

THE BUSINESS ANALYST: THE PIVOTAL IT ROLE OF THE FUTURE

Kathleen B. Hass, PMP
Project Management Practice Leader
Management Concepts
8230 Leesburg Pike
Vienna, Virginia 22182
tel 303.663.8655
KHass@managementconcepts.com

Abstract

Change is the norm, fierce competition is the driver, and lean thinking is the latest call to action. Information Technology (IT) has finally come into its own, now viewed as a value provider as opposed to a cost drain. With the stakes so high, IT organizations are faced with an extraordinary combination of pressures to deliver value to their organizations in terms of added revenues, avoided costs, lower taxes, higher productivity, less employee turnover, less risk exposure, etc. Competitive advantage is now linked to an organization's ability to rapidly deploy IT solutions, and to change those systems as the business need evolves. IT projects must not only deliver high quality products faster, better, and cheaper (traditionally the responsibility of the project manager), they are also under intense scrutiny to positively impact the bottom line (increasingly, the joint responsibility of the project manager, project sponsor, and the business analyst).

Introduction

Projects play an essential role in the growth and survival of organizations today. It is through projects that we create value in the form of improved business processes and new products and services in response to changes in the business environment. Since data and information are the lifeblood of virtually all business practices, IT projects are often the key mechanism used to turn an organization's vision and strategy into reality. Executives have their eye on the IT portfolio to ensure that they: (1) invest in the right mix of projects, (2) optimize their resources, (3) develop expert capabilities to deliver flawlessly, and ultimately, (4) capture the expected added value to the business.

Since there appears to be a neverending demand for new IT products and services, executives across the spectrum are adopting the practice of *superior business analysis* to increase the value IT projects bring to the business. The talents, competencies, and heroics of project managers and technologists alone cannot drive value into the organization. For business needs and goals to be converted into innovative solutions that truly bring wealth to the enterprise, a stronger bridge must be built between the business and the technical communities.

The Project Performance Partnership

Every IT project, whether in-house or outsourced, COTS purchase or custom development, or complex systems integration, needs exceptional requirements management. In the spirit of high-performing teams, business analysts align themselves with professional project managers, the best developers, and business visionaries to determine the most appropriate, cost-effective, and innovative solution. As this core team forms, a *project performance partnership* emerges that rivals the world's great teams, (e.g., tiger teams, special operations teams, professional sports teams, and parametric teams). At the center of the team is the dynamic twosome: the project manager and the business analyst. One has an eye on the management of the *project*, while the other focuses on management of the *business requirements*. The wise project manager welcomes this teaming trend, understanding that inadequate information relating to requirements leads to poor estimates, and makes time and cost management virtually impossible.

Why is the business analyst emerging as *the* central IT competency of the future? Because requirements play a vital role in engineering IT systems, and five of the top eight reasons why projects fail are tied directly to poor requirements (The Standish Group, 1998). It is the business analyst who manages the entire *Systems Requirements Life Cycle*, from understanding the business need to ensuring that the delivered solution meets the need and adds value to the bottom line. Clearly, the



business analyst has a critical role throughout the Business Solution Development Life Cycle, not simply during the requirements phase. (See Figure 3, *Business Solution Development Life Cycle Mapped to the Systems Requirements Life Cycle*, on the final page of this White Paper.)

It is through the leadership of the business analyst that requirements are captured and *fully understood by the technical team* before solutions are designed and implemented. The business analyst serves as the liaison between the business community and the technical solution providers throughout the project life cycle. As projects become larger, cross-functional, global, and more complex, organizations are realizing that requirements management skills are indispensable. With the current trend in outsourcing IT development, the role of the business analyst becomes even more critical in today's IT environment.

Enter the Professional Business Analyst

When you hear about far-reaching innovation, cutting-edge technology, and high-growth IT careers, don't just think in terms of architecture and development prowess. We are discovering that technical skills can be relatively easy to outsource, but organizations cannot abdicate control of their business requirements. In virtually every organization, the pivotal leadership role of the business analyst is beginning to shape the future of IT. So don't blink — or you'll miss out on the IT opportunity of a lifetime.

Once management has developed a portfolio of valuable IT projects, the focus is on flawless project execution to maximize the value delivered to the organization. All too often, however, IT project success is elusive. Projects are late, over budget, or may never even be delivered. Sometimes work is incomplete, does not meet requirements or expectations, and does not deliver the benefits or returns on investment expected by the organization. This rather dismal IT delivery record can no longer be tolerated.

What are the key *organizational capabilities* executives must build to improve delivery performance and meet business expectations? For project success, several capabilities are essential:

- Effective and targeted project management and systems engineering processes, tools, and techniques
- Appropriate executive decision making at key control gates
- Exceptional project leadership and high-performing teams
- Collaboration and respect between the business and IT communities, and
- Business analysis processes that ensure the development team will have a clear understanding of the customer's overall business and information needs.

Where do we get the exceptional business analysts who can bridge the chasm between the business and technical communities? As the project management discipline matures

into a strategic business practice, so must our business analysts evolve into strategic leaders of change.

Will the Real Business Analyst Please Stand Up?

