The Requirements Communication Problem

Process Models Bridge The Requirements Gap

John Forrest
Chief Executive Officer
Holocentric Pty Ltd

Revised January 2007

IT systems will only deliver the benefits which business users expect if they are based on a set of well-documented and traceable requirements. These requirements need to be traceable up to the desired business outcomes and down to the IT system's functionality to eliminate the translation stages which result in the introduction of most of the errors.

Business process models are an effective means of binding the business outcomes to the system requirements in a form which is easily understood by business and IT people alike, removing the potential for requirements to be misunderstood and a poor reflection on the actual business needs.

Executive Overview

Defining business requirements accurately is one of the most important success factors for technology projects. Rather than focus on solutions that satisfy a list of requirements, we need to focus on solutions that satisfy desired business outcomes.

The best way to achieve this is by performing business process modeling. Employing a visual model to capture people's roles and derive requirements produces much more effective requirements. Furthermore, requirements based upon roles lead to higher quality user acceptance, testing and implementations.

Process modeling achieves this by providing a common language for the business and IT, creating a platform to:

- effectively capture requirements from business people in a way they understand
- document how each requirement relates to an actual business process
- communicate how each process relates to a desired business outcome.

A business model unifies requirements gathering into a coherent and managed process. The ability to relate each stage in the process with the stage before it or after it, gives us an auditable account of how technology has satisfied desired outcomes.

Building or buying software is a significant investment. Tying requirements directly to desired business outcomes with a business process model shows much promise in improving project success rates.

The record so far is poor

Let's face it – the record of information technology in satisfying business needs is increasingly met with cynicism, often deserved.

This perception is supported by statistics. In 2004, the Standish group announced that software project failure rates were 69%[1]. Other studies don't provide much comfort[2].

The majority of software build projects fail

Yet, the Standish Group considers a project a success if it "… is completed on-time and on-budget, with all features and functions as initially specified."[3] This is a fairly generous interpretation of success because it takes no account of the realization of the project's benefit – even 'successful' projects often fail to support the delivery of the anticipated business outcomes.

A more useful measure of project success would be:

A project that satisfies the desired outcomes of the business, on-time and on-budget.

This more rigorous definition allows us to judge success against business needs. It places an emphasis not only on delivering technology that meets the requirements, but also on successfully defining requirements that meet business needs in the first place.

Project success should be measured according to desired business outcomes

There are many areas where we can improve, including project scoping and estimation. However, one of the ripest areas for improvement is getting requirements right. Good requirements have one of the largest impacts on project success. One estimate claims that 85% of defects in software projects originate in the requirements phase.[4] Another estimate places the cost of correcting an error post-implementation at 100 times higher than during development[5].

[1] The Standish Group, "2003 CHAOS Chronicles", 2003, available at www.standishgroup.com.
[2] Robert N Charette, "Why Software Fails", IEEE Spectrum, 2005. Available at www.spectrum.ieee.org/sep05/1685
[3] "The Standish Group Report – CHAOS", 1995, page 2.
[4] Young, Ralph R. Effective Requirements Practices. Boston: Addison-Wesley, 2001
[5] Charette, 2005

Incomplete understanding & poor translation

Requirements are poorly defined because:

- the actual business needs are translated by business analysts or consultants into a form which is understood by IT developers
- business people are asked to authorize requirements which have been written with the needs of IT in mind, in a form which is not natural to them.

