

Stakeholder Identification in the Requirements Engineering Process

Helen Sharp
Centre for HCI Design,
School of Informatics,
City University, Northampton Square, London,
EC1V 0HB, UK,
h.c.sharp@soi.city.ac.uk

Anthony Finkelstein & Galal Galal
Computer Science Department,
University College London,
Gower Street,
London WC1E 6BT, UK
(a.finkelstein, g.galal)@cs.ucl.ac.uk

Abstract

Adequate, timely and effective consultation of relevant stakeholders is of paramount importance in the requirements engineering process. However, the thorny issue of making sure that all relevant stakeholders are consulted has received less attention than other areas which depend on it, such as scenario-based requirements, involving users in development, negotiating between different viewpoints and so on. The literature suggests examples of stakeholders, and categories of stakeholder, but does not provide help in identifying stakeholders for a specific system. In this paper, we discuss current work in stakeholder identification, propose an approach to identifying relevant stakeholders for a specific system, and propose future directions for the work.

1. What is a ‘stakeholder’?

There is a large body of literature in the strategic management area which discusses organisations in terms of a stakeholder model. Stakeholder analysis, it is claimed, can be used to analyse an organisation’s performance and determine its future strategic direction. An oft-quoted definition of ‘stakeholder’, taken from a key reference in this literature is:

‘A stakeholder in an organisation is (by definition) any group or individual who can affect or is affected by the achievement of the organisation’s objectives.’ [10]

A much broader definition, which has also been attributed to Freeman, is that a stakeholder is ‘anything influencing or influenced by’ the firm, but it has been claimed that this definition is problematic because it leads to the identification of a very broad set of stakeholders. It is important to distinguish between influencers and stakeholders because while some potential stakeholders may indeed be both stakeholders and influencers, some who have a real stake in an enterprise may have no influence, e.g. a job applicant, while some influencers may have no stake, e.g. the media [8].

Information systems (IS) researchers have also taken up the idea of stakeholders:

‘We define stakeholders as these participants <in the development process> together with any other individuals, groups or organisations whose actions can influence or be influenced by the development and use of the system whether directly or indirectly.’ [22]

In software engineering, stakeholders have been defined as:

‘The people and organisations affected by the application’ [3]

‘System stakeholders are people or organisations who will be affected by the system and who have a direct or indirect influence on the system requirements’ [16]

‘Stakeholders are people who have a stake or interest in the project’ [4]

A more explicit refinement of this definition is:

‘... anyone whose jobs will be altered, who supplies or gains information from it, or whose power or influence within the organisation will increase or decrease.’ [7]

They go on to say that

‘It will frequently be the case that the formal ‘client’ who orders the system falls very low on the list of those affected. Be very wary of changes which take power, influence or control from some stakeholders without returning something tangible in its place.’ [7]

When faced with the practical problem of how to identify the set of stakeholders relevant to a specific project, these definitions are not particularly helpful. The main concern is that, although such definitions are usually accompanied by example groups of stakeholders, they are vague and may lead to consideration of inappropriate or incomplete groups of stakeholders.

Categories of stakeholder include end-users, managers and others involved in the organisational processes influenced by the system, engineers responsible for system development and maintenance, customers of the

organisation who will use the system to provide a service, external bodies such as regulators, domain experts, and so on. They will each have different goals, and will try to satisfy their own without recourse to others [16].

Cotterell and Hughes [4] suggest that stakeholders might be in one of three categories: internal to the project team; external to the project team, but internal to the organisation; and external to both the project team and the organisation.

Newman and Lamming [20] suggest a different division: into those who will use the system directly or indirectly, and those who will be involved in developing the system. This distinction is also taken up with respect to the development of knowledge-based systems [22]. The set of stakeholders in the knowledge acquisition *process* and the set of stakeholders in the *use* of a knowledge-based system, are not necessarily identical; they are likely to vary in membership, and for those members in common, the type and level of stake they have is likely to vary. The IS literature suggests a different division again, into ‘hubs’ or ‘sponsors’ and ‘spokes’ or ‘adaptors’, where the former are those initiating and sustaining the system, while the latter are those participating [21]. Macaulay [19] identifies **four categories of stakeholder** in any computer system:

1. Those responsible for design and development;
2. Those with a financial interest, responsible for its sale or purchase;
3. Those responsible for introduction and maintenance;
4. Those who have an interest in its use.

But again, she offers no guidelines for identifying specific stakeholders for a given system.

So far, we have not distinguished between individuals or groups and roles. As with other situations, the mapping between stakeholders and individuals or groups is not one-to-one. It is therefore more appropriate to think of stakeholders as *roles* rather than as specific people. In subsequent discussion, we shall use ‘stakeholder’ to mean ‘stakeholder role’.