Frequently, expertise in the technical area of the project is the key requirement for the position of business analyst. In this case, business analysis is treated as a subset of the technical discipline. Time and again, projects encounter difficulties not from lack of technical expertise, but from an inability to gather, understand, analyze and manage business requirements, and convert them into useable system specifications. Projects are often initiated, and design and construction of the solution is underway, before IT team members have a clear understanding of the business need. Often, tolerance is low for technical failure and high for inadequate and ever-evolving requirements. All too often, IT projects suffer from requirements creep due to the "Let's start coding and see how it turns out" syndrome. While this may be appropriate when conducting agile development, it often falls short for complex business systems initiatives.

Business requirements analysis differs from traditional information systems analysis because of its focus, which is exclusively on *adding value to the business*. In particular, project managers rely on business analysts to assist in providing more detailed project objectives; business needs analysis; clear, structured, useable requirements; trade-off analysis; requirement feasibility and risk analysis; and cost-benefit analysis. Poor requirements definition emerges without this key liaison between business and IT departments, resulting in a disconnect between what IT builds and what the business needs.

To meet the challenge, technically adept engineers often are asked to make the professional transition to the disciplines of project management and business analysis. Often, these individuals assume a trio of leadership roles on projects: technical lead, project manager, and business analyst. Inevitably, after requirements are captured at a high level and the project plan is being executed, technical activities tend to elicit the majority of attention. When that happens, requirements and project management suffer, and the initiative is positioned to become a runaway project.

From Analyst to Project Leader

It is increasingly clear that while technical knowledge areas are necessary, they are insufficient for successfully managing requirements on the large, enterprise-wide, complex, mission-critical projects that are the norm today. Just as a business leader must be multi-skilled and strategically focused, business analysts must possess an extensive array of leadership skills. The business analyst is now assuming a leadership role, and is quickly rising to a senior position in the enterprise (whether placed in the business unit or the IT organization). As the IT contribution moves beyond efficiency to business effectiveness, the business analyst becomes the central figure on the project team who must be "bi-lingual" in speaking both business and



technical languages. To perform in this pivotal role, the business analyst must possess a broad range of knowledge and skills. Browsing through the more than 5,000 job postings for business analysts on *Monster.com* turned up this job description:

“The main purpose of the role will be to design and specify innovative solutions which meet the business requirements allowing the business benefit to be attained; and to facilitate divisional communication and awareness of the standards and quality expectations within the System Analyst teams.”

Many job titles were also uncovered, including business analyst, business systems analyst, business system planner, and even principal solutions architect. Regardless of the job title, a strong, experienced business analyst is critical to IT project success. Simply put, without a well-understood and documented requirements baseline, it is virtually impossible to meet project objectives. A *baseline* is the set of functional and supplemental requirements that the project team has committed to implement in a specific release (Wiegers, 2003). It has been said that if an IT organization only has resources

and budget to put into a single life cycle area to improve project performance, that area should be requirements definition and management. Depending on the level of responsibility and placement in the organization, business analyst duties include the following:

- Identify and understand the business problem and the impact of the proposed solution on the organization's operations
- Document the complex areas of project scope, objectives, added value or benefit expectations, using an integrated set of analysis and modeling techniques
- Translate business objectives into system requirements using powerful analysis and modeling tools
- Evaluate customer business needs, thus contributing to strategic planning of information systems and technology directions
- Assist in determining the strategic direction of the organization
- Liaise with major customers during preliminary installation and testing of new products and services
- Design and develop high quality business solutions

Technical	Analysis	Business	Leadership
Systems engineering concepts and principles	Fundamentals of business analysis	Business process improvement and reengineering	Fundamentals of project management
Complex modeling techniques	Ability to conceptualize and think creatively	Strategic and business planning	Capacity to articulate vision
Communication of technical concepts to non-technical audiences	Techniques to plan, document, analyze, trace, and manage requirements	Communication of business concepts to technical audiences	Organizational change management; management of power and politics
Testing, verification, and validation	Requirements risk assessment and management	Business outcome thinking	Problem solving, negotiation, and decision-making
Technical writing	Administrative, analytical, and reporting skills	Business writing	Team management, leadership, mentoring, and facilitation
Rapid prototyping	Cost / benefit analysis	Business case development	Authenticity, ethics, and integrity
Technical domain knowledge	Time management and personal organization	Business domain knowledge	Customer relationship management

Figure 1, Business Analyst Knowledge and Skill Set Requirements

While the business analyst is fast becoming a relatively senior position in the business world, historically it has been considered a mid- to low-level role. A recent survey revealed an increasing demand for senior individuals who can perform the ever-widening range of business analysis functions. Since business analysts walk in both business and IT worlds, they arrive from various fields. Some come from the ranks of programmer/analyst positions, while others have conventional business expertise supplemented by some IT training. To successfully fill the business analyst role, one must acquire

mastery of a unique combination of technical, analytical, business, and leadership skills. (See Figure 1, Business Analyst Knowledge and Skill Set Requirements.)

Business Analysis in Practice

During the requirements discovery and definition, requirements are determined and documented at a very high level. The requirements gathering process explores the solution without

commitment to any specific product selection. Requirements definition is most effective as a joint effort among users, customers, stakeholders, and developers (Hadden 2003). Defining, analyzing, and documenting requirements is a highly creative and iterative process that is designed to show *what* the system will do, but not *how* it will be done. Therefore, the requirements in their textual and graphical form represent a *model* of the system, serving as an intermediate step between the business need and the solution design.

