Requirements Eng (2007) 12:121–123
DOI 10.1007/s00766-007-0043-3

ORIGINAL ARTICLE

Business analysis and requirements engineering: the same, only different?

Jason Rubens

Published online: 11 April 2007
© Springer-Verlag London Limited 2007

1 Introduction

This essay has its origins in a conversation at a recent requirements engineering (RE) conference, where I suggested that much of the material under discussion would be familiar to industry practitioners in Australia, but that this work would be recognised as part of the job role of ''business analyst'' (BA). The purpose of this essay is to describe what it is that a BA does, and to compare this with the role of a ''requirements engineer''.

My contention is that, although there is patently overlap with RE, the practice of business analysis includes other activities that appear to be outside the scope of RE. Requirements engineers reading this essay may disagree—they may consider that the job functions I describe are all legitimate parts of current RE practice. Rather than form an exact conclusion about the extent of the business analysis domain, this essay aims to describe the focus of my role as a BA, and to highlight a particular approach to the job of business analysis, that may be of use in the work of other BAs and requirements engineers.

2 Business analyst job description

It should be mentioned that I work in software development, within a consulting environment, and this gives me a particular view of business analysis, which is reflected in this essay. Other readers may be working in product development, research, or other disciplines, and/or work more directly for the ''business'' parts of their organisations. These differing backgrounds and experiences will colour our various perceptions of business analysis or RE.

I suspect, though, that even if we had the same backgrounds, we would still have had very different experiences, and hold very different opinions on being a BA. This is because what someone with the title of ''business analyst'' does is so loosely defined within the industry. My colleagues and I have previously attempted to define what it means to be a BA, by creating a job description that included a wide range of practice areas, all potentially conducted by a BA as part of the system development life cycle (SDLC). I want to emphasise that any individual BA was not expected to perform in all of these practice areas, but that every BA was expected to work in some of these areas during their careers.

- *Functionality*: The definition of the behaviour of a ''system'', usually software or a physical product. This patently overlaps with the requirements gathering and management activities most often associated with RE.
- *Business strategy*: Understanding and defining the wider business context of the client, including the definition of mission and vision, and of specific strategies to achieve them.
- *Process*: Process modelling, innovation and design, similar to the business processes reengineering that has been fashionable in the process and manufacturing industries, although in our case more specifically applied to business processes that involve the use of software.
- *Finance*: Dealing with the financial value of business activities, particularly the cost/benefit analysis of new initiatives, business case development for new work, etc.

J. Rubens (✉)
Melbourne, Victoria, Australia
e-mail: jason.rubens@aanet.com.au

- *Organisation*: Defining organisational structure, including job roles, skills and training needs, best suited to accompany the system and process changes defined in the categories above.
- *Change management*: Managing the people side of change associated with system implementation.

There is a commonality in all of these practice areas performed by BAs: the topics are all concerned with analysing, codifying or implementing some change to the client business. In our job description we summarised that the BA works to ''ensure the delivery of a solution that meets the client's business needs''. That is to say, the ''business'' part of being a BA is in the identification and attainment of some benefit for the client. This means that the BA is focussed on delivering against some particular business goal, regardless of the method of delivery, which might be software, hardware, organisational change, etc. The domain of business analysis shows us a focus on, and even a responsibility for, delivering a solution that achieves some defined business outcome.

In the following sections, I would like to exemplify some of the ways in which this focus on business outcomes can be expressed by the BA, by looking at: whole system delivery; relationship management; and thought leadership.

3 Whole system delivery

Business benefits can be better delivered by engaging BAs across the entirety of the SDLC, not just for up-front requirements gathering. BAs can provide their knowledge of the requirements into other phases and other deliverables. Ideally, the BA acts as the ''interpreter'' during design and development, asking: What are the implications for the business, of using a given piece of technology? What should be built to satisfy a given business requirement? BAs help to raise functional issues from the developers to the stakeholders, and resolve them. BAs can also contribute to the internal process improvement of the development effort itself.

Having BA skills available during the whole of the system development process expands the focus of the requirements that they create. ''Good'' requirements will thus have the additional feature that they can be re-used by many other parts of the SDLC—development of system functionality, creation of test scripts, creation of training materials, and so on. Good requirements will also address the specific needs of other implementation activities, such as the need for specific testing and training environments, support of once-off data migration, etc.