2. Identifying stakeholders

There has been little focus on the participants of the requirements engineering (RE) process, for example in terms of how to trace participants in the RE process and how to identify stakeholders [13]. All of the references cited above emphasise the importance of identifying stakeholders, and although they provide examples, or broad guidance for identifying them, none describes a model or a concrete approach for identifying stakeholders for a specific project.

This deficiency has been noted in the management literature, and in the IS literature [21], where the

approaches have been criticised for either assuming that stakeholders are ‘obvious’, or for providing broad categories which are too generic to be of practical use. Expert identification for knowledge-based systems development has similarities with stakeholder identification in RE, although here too there has been an assumption that experts are readily identifiable [22].

Pouloudi and Whitley [21] suggest four principles of stakeholder identification, and describe an approach which is based on these principles, and which they used to identify stakeholders in the drug use management domain. Lyytinen and Hirschheim [17] also provides some guidance for stakeholder identification for IS, while acknowledging that ‘the identification of the set of all stakeholders is far from trivial’.

Methods for requirements engineering, e.g. KAOS [5], do not directly support stakeholder identification; it seems to be assumed that it is straightforward. The task of identifying actors for use case development [15] has similarities with stakeholder identification, but is targeted only at a fraction of system stakeholders.

Stakeholders are related to each other and interact with each other [22,11,17]. Interactions between them include: exchanging information, products, or instructions, or providing supporting tasks. Information about the stakeholders and the nature of their relationships and interactions needs to be identified and recorded. Dimensions of importance are: **relationships between stakeholders, the relationship of each stakeholder to the system, and the priority to be given to each stakeholder’s view**. This information is needed to manage, interpret, balance and process stakeholder input. Reconciling different stakeholder views is beyond this paper, but is considered in [12].

3. An approach to identifying stakeholders for requirements engineering

Here, we propose an approach to discovering all relevant stakeholders of a specific system that we believe is domain-independent, effective and pragmatic. We draw on the literature cited above, and on aspects of ethical decision-making in software systems [18].

Our starting point is a set of stakeholders that we refer to as ‘baseline’ stakeholders. From these, we can recognise ‘supplier’ stakeholders and ‘client’ stakeholders: the former provides information or supporting tasks to the baseline, and the latter processes or inspects the products of the baseline. Other stakeholders that we call ‘satellite’s interact with the baseline in a variety of ways. ‘Interaction’ may involve communicating, reading a set of rules or guidelines, searching for information and so on. Our approach focuses on interactions between stakeholders rather than relationships between the system and the stakeholder,

because they are easier to follow. Figure 1 illustrates the main elements of stakeholder identification.

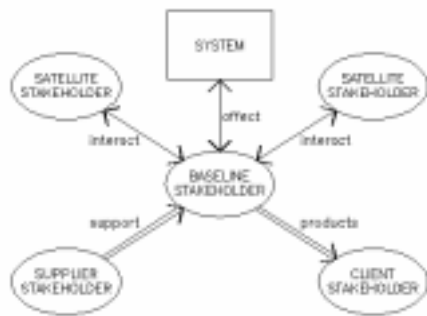


Figure 1. The main elements of stakeholder identification addressed by our approach

3.1 Baseline stakeholders

We have identified four groups of baseline stakeholder: users, developers, legislators and decision-makers. We have dubbed these ‘baseline’ because the web of stakeholders and their relationships can be identified from them. The nature of each group is explored below.

3.1.1 Users. The term ‘user’ has many interpretations. For example, Holtzblatt and Jones [14] include in their definition of ‘users’ those who manage direct users, those who receive products from the system, those who test the system, those who make the purchasing decision, and those who use competitive products. Eason [9] identifies three categories of user: primary, secondary and tertiary. Primary users are those likely to be frequent hands-on users of the system; secondary users are occasional users or those who use the system through an intermediary; tertiary users are those affected by the introduction of the system, or who will influence its purchase.

Here, we shall assume that ‘users’ are the people, groups or companies who will interact with the software and control it directly, and those who will use the products (information, results etc) of the system.

3.1.2 Developers. The developers of the system are stakeholders in the RE process, but their stake in the final requirements specification, or indeed in the system itself, is different from the users described above.

3.1.3 Legislators. Professional bodies, government agencies, trade unions, legal representatives, safety executives, quality assurance auditors and so on may produce guidelines for operation that will affect the development and/or operation of the system. Some of these will be local to the domain, others will be national or international; others will be local to the organisation.

3.1.4 Decision-makers. Within the development organisation and the user organisation, there will be decision-making structures that relate to the system under development. The kinds of stakeholder here would include managers of the development team, user managers and financial controllers in both the developer and the user organisation.