The requirements development process is typically subdivided into *business need identification, scope definition, elicitation, analysis, specification, documentation, validation, management, and maintenance and enhancements*. These subdisciplines encompass all the activities involved with gathering, evaluating, and documenting requirements (Young, 2001).

Business Need

As initiatives are selected, project sponsors and project managers are assigned to new programs and supporting projects. Pre-project analysis is required to determine the most appropriate solution to achieve the strategic goals. Activities include more detailed analysis of the business need, feasibility studies, solution trade-off analysis, and development of high-level business requirements. The results of these analyses are often captured in *Feasibility Studies, Benchmark Studies, Competitive Analysis Reports, Needs Analysis* and high-level *Business Requirements* documents.

Business Domain Scope Definition

Initial requirements definition typically originates in the early stages of the project when the product description is created, and is ideally captured in initiating documents: the *Business Case, Project Charter, or Statement of Work*. All requirements should be traceable to these original sources. Prior to eliciting requirements, the business analyst and project manager partner to conduct initial planning and scoping activities to: (1) gain perspective of the needs and environment of customers, users, and stakeholders; (2) review, or *create if non-existent*, the Business Case, Project Charter, and Statement of Work (or similar scope definition document); (3) understand the business vision, drivers, goals, and objectives for the new/changed system; (4) assemble and educate a requirements team comprised of key business and technical stakeholders; (5) understand and document the scope of the project; (6) define the documents and models to be produced, and begin to develop the *Requirements Management Plan*; (7) define/refine the checklist of requirements activities, deliverables, and schedule; and (8) plan for change throughout the life cycle.

Requirements Elicitation and Discovery

Requirements are always unclear at the beginning of a project. It is through the process of progressive elaboration that requirements evolve into maturity. Requirements elicitation involves conducting initial requirements gathering sessions with customers, users, and stakeholders to begin the documentation process. Requirements gathering techniques include: discovery sessions, interviews, surveys, prototyping, review of existing

system and business documents, and note taking and feedback loops to customers, users, and stakeholders.

Business requirements are the essential activities of the enterprise that must be supported by the system. Business requirements are derived from business goals. The development of *business scenarios* is an effective method for understanding business requirements. A critical success factor in the value of the system after deployment is the extent to which it supports business requirements and facilitates the organization in achieving business goals. Elicitation and discovery activities include:

- Identifying the customers, users, and stakeholders to determine who should be involved in the requirements gathering process
- Understanding the business goals and objectives to identify the essential user tasks that support the organizational goals. Successful systems support the business requirements and facilitate achievement of organizational goals
- Identifying and defining requirements to understand the needs of the users, customers, and stakeholders. This is the activity of capturing the business requirements for a target system, as viewed by the customers, business users, and stakeholders. Business requirements are the critical activities of an enterprise that must be performed to meet the organizational objectives, and they are solution independent. Business scenarios (a.k.a., usage-based scenarios) are often used as a technique to ensure an understanding of business requirements.
- The *Business Requirements Document* and the *Requirements Management Plan* are the key outputs of this activity.

Requirements Analysis

Requirements are first stated in simple terms, and are then analyzed and decomposed for clarity. *Requirements analysis* is the process of structuring requirements information into various categories, evaluating requirements for selected qualities, representing requirements in different forms, deriving detailed requirements from high-level requirements, and negotiating priorities. Requirements analysis also includes the activities to determine required function and performance characteristics, the context of implementation, stakeholder constraints and measures of effectiveness, and validation criteria. Through the analysis process, requirements are decomposed and captured in a combination of textual and graphical formats. Analysis represents the middle ground between requirements and design (Ambler, 2005). Analysis activities include:

- *Studying* requirements feasibility to determine if the requirement is viable technically, operationally, and economically
- *Trading off* requirements to determine the most feasible requirement alternatives
- *Assessing* requirements feasibility by analyzing requirement risks and constraints and modifying requirements to mitigate identified risks. The goal is to reduce requirement risks through early validation prototyping techniques



- **Modeling** requirements to restate and clarify them. Modeling is accomplished at the appropriate usage, process, or detailed structural level
- **Deriving** additional requirements as more is learned about the business need
- **Prioritizing** requirements to reflect the fact that not all requirements are of equal value to the business. Prioritization may be delineated in terms of critical, high, average, and low priority. Prioritization is essential to determine the level of effort, budget, and time required to provide the highest priority functionality first. Then, perhaps, lower priority needs can be addressed in a later release of the system.

Requirements Specification

Requirement specifications are elaborated from and linked to the structured requirements, providing a repository of requirements with a completed attribute set. Through this process of progressive elaboration, the requirements team often detects areas that are not defined in sufficient detail, which unless addressed can lead to uncontrolled change to system requirements. Specification activities involve identifying all the precise attributes of each requirement. This process ensures an understanding of the relative importance of each of the quality attributes. The system specification document or database is an output of the requirements analysis process.

