Where Intangibles Collide – Stakeholders and Software projects

Alistair Duffy and Hugh Sasse
De Montfort University, UK
apd@dmu.ac.uk, hgs@dmu.ac.uk

There is a school of thought that says software systems do not work properly until they have failed repeatedly in action. There are various methodologies for software development, some very academic, some less so. In all cases, a limiting factor on whether the software will do what it needs to do is how well stakeholders have been included in the project. This article addresses some of the stakeholder factors pertinent to software-heavy (computational electromagnetics) projects. The importance of identifying and addressing the needs of stakeholders can be clearly seen in a wider context by looking at Concorde and the Brent Spa oil rig [1] although neither were computational electromagnetics projects, they do illustrate a point. The Concorde design is perhaps best described as iconic, but as a project it was not quite as successful because although it was designed for transatlantic flights, the USA had virtually no involvement an as a result it took quite some time for the routes to be opened up. Similarly, if the disposal of the Brent Spa oil rig had involved negotiations with environmental groups from the outset, substantial amounts of money could have been saved, not to mention avoiding a public relations disaster.

From a software angle, "In the real world, stakeholders are interested in solutions, not some abstract developer-centric set of requirements" [2]. So it is important to identify who the stakeholders are, what their interaction is with the project, what they want to get out of it and what the conflicts are between stakeholder requirements. This article will look at defining and identifying stakeholders and then how the development process can benefit from this.

The benefits of identifying stakeholders are both direct and indirect. Direct benefits include ensuring that all the groups who should be involved in the project have their contributions and expectations clearly built into the project plan, this includes involving the stakeholder groups when required and identifying conflicting stakeholder requirements that need managing. Indirect benefits include the resultant awareness of possible weak signals that, as a project team member, being sensitised to "stakeholders" can produce. Weak signals are deviations in the noise floor that can signify a growing signal, spot it early enough and you can deal with it, leave it too long and it can become a problem: the problem with the noise floor is that you need to be looking at it carefully and you need to adopt an appropriate filter. However, by cycling through stakeholder requirements and negotiating conflicts, many of these issues can be minimised.

Stakeholders

There are probably as many definitions of stakeholders as there are people discussing it. Even having decided on a definition, it can be difficult, if not impossible, to account for all stakeholder relationships, but as noted above, being sensitised can be a benefit in itself. Some approaches to stakeholder identification include lists of possible involvements. Such as in [3] where the author cites a World Bank view of stakeholders including those:

- Who might be affected by the development concern to be addressed
- Whose behaviour has to change for the effort to succeed?

We find these very helpful but prefer a simpler push-pull view of stakeholders: a stakeholder is someone (or a group) who contributes something to the project in order to get something out of it. This loose definition has the disadvantage that it requires more than a tick list to identify the stakeholders but has the advantages that virtually all stakeholders can be identified this way by thinking creatively about the push and pull. Internally, stakeholders will include the programmers (contributing their effort and giving up the opportunity to work on other projects in order to gain remuneration, job satisfaction and intellectual challenge). Externally, the stakeholders will include (but not be limited to) customers who will contribute money, possible IPR and give

up the potential for working with another organisation in order to receive a product that allows them to do the job they want doing.

A helpful tool in identifying stakeholders is to group them [3] into primary, secondary, external and extended stakeholders.

- Primary stakeholders – who have authority or resource responsibilities, or who have the power to influence collaboration outcomes. Their involvement is vital otherwise problems may ensue. Naturally, a customer for a solver will be intimately involved in the project planning process.
- Secondary stakeholders – have an indirect interest in the outcome such as with employees or consumers. They should be involved in elements that impact on them but not in other areas. Here, for example, an end-user may be involved in discussions about front-ends but not necessarily data structures.
- External stakeholders – are not part of the project but will have an impact on the project. An example of these may be regulators or grant awarding bodies. They will be expecting something out of the project and need to be involved carefully, possibly by lobbying, letters or discussions.
- Extended stakeholders – are more likely to influence the overall impact of the finished product (a conference audience, perhaps, or the readers of this article (?)). It is difficult to account for this group as easily as the other three groups. Their involvement may apparently be one way: they consume what has been output from the project. Certainly, their input to the project itself will be limited, the danger being that any input they have may not be predictable *a priori*.

The three main categories that software projects fall into are [4]:

- Standard package
- Customised
- Bespoke

In a rather more ordered world than the one we currently live in, these three categories would exist separately. However, for CEM, they can become somewhat intertwined. In the extreme case, a bespoke package will be developed for a customer, based on the customising of a current package that will also be included in a standard product. All the stakeholders need to be considered but possibly with different emphases.