3.2 Exploring the network of stakeholders around the baseline

Below, we suggest a five-step procedure which can be used to explore the web of stakeholders around each of the four baseline groups. Before presenting the procedure, we make four further observations:

1. Identifying exactly what happens in a work place situation is not easy ‘out of context’ and we suggest using participatory techniques such as contextual inquiry [14]. The new software system may result in user roles being changed or created. It is not possible to observe these in context since they do not yet exist. However, existing practices can be observed, and the work design for any new role can be studied. For example, software system maintenance will not be observable at this stage, but observing maintenance activity on other systems may be possible.
2. Stakeholders may be internal to the team, internal to the organisation, or external to both.
3. Each business has its own cycle of events, and it is important to consider the *full cycle of activities* when trying to identify stakeholders, e.g. a full financial year, a full billing cycle, a complete mission, etc.
4. Software development has a lifecycle, and *the complete lifecycle* should be considered when identifying stakeholders around the developer baseline.

The approach basically uses the same five steps for each of the four baseline stakeholder groups mentioned above (see Figure 2). However, because the stakeholder groups represent different kinds of involvement, there is some customisation required for each one.

Notes specific to each baseline case are given below.

Users: The starting point for the user baseline group is the direct user roles, i.e. people who will actually operate the system. There are likely to be different groups of direct users who might be identified by, e.g. frequency of use of the system, experience of computer use, expected goals, position in the organisation, internal or external to the organisation, etc;

1. Identify all specific roles within the baseline stakeholder group;
2. Identify ‘supplier’ stakeholders for each baseline role;
3. Identify ‘client’ stakeholders for each baseline role;

4. Identify 'satellite' stakeholders for each baseline role;
5. Repeat steps 1 to 4 for each of the stakeholder groups identified in steps 2 to 4.

Figure 2. The basic five steps to identify the web of stakeholders from a baseline

Developers: The roles within the developer baseline group could include analysts, designers, programmers, testers, quality assurance representatives, maintainers, trainers, project managers and so on. The starting point for this section of the investigation could be any of these that are available. The influences on development teams are many and complex. For example, the nature of teams and their composition and the likely effect this will have on development is well-known (e.g. [6]). While we are not suggesting that all influences on the team should be considered as stakeholders of the final system, they need to be recognised as stakeholders in the requirements engineering process. These people will include other customers of the software development company, and those supplying this company, such as tool vendors and developers.

Legislators: Identify legislator roles and the remit they have, i.e. do they cover the operation of the system such as data protection legislation, or the development of the system, e.g. defence standards. These could be quality managers within the developer's organisation, auditors from an external agency, guidelines and rules, etc. Note that evidence of the existence of a legislator may be a document such as the quality manual.

Decision-makers: A good starting point for this group is the stakeholder in the user organisation responsible for commissioning the system. Identify decision-makers within the development organisation and the user organisation (which may be the same, or different) who have any power over the decision to build the system, or over any processes, people or standards identified in earlier stakeholder enquiries.

3.3 Potential benefits of this approach

It starts at a known stakeholder centre and works outwards. Hopefully this means that all important stakeholders are captured, and yet that irrelevant actors are not included. Consistency-checking is built into this process because the supplier stakeholders and the client stakeholders must overlap; if there is a product of the system for which there is no user, then something has been missed from the analysis.

3.4 Potential flaws in this approach

There is a danger that too much time is spent on identifying roles and relationships, and the team is swamped with data. Knowing when to stop looking is

just as important as knowing where to look. Guidelines for this question are likely to emerge from future work.

This approach might lead to a model in which stakeholders are identified in 'serial strings', thus ignoring inter-linking across the strings. The approach includes some self-checking, though, so these crossovers should become clear.

4. Future directions

4.1 Validating the approach

The approach presented here has not yet been validated. The next step is to deploy it on a real project. It may be possible to make some headway by considering a fictitious (but realistic) project such as the Meeting Scheduler [25], but we want to ensure that our approach is pragmatic, and a suitably realistic appraisal requires a real project. Issues to be addressed in this appraisal are:

1. When to stop collecting data and recognise that the scope of stakeholders is sufficient?
2. What is the most effective way to document the findings?
3. The approach described here needs to be refined in order to specify more pragmatically what to look for in work situations so as to uncover stakeholders and influential members of staff.
4. Suitable schemes for characterising and measuring stakeholder relationships need to be devised. One possible scheme might be: 'in charge of', 'supports', 'is crucial to', 'provides information for', and so on.