Attributes are used for a variety of purposes including explanation, selection, filtering, and validating. In addition, attributes enable the association of data with objects, table markers, table cells, modules, and projects (Stevens, Brook, Jackson, & Arnold, 1998). Attributes may be user defined or system defined. Attributes allow the requirements team to associate information with individual or related groups of requirements, and often facilitate the requirements analysis process by filtering and sorting. Typical attributes attached to requirements may include:

- **Unique identifier** that does not change. The reference is not to be reused if the requirement is moved, changed, or deleted
 - **Acceptance criteria** describe the nature of the test that would demonstrate to customers, end users, and stakeholders that the requirement has been met. Acceptance criteria are usually captured from the end users by asking the question, “What kind of assessment would satisfy you that this requirement has been met?”
 - **Author** of the requirement refers to who wrote it
 - **Complexity** indicates how difficult the requirement will be to implement
 - **Ownership** specifies the individual or group that needs the requirement
 - **Performance** addresses how the requirement must be met
 - **Priority** of the requirement rates its relative importance at a given point in time
 - **Source** of the requirement identifies who requested it. Every requirement should originate from a source that has the authority to specify requirements
- **Stability** is used to indicate how mature the requirement is. This is used to determine whether the requirement is firm enough to begin work on it
 - **Status** of the requirement denotes whether it is proposed, accepted, verified with the users, or implemented
 - **Urgency** refers to how soon the requirement is needed

Requirements Documentation

Requirements documentation must be clear and concise since it is used by virtually everyone in the project. Selected types of requirements may need to be expressed formally using technical language, e.g., legal, safety, and security requirements. This is acceptable, as long as they are mapped back to the requirements that are more easily understood. However, in most cases the language used to document system requirements should be as non-technical as possible. A diagram can express structure and relationships more clearly than text, whereas for precise definition of concepts, clearly articulated language is superior to diagrams. Therefore, both textual and graphical representations are essential for a complete set of system requirements. Transforming graphical requirements into textual form can make them more understandable to non-technical members of the team. This is one of the few times in the system life cycle when duplication is advisable.

Requirements are categorized into types depending on their source and applicability. Understanding requirement types helps in analyzing and prioritizing requirements. While some requirements are mandatory, others may be nonessential. Understanding requirement types also enables the technical team to conduct trade-off analysis, estimate the system cost and schedule, and better assess the level of changes to be expected. Finally, reviewing the list of requirements types can aid the business analyst in identifying areas that may require further investigation. Typically, requirements are broadly characterized as *functional* or *supplemental* (a.k.a. nonfunctional).

- **Functional requirements** describe capabilities the system will be able to perform in terms of behaviors or operations – a specific system action or response. Functional requirements are best expressed as a verb or verb phrase. Functional requirements are written so as not to unnecessarily constrain the solution, thus providing a firm foundation for the system architects.
- **Supplemental requirements** stipulate a physical or performance characteristic and serve as constraints on system capabilities. Constraints pose restrictions on the acceptable solution options. Constraints may include the requirement to use a predetermined language or database, or specific hardware. Constraints may also specify restrictions such as resource utilization, message size and timing, software size, maximum number of and size of files, records, and data elements. **Business constraints** include budget limitations, restrictions on the people who can do the work, skill sets available, etc. **Technical constraints** include any enterprise architecture standards to which the system must adhere.

Documentation activities involve translating the collective requirements into written requirements specifications and



models in terms that are understood by all stakeholders. This task typically involves substantial time and effort as each stakeholder may have different expertise, perspectives, and expectations of the requirements.

Requirements Validation

Requirements validation is the process of evaluating requirement documents, models, and attributes to determine whether they satisfy the business needs and are complete enough that the technical team can commence work on system design and development. The set of requirements is compared to the original initiating documents (business case, project charter, or statement of work) to ensure completeness. Beyond establishing completeness, validation activities include evaluating requirements to ensure that design risks associated with the requirements are minimized before further investment is made in system development. An often-used analysis technique to validate requirements is prototyping.

Requirements Management

A major control gate for projects occurs upon exiting the requirements phase and transitioning to requirements management. This involves presenting requirements for review and approval at a formal control gate review session. At this point, the project schedule, cost, and scope estimates are updated, and the business case is revisited, to provide the salient information needed to determine whether continued investment in the project is warranted. Upon securing approval to proceed, the business analyst baselines the requirements, implements a formal requirements change control process, and transitions into requirements management activities in support of solution design efforts.

Who is responsible for managing requirements? All too often, there is a fatal flaw in effectively managing IT project requirements. Once defined, requirements changes must be managed throughout the business solution life cycle. In addition to managing requirements, the relationship between the product scope and project scope must be understood and managed. For example, a critical project often merits formal, thorough, and time-consuming scoping activities, while a routine project might require substantially less documentation and analysis. The business analyst and project manager work collaboratively to define and manage the product and project scope. Requirements management activities include:

- *Allocating* requirements (also referred to as partitioning) to different subsystems or sub-components of the system. Top-level requirements are allocated to components defined in the system architecture such as hardware, software, manual procedures, training, etc.
- *Tracing* requirements throughout system design and development to track where in the system each requirement is satisfied. As requirements are converted to *design documentation*, the sets of requirements documentation, models, specifications, and designs must be rigorously linked to ensure that the relevant business needs are satisfied.
- *Managing changes* and enhancements to the system. Managing requirements involves being able to add, delete, and modify requirements during all project phases.

- The business analyst continues to facilitate the *validation and verification* of requirements throughout the project. This purpose of verification and validation is to ensure that the system satisfies the requirements, as well as the specifications and conditions imposed on it by those requirements.
 - *Validating* requirements to provide evidence that the designed solution satisfies the requirements through user involvement in testing, demonstration, and inspection techniques. The final validation step is the user acceptance testing, led and facilitated by the business analyst.
 - *Verifying* requirements to provide evidence that the designed solution satisfies the requirements specification through test, inspection, demonstration, and/or analysis.

