



Executive Guide to Business and Software Requirements

EXECUTIVE SUMMARY

This paper is designed as an executive briefing on the issues surrounding getting business and software requirements right. The white paper is structured so that each section is its own 'mini-white-paper' but each is also part of a total look at better structuring requirements documents, processes and content completeness.

At the end of this paper is a one page 'tear-out' summary. This summary is both a shortened distillation of the content, and a diagnostic tool for reviewing your organization's performance in extracting and documenting business requirements.

In short: we propose that Business and Software Requirements are a process that needs executive attention. Not only is this a central enabling process for business strategy execution, it is one that requires discipline and commitment from the organization to be able to make it perform well.

In addition to the wealth of statistical detail to support the monetary contribution of this process, we propose that business requirements are more than a template to be completed. Effective and efficient requirements must:

- ***Start with a Template:*** that clearly defines a format for content designed for use by a specific user. And also has;
- ***Complete Content:*** this content has specific characteristics, and is measurably complete. And;
- ***Uses an Efficient Process:*** since a company's subject experts are its most limited asset.

CONTENTS

EXECUTIVE SUMMARY	2
CONTENTS	3
WHY FOCUS ON GETTING ‘BUSINESS REQUIREMENTS’ RIGHT TODAY?	4
HOW DOES BUSINESS PERFORMANCE IN DEFINING BUSINESS REQUIREMENTS AFFECT OTHER OPERATIONAL AREAS?	5
UNDERSTANDING OF EFFECTIVE BUSINESS REQUIREMENTS – THE DEFINITION	5
BUSINESS REQUIREMENTS: THE HOLISTIC VIEW	6
WHAT MAKES GETTING BUSINESS REQUIREMENTS RIGHT A COMPLEX PROBLEM?	7
ASSURING CONTENT CLARITY IN BUSINESS REQUIREMENTS	9
EFFECTIVE INFORMATION FORMAT RECOGNIZES THREE AUDIENCES	10
EXECUTIVE BRIEFING ON FORMATTING STANDARDS AND ERRORS	10
EXECUTIVE BRIEFING ON COMPLETENESS: BUSINESS REQUIREMENTS TESTING AND CONTROL PROCESSES	12
EXECUTIVE BRIEFING ON COMPLETENESS: CONTROLLING FINANCIAL RISKS.....	14
BUSINESS REQUIREMENTS AS A PROCESS	15
KEY SUCCESS FACTORS AND PERFORMANCE ASSESSMENT OF BUSINESS REQUIREMENTS.....	17
ABOUT IAG CONSULTING	19

WHY FOCUS ON GETTING 'BUSINESS REQUIREMENTS' RIGHT TODAY?

The concept of getting business and software requirements right sounds straightforward. However, most companies would admit they need to improve - in fact, flawed requirements trigger 70% of project failures¹. Consider the statistics from recent studies:

1. While fewer than 15% of projects are considered outright failures by those involved, 21.6% of all spending on projects is still wasted today – either on failed projects, or poorly controlled overruns.²
2. As the projects get more interdepartmental and complex, the failure rate rises. For example, 51% of companies implementing an ERP viewed their ERP implementation as unsuccessful.³

The ERP statistics demonstrate an important relationship: the larger the number of stakeholder groups involved, and the more complex the processes, the more likely a project will fail due to business requirements-related problems. It is not so much that over 20% of all money spent across North America on new projects will likely be wasted, it is that the more strategic a project becomes, the more likely it is to fail.

From a cash flow perspective: more money will be saved by improving your processes in business requirements, than by outsourcing the entire application development organization offshore. Fixing business requirements processes, however, requires more organizational discipline to achieve results.

¹ Info-Tech Research Group (2006)

² Standish, Chaos Report (2004)

³ Robbins-Gioia Survey (2001)

HOW DOES BUSINESS PERFORMANCE IN DEFINING BUSINESS REQUIREMENTS AFFECT OTHER OPERATIONAL AREAS?

An Analogy: Evaluating the Effectiveness of Your Requirements Development Processes

Even mid-sized organizations have hundreds of projects that are run annually, maybe 40 of which are significant and risky. An executive needs to think of getting business requirements done as a manufacturing process, done in an identical way, 40 times over the course of a year.