4.2 Longer term goals

The approach described in this paper will uncover a network of stakeholders, with each stakeholder being a node and each edge being the relationship between the two stakeholders. If a suitable network theory can be identified, it may be possible to automatically calculate the weight of a stakeholder's view in relation to the system, based on the inter-stakeholder relationships identified through the investigation. This in turn leads to the potential for developing tool-based support to identify important stakeholders and their level of influence.

Towards this goal, literature on network approaches to stakeholder analysis in other domains is worthy of investigation. For example, Social Network Theory [24] and Industrial Networks [1]. Work in domain networks [2] and goal reduction [5] may also provide useful techniques. Other work, worthy of investigation, includes algorithms built into space planning software for interior design which incorporate arithmetic for dealing with similar options.

5. References

- [1] Axelsson, B. and Easton, G. (eds) (1992) *Industrial Networks: A New View of Reality*, Routledge, London.
- [2] Bolton, D., Jones, S., Till, D., Furber, D. & Green S. (1994) 'Using domain knowledge in requirements capture and formal specification construction', in *Requirements Engineering: social and technical issues*, Jirotko, M. & Goguen, J. (eds), Academic Press, pp 141-162.
- [3] Conger, S. (1994) *The New Software Engineering*, International Thomson Publishing.
- [4] Cotterell, M. and Hughes, B. (1995) *Software Project Management*, International Thomson Publishing.
- [5] Dardenne, A., van Lamsweerde, A. and Fickas, S. (1993) 'Goal-directed Requirements Acquisition', *Science of Computer Programming*, **20**, 3-50.
- [6] DeMarco, T. and Lister, T. (1987) *Peopleware: productive projects and teams*, Dorset House Publishing, New York.
- [7] Dix, A., Finlay, J. Abowd, G. and Beale, R. (1993) *Human-Computer Interaction*, Prentice-Hall.
- [8] Donaldson, T. and Preston, L.E. (1995) 'The Stakeholder Theory of the corporation: concepts, evidence and implications', *Academy of Management Review*, **20**(1) 65-91.
- [9] Eason, K. (1987) *Information Technology and Organisational Change*, Taylor and Francis.
- [10] Freeman, R.E. (1984) *Strategic Management: A stakeholder approach*, Pitman, Boston.
- [11] Galal, G H and Finkelstein, A (1999) 'Thoughts on Identifying and Managing Stakeholders', Department of Computer Science Research Note RN/99/16, University College London, University of London.
- [12] Galal, G.H. and Paul, R.J. (1999) 'A Qualitative Scenario Approach to Managing Evolving Requirements', *Requirements Engineering Journal* (in press).
- [13] Gotel, O.C.Z. and Finkelstein, A.C.W. (1995) 'Contribution Structures', in *Proceedings of the 2nd International Symposium on Requirements Engineering (RE '95)*, IEEE Computer Society Press, 100-107.
- [14] Holtzblatt, K. and Jones, S. (1993) 'Contextual Inquiry: a participatory technique for systems design', in *Participatory Design: principles and practice*, Schuler, D. and Namioka, A. (eds), Lawrence Erlbaum, New Jersey, pp 177-210.
- [15] Jacobson, I., Christerson, M., Jonsson, P. and Overgaard, G. (1992) *Object-oriented Software Engineering: a use case driven approach*, Addison-Wesley, Reading, Mass.
- [16] Kotonya, G. and Sommerville, I. (1998) *Requirements Engineering: processes and techniques*, John Wiley.
- [17] Lyytinen, K. and Hirschheim, R. (1987) 'Information systems failures- a survey and classification of the empirical literature', *Oxford Surveys in Information Technology*, Vol 4, 257-309.
- [18] M880 (1997) *Software Engineering*, CCI Postgraduate Course, The Open University.
- [19] MacCaulay, L. (1994) 'Cooperative requirements capture: Control room 2000', in *Requirements Engineering: social and technical issues*, Jirotko, M. and Goguen, J. (eds), Academic Press Ltd., London, 67-85.
- [20] Newman, W.M. and Lamming, M.G. (1995) *Interactive System Design*, Addison-Wesley.
- [21] Pouloudi, A. and Whitley, E.A. (1997) 'Stakeholder identification in inter-organisational systems: gaining insights for drug use management systems', *European Journal of Information Systems*, **6**, 1-14.
- [22] Pouloudi, A. (1997) 'Stakeholder Analysis as a Front-End to Knowledge Elicitation', *AI & Society*, **11**, 122-137.
- [23] Scott, J. (1991) *Social Network Analysis*, Sage, London.
- [24] Van Lamsweerde, A., Darimont, R. & Massonet, Ph (1993) 'The Meeting Scheduler System - Preliminary Definition', Internal report, University of Louvain.