ACKNOWLEDGING THE IMPLICATIONS OF REQUIREMENTS

Ken Boness, Rachel Harrison, Kecheng Liu
Department of Computer Science,
The University of Reading, Reading, Berkshire, RG6 6AY, UK

Keywords: Natural Language Requirements, Validation, Profiling, Goal Orientation, Semiotics.

Abstract: The traditional software requirements specification (SRS) used as the principal instrument for management and planning and as the foundation for design can play a pivotal role in the successful outcome of a project. However this can be compromised by uncertainty and time-to-market pressures. In this paper we recognise that the SRS must be kept in a practical and useful state. We recognise three prerequisites to this end and introduce a programme of research aimed at developing a Requirements Profile that changes the emphasis of requirements engineering from defining the requirements to defining *what is known* about the requirements. The former (being a subset of the latter) leaves the traditional idea of a SRS unaffected whereas the latter adds much to the avoidance of misunderstanding.

1 INTRODUCTION

This position paper describes work in progress. Consideration of the practicalities of working with a software requirements specification (SRS) leads us to propose a requirements profile. Our expectation is that it would be used to annotate a SRS and cause a shift in emphasis from simply defining requirements to also quantifying our assumptions about the requirements; e.g. “we are 95% certain that we understand the requirements well”.

A software requirements specification defines what is wanted from a new or modified software product that is expected to function in a particular environment of users, organisations and machines. In traditional requirements engineering a SRS written in a format such as recommended in the IEEE standard (IEEE 1998) would be used as the principal instrument of specification, negotiation and management. In this role it would ideally provide benefits to the project such as the basis for:-

- realistic estimation of cost and risk;
- avoiding wasted or misguided effort;
- architecture and design;
- testing and acceptance;
- management of releases;
- a clear understanding shared by all stakeholders.

However often in real projects it is the case that requirements remain uncertain until late on. Furthermore, often in industry there is the urgency of a time-to-market deadline. Both factors can make the benefits of working with a SRS seem unattainable.

This conflation of urgency with uncertainty is often the reason why iterative-incremental development methods are chosen over the waterfall method. However such methods can be consistent with the diligent use of a SRS.

When these conditions are severe the role of the SRS may break down. Some developers take the view that the pursuit of a SRS is futile and instead adopt a more radical method of development such as one of the Agile methods. Worst of all a SRS may be written and then set aside in the performance of the project. Clearly these practices call into question the very nature of the SRS. Recently published papers bear witness to this with a debate about the Agile methods (Goetz; Jepsen; Pinheiro; Tomayko; Berry; Eberlein & Leite; Kohler & Paech 2002). Similar interest is found also in, for example, (Baskerville et al 2003). However the debate exposes a wide range of situations where the use of some kind of SRS remains appropriate for the primary instrument of specification and

Boness K., Harrison R. and Liu K. (2005).
ACKNOWLEDGING THE IMPLICATIONS OF REQUIREMENTS.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 337-342
DOI: 10.5220/0002548303370342

management. Our interpretation is that this is where there is one of the following:-

- a fixed price contract;
- a need for a wide range of stakeholders to be involved;
- a probability of significant additional development in the future;
- where the system's architecture must be decided from very early in the project.

Published studies concerning the tendency for IT development projects to fail to deliver what is wanted in time and budget all report deficiencies in the preparation and use of the SRS as one of the main causes of failure; see (Taylor 2001; Keil et al; POST 2003). It is noteworthy that a SRS is usually presumed to be in use and the criticism reinforces the role of the SRS described at the beginning of this section along with the expected benefits of its use.

The above arguments suggest that there is a significant commitment to the use of the SRS but its use is potentially compromised by a lack of practicality in the face of uncertainty combined with urgency. Hence we suggest that:-

It is desirable in software development projects, where practical, to use a SRS with diligence as the primary instrument of specification, negotiation and management. (Premise 1).

Practicality and diligence are the keys. Without the former the latter is unlikely. Without the latter the value of the SRS is lost.