How could a person manufacturing computers be random about what the computer looked like, what was in the computer, and how the computer was made and still expect a consistent user experience from using that computer?

1. Effective business requirements are both a business organizational discipline issue, and a technology issue. Requirements are the bridge between business and technology wherein the maturity of the processes in place directly impact the effectiveness of a company in its use of technology or technology suppliers.
2. High quality processes are also efficient and streamlined. This means that implementing better processes should come with the future benefit of time savings and better execution. Today, most business strategies end with some form of technology solution. Making this transition between business and technology faster and more efficient is a goal that can be profitable at the highest level of a business.
3. Poor requirements have extreme organizational financial risk – particularly on large technology initiatives. There are processes for testing the completeness of requirements. There are triage tests to determine if it is probable that less intuitive processes were missed. There should be little excuse for a project missing its financial targets by orders of magnitude because the requirements were poorly specified.

Requirements are not simply a document or template that needs to be filled in. Effective business requirements are achieved only by getting the right content, in a consistent format, using a clear process. The process is a discipline that requires commitment at the highest level.

UNDERSTANDING OF EFFECTIVE BUSINESS REQUIREMENTS – THE DEFINITION

Officially: “A requirement can be thought of as something that is demanded or obligatory; a property that is essential for the system to perform its functions. Requirements vary in intent and in kinds of properties. They can be functions, constraints, or other elements that must be present to meet the needs of the intended stakeholders. Requirements can be described as a condition or

capability a customer needs to solve a problem or achieve an objective. For clarification purposes, a descriptor should always precede requirements; for example, business requirements, user requirements, system requirements, operational requirements, contract requirements, and test requirements.”⁴

In short: business requirements are the description of the business needed by all parties involved in using, implementing or maintaining an application.

Unfortunately with business requirements, much of the documentation generated is simply ineffective. The difficult part is describing all the details of the business in a way that is absolutely comprehensive and which presents an exact and consistent specification for each relevant piece of a business. Effective requirements have certain characteristics and information content. Specifically, effective Business Requirements in a project have three characteristics:

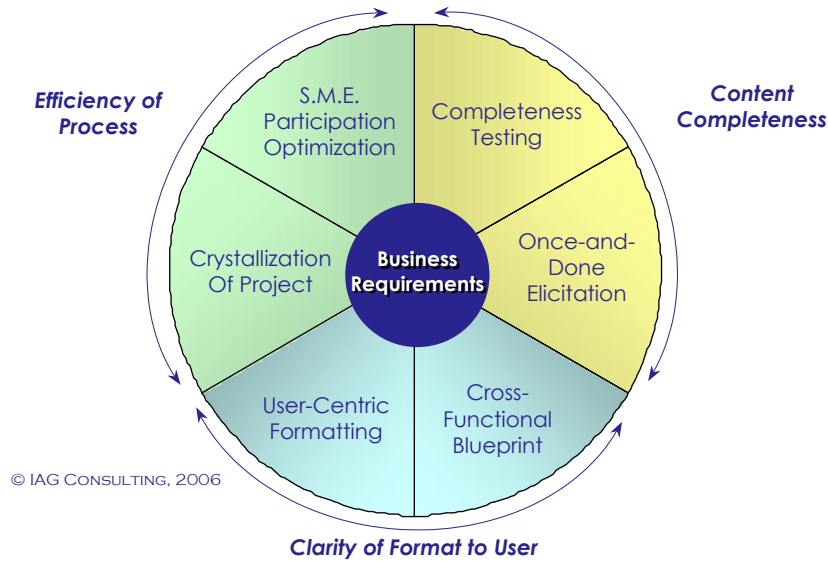
- **Clarity of Format:** information is correctly and consistently structured in a useable form.
- **Content Completeness:** certain types of information are persistently available.
- **Efficiency of the Process:** it is clear how analysts will elicit the information needed.

BUSINESS REQUIREMENTS: THE HOLISTIC VIEW

Geoffrey Moore first coined the term ‘whole product’ in 1991 (Crossing the Chasm), and we see this concept as useful in assessing business requirements. *Business and software requirements are ineffective unless an efficient process is adopted to ensure that content is both complete and clear.* A holistic approach is more than simply producing a nice document – it is a better way of creating a symbiotic relationship between business and IT.

