

The First Key to Project Success Is Collaborative Requirements Definition and Management

Matt Light

This research examines the continued importance of requirements definition in management, even as iterative development has become mainstream and accelerated agile methods are leaving the "early adopter" phase. How requirements are gathered, then managed, is among the first and most important activities in delivering information systems projects.

Key Findings

- Active collaboration is essential for successful information system and service projects.
- Business sponsors often fail to express clear judgments about priority requirements.
- Requirements definition presents a project team (including stakeholders) with one of its earliest — and, perhaps, most important — choices as to the level of capability maturity it will bring to the project.

Recommendations

- Adopt measures to drive active collaboration on project requirements.
- Insist on clear business-sponsor guidance as to whether scope, schedule or budget is the No. 1 priority.
- Become adept at multiple modes of requirements definition, and capture requirements in manageable form.

WHAT YOU NEED TO KNOW

This document was revised on 12 August 2008. For more information, see the [Corrections page](#) on gartner.com.

Project leaders should assert the need for active (rather than passive) collaboration on requirements from all project stakeholders. Leaders should size and schedule iterations based on a project's prime driver, which is a requirement that must come from the business sponsor. Multiple techniques and tools should be used to gather manageable requirements.

STRATEGIC PLANNING ASSUMPTION(S)

By 2012, the requirements definition and management (RDM) market will grow at more than 10% annually, from approximately \$200 million in 2008 to more than \$300 million, with new entrants and continued merger and acquisition activity. We expect that RDM will further tie into application life cycle management (ALM) environments, or be featured in ALM suites, to help generate downstream development artifacts, which updates to requirements can then flag for updating.

ANALYSIS

Efficient, accurate delivery of information systems is impossible without "active collaboration" in defining and managing requirements — collaboration among IT project team members, business and executive stakeholders, intended users and, often, third-party contractors. We distinguish active from passive collaboration; most project teams take collaboration for granted, adopting the attitude that: "Of course, we'll collaborate as needed when project tasks call for it." Such a passive approach causes a lack of communication and rampant miscommunication, creating much room for missed requirements and inaccurate requirements, often costing significant rework or even causing rejection of the deliverable.

We define active collaboration as measures that enable a spirit of cooperation through intentional, defined processes; assigned responsibilities; and tool support that is structured (for example, in formal requirements or in configuration management tools with defined workflows) and unstructured (such as in e-mail, instant messaging and wikis). Ideas from agile development emphasize active, intentional collaboration, as in the way requirements are discovered incrementally and may change during the course of a project (see "Pairing Agility With Quality: Gartner's 10 Principles of NeoRAD").

In all this, the project manager should take the lead, focusing first on ensuring that the team knows what to do — that is, on gathering requirements — including functional requirements and nonfunctional requirements, such as business drivers. The first step in active collaboration is working with the project sponsor to identify the project's primary driver.

What's Driving the Project?

For many project managers in IT organizations, their "default" primary driver is usually on-time delivery — largely because on-time delivery is generally easiest to ascertain and most visible to their managers and executive sponsors. In some cases, the primary driver is budget. In the former case, some extra costs are often accepted to ensure on-time delivery; in the latter, late delivery is preferred to doubling (for example) the head count of the project team to be on time. However, most business-side stakeholders would usually give priority to the third side of the project triangle: functional scope.

Because features are often numerous, difficult to measure and sometimes implemented partly (or poorly), functional scope is not as clearly visible as a missed due date or a cost overrun. Whether functional scope has been addressed satisfactorily in a deliverable can be difficult to ascertain. Nevertheless, stakeholders whose jobs are affected by the IT deliverable generally care most about the features and functions of systems they must use. If their functional requirements are poorly met, then on-time and on-budget delivery provides scant satisfaction. On the other hand, if the expected functionality is well-implemented, then they will express higher levels of satisfaction, even if the deliverable arrives somewhat behind schedule and over budget.