System Maintenance and Enhancement

The business analyst's job does not end when the IT solution is delivered and operational; she also maintains key responsibilities in the following areas (Mooz, Forsberg, & Cotterman, 2003):

- *Maintenance* – service provided to prevent and correct defects in the IT system
- *Enhancements* – changes to increase the value provided by the system to the business
- *Operations and Maintenance* – the phase in which the system is operated and maintained for the benefit of the business. Documentation produced in this phase includes system validation procedures, system validation report, maintenance reports, annual operational reports, and deactivation plan and procedures. The business analyst plays a major role in managing enhancements to the system, and in determining when the system should be replaced and therefore deactivated.

Requirements Engineering Considerations

To understate the obvious, requirements engineering is a difficult and risky business. Ideally, we would like to get a clear and thorough picture of the requirements before development, obtain customer sign-off on these requirements, and then set up procedures that limit requirements changes following sign-off. However, regardless of the care taken in requirements engineering, requirements are going to change due to several circumstances:

- The business environment is dynamic. In today's economy, fundamental business forces are rapidly changing the value of system features. What now might be a good set of requirements may not be so in a few months or a year.
- Everything in IT systems development depends on the requirements. The assumption that fixed requirements are not the norm also means the baseline plan is subject to change.
- Estimation is difficult for IT projects as they are basically research and development endeavors. The nature of IT systems is intangible, and the real value is difficult to predict.

Realizing the difficulty in defining and managing requirements, and having described the business analysis practices, it is imperative that we discuss three additional concepts: *agile*



development, the *iterative* nature of requirements generation and system development (especially for software-intensive systems), and the element of *scalability*.

Agile Development

Over the past few years, there's been a rapidly growing interest in agile (a.k.a. "lightweight") methodologies. Described as an approach to rid IT development of burdensome bureaucracy (Fowler, 2003), or alternatively a license to hack, agile methodologies have generated interest throughout the IT world.

The emphasis in agile methods differs substantially from traditional, heavyweight engineering methods. The most notable divergence is that they are less document-oriented, usually emphasizing a lesser amount of documentation for a given task. Moreover, agile methods often spotlight source code as a key part of documentation. Additionally, there are two more fundamental distinctions:

- **Agile methods are adaptive rather than predictive.** Engineering methods plan out a large part of the solution in great detail, and then manage changes throughout the project. Agile methods attempt to adapt and thrive on change.
- **Agile methods are people-oriented rather than process-oriented.** The goal of engineering methods is to define a process that is repeatable and independent of the development team. Agile methods focus on the skill of the development team, trying to make the process more tightly support the team in its work.

The world of agile analysis challenges business analysts to become the *communication mentors* and *coaches* of project teams. To do this, one of the tenets of agile development must be followed: that of *active stakeholder participation* throughout the project life cycle. The focus changes from our venturing forth to find out what customers want, to helping them determine what they want and *need*. The obvious enabler to active stakeholder participation is co-location of the business and development team. However, the business community cannot always free critical resources to work with the development team on a full-time basis. In this case, the business analyst will conduct interviews and workshops with the business community in its own environment, with key members of the development team present to hear "the voice of the customer."

What is agile analysis? The business analyst follows the same practices outlined above, while incorporating these traits (Ambler, 2005):

- **Communication Rich.** Analysis is communication rich, valuing face-to-face meetings and teleconferencing over documentation and e-mail.
- **Highly Iterative.** Agile analysis is highly iterative. Analysis and design activities are dependent on each other, and in practice are matured in an iterative manner. Indeed, since estimating is part of analysis, it is impossible to estimate the cost of a solution without knowing the solution design.
- **Constant Feedback.** Agile analysis is highly incremental, so that components of the solution can be implemented for customer feedback before committing to further investment

in development. Hence, estimation and prioritization of requirements in increments is a must. This approach facilitates trade-off analysis and critical decision making on the part of the customer.

- **Just Enough.** Agile analysis follows the premise that good is good enough. It is the art of applying just the right amount of rigor—no more and no less.

Ambler presents this definition of agile analysis:

"Agile analysis is a highly iterative and incremental process where developers and project stakeholders actively work together to understand the domain, to identify what needs to be built, to estimate that functionality, to prioritize the functionality, and in the process, optionally producing artifacts that are just barely good enough."

So, when is it appropriate to use agile methods (Fowler, 2003)? Current thinking suggests that these methods should be used when the following conditions are present; the absence of one or more of these circumstances will likely put the agile approach at risk.

- **Transitioning to More Rigor.** When you have been following the code-and-fix method, using agile methods will apply some discipline to the process. The agile approach has the advantage of being easier to implement than a more rigorous method. Much of the advantage of agile methods stems from their light weight. Simpler processes are more likely to be followed when little or no process had been employed in the past.
- **Small Core Team.** The development team must be small, high-performing, dedicated full time, highly skilled, and empowered to make most project decisions.
- **Unknown Requirements.** Agile approaches are appropriate when requirements are uncertain or volatile. Logic dictates that if requirements are unstable, you cannot have a stable design, or be able to rigidly adhere to a planned process.
- **Highly Invested Stakeholders.** It is important for the customer to understand that when requirements change, following a predictive process is risky. In addition, the customer must be willing to be involved during the entire development process.
- **Incremental Development.** Agile methods work well when you are working iteratively and incrementally.