THERE ARE 6 ATTRIBUTES THAT MAKE AN EFFECTIVE AND EFFICIENT PROCESS.

⁴ International Institute of Business Analysis, Body of Knowledge v1.4 (2006)



WHAT MAKES GETTING BUSINESS REQUIREMENTS RIGHT A COMPLEX PROBLEM?

Simply because a company or project is smaller, does not mean that the business requirements for that project are simple. The two are only loosely correlated. Requirements risk⁵ is better assessed by looking at the number of stakeholder groups involved in a project and the relative complexity (newness to the requirements team) of the processes these people will be reviewing. Hence, there is a higher probability that the business requirements will never be properly defined for a system with many new processes that impact many divisions.

⁵ Digital Mosaic, 2005: risk that the process of gathering business requirements will overrun by 2 times in time or cost, or that requirements documentation derived from this process will be unusable to control system design or implementation.

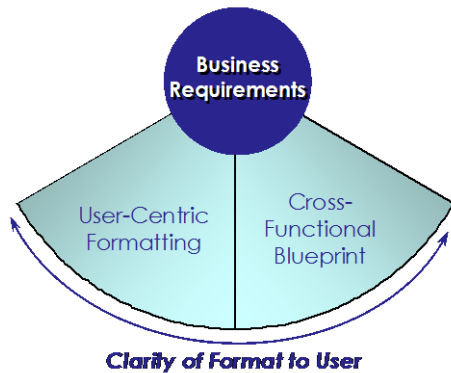
WHAT ARE THE ISSUES THAT CAUSE THIS FAILURE PROBABILITY?

1. As the number of stakeholders rise, so to does the need to focus on what happens *between* departments rather than within a department. The need to achieve consensus without a process that drives consensus will cause the project to spin endlessly.
2. As complexity rises, analysts who may have relied on their personal subject-matter expertise are no longer able to define requirements or effectively ask questions of stakeholders in order to elicit requirements. Hence, a company cannot rely on industry experts alone as each is only able to contribute to a small part of a large system. Lacking a method to drive questioning, the team will develop a list of requirements that are riddled with information holes, or the process will lack a feeling of being proactively driven.
3. As both variables and the project size rise the information contained within the documentation becomes more interdependent, complex, and voluminous. The structure of the information, its findability, and the interpretation of content all have an impact on the ability of analysts to simply maintain the documentation. As control is lost, each subsequent iteration of the business requirements will raise more questions than are answered – closure or accuracy is never achieved.
4. Lack of completeness kills projects – either by overweighting the elicitation process and making it inefficient, or providing less-than-adequate detail.

The process of business requirements discovery is a collaborative process between business and technology. It must clearly describe the “WHAT”, not the “HOW” of business needs, with the content serving two audiences. The skills and techniques needed to keep the discussion narrowly focused, proactive, and, through which information is organized are unique competencies. A failure to utilize these skills during a larger requirements engagement increases the risk that a project will fail in the requirements stage of development. Sometimes this failure in requirements is only realized after millions have been spent in delivery.

Simply because there is a somewhat ‘creative’ element to the process of doing business requirements does not mean that the ‘manufacturing process’ should also be ‘creative’ (poorly controlled). First manufacture the car right; then, customize it with feature needs or wants.

ASSURING CONTENT CLARITY IN BUSINESS REQUIREMENTS



Format is an executive issue? Absolutely. If the content of a business requirements document is not well structured, it would be like reading a novel with no page numbers and the pages in random order. A persistent person will eventually figure out the story, but the process is extremely inefficient. Organizations sometimes fix this problem with templates that are over-engineered creating so much paperwork there is a high degree of redundancy. Fortunately or not, these templates will be on the critical path of every executive in your company. Why? Because at some point every strategic initiative they are driving is tied to a business or software requirement.