Requirements fail because business processes are poorly captured

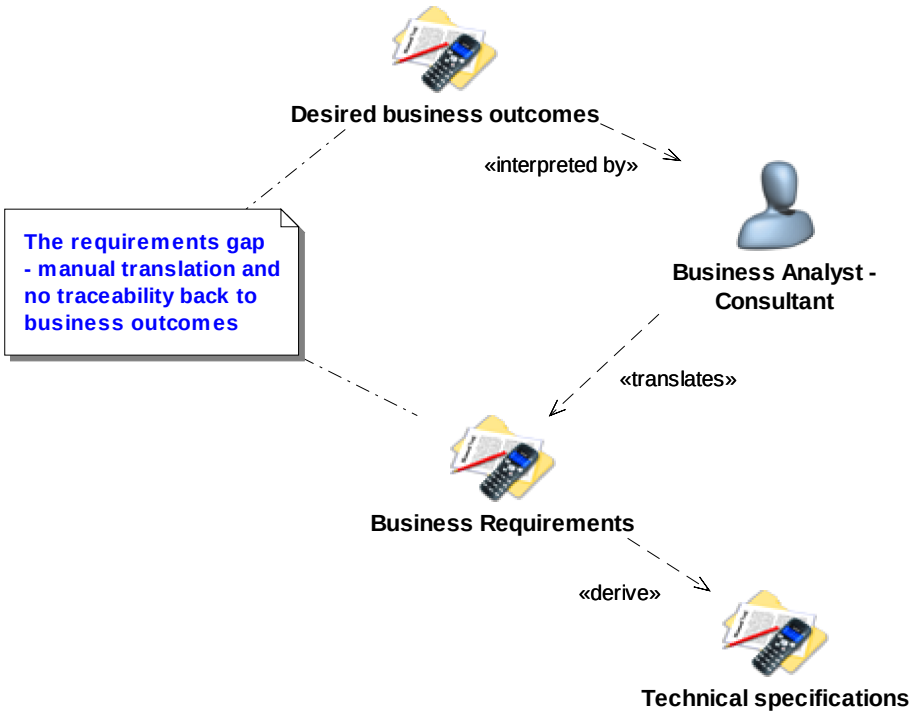
To understand how these issues arise, it is worth considering how typical approaches to requirements gathering fall down.

Failure of traditional requirements gathering

Typically, gathering requirements has been conceived only as documentation based upon conversations between business analysts and workers on how a system should work. It often starts with high level process descriptions that provide a context to the project. Business experts in each particular area are identified and brought on to fill in the details.

The responsibility for gathering the detailed requirements from business experts is delegated to internal business analysts or consultants. They interview business experts, to translate their detailed understanding of their role into requirements.

Yet business analysts tend to ask representatives of the business "what do you want the system to do?" This approach to requirements gathering demands people conceptualize their role in abstract terms, something they understandably struggle to do. Nor should we expect them to be able to - it is a particular skill that takes years to develop. This breakdown occurs at a critical juncture, where requirements move from the business sphere to the IT sphere – we shall call this the 'requirements gap'. The desired business outcome, such as to improve a process, or increase margins or increase sales, is translated into a series of solutions such as improve training, implement a CRM system or automate processes.



Business people find it hard to conceptualize their role via a list of requirements

Another problem is the language that is used to document requirements. The language used is designed to allow requirements to be easily translated into a functional specification for a system to either be purchased or built. The tools currently used are designed to meet the needs of the IT department, not the business user (the system owner and customer).

Yet the members of the business are then required to validate that the documented requirements reflect the actual needs of the business. Representatives of the business struggle to do this because there is no direct correlation between the requirements statements and the desired business outcomes.

Lastly, the requirements often make no mention of the roles performed by people within the business and the responsibilities for each of these roles. People in these roles will need to change the way they work in order to use the new system.

Once the system is implemented, users of the system need training in use of the new system. Unfortunately, training tends to focus on the system functionality and not on how a user's role has changed. This is because those implementing the system

can't relate the functionality back to the business processes (and hence roles), because they were not documented during the requirements phase.

This has a number of effects downstream. Users will struggle to use the system to its potential, compromising the desired outcomes which were envisaged at commencement of the project. Also, buy-in from the user base is harder to achieve when users don't fully understand how the new system impacts their job.

As an example, a marketing executive has a business problem such as needing to improve customer interaction and customer contact. In scoping a CRM system they will likely discuss the need for a single view of customer and the need to integrate with various touch points within the organization which the customer comes in contact. However, the business analyst will ask them to approve a document that lists requirements including active directory, dynamic HTML, financial systems table definitions, that are too technical for the business user to understand how they will improve the return from existing customers.

