Requirements Overview

BASIC INTRODUCTION & THEORY

There are not many good examples of requirements to learn from, partially because so few people know how to write good requirements, they can be interpreted in so many different ways, the customer who wrote them is no longer around or available and it was deliberately written to give them wriggle room, and finally they are often so tied up to commercial contracts nobody wants them to be in the public domain. Those that have investigated requirements in detail, know that the good ones follow a prescriptive formulae, in terms of they are:

- **Correct** – unique, no-conflict and accurately describing delivered functionality;
- **Feasible** – implementable in terms of known capabilities and system limitations;
- **Necessary** – something the customer needs, not what they want, plus conformance;
- **Prioritised** – listed in terms of customer value, implementation cost and technical risks;
- **Unambiguous** – everyone who reads it must draw the same interpretation of it;
- **Verifiable** – how can it be tested, inspected, demonstrated, supported, maintained?

You must not forget other factors such as:

- **Complete** – at an individual level, system and enterprise level, including business level;
- **Consistent** – making sure there are no conflicts, disagreements or inconsistencies;
- **Modifiable** – make sure you can modify the change history, group/cross-reference them;
- **Traceable** – be able to link them hierarchically and to design elements, test cases/reports;
- **Tradeable** – be prepared to trade them as they don’t all have the same value/importance.

We should all be aware of the MoSCoW Prioritisation Method used to help reach a common understanding with stakeholders, which is easy to remember but very hard to put into practice:

- **Must Have** – critical to be delivered in the current timebox as the Minimal Viable Product;
- **Should Have** – important but not in the current timebox or time critical;
- **Could Have** – desirable elements but not necessary, used to improve user experience;
- **Won’t Have** – agreed as the least critical, lowest payback or value or not at this time.

Get the basic requirements grouped into the right buckets is a great start, and you can create a firm baseline from which to move forward into the specification and documentation phase (which are both collections of information about the end product at various times in the product lifecycle).

- **Stakeholder Business** – linked to what must be delivered, accomplished or enablers;
- **Stakeholder Customer** – what the paying and accepting into service communities specify;
- **Stakeholder End User** – often a neglected group how actually use the products/process;
- **Functional Product** – linked to the properties and capabilities of the product or system;
- **Functional Process** – linked to specific methodologies or organisational constraints;
- **Non-Functional Service** – linked to how well it performs its function and/or quality;
- **Constraints** – imposed limits on the project/process operations, alternatives or technology;
- **Implementation & Delivery** – similar to constraints but related to use, evolution and approvals.

When building or buying complex systems there are two types of complexities:

- **Accidental complexity** – focus is on reducing this by choosing the right tools and software;

- **Essential or inherent complexity** – can only be managed often through an incremental way.

It is important to realise this distinction, because some issues can be reduced through standardised processes and procedures, and others need to be personally managed carefully as if creating a work of art, with focus, attention to detail, feedback loops, regular reviews and an enquiring mind.

The requirements management process is traditionally broken down into a linear sequence as follows:

- **Discovery or Elicitation**
- **Analysis**
- **Modelling**
- **Documentation**
- **Communication**
- **Validation**

THE LENSE OF REALITY

Now bring in reality where Customers don't often know what they need at the start, there is scope creep and change throughout the project/programme that impacts budget, schedule and deliverables, and the whole process of eliciting detailed requirements at the start, which if you get it wrong, cripples the resultant outputs and outcomes. The process of evaluating proposed changes to requirements (i.e. cost, impact on work already completed, system/enterprise level and the delivery milestones, the additional risk being brought into the project, the effect on the company resource pool and knock on impact to other requirements, projects and programmes, the political and external factors), approving and incorporating the change into project/programme and the effect on re-baselining, then tracking the degree to which the requirements have been implemented, can be substantial. This is coupled with the fear that Stakeholders may have to re-prioritise, come up with more budget and not get what they want in the required time period with the allocated resources. Re-negotiations are rarely win-win environments.