Example: a set of requirements for a *scheduling system* might look like the following:

- Must post schedule to the regional offices
- Must have regional managers verify that the schedule is complete
- Must have wireless access for crew management offices
- Must be able to handle 50 transactions to 25 sites per day
- Must be available 7X24
- Must take results of schedule and post to payroll system
- Must be compliant with union work rules

While these may be valid requirements, they are almost useless in format. To see the problem, assume that easily 10 times as many equally valid statements would exist for the scheduling office. Visualize that this is one of just 10 divisions defining requirements for a major ERP system - of which the transport scheduling is only a part. Now, try to spot the inconsistencies between the divisions from this list of 700 statements (all of which look individually valid). Candidly, assessing the interdependency of such a large number of statements cannot be done, and as the system scaled toward an enterprise ERP such as an SAP, the listing might be closer to 7,000 statements if the requirements work was done this way.

EFFECTIVE INFORMATION FORMAT RECOGNIZES THREE AUDIENCES

What's been missed above: while information formatted as above may be very useful in format to a business professional, the format itself makes the information inefficient for analysts to use, and next to worthless for a technology specialist that simply cannot translate the requirements into precise process flows, data flow and business rules. The opposite problem might be to adopt a very technology centric format for defining requirements (such as UML Diagramming or even overly formal System Use Cases), which are simply too difficult for non-technical people to readily understand, observe implications, and readily sign-off as complete.

The format **MUST** match the 3 very different user needs: business professional, IT professional and analyst/oversight role. The issue is to ensure that the format for early stages of analysis can be used seamlessly in latter stages with little or no rework. Otherwise, there is redundancy and inefficiency.

A large company likely has over 20% of its expenditure on the analyst function tied up in ineffective templates.

EXECUTIVE BRIEFING ON FORMATTING STANDARDS AND ERRORS

There are three fatal formatting errors:

1) Ambiguity of any kind. For instance, in the scheduling system example on the prior page: when is it time to post a schedule to a regional office? Who does this? What information must this person (or the system) have in order to complete this activity? How does the system know that the activity is completed? What happens if the person trying to post does not have a schedule? It must be clear.

2) Stove-piped requirements reporting. If requirements are collected and reported divisionally rather than cross-functionally, the analyst will be unable to determine interdepartmental conflict (as in the example above).

Diagnostic Questions for Format

A quick set of questions, degree of importance and issues to identify:

1. How will the team compare requirements extracted from one division with requirements extracted from another division to detect inconsistencies? (Very high importance: be 'from Missouri' because it is easy to mask information that is not, in fact, comparable.)
2. Is the format used identical on every project? (Moderate importance: poor format consistency indicates weak discipline. Enough flexibility and the "software" manufacturing process may be broken.)
3. Is the information given at the earliest stage of project inception sufficient to be able to scope the effort needed for the next stage? (getting this right takes a company from a state of 'basically performing right', to 'performing well'. Most companies do very early stage scoping very poorly and thus the process quality never changes.)

3) Poor information reference-ability. It sounds simple, but often, the information within requirements documents is just too difficult to find to make documentation useable.

A company should even have a style-guide put together to define what gets underlined, what is bold italic, for every major document it produces. It is important to ingrain the discipline of effective Business Requirements, and each major systems development activity will have AT LEAST 100 unique sets of eyes looking at a requirements document. Each of these people must also correctly interpret the document to their job context, and perform their job function more efficiently by using it.

The way information is put together in a format that eliminates ambiguity, makes comparability between divisions possible and provides information reference-ability is the 'cross-functional blueprint'. These are the highest level of requirements defined interdepartmentally.

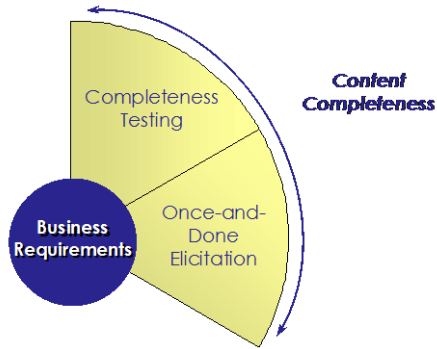
FORMATTING STANDARDS

If you've heard the terms Use Cases or UML (unified modeling language). These are one of a number of standardized ways of formatting requirements information for greater levels of information organization. The purpose of these formats is to facilitate the communication of requirements from users to technologists (Use Cases), while also simplifying the task of managing requirements through the development process into code.

