confident that the solution they develop will meet real business needs.

For the project manager, the requirements life cycle process provides exceptional benefits throughout the life-cycle – minimizing risks while providing a central framework for describing what's being proposed, developed, and delivered. A consistent, controlled approach means that there are clear expectations on how requirements are uncovered, communicated and maintained, as well as guideposts for major activities and tasks.