The following table gives a starting point for identifying stakeholders for a project (some suggestions are in place). What it really suggests is that by creatively thinking about those involved, they can be categorised and treated in a way that is relevant to their impact on the project

	Primary	Secondary	External	Extended
Standard		Sales team	Government funding body	
Customised	Customer			Conference attendees
Bespoke	Customer (contract placer)	End-user		Buyer's customers

This approach can be further elaborated by considering the stakeholders as *personas* [5] although we are not going to discuss this further here.

Stakeholder involvement

We now have a potentially simple structure to identify the stakeholders and what they want out of their relationship with the project. In each case the following need to be clearly identified

- What are they contributing or giving up?
- What are they getting back from the project?

- How should they be involved? What is the best way of communicating with them
- Are their any conflicts with other stakeholders?

Naturally, major consideration needs to be given to those with the greatest ability to influence the outcome: the primary or secondary stakeholders. It must be remembered that all stakeholder groups can have an impact on the success or failure of a project, or on whether it comes in late and over budget. This section looks at how this involvement can be effectively utilized.

There is a certain amount of negotiation required to identify the most relevant stakeholders, their push-pull attributes and how they can and should be involved. A useful framework for undertaking this negotiation is the WinWin Spiral [6, 7]. The method is also claimed to enhance trust between stakeholders and to encourage flexibility within the team.

In essence, the WinWin Spiral Model [7] is based on the following activities:

1. Identify the key stakeholders.
2. Identify the stakeholders' win conditions (their goals from the project). If their win conditions are non-controversial, i.e. they do not conflict with other win conditions, they are accepted as they are. Otherwise, an 'issue artefact' is created and this becomes the topic of negotiations to resolve the issue.
3. Negotiate win-win reconciliations.
4. Establish objectives, constraints and alternatives.
5. Evaluate the alternatives and identify and mitigate risks.
6. Define the next level of product
7. Validate the product / process definitions
8. Review and commit
9. return to #1 for the next cycle to deal with the next level of detail.

Several cycles at increasing levels of complexity may be necessary. One example [7] demonstrates a four cycle approach for a multimedia project, although the general thrust can be a helpful starting point:

- Cycle 0 – determine feasibility of general approach.
- Cycle 1 – develop prototypes, plans, etc. and verify the existence of at least one feasible solution.
- Cycle 2 – develop more detailed plans and verify that there are no major risks in satisfying the specifications
- Cycle 3 – develop operational plans including user support plans.

Any other implementation of this may require more or fewer cycles, but consideration of the four cycle approach is a good starting point.

It should be noted that there are other software development methodologies which address similar issues. For example Agile Methods such as Extreme Programming [8], which develop, in short iterations, only a limited number of User Stories with customer involvement on the development team throughout.

However, embedding the identification of stakeholders as discussed earlier into the WinWin Spiral model should help in minimising 'nasty surprises' and maximising the quality of delivery, even if a formal software requirements specification (SRS) approach is being used [e.g. 9]. So, hopefully, repeated failure will not be as important in order to prove that the software works.

References

[1] S. Mann, "Stakeholders can ruin your day", Professional Manager, November 2004, pp 26 – 28.
[2] D. Thomas and A. Hunt, "Nurturing Requirements", IEEE Software, 2004, pp 13 – 15

[3] J. McManus, “A stakeholder perspective within software engineering projects”, IEEE International Engineering Management Conference, 2004, pp 880 – 884
[4] J. Giesen and A Volker, “Required interdependencies and stakeholder preferences”, IEEE Joint International Conference on Requirements Engineering (RE ’02), 2002, pp 1 – 4.
[5] A Cooper and R.M. Reimann, “About Face 2.0. The essentials of interaction design”, Wiley, 2003
[6] B. Boehm and A Egyed, “Software requirements negotiation: some lessons learned”, Proceedings of the 20th International Conference on Software Engineering, April 1998, pp 503 – 506.
[7] B. Boehm, A. Egyed, J. Kwan, D. Port, A. Shah, R. Madachy, “Using the WinWin spiral Model: a case study”, Computer, IEEE, July 1998, pp. 33 – 44.
[8] K. Beck and C. Andres, “Extreme programming explained 2/e”, Addison-Wesley Professional, 2004
[9] IEEE std 830 - 1998 Recommended Practice for Software Requirements specification