A high quality format for business requirements enables content to be easily decomposed into necessary content in the next stage of analysis. If this cannot happen, then the project loses cohesion (the analyst term is 'traceability') and what results from the development effort may or may not be what was specified as the intent of the development effort.

A test: can your team exactly predict the cost and effort of the next stage, based on the completed template of the prior stage?

EXECUTIVE BRIEFING ON COMPLETENESS: BUSINESS REQUIREMENTS TESTING AND CONTROL PROCESSES



“DONE” needs to have specific meaning for business and software requirements – not just that stakeholders got tired and signed-off on a document. For requirements to be complete, *ALL* in-scope business activities must be analyzed, and for each, there must be sufficient detail that is unambiguous, accurate, well-organized and validated. Regardless of the development type – procuring packages, outsourcing, or in-house development, information content for business requirements should have at least the content listed in the call-out box below.

“Incomplete requirements”, where documentation does not meet certain minimum standards, may squander all time and resources expended on the requirements effort to date. What happens to a project of \$5 million if \$500,000 of project expenditure has only nominal benefit to project delivery? It overruns. Controls can be implemented in three areas to assure a higher level of completeness: analyst standards for modeling requirements, peer reviews and enhanced client review standards.

Minimum Content Needed for Business Requirements

At the highest level, the following is the minimum set of information needed for defined business requirements:

1. Every process within the system identified (as either in scope or out of scope) and each business activity in scope described with a standardized task-level description of how information moves between people and/or within the system.
2. Trigger(s) to the activity, post-conditions (what is the outcome of an activity) and exceptions/exceptions/exceptions.
3. Actor(s) of the process (i.e., who interacts with the system?)
4. Well-formed requirement statements that define the required capability (i.e., “the system must...”)
5. Business rules that support the process need to be documented
6. Data required to support the process (data attributes) identified and relationships among data required in the process
7. Non-functional requirements such as volumes, security/ access, specific usage issues, etc.

EXECUTIVE ISSUES IN MANAGING COMPLETENESS:

1. At what point in time during the systems development process are requirements 'defined'? The answer here directly impacts the technology team's ability to forecast and budget.
2. How complete is the information content once received? The answer here directly impacts the amount of rework, wastage, or scope creep on a project.

TIMING OF REQUIREMENTS AND THE SYSTEMS DEVELOPMENT LIFECYCLE

There are a number of development methodologies that have become buzzwords. Agile, Rational Unified Process (RUP), RAD (rapid application development), waterfall, or even to some extent design-oriented philosophies like object-oriented development all prescribe an evolutionary process of moving from requirements through to implementation. What is different in these approaches is the point at which a full understanding of requirements is garnered.

- Waterfall, RUP, Use Cases all prescribe >80% understanding of business requirements documentation prior to moving into design.
- Agile or Extreme methods have little or no formal business requirements definition and rely on iterative design to achieve the result.

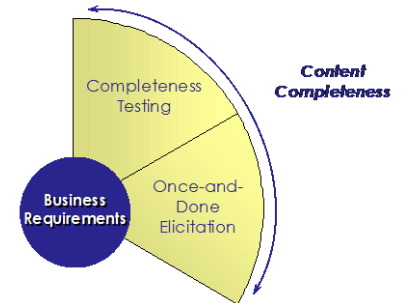
In a small project, the selected method is somewhat irrelevant (except for the impact on enforcing a singular disciplined approach) since there is less cost associated with rework and limited task interdependence. But, in a \$2 million project we can demonstrate that there is over \$300,000 to be saved by simply doing a better job up front.

Success in large projects is not just about completeness, but about getting complete information earlier in the development process if you hope to use it to control cost and scope.

EXECUTIVE BRIEFING ON COMPLETENESS: CONTROLLING FINANCIAL RISKS

There are two financial risks when it comes to completeness:

1. Will the company ever get complete business requirements? Remember, something in the order of 93 out of 100 IT projects will restart at some point so this is a very real issue.
2. Risk that the project failed to identify all elements that should have been in scope early enough for management to make a decision about cost.