It is worth noting the wider definition of a ''requirement'' or ''traceability item'' provided by the Rational Unified Process (RUP). The RUP defines a ''traceability item'' as any entity whose dependencies are being managed across the life cycle [1]. This is an extension of what we traditionally describe as a requirement: ''a condition or capability to which a system must conform'' [1]. This extended description alludes to the potential power of traceability: if the requirements model is used to trace entities across the breadth of the SDLC, it can answer the fundamental questions of implementation:

- Are we delivering what we said we would deliver?
- Are we delivering the right functionality, based on the agreed scope?
- Can we demonstrate our successful delivery through audit trails?

4 Relationship management

BAs can help projects succeed by helping to manage the human side of the requirements process. BAs are at the front line of client interactions, heavily involved with the stakeholders across multiple levels of a client organisation. BAs work with the client's executive management (who provide funding), the visionaries and change leaders (who lead the project and set its direction), and the staff and customer communities who form the user base (and who provide much of the detail of the requirements). This places the BA in a unique position to assist development of the client/vendor (or business/IT) relationship.

The BA helps to build trust that the vendor (or IT business unit) understands the business and the requirements, and will deliver against them. This is why we use traceability, so that we can prove we know what the client wants (or does not want), have documented it, and are managing against it.

Typically the more difficult client discussions centre on scope management. The BA works to ensure that requirements stay within the agreed scope, all the way to delivery of a live system, by managing the client's expectations of what will be delivered and when. This often means identifying when a new requirement is stated, and beginning the requirements management/project change management processes for it. It may also mean saying ''no'' to the client, but still maintaining their trust and belief in the solution.

5 Thought leadership

There has been a view in the IT/consulting industry that BAs provide only ''soft'' skills—that BAs are experts in leading creative discussions, workshop sessions, gathering

requirements, etc, but that BAs do not have (or for that matter do not need) any knowledge of the problem domain itself. Fortunately, this view is changing—certainly we are trying to change it in the consulting space—to one where BAs are engaged to also provide ''thought leadership''. That is, BAs will also provide expertise in both the business domain, through client and industry knowledge, and in the technical domain, through understanding the capabilities of the technology being used.

Ideally, a BA will understand the specific client's business, their competitors' businesses, and industry-wide trends and techniques that are being used in that industry. BAs will also have some expertise in the technologies being used, and so can provide guidance on what can be achieved technically—what the opportunities are and what the limitations are.

This means that the BA is not just a note-taker during the requirements and delivery process; BAs guide the process into directions they feel will be fruitful for the development of feasible, beneficial solutions. BAs try to steer clear of areas that are not viable, by using their experience and understanding of the business and of the technology. The BA aims to provide a balance between a whole world of possibilities, what the client actually wants, and what can be practically achieved.

6 Promoting the benefits

Having considered how BAs can express their business focus, it will be impossible for BAs to provide these beneficial inputs unless they are in fact engaged on implementation projects! Getting engaged means selling the value of BAs on a project, over the long term, to client stakeholders and project management.

I am seeing a growing level of understanding of the BA in the IT industry, but there is a still a lack of respect for the BA as a career decision: the role is seen as a set of general tasks that any team member should be able to perform. My view is that a BA has a particular focus that differs both from the developers and user communities within which BAs typically start their career. Business analysis represents a large body of practice, in which it takes experience and dedication to become expert.

A useful corollary is the work of project managers (PMs)—also a skill that has not always been so widely accepted as useful in the IT industry, but which has already attracted an increasing level of respect. I expect BAs will have to mimic some of the changes that have occurred for PMs, so as to gain better recognition and understanding of their job role, such as:

- Greater movement of innovative, useful and practical techniques between industry and research.
- More mature, widely known and accepted methodologies and Bodies of Knowledge, perhaps with training and industry certification against these.
- IT consultancies offering business analysis as a saleable skill, even to the extent of specialising in BA work.

With these sorts of actions we should imbue a better understanding in other IT practitioners of what BAs actually do and of the value of BAs' work in delivering business benefit. Eventually, we would look to create a demand for BAs by client business, where clients understand that having BAs will improve the delivery of the business benefits of development projects.

7 Conclusion

I have tried to show in this essay what the BA role might be: that the role includes established RE subject matter, but can also include tasks from wider areas of practice. Most significantly, business analysis has a particular focus: the delivery of beneficial change to the business. I trust that this attempt to highlight such a focus on business outcomes has helped to identify work that we, as BAs or requirements engineers, can do to provide solutions that actually deliver their promised benefits.

Reference

1. Rational Software Corporation (2003) Rational Unified Process® Version 2003.06.12. Rational Software Corporation