The project manager is responsible for actively collaborating with the executive sponsor and other stakeholders to identify which high-level project parameter — schedule, budget or scope — is the primary driver. In fact, it is the role of the project manager to insist on a primary driver. Too often, sponsors abdicate their responsibility to define this high-level requirement, with such weak disclaimers as, "They're equally important," or "We need more information."

Certainly, the secondary drivers are important, and adjustments may be necessary as more information is gained (as in agile approaches). However, the project manager cannot do the job of scoping within schedule and budget constraints without knowing the sponsor's priority — and the project manager should not be the one making this business decision. Project sponsors should take responsibility for their roles in project collaboration, and should not render others' jobs more difficult by avoiding this responsibility.

The project manager's role is to plan the activities, including a series of requirements definition sessions, which will help him or her set such project parameters as the size of a first iteration, given time constraints. In an era of iterative projects, strong capabilities in RDM are key to sizing and sequencing iterations in keeping with business drivers. The growing adoption of iterative (and agile) approaches to delivering IT projects has required closer, faster collaboration (see "Agile Requirements Definition and Management Will Benefit Application Development").

RDM: The "Golden Mean"

Once you've ensured that your project links to your business strategy, work with the people who will use the deliverable. Identify individuals experienced in the business who truly understand users' needs to help define the requirements.

A leading indicator of IT project success is whether requirements are specified in a manageable fashion. Gathering the requirements of a new IT system (or a major enhancement to one) presents a project team with its first — and, in many ways, most important — choice. After a conversation or an interview with a project sponsor or user, many project teams proceed directly to building a solution. Such a rush to programming is undesirable, because it often results in systems so inaccurate or incomplete that they are unusable. However, project managers should also avoid a rigid engineering approach taken to the extreme of attempting to specify more than 90% of requirements in detail, before then proceeding to the building phase.

Gartner advises flexibility in finding a "golden mean," depending on the project's primary driver, which often means that the project manager should try to understand about two-thirds to three-quarters of the business requirements (although only at a high level) to drive initial budget and schedule estimates. Because 100% of the requirements are not known, this "rule of thumb" is obviously a rough estimate — one might reasonably ask two-thirds to three-quarters of what?

From this foundation, trade-offs can be made. Many IT projects are an exercise in balancing immediate priority requirements with the need to identify future design requirements and dependencies from half-seen needs (which come into focus only during the course of development). The purpose of this rule of thumb is simply to ensure that the project manager has

paid sufficient attention to the overall range of requirements to reasonably begin building cost and schedule estimates, based on the primary business driver.

The apparent clarity and stability of requirements should provide guidance as to the best-fitting delivery approach providing "just enough process" (see "Waterfalls, Products and Projects: A Primer to Software Development Methods"). Rapid application development (RAD) and agile (NeoRAD) methods used on projects do not discard RDM, but balance the requirements and the code that implements them. Using tools to provide traceability ensures that requirements are not overlooked, and that dependencies are understood. Keep in mind that requirements can be discovered and expressed in a variety of forms; for example, prototypes provide one method of enduring that requirements are solidified.

Requirements Definition Techniques

Various approaches belong in an IT organization's repertoire, with versions of waterfall, iterative development and NeoRAD approaches for high-cost and risk package implementations and for new development, as well as for less-costly and less-risky efforts, such as enhancements. If early requirements-gathering sessions reveal clear and stable requirements, a waterfall approach might be the best; if requirements are less clear, more likely to change — that is, if the project is, in part, a "voyage of discovery" — then a more-iterative or agile approach will be more appropriate (or at least a shorter waterfall).

Project managers and analysts should aim for a multidimensional view of project requirements, and should not rely on a single, narrow approach for defining them. The following techniques are useful ways to define requirements, with best results when multiple techniques are used:

- Open-brainstorming workshops, rather than closed interviews
- User-directed role playing
- Storyboarding to capture application flows
- Modeling business processes
- Developing use cases
- Simulation
- Prototyping

Repetitive requirements definition sessions should move from brainstorming to validation and should have ample user participation. Early sessions should include mostly user-side stakeholders, as well as a dedicated facilitator and a "scribe" to capture requirements in pictures, models and text. Various deliverables are outputs of the requirements process, such as business process flow diagrams, use cases, activity diagrams, context models, textual requirements documents and so forth.