The first issue is controlled by changing the structure of elicitation such that re-work is minimized by gaining a richer set of information from subject experts on the first pass through. The second issue will be the more costly on larger projects, and may result in a company spending millions through an implementation and realize that the company has inadequate business requirements. The failure in completeness surrounding requirements costs millions.

On larger projects, here's a startling statistic: 30% of requirements content is missing (on average) prior to development. This means that an average project over \$5 million, has, on average, \$1.5 million of development cost associated with business requirements that are totally undocumented. What should frighten an audit committee is: what happens if the \$1.5 million of unknown requirements turns into \$3 or \$5 million? In 2001, the Conference Board did a survey and found that the average project is 25% over budget with support costs underestimated in the year following implementation by an average of 20%.

Diagnostic Questions for Content

A quick set of questions, degree of importance and issues to identify:

1. Does the quality of business requirement detail vary based on the analyst performing the work? (Very High importance: fixing this issue is step 1)
2. At what point in time of a project's lifecycle does the project team have 'full' requirements? (Moderate importance: Earlier is better. Too late in the cycle indicates a broken process.)
3. How is content tested for its completeness? (getting this right takes a company from a state of 'basically performing right', to 'performing well'. Without the ability to objectively measure quality, there is less ability to radically improve the process.)

REQUIREMENTS COMPLETENESS FOR SOFTWARE SELECTION

There is also a common misperception in the purchasing of packaged software that requirements are essentially unnecessary. Remember, 51% of ERPs fail. Poorly defined requirements mean:

1. Lawyers engaged to write the contract will not be able to define the scope. Therefore a loose statement of work follows with poor client recourse and definition of failure.
2. Each executive awaiting the implementation of the system will each have a different perception of what the system is, and will do for them.
3. The organization will be eagerly awaiting the 'best practices' contained in the system. Some of these may conflict with the way this organization must do business to the extent that the organization becomes dysfunctional after implementing these practices.
4. Technology people will still need the requirements information to configure the system.

If a system is over \$5 million, have a formal requirements review and validation protocol or "business and software requirements audit" with report to executive. The alternative is unnecessarily expensive.

BUSINESS REQUIREMENTS AS A PROCESS

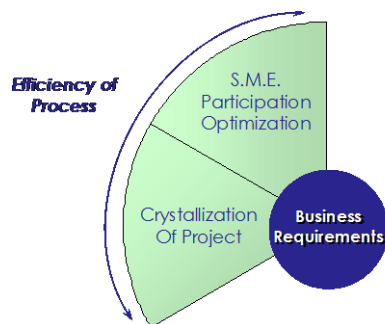
At my last speaking engagement, I asked the question: "who thinks their business requirements elicitation is efficient?" (elicitation is the process of requirements extraction or discovery) Only 1 out of 200 people put up their hand.

Few organizations see the development of business requirements as an efficient systematic process. In the system development lifecycle's (SDLC) of many companies, it seems semi-mystical or magical occurrence. Requirements are not just 'done', business requirements evolve through a process. Again, the idea of a manufacturing line is useful to visualize the efficiency of 40 business

requirements projects annually being produced with a randomized assembly line. If analysts engage stakeholders in an unstructured way, with stakeholders only giving unstructured attention, expect an unstructured result.

THERE ARE TWO ELEMENTS ON THE CRITICAL PATH FOR FIXING THIS ISSUE:

1. Create a plan for requirements definition that crystallizes every project. This process should take no more than a few days. The outcome should precisely estimate the amount of time and resource for business and software requirements.
2. The rate-determining step for producing business and software requirements (for most companies) is the availability of subject experts. Any methodology adopted must, BY DESIGN, optimize the use of these professionals and management or risk failure.