Section §2 discusses practicality and diligence in more detail. In §3 we propose prerequisites for a practical SRS. §4 discusses how we can measure a SRS with a new requirements profile. We conclude with a brief discussion.

2 PRACTICALITY & DILIGENCE

Practicality is at the heart of motivating the stakeholders to work with a SRS and to take responsibility for it. In some cases this is to plan and design from it and in others it is to verify and validate it. These responsibilities take effort. It is essential that this effort is perceived as worthwhile.

Berry (2002) points out that pain (i.e. tasks seen as chores) can be a practical barrier to the diligence of developers (and other stakeholders). He reminds us that the perceived value of any task matters if the task is to be depended upon. To some extent a disciplined process may suffice but just as in any regulated system personal motivation will remain significant. McPhee and Eberlein (2002) concluded that it is important to stakeholders that requirements are unambiguous and usable. From a project management perspective, Keil, Rai, Mann and Zhang (Keil et al 2003) stress the importance of the clarity of requirements. Clearly important qualities of the SRS that affect motivation include: usability and understandability.

A further part of practicality is the timeliness of the SRS. If it is to take the role of principal instrument of negotiation and planning it must be available and usable early in the project. Urgency and uncertainty make it almost inevitable that when the SRS it is needed early in the project it is highly unlikely that the requirements can have been specified with much rigour. Consequently its content may, to some extent, be ambiguous, inconsistent and structurally incomplete. This makes it very unlikely to be understood consistently by all the stakeholders.

Note: In this paper we use the word *coherence* when referring to the understandability, consistency, and structural-completeness of an SRS. This extends the usage in (Nusibeh & Easterbrook 2000).

Given time and careful requirements elicitation, verification and validation along with good and appropriate drafting the SRS should become more coherent. However any lack of coherence would increase the chances of misunderstanding.

Thus for a SRS to be used as in premise 1 it must be available early in the project and should be qualified in some way so as to reduce the potential of misunderstanding. This is summarised in a second premise:-

The SRS must be produced rapidly with emphasis on its usefulness to all stakeholders and should make clear to any stakeholder reading it the extent of what is known about what is wanted (Premise 2.)

3 PREREQUISITES

Pinheiro (2002) offers three principles of requirements. They should be:-

1. Purposeful (there should be an objective to be fulfilled);
2. Appropriate (requirements should express what

is necessary to achieve the system's objectives;
3. Truthful (requirements should express what is actually required).

From our premise 2 we propose three prerequisites for a practical SRS over and above the principles of Pinheiro:-

1. Rapid production;
2. Using a format that is understandable to all stakeholders;
3. A means for making "what is known about what is wanted" clear to all stakeholders.

The need for **rapid production** comes from the need to have requirements in place early enough for the SRS to be used for estimation, scoping and initial sketching of architecture; possibly also to be bound into a contract. This usually implies an early deadline making it likely to be a "rough sketch" that can be subsequently improved to approach correctness (Zowghi & Gervassi 2003). How much can be elicited in the available time will depend upon the effectiveness and efficiency of the techniques of elicitation adopted. In this work we are not concerned with the details of different techniques of elicitation; we presume that suitable methods will have been chosen in a groundwork phase (Finkelstein 1993) to suit the project.

The need for the format to be **understandable** to all stakeholders is self-evident if the SRS is to be shared for common understanding and if it is to be used for negotiation between stakeholders. The question is what limitations ought to be applied on the format of the SRS in order that it is understandable to the stakeholder community as a whole. In some communities the efficient and effective means of communication may be formal (for example using predicate logic, mathematics and/or other symbolic representation). However it is likely to be the case that key stakeholders are effectively lay people for whom such methods are not properly understood. Possibly the ideal is to have the SRS in multiple versions; using natural language to share and using formal versions to validate. Rolland and Proix in (1992) described an exercise to bridge the problem of using natural language for communication among the stakeholders whilst using a parallel formalised representation and subjecting it to the rigour of formal analysis. They developed a system that accepted an initial SRS written in natural language and parsed it to represent it in a formal equivalent which is then analysed by formal methods. The errors discovered in this process are corrected in the formal model and then a new version of the SRS is synthesised in natural language for all stakeholders to review.