If our ultimate aim is to support business outcomes we need to define the processes that will achieve these outcomes. Yet we cannot do this without a good understanding of how the business works.

Requirements management does not provide the fix

It's important to recognize the difference between requirements management and requirements gathering.

Requirements management is useless if managing the wrong requirements

Good requirements are those that describe the sequence of work required to achieve a desired business outcome. Usually, people perform this work with the help of technology. Thus requirements definition establishes what is required to support the work necessary to achieve desired business outcomes.

The usual way to define these requirements is to create long lists of functional needs, divided into logical groups. These lists of requirements usually become too large to comprehend and requirements management becomes necessary. Requirements management refers to tools and methods used to manage requirements once they are defined, and track efforts to convert

them into technical specifications for IT systems.

Managing requirements is obviously an important part of any project, but it does not help us ensure our requirements are correct in the first place. Focusing on management of poor requirements is putting the cart before the horse.

So before we start to manage a project, more effort is needed in getting the requirements right. To do this, we need a better understanding of where the requirements definition process fails.

Requirements redefined

No matter how well we record, manage and build requirements, if they are not captured properly in the first place the project will struggle to satisfy the business. What is needed is a way of accurately capturing business needs and referencing them to requirements, so that services delivered by technology are more closely aligned with actual business activity and deliver the desired result.

A good start is in redefining the requirements gathering process to be much more comprehensive. A better definition of software requirements would be:

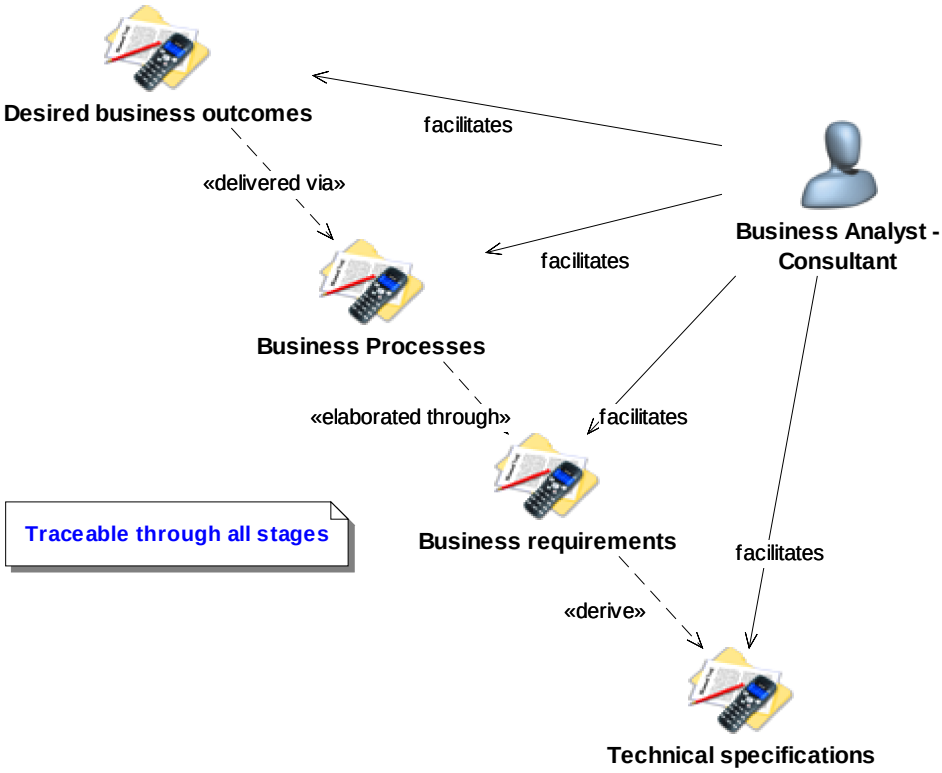
The statement of desired business outcomes that need to be supported by technology.