Ideally, a dedicated business analyst from the user side will assist in clarifying "as is" and "to be" processes, typically through process models, and provides perspective on the business process architecture context. Although it is increasingly common to assign the project manager and analyst roles to one individual, this should be avoided on more-complex or critical projects, because one individual rarely has a strong background in both disciplines. If requirements analysis and project management are to be done by a single person, then the project sponsor and IT management should explicitly state this in the project charter, take steps to assess that person's background and skills in each role, and allow for training and coaching as needed.

During a period of transition, when a project manager/analyst job may be emerging, it will be necessary to assess and train professionals to fit the position (see Note 1), and perhaps to supplement internal capabilities with those of external service providers, including some of the boutique providers of requirements-related services (see Note 2). During requirements gathering, the more-technical system analysts and designers from the IT organization should watch, listen and learn to gain the necessary business understanding to supplement their technical skills. This often includes direct observation of users doing their jobs, as well as usability testing as development proceeds (see Note 3).

Requirements involve attributes beyond functional specifications, such as performance levels, security or just a simple version number. Other examples of nonfunctional requirements that are also security or regulatory requirements include time constraints, infrastructure constraints, budget constraints, installation and implementation requirements, performance, standards and usability requirements, and safety and environmental concerns.

Requirements, once gathered, should be tested or validated. Validation can occur by reviewing detailed use cases, or simulations, with stakeholders. During development, at least one experienced user-side product manager — the "voice of the customer" — should be chosen to help validate requirements and changes, and to be readily available to address questions about requirements. Less-experienced substitutes or IT representatives who supposedly know the users' needs are insufficient.

Users and other stakeholders best understand their systems' performance, reliability, security and other needs. In most NeoRAD approaches — including such agile methods as scrum and Dynamic System Development Method (DSDM) — validation is performed by giving users an accelerated implementation of working software — usually a small increment of the desired functionality, in short iterations.

Ongoing Requirements Management

Whether in documents or process models, requirements definition should not be viewed merely as a step in development that, once completed, feeds the next step. Rather, requirements definition should be part of ongoing requirements management. Requirements should be traceable to system functions, and each function should be traceable to a requirement, or it could yield unspecified behavior.

As requirements are revised, changes can have unforeseen consequences, unless traceability allows impact analysis — for example, among parent-child relationships and dependencies, including links to non-IT projects. This process can be simplified, if requirements are captured in a database-based requirements management tool. Such tools enable a collaborative review for completeness, use-case creation, test case creation, traceability and facilitation of versioning/change control.

The most rigorous traceability is practiced by makers of high-reliability systems, embedded systems and manufactured goods, including the U.S. Department of Defense and healthcare industry contractors, to ensure that no unintended system behavior occurs. These have been major customers of traditional requirements management tools.

Requirements definition is a new capability offered by various innovative tools, as well as by more-traditional modeling tools. A converged market for RDM is emerging, as providers of requirements management tools increasingly offer features or complementary tools for requirements definition, and some pure-play requirements definition providers have begun to build or integrate with requirements management tools (see Note 4 and Note 5).

We expect the RDM market to grow at more than 10% annually, from about \$200 million in 2008 to more than \$300 million by 2012, with new entrants and continued merger and acquisition activity. We also expect that RDM will further tie into ALM environments, or be featured in ALM suites, beyond integration with test management or other modules. RDM will also generate downstream development artifacts (such as use cases, test cases and schedules), which updates to requirements can then flag for updating.

RDM is not limited to individual projects. It's a process that must be measured to be improved. Various metrics are worth considering (see Note 6).