In the absence of these two elements, a company cannot hope to make the process highly efficient. It simply cannot estimate with any credibility the time requirements of the process, nor is it working every engagement forward from a common starting point. Unfortunately, this initial scoping and elicitation planning activity tends to be poorly done even by organizations well defined and mature overall requirements processes. Fixing this process is central to:

1. Expectation management
2. Managing initial estimates of scope
3. Managing subject expert participation on projects
4. Creating forecasts of analyst loading
5. Being able to measure analyst performance

DEFINING A SOLID PROCESS FOR BUSINESS REQUIREMENTS

Diagnostic Questions for Process

A quick set of questions, degree of importance and issues to identify:

1. Can we predict how much time will be needed from subject experts on a project? (Very High importance: fixing this issue is step 1)
2. What is our variability in performance against forecast? (Moderate importance: without a forecast, the organization cannot be proactive in addressing issues)

The process of getting the business requirements information extracted from the heads of subject experts is called an 'elicitation methodology'. This is different than the way the analyst chooses to write the requirements information down (a 'documentation methodology'). It should include:

1. **Requirements planning** – scope and plan and set strategy
2. **Requirements elicitation** – gathering of requirements from stakeholders
3. **Analysis and documentation** – Functional, informational and non-functional requirements.
4. **Verification and review** – verifying completeness with peer and client reviews.
5. **Validation and acceptance** – confirm, prioritize and obtain approvals.

KEY SUCCESS FACTORS AND PERFORMANCE ASSESSMENT OF BUSINESS REQUIREMENTS

Effective business requirements are a complex amalgamation of format, content and process, just like any major operational function of the company. Similarly, like accounting practices, there should be an audit and a standardized set of deliverables which everyone in the company (or key vendors outside the company) agrees constitute complete requirements.

Here's the painful part: there is a preconceived notion that getting well-defined requirements is, or should be, painful. This is simply not true. This is merely evidence of a sub-optimal process.

Like any quality process, the higher the quality and repeatability, the *more efficient* the process. Perfecting the manufacture of goods is an excellent analogy. Use these same principles to perfect the manufacture of business requirements into an efficient process.

CRITICAL SUCCESS FACTORS FOR BUSINESS AND SOFTWARE REQUIREMENTS

1. It's fine if users do not know all the answers. As long as we know what we don't know and can limit the impact of these imperfections on other areas of the SDLC.
2. A valid result of a business requirements assignment is to realize that a project is simply not ready to move forward. It is better to determine this with \$50,000, than with \$500,000.
3. We can have 6 inches of documentation - As long as it did not take us 6 years to get there and the reference is easy to use.
4. Always separate 'what' the system must do (this is business requirements) from 'how' it will do this (this is technical design).
5. Crystallize everything.
6. Ensure that the interdepartmental blueprint is done first when eliciting requirements. Without first working out how processes cross boundaries in a comparable format, the content is stove-piped and unusable.
7. Quash Inconsistent levels of information content and quality.

STAY FOCUSED ON THE BUSINESS CASE FOR MAKING IMPROVEMENT

It is 80 times less expensive to fix defects when found in the specification stage⁶. Research from a number of sources shows that an investment in software productivity improvement can yield:

- 67% reduction in rework costs.
- 30% to 40% reduction in schedule lengths⁷
- 90% reduction in defects.⁸
- 350% increase in productivity gain.

⁶ Doolan, E.P., (1992) Experience with Fagan's Inspection Method, Software Practice and Experience (Vol. 22(2))

⁷ Air Force Research Laboratory - Information Directorate (1999, McGibbon), A Business Case for Software Process Improvement Revised - Measuring Return on Investment from Software Engineering and Management

⁸ Jones, C. (1996) "Software Defect Removal Efficiency", Computer (Vol 29, No 4)

Interestingly, when SEI researched organizations moving from CMM level 1 to level 2 and from CMM level 2 to CMM level 3, the greatest single increase in performance was reported by an organization which principally focused on improving requirements definition and management.

ABOUT IAG CONSULTING

IAG specializes in business and software requirements. Since 1997, our company has worked with 300 of the Fortune 1000 companies, completed over 1,300 requirements projects, and trained more than 100,000 business analysis professionals. Our organization focuses on a practical and practiced approach that is efficient for all stakeholders in both business professional and information technology departments. We bring measurable gains to our clients:

- Reducing time needed to complete requirements
- Ensuring completeness in documentation and reducing change requests
- Issuing RFPs where vendors can bid accurately and clients get better terms
- Reducing costs in systems development
- Salvaging troubled projects

CONTACTING AN IAG CONSULTING SPECIALIST

Email us at: info@iag.biz or

Call our North American Toll Free line: 800-209-3616