Requirements should reference desired business outcomes

To use this definition in practice requires an expansion of the requirements definition process. We need to ensure that desired business outcomes are described via current and future business processes.

Indeed, the requirements process can be better described as an iterative flow through four interdependent steps:

1. Define the desired business outcomes
2. Describe the processes that will deliver the desired outcomes
3. Elaborate the business requirements of each process step
4. Derive technical specifications and/or procurement requests.



In order for systems to genuinely support the business, the detailed technical specifications must be judged against how well they enable work practices to deliver the desired results.

Understanding the business

It pays to define the desired business outcomes - a definition of what the project should achieve – right up front. This is ultimately what success should be measured against, so a satisfactory definition is vital. The earlier we define this, the better, because it is the vision that drives the rest of the project. It also helps in the decision between building a solution or buying a package.

A good understanding of the business is vital

The initial benefit to gaining this understanding is that the business case, which positions the outcomes against the project investment, can include specific responsibilities and control points, establishing clear accountability for delivering outcomes.

The second benefit is that it demands that IT system requirements fulfill the needs of specific people in the organization. Requirements expressed in terms of their roles and their activities are far easier to manage than catalogs of requirements written in a technical language with no accountability.

A "business" is a series of processes, carried out by people, and supported by IT systems

One way to ensure we understand and document the business well enough to satisfy desired business outcomes is to perform business process modeling.

A living blueprint of your business – a model

Creating a business process model of the business creates a living blueprint of how the organization works. It shows how different processes interconnect by modeling the outputs of one business process as the inputs of others, and so on.

Models can be constructed top down – starting from the vision down to the details, or bottom up – from a single process up to the strategy.

This makes a business model an ideal way to capture and document the inherent complexity of business processes. It can conveniently display different levels of granularity that a written document cannot, making it a much richer and fuller account of the business.

Business process models are an excellent way to understand the complexity of modern businesses

Systems based upon requirements derived from a business model are much more sensitive to context than those derived by traditional means. This translates to fewer surprises downstream, avoiding those all-too-common scenarios where a system is delivered that won't work with other systems or doesn't satisfy the business as a whole. This ability to see the entire end-to-end perspective of how each discrete activity contributes to the business allows project teams to deliver requirements for a business outcome, rather than for a set of loosely connected function points.

Secondly, business models are very effective mediums for communicating this knowledge. For example, imagine verifying with key stakeholders that documentation for a business process

is correct. Traditional methods require stakeholders to read a document containing thousands of individual requirements and combine these in their mind to determine how accurately they represent the business process.

In contrast, a business model presents the entire process as a visual model, both at a high level and by drilling down into the details. It is far easier to comprehend a business process when available in this format. It is also easier to verify its accuracy because it is possible to view different perspectives of the same information.

The traditional medium for capturing and communicating requirements – a set of discrete electronic documents - is simply inadequate for capturing and communicating the complexity of business processes.

Business model as the common language

When the requirements dialogue between the business and IT goes wrong, neither party is wholly to blame. What is missing is an intermediary, a common vocabulary language, which facilitates the encoding from the real world operation of the business into the technical specifications.

This is where creating a business process model helps in defining the processes that will support desired business outcomes.

Business process models are an excellent medium for communicating requirements

When capturing business activities, business analysts work with business people to model how they do their job. Because the business model is visual and can be manipulated to display different views of the same information, it is much easier for business analysts to verify with a person if they have captured their role faithfully.

The ongoing dialog between those who create the model and workers participating in the processes being modeled ensures an accurate representation of how the business works. Stakeholders can verify the accuracy of the model, in an iterative and exploratory process. It also means that the business is involved in the requirements process at each step.

The business model is the common vocabulary for describing the

business that we have been looking for. Because it is visual, it enables members of the business and business analysts to communicate effectively. This is a much smaller cognitive leap than expecting a person to articulate how they would like a system to work. Business analysts can focus on documenting how people actually do their job and encoding this into requirements.