Iteration

Although the steps appear to be sequential in this discussion of business analyst practices, they are unquestionably performed iteratively. Iterating is the best defense when attempting to control an unpredictable process. The business analyst needs to build in candid feedback mechanisms at frequent intervals in order to reveal the status of requirements and development.

The key to this feedback is an iterative approach to requirements generation. During the requirements phase, the IT architects are working on early iterations of the solution design. As the business analyst conducts requirements trade-off



analyses, the architect does the same on solution options. Thus, prototyping is the first step in iterative development.

Whether called incremental, evolutionary, staged, or spiral methodology, these techniques are iterative in nature. Early prototypes are produced, followed by incremental working versions of the final system containing a subset of the required features. These working sub-systems possess limited functionality, but are otherwise true to system requirements. The value of iterative development is found in regular customer reviews and feedback following each iteration.

The best validation that requirements have been met is a tested, integrated system. Documents and models often contain undetected defects. When users actually work with a system, flaws become evident, whether caused by a system defect or a misunderstood requirement.

For the project manager, a new approach to planning is essential. Rolling wave planning is the order of the day, where short-term plans cover a single iteration and are quite detailed, while later iterations are planned at only a high level. Iterative development provides a firm foundation in each increment that becomes the basis for later waves of plans.

Iterative, agile, incremental development is the latest defense in the quest for IT achievement. Results can be dramatic. Business value is delivered faster and cheaper. Customers can see constant progress. Frequent feedback keeps the project aligned with business needs since flexibility and change are built into the project. Value-based prioritization ensures the most important features of the solution are delivered first. Feedback is central; it's about learning faster, not working faster.

Scalability

Whether a light or heavy methodology is used, all the activities discussed above are performed. They are likely to be executed in a broad sense at project initiation, and progressively elaborated as the project traverses its life cycle. For small, straightforward projects that are easily understood, a minimal amount of requirements documentation and models is appropriate. Indeed, the rule is that the smaller the team, the less formal the documentation. However, for significant, complex, high-risk projects, a full set of approved requirements documentation is in order. For low-to-moderate risk projects, the rigor should be scaled appropriately, applying more formality and structure in the higher risk areas of the project. See *Figure 2, Project Sizing Grid*. The related sizing formula appears after the grid.

		Project Type		
		Low Risk (Small)	Low-to-Moderate Risk (Medium)	Significant, High Risk (Large)
Project Attribute	Estimated Time	< 6 Months	6 – 12 Months	12 – 24 Months
	Core Team Size	< 7	7 – 12 Core Team Members	> 12 Core Team Members
	Cost	< \$750,000	\$750,000 – \$1 million	> \$1 million
	Schedule	Schedule is flexible	Schedule can undergo minor variations, but deadlines are firm.	Deadline is fixed and cannot be changed. Schedule has no room for flexibility.
	Complexity	Easily understood problem and solution. The solution is readily achievable.	Either difficult to understand the problem, the solution is unclear, or the solution is difficult to achieve.	Both problem and solution are difficult to define or understand, and the solution is difficult to achieve.
	Strategic Importance	Internal interest only.	Some direct business impact and/or relates to a low priority initiative.	Relates to key strategic initiatives.
	Level of Change	Impacts a single business unit.	Impacts > 1 business units.	Enterprise impacts.
	Dependencies	No major dependencies or interrelated projects.	Some dependencies or interrelated projects, but considered low risk.	Major high-risk dependencies or interrelated projects.

Figure 2, Project Sizing Grid

Project Sizing Formula

Significant, High-Risk Project	Low-to-Moderate Risk Project	Low Risk Project
Level of Change = Enterprise, or two or more categories in Large column	Four or more categories in Medium column; or one category in Large column, and three or more in Medium column	Remaining combinations

Recipe for Project Success

Whatever the nature of your project, a business analyst can't go wrong if she remembers to scale the project by following the recipe for success adapted from The Standish Group International, Inc. in their 1999 report: *Unfinished Voyages*, a Follow-Up to the CHAOS Report. Their message is clear: size matters. When structuring a project or major project phases, the project manager and business analyst should strive to follow these guidelines to reduce project risk.

Ingredients:	Minimization, communications, standard processes
Mix with:	Full-time core team members: business analyst, project manager, business visionary, architects and developers, coached by an involved project sponsor
Bake:	No longer than six months, no more than six people, at no more than \$750,000

Business Analyst Challenges

Any discussion of the business analyst role is incomplete without addressing the pitfalls of filling this difficult and vital function. Although the business analyst is expected to scope the system, translate business needs to developers, translate technical issues to business representatives, model and document requirements, act as communication broker, serve as political mentor, test and validate the solution (and in general, represent customers, users, and stakeholders), the position can sometimes wield too much power (Ambler, 2005). For many organizations, the addition of senior business analysts has greatly improved the elicitation and documentation of requirements. However, there are common problems that often occur. It pays the business analyst to take heed of these pitfalls.