Note 1 Business Analyst Certification

The role of the business analyst is evolving from what was once known as the system analyst role. Business analysts help build a clear understanding of requirements by modeling established processes and redesigning them in support of business change. Demand for business analyst assistance in requirements definition increases when business processes face substantive changes or redesign; when business processes must be coordinated, shared and intertwined across the company; and when performing custom application development, for example, in the fields of insurance and commercial banking.

Reflecting the growing prevalence of business analysts, new means of training and certifying business analysts have appeared. The International Institute of Business Analysts (IIBA) offers a certification to business analysts who can show five years of work experience, including direct experience in four out of six knowledge areas, as defined in the IIBA's "Business Area Body of Knowledge" (BABOK). The six areas are

- Business Analyst Plan and Monitor
- Enterprise Analysis
- Elicit
- Requirements Analysis
- Solution Assessment and Validation
- Requirements Management and Communication.

Two references and 21 hours of professional development are also required.

Several academic sources of training and certification have also emerged. These include Boston University, George Washington University, Northwestern University, Teachers College-Columbia Corp. Ed., University of California Irvine, University of North Carolina Charlotte and Villanova University.

Note 2 Boutique Providers of Requirements-Related Services

- Advanced Management Services
- Atlantic Systems Guild
- Doreen Evans Associates
- IAG Consulting

- Pierson Requirements Group
- Process Impact
- Requirements Quest
- Requirements Solutions Group

Note 3

The Users' Bill of Rights

- We shouldn't have to read a manual — certainly not a huge one.
- We should be able to accomplish every task and entry with the fewest possible keystrokes.
- We should be able to do things out of order without being penalized.
- We should be able to make mistakes without being terminated, executed, canceled, rebooted or erased.
- We should be able to understand why the program does what it does.
- We should be able to expect that all of what we type into the computer is saved, by default.
- We should be forewarned when any work is overwritten, undone or erased.
- We should expect to have most of our work retained after power is interrupted.

Source: Usability Institute

Note 4

Sample Requirements Management Tool Providers

- Borland — Caliber Analyst (RM Module)
- Compuware — OptimalTrace
- IBM — RequisitePro (Rational), DOORS (Telelogic)
- Jama Software — Contour
- MKS — Integrity
- Serena — DimensionsRM
- TCP Sistemas e Ingeneria — IRqA
- Siemens PLM Software — Teamcenter

Note 5

Sample Requirements Definition Tool Providers

- Atlantic Systems Guild — Volere
- Axure Software Solutions — RP Pro

- Borland — Calibre Analyst (DefineIT module)
- Gorilla Logic — GXE
- iRise — Application Simulator
- Ravenflow — Raven
- Serena — Composer (Mashup Composer; Prototype Composer)
- Blueprint — Blueprint

Note 6
Improving RDM

To improve in any area, a baseline must be measured. In RDM, metrics include:

- Scope creep as a percentage of budget or requirements
- Rework due to requirements errors of commission
- Functional fit as a customer satisfaction survey question

REGIONAL HEADQUARTERS

Corporate Headquarters

56 Top Gallant Road
Stamford, CT 06902-7700
U.S.A.
+1 203 964 0096

European Headquarters

Tamesis
The Glanty
Egham
Surrey, TW20 9AW
UNITED KINGDOM
+44 1784 431611

Asia/Pacific Headquarters

Gartner Australasia Pty. Ltd.
Level 9, 141 Walker Street
North Sydney
New South Wales 2060
AUSTRALIA
+61 2 9459 4600

Japan Headquarters

Gartner Japan Ltd.
Aobadai Hills, 6F
7-7, Aobadai, 4-chome
Meguro-ku, Tokyo 153-0042
JAPAN
+81 3 3481 3670

Latin America Headquarters

Gartner do Brazil
Av. das Nações Unidas, 12551
9º andar—World Trade Center
04578-903—São Paulo SP
BRAZIL
+55 11 3443 1509