Because we are starting with a much better understanding of the business, we are much better placed to leverage this understanding to derive functional requirements that will directly support desired business outcomes. Business analysts can facilitate collaboration between the business and IT using the process model as a common language.

A common language is essential for developing a common understanding

So creating a business process model before we start gathering requirements has many benefits, and is likely to result in either better written software or superior selection processes for packaged software. The model helps us convert a desired business outcome into desired business processes, and from there into a set of functional requirements.

In addition:

- if the system cannot meet the requirements as defined, the impact can easily be understood and discussed with the affected people, and tradeoffs worked through sooner rather than later
- the model can be used to relate delivered functionality back to business users, identify training needs and implement changed procedures.

Bringing it all together - traceability

We have seen how a process model helps us base technical specifications on a better understanding of the business. But what about relating technical requirements to the business outcomes?

One of the key activities in requirements gathering is encoding – translating desired business outcomes into technical specifications.

We have used the business model to help encode each stage into a different perspective of the same problem – achieving the

desired business outcomes. So we have a record of how each perspective was abstracted into a particular version of a solution.

Until now we have only talked about encoding one way – from the desired business outcomes through to technical specifications. But of course we can go the other way. We can *de*code from technical specifications to business outcomes.

While this idea is usually neglected, it can provide many valuable outcomes.

The reason is that both business processes and the software based upon them will inevitably contain compromises. Using a business process allows us to relate functional requirements back up to the desired business outcomes.

This provides an auditable realization as to what degree the software satisfies the business need. Finally, we can now judge project success according to satisfaction of desired business outcomes, rather than only the degree to which the project delivered on requirements.

Tracing back up from requirements to processes and desired business outcomes allows us to measure project success

Tracing back through the different perspectives also allows us to assess the impact that changes in one perspective has on another.

Unifying requirements gathering

Perhaps most importantly, the business model unifies these four steps – business outcome, business process, technical requirement and technical specification - into a coherent, managed process within a single repository.

Most of the activities we have described here, such as documentation of business processes or delegation, will already be assigned to individuals in the organization. However, these activities are rarely unified into a managed process. The business model allows us to pull together these activities and store them in a single repository, connecting them together.

So there is not much more effort in creating a business model and using it to generate more appropriate requirements.

Furthermore, the ability to visualize roles and how they connect with other processes helps people see how their personal efforts contribute to the company's goals. They are also much more likely to support a system derived from a visual model of their role that they were able to understand and verify.

Even if the model is only used for one project, the way it improves understanding of that part of the business is invaluable. Yet the model can become a substantial corporate asset for the life of the business, useful as a reference of how the business works or as an exploratory tool to predict the impact of future changes.

Furthermore, a good business model can be used for many other things, so the return on investment is substantial.

For example it is possible to use the model to help design test cases that test not just individual function points, but end-to-end processes as well. Or it might be used to re-engineer existing business processes.

Conclusion

Business people are not satisfied with the IT systems being delivered as they do not deliver the desired business outcomes. Many of the issues relate to the communication between the business people and IT systems providers.

Relying on a translation of business outcomes to business requirements by a business analyst or consultant increases the likelihood that requirements will not be accurately represented. Thereafter, even the most sophisticated requirements management approaches will not deliver the business outcomes. At best, the "interpreted" business requirements will be met.

A business process model provides a framework to manage this extended requirements process. The process model relates desired business outcomes to business processes, and documents the roles responsible for delivering on those processes. The model makes describing actual business activities into requirements more effective - business analysts can verify and cross-check with stakeholders that the model faithfully reflects reality.

Equally important, a business process model allows us to relate all the stages of the requirements gathering process together. This gives us an auditable account of how desired business outcomes were related to business processes, and how these processes were converted into functional requirements.

To anyone interested in improving how technology can service business needs, business process modeling shows the promise to significantly improve the accuracy of system requirements and bridge the 'requirements gap'.