- **Knowledge and Skills.** Business analysts often lack necessary skills due to the IT practice of thrusting strong technologists into managerial positions.
- **Barrier vs. Bridge.** Business analysts sometimes assume too much influence over project decisions. Some business analysts may stand between the technical and business communities, instead of serving as a facilitator and enabler by bringing them together at the table. In such cases, the business analyst acts as a communication barrier versus a bridge.
- **Lagging Business and Technical Acumen.** Business analysts quickly become generalists in both the business and technical arenas. Maintaining credibility with both communities requires staying current with both technology advances and business trends.
- **Analysis Paralysis.** Business analysts are often tempted to overanalyze, stretching out the analysis period rather than iterating. Several iterations with feedback from both business and technical communities are worth more than a single, all-encompassing analysis. Sometimes this leads the business analyst to make promises based on theoretical models that may not work in practice.

The Grooming of the Business Analyst

How do IT organizations develop the exceptional business analysts needed to bridge the chasm between the business and technical communities? As with any other leadership role, competency comes from acquiring education and training, seeking mentoring and coaching, and jumping in headfirst to learn the discipline.

Because the role requires both business and technical expertise, formal qualifications for business analysts include studies of computing and management information systems, coupled with traditional business administration courses. In addition, business analyst education, training, and real-world experience should include the following (Sodhi & Sodhi, 2003):

- Knowledge of the overall requirements generation process
 - Requirements initiation
 - Systems requirements elicitation
 - Requirement types
 - How to write good requirements
- Documenting requirements
 - Requirements definition
 - System requirements documentation
- Requirements feasibility and reliability
 - Cost/benefit analysis
 - Alternative solution analysis
 - Feasibility and reliability risk analysis
- Managing system requirements
 - Tools and techniques
 - Technical specifications
 - Test plans
 - Requirements traceability process
 - Requirements and system development
 - Requirements and systems engineering processes
- Systems engineering planning
 - Requirements management controls
 - Requirements analysis for IT systems
 - Requirements functional analysis and allocation
 - Requirements and systems design
 - Requirements and systems implementation

Map Out Your Business Analyst Development Program

Look for leading-edge business analysis training offerings focused on increased performance, best practices, and project results. The courses should be based on sound systems engineering principles, focused on leadership and facilitation skills, rich in lean-thinking, agile tool sets, targeted toward real-



world IT situations emphasizing outsourcing challenges, and filled with tailoring techniques for small, medium, and large, high-risk projects.

The course offerings should be designed to provide practical guidelines and skills that lead to immediate results for writing, defining, analyzing, and managing IT systems requirements. Based on real-world experiences and case studies, courses for the business analyst should offer practical strategies, well-tested methods, and tools for implementing requirements management techniques.

Whether you are an individual developing your career or a manager advancing your organization's IT business analysis capability, select education and mentoring offerings that will increase the value your projects contribute through advanced business analysis. Seek consultants who will work with you to select the best mix of courses and reinforcement strategies that will provide immediate impact to your organization.

Final Words

Gaps in technology, techniques, and tools are not the fundamental reasons why projects fail. Rather, project failure most often stems from a lack of leadership, and poor choices made by people. Undeniably, the business analyst and project manager are evolving into the IT project leaders of the future. But team leadership is different from traditional management, and teams are different from operational work groups. The key issues are no longer centered on control and *management*, but rather collaboration, consensus, and *leadership*.

Team leaders must have an understanding of how teams work, and the dynamics of team development. They must develop specialized skills that are used to build high-performing teams. When building software-intensive systems, well managed teams undoubtedly accomplish more work in less time than do poorly managed teams (Bechtold, 1999). Traditional business managers and technical leads cannot necessarily become effective team leaders without appropriate training and coaching. There is no shortage of roles that the business analysis leader of teams can play. Some suggest that they are the new wave of IT management, assuming roles that include sponsor, promoter, trainer, teacher, team member, inventor, and entrepreneur.

Leading researchers and scholars consider project success as the way to ensure organizational success. Conversely, significant project failure often leads to the demise of companies. Organizations on the leading edge are improving project performance by better training and equipping their business analysts. A specific career path for business analysts leading to senior executive positions is one strategy to retain key talent, while training those who will follow. Enlightened organizations in both the public and private sectors are creating cohesive career management plans for business analysts to develop their potential, match their skills to assignment, track performance, and reward them appropriately.

Unlike the past, when organizations embarked upon professional training and mentoring only after a crisis or in conjunction with a major reorganization, IT organizations are now rising to the challenge and valuing less technical, more business and leadership-focused competencies. Whatever the driver, components of world-class, professional business analysis career programs include most of these elements: formal off-the-shelf and customized training; mentoring and on-the-job training, advancement based on education, testing, and experience; defined evaluation and compensation processes; and suitable titles and advancement opportunities.

