



Why requirements engineering is so much easier for a trained specialist.

First of all, it may be a much easier task than you anticipated. A professional will have handled projects like yours before and will understand many of the requirements that are common to all such projects.

For example a typical CMS project will produce 80 to 120 requirements of which only about 25 are likely to vary very much. It is not all that different for CRM, DRM, ERP or any other well established COTS system. In the above scenario, a professional, beginning with a well worn template will do a better job in a fraction of the time than anybody without this experience, knowhow and intellectual capital. The fastest way to a successful project is to have requirements recorded professionally by someone who has the above collateral at his or her disposal and knows what to look for and what to avoid..

When you, or a business analyst talk to your people about how they work “As is” or how they are likely to work “To be”, they will either be describing something that will no longer be relevant-the very reason you are doing this exercise, or guessing at something, of which they have no experience.

An experienced professional knows how to make the judgements, what to discount and how to generate the right mind frame through proven techniques to elicit real useful requirements that won't create problems with stakeholders or lead to issues at implementation time. A professional also knows how to verify requirements in order to reduce and manage risk to your project.

The importance of requirements to the success of your project.

The success of every software project, be it corporate change, or a product, relies on delivering value to stakeholders by satisfying and supporting users in their tasks.

Achieving this goal relies on gaining a deep understanding of what tasks are required to be supported and what the issues, problems, risks and pain-points are for users, while bearing in mind stakeholder needs. If the requirements fail to interpret the needs then the system will not deliver value..

Requirements form the absolute basis on which the functional design is based and on which the technical specifications are drawn and ultimately the finished system.

When the testing team are ready to begin work, it is on these very requirements that they will base their test cases.

When the supplier demands payment it will be on the basis of requirements met that they will expect payment and refuse to make changes.

Requirements therefore are the foundation of everything a software project, or for that matter any product is based on. Get this wrong and everything else is doomed to fail.

Finally many software professionals fail to recognise that requirements elicitation is the first and most important activity in the change management element of the program to ensure a smooth roll out and realisation of benefits. Handled correctly, a requirements engineering exercise enables people to voice misgivings, earn trade-offs and gain real involvement and buy-in to the process of change.

Who are the critical stakeholders?

Every project and every product is a fine balance between three key groups, **stakeholders** who finance and ultimately expect to gain from the efforts, **users** who must receive perceived value if they are to use the product, and the **delivery team** who must perform to a certain level of quality in order to deliver value as promised.

When launching a product the users are members of the public who have to be reached by a careful combination of consultation methods to ensure a representative sample have been consulted and their answers verified. In a software project, the job is easier because it generally involves consulting employees. There is always a dynamic initial period of potential

trade-off or even abandonment as differing user needs are communicated back to stakeholders who may no longer want to proceed or may have to rethink the product and revisit the business case. In the case of a software project it is important that the requirement is feasible and that the final deliverable can be reasonably expected to meet expectations. In the case of a product the team have to be able to deliver consistently to the expected level of quality. It is important therefore, that before the requirements are signed off, they have been discussed as appropriate and feasible with all three interested parties.

Language is everything when defining requirements.

Most software projects, especially the well publicised failures begin with a programmer or middle manager who is well known for “being well up on this stuff” taking on the role of writing out requirements. In either case the requirements will invariably be a Pseudo Technical design and nothing, that even closely resembles requirements. It will probably have been done with a quite deliberate disregard for certain important people who might disagree with the hypothesis and may well be loaded with subtle political tripwires. Hardly surprising then, when the project doesn’t work.

The correct representation of requirements is in the voice of a user advocate. The writer is firmly in the corner of the user when writing out requirements and makes a very distinctly noticeable point of never so much as hinting at how this functionality might be delivered, simply what it will enable the user to do.

This is vitally important because it allows the system architect, designers and developers to sit together with a clean sheet and the minimum constraints and to design the most efficient solution possible.

Knowing when to stop is a must have.

The trouble with successful requirement gathering exercises is that they generally stir up an avalanche of “bright ideas” and this is not always from the lower end of the management food chain. Would that it were.. It is important therefore that before beginning the exercise the constraints are acknowledged and the scope is well set out so that when the job is done, change control can be put in place and rigorously managed. One of the commonest causes of project failure is requirements that are missed, interpreted, or remain completely fluid throughout the process.

The cheapest and quickest way to a successful outcome is to get the requirements right, get them all, communicate them effectively and be prepared to make small changes if necessary.

In summary

Although requirement management is a critical part of any product or software project development and full of pitfalls for the inexperienced, it can be relatively painless as well as infinitely productive when handled by a professional. Whether you benefit from using a consultant will always depend heavily on the experience and tools they bring with them to leverage extra value in your project. In requirements management there are many opportunities to achieve savings and limit risk through partnering with a competent consultant.

Copyright Edward Taaffe © 2006

Edward Taaffe is a business consultant with a track record in high tech and with broad experience of business analysis, product management, knowledge management and change management in the private and public sectors.

Links:

In the news : <http://www.siberlogic.com/pressreleases/2006-01-27/default.asp>

Website: www.thebridger.co.uk

Email: ed@hebridger.co.uk