

Tips and Traps on Requirements Gathering

Never before has requirement collection been so important to businesses. This function now accounts for approximately 10% of all spending in the systems development lifecycle. How companies link their requirements gathering to their system development and project management methodologies, is significant in terms of cost, scope control and time to market.

The toughest job in the requirements definition stage is to get stakeholder consensus as a systematic, expected deliverable in the project cycle. Yet it is also absolutely essential for a team to have consensus if it is to lock a requirements specification at a point in time and after that to manage change control against this specification.

So, here are some Tips and Traps on Requirements:

Tips for Accelerating Consensus Building

Stakeholders must OWN the requirements

It's almost trite to say that stakeholders must feel a sense of ownership of the requirements, but we see all too often a process for requirements elicitation that actually diminishes this sense of ownership. A good example is when requirements are framed in the technical jargon of IT architecture. It is impossible to get stakeholders to take ownership of requirements they don't understand, but may be reluctant to admit. A sound process for requirements elicitation includes:

- Holding facilitated sessions where all stakeholders are present
- Using techniques and tools that engage the non-technical participant
- Setting the expectation that the stakeholder representatives will sign off on the detailed requirements specification.

Speed and efficiency are the essence of the process

Stakeholders will not participate if they feel their time is being wasted. Companies need to cut in half the amount of time they demand from users to define their requirements.

After this, if it takes more than a week to produce and approve documentation of the requirements, it is likely that the original requirement will have shifted somewhat. Speed is fundamental. Any approach that is not highly accelerated is unlikely to be consistently applied over a wide range of project types. The essence of getting speed is to use a highly disciplined approach to managing the elicitation sessions. Such an approach forces the right questions to be asked at the right times, and prevents a group from going backward to rehash decisions that have already been made. A disciplined approach is productive, comprehensive, and exciting for participants.

Defined beginning and end point in the requirements gathering stage

Stakeholders will participate in a process in which there is a clear beginning, clear momentum during the process, and a valuable product at the end. If the requirements definition process starts to wander, stakeholders lose interest, and then it is extremely difficult to rebuild their motivation and energize the project.

Traps that Get in the Way of Consensus Building

Getting too focused on a technology too early in the process

The focus of requirements definition must be 100 percent on what is the business objective of the system. Don't make these common mistakes:

- While the requirements may include non-functional specifications such as responsiveness, they should be technology-agnostic
- Arguing the merits of one technology over another, too early in the process distracts from the need to have absolute clarity on what the technology needs to do.
- Getting too technical muddies the water on the ownership of requirements since it forces ownership away from the business stakeholder.
- Selecting a particular application package too early in the process will tend to disengage consensus building.

Insufficient detail in how the data flow needs to be handled

It is relatively easy to get consensus on all the high-level functions that a system needs to contain and to assign priorities to them. Even for a scope as large as Flight Operations for a major carrier, we can always complete this in less than one week. But this level of detail is insufficient to compare application vendors reliably, and it is a trap to think that consensus built at this level is adequate.

The issue is that the application may not manage the data flow in the same way that your people currently perform the work, and this won't be apparent from a high-level view of the requirements and the proposed solution. One of three things then happens:

- The application is implemented "vanilla" and people change the way they work to accommodate the application
- The application is customized to address the gap
- The functionality needed cannot be implemented.

In all three cases, unless the stakeholders anticipated and accepted this change of course, then the scope of the project will increase and stakeholder satisfaction with the result will decrease.

"Best Practices" Trap

One of the claimed advantages of a package is that it embodies industry best practices. Focusing on this possibility can be a subtle trap. Most organizations obviously want to make a decisive leap forward in process efficiency through the implementation of a new package, so it is easy to get distracted by this prospect. However, without knowing the degree of change from the existing process that this implies, it is difficult to precisely control scope and ensure that essential aspects of functionality are not lost in shooting for this "uber-function."

When looking at best practices, keep le Châtelier's Principle in mind: the greater the stress you put on a system, the more the system fights back to return to its state of comfortable equilibrium. Achieving great benefit then means dealing with significant organization resistance unless consensus is first reached on how the change will impact at the grass roots of an organization.