References

- Ambler, S. (n.d.). *Agile Analysis*. Retrieved Apr. 08, 2005, from Ambysoft Web site:
<http://www.agilemodeling.com/essays/agileAnalysis.htm>.
- Ambler, S. (n.d.). *When Does (n't) Agile Modeling Make Sense?* Retrieved Apr. 08, 2005, from Ambysoft Web site:
<http://www.agilemodeling.com/essays/whenDoesAMWork.htm>.
- Bechtold, R. (1999). *Essentials of Software Project Management*. Vienna, VA: Management Concepts.
- Fowler, M. (2003). *The New Methodology*. Retrieved Apr. 08, 2005, from martinofowler.com Web site:
<http://www.martinofowler.com/articles/newMethodology.html>.
- Hadden, R. (2003). *Leading Culture Change in Your Software Organization*. Vienna, VA: Management Concepts.
- Mooz, H., Forsberg, K., & Cotterman, H. (2003). *Communicating Project Management: The Integrated Vocabulary of Project Management and Systems Engineering*. Hoboken, NJ: John Wiley and Sons.
- Sodhi, J., & Sodhi, P. (2003). *Managing IT Systems Requirements*. Vienna, VA: Management Concepts.
- The Standish Group, (1999). *Unfinished Voyages: A Follow-up to the CHAOS Report*. Retrieved Apr. 08, 2005, from The Standish Group Web site:
http://www.standishgroup.com/sample_research/unfinished_voyages_1.php.
- Stevens, R., Brook, P., Jackson, K., & Arnold, S. (1998). *Systems Engineering: Coping with Complexity*. Indianapolis, IN: Pearson Education, Prentice Hall PTR.
- Wiegers, K. (2003). *Software Requirements: Practical Techniques for Gathering and Managing Requirements Throughout the Product Development Cycle*. 2nd ed. Redmond, WA: Microsoft Press.
- Young, R. (2001). *Effective Requirements Practices*. Addison-Wesley information technology series. Boston: Addison-Wesley.



Business Solution Life Cycle

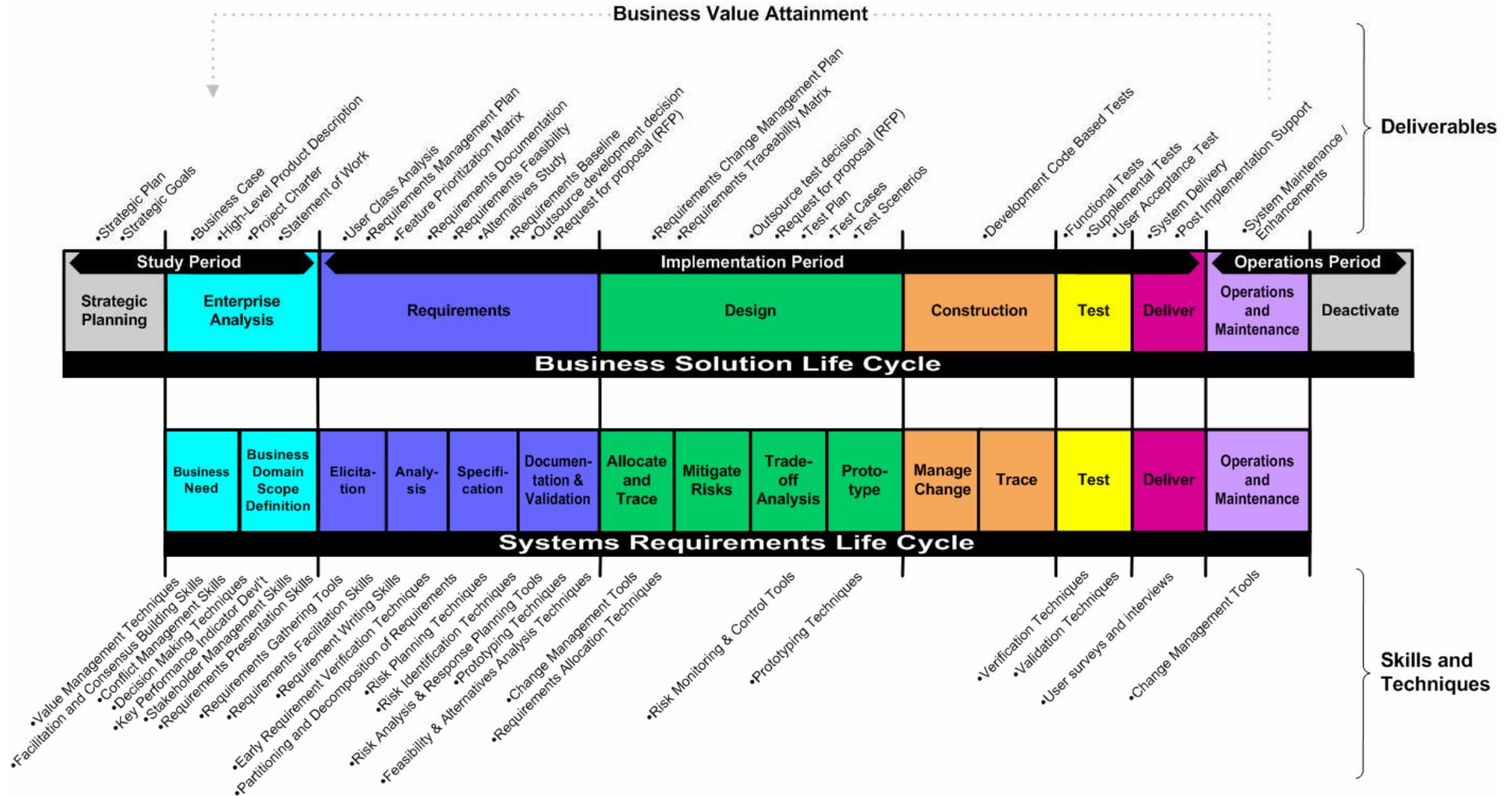


Figure 3 – Business Solutions Life Cycle Mapped to the Systems Requirement Life Cycle