This bring into a focus an often cited metric (credited to Jim Johnson, chairman of Standish Group based on four internally developed products at four companies) that states 45% of features in a product are never used; 19% are rarely used; 16% are sometimes used; 13% are often used; and finally 7% are always used. Now add in Paretos' Law that states that 80% of the results may actually come from only 20% of your efforts.

Let us dissect the 'Must Have' requirements into two parts and it is often the unwritten key to succeeding in business:

- **Needs** – Essential to life, Indispensable, A Necessity which equals the 7% always used and solves a real or customer imagined problem. This can be very difficult to extract from customers and users, because they often don't know what they need, take it for granted that people already know or need to see it created before they can determine how to use it properly and know what they are really going to use it for to create value ;
- **Wants** – Simply something that would nice to have, however it is high up there attached to meeting or exceeding customer expectations, a very powerful motivator and helps to build lucrative ongoing relationships where they feel they are in control. In the above example, I will attach this to the 13% of the features of the product that are often used.

This then leaves the 'Should Have' category which neatly fall into the lower expectation category (i.e. Customer expects them later) and I will attach this to the 16% of sometimes used product feature capability. I have deliberately left the rarely or never used functionality and their associated requirements out of the mix, because investing in these areas provides limited value add and just adds complexity, risk and delays to most projects/programmes. If you are doing something rarely or never, you should really ask yourself two questions: (1) Do I really need to do it and what is the impact if I don't do it? (2) Is there another way of accomplishing the output or outcome without bring it into the mainstream project/programme in an effective and efficient manner? Rare things should be left in the museum or brought out as needed/completed by Subject Matter Experts who can undertake the task far more effectively than trying to train everyone, and investing resources to reduce learning decay.

THE WHAT IF THOUGHT PIECE

Assume there is traditional £100m programme that needs to be delivered based on 1,000 requirements that have been created in the traditional method with the assumption that they all need the same amount of effort to solve. Based on the evidence presented above, not all the requirements are needs (7%) or wants (13%), which equates to 20% of the requirement list. If you were to really distil the requirements baseline down to that 20% of needs and musts (it will not be easy), it could give 80% of the solution at 20% of the cost (£20m). If you could find other smaller and cost effective solutions or alternatives for the 16% of the sometimes used capability, and couple this with additional processes rather than systems for 19% rarely used functionality (assuming this second stage costs half of the first stage, i.e. £10m), and add a final optimisation/improvement stage (£5m for 10% functionality improvement) that adds further need requirements once the users have discovered what the system does and requested additional modifications to determine their true needs. This all means that from a Customer point of view, a £100m programme could in theory be deliver a 90% solution for £35m. Traditional programmes it would seem typically could spend 80% of their budget (£80m this case) on the last 20% of capability, when research indicates that 45% of functionality is never used, and a lot of the budget could be funding this gross waste of resources. The last factors to consider are:

- **Revenues and Profits** - companies need revenue and profit to survive, so it will be very hard for the company to say that they can deliver 90% capability for 35% of the budget;
- **Brave Enough** – what customers are brave enough to say enough is enough to their end users, and say they will not be getting 10% of their requirements. We have all been conditioned to give or reach 100%, sportsmen promise 110% and if you really want to try hard, you aim for 120%. They are all clichés and we need to think of better ways of expressing our ideas and of applying our resources;
- **257%** - if £35m can deliver a 90% solution, you will still have £65m left in your original budget to deliver another 1.57 similar sized programmes. This means that taking a pragmatic approach to requirements, you could in theory get a 257% solution or additional capabilities based on not trying to achieve a 100% solution, by taking this different approach. If we psychologically reset our aim point, we could achieve extraordinary things. Leave the 100%, 110% and 120% for the Olympians to chase after, whilst the ordinary businesses deliver capabilities beyond the wildest dreams of their customers, users, shareholders, stock markets, business owners and employees. What a place that would be to work for!!!