Rolland and Proix chose natural language for sharing, communicating and agreeing the SRS and chose the formal version for specialist and computed validation. This is a significant stance. All systems that have multiple views or versions of the SRS (even if they are derived from a single formal core) have the problem of documentary precedence. The same pertains to legal documents needing agreement. Any discrepancies in translation between the different forms potentially cause misunderstanding. For example if a contract is at stake then the reasonable interpretation of the shared SRS (in natural language) would probably hold precedence over specialist formal versions; (Ambriola & Gervassi 1997).

(Leveson 2000), in a broad view of requirements indicates a continued strong role for natural language in specification. Further in their review of requirements management tools Finkelstein and Emmerich (2000) give natural language a powerful endorsement whilst implying a mixed future:

"We argue that the reason that the use of natural language persists in requirements documentation is that it plays a valuable role and furthermore that it is unlikely to be supplanted. This role goes beyond simply providing a means for stakeholders to validate the specifications, though this is in itself very valuable, but is concerned with the essence of the specification task. Requirements refer to the real-world, for the models that result from analysis to be comprehensible it is essential that the correspondences between the components of the model and the real-world phenomena are explicated. Without this the models are airy abstractions."

Thus notwithstanding the acknowledged power and rigour of other forms for expressing requirements our work concentrates on the natural language forms of a SRS:-

Requirements expressed in natural language remain important to a significant extent in contemporary practice as the contracted and shared expression of need. (Premise 3)

We are testing this premise through a survey of current industrial practice. Early indications are positive. A related market survey in 2002 (Mich et al 2002) reported circa 70% of SRS documentation being substantially in natural language.

Our interest does not rule out situations where translations to and from formal representations are used to add modelling rigour. The point is that the objects of our work are requirements specifications expressed in natural language for the purpose of agreement by all stakeholders. Of course whilst they will usually have been written by humans they may have been machine generated.

We expect that requirements will be presented in eclectic forms dominated by natural language texts. These are likely to involve mixtures of: goals; use narratives (such as stories[i], use cases, and scenarios); assumptions; and specific requirements (both functional and non functional). Hence:-

The objects of our study are software requirements specifications written in natural language and broadly compliant with the IEEE standard (IEEE 1998). They may also have diagrams (or other non natural language inclusions) as sign surrogates for text. (Premise 4)

This premise stresses the dominance of natural language but admits the use of diagrams such as literate modelling (Arlow et al 1999; Finklestein & Emmerich 2000).

Using the idea of *signs* from semiotics, (Pierce 1935), the SRS could be construed as a collection of signs where the signs are chosen (written) by the author to signify ideas to the reader. Sometimes the author may find it necessary to use a means other than natural language, such as a diagram or an equation, to signify a subset of ideas. In this sense the signs are surrogates for missing and potentially more complex natural language text. Thus such signs are regarded here as possible infill to otherwise missing information. In principle the reader could be asked questions about each sign and their answers then be used to complement and possibly complete the natural language body of the SRS.

Semiotics also cautions us that it is an incomplete understanding of communication to consider solely the author's use of signs to signify an idea. It is necessary to consider the potential elusiveness of what is being signified and how the reader responds to the signs (Pierce 1935). The author's choice of sign may be an imperfect signifier of a clear idea on one hand; it may be a good signifier for a vague idea on the other. It is also possible that the reader will interpret the sign differently to some degree from what has been intended by the author. Thus the SRS may well induce misunderstanding and divergent assumptions between different stakeholders by the author's choice of signs. Add to this the likelihood of poor drafting induced by time pressures then the likelihood of misunderstanding is increased.

Clearly the possibility of misunderstanding constitutes a severe jeopardy to projects depending on the SRS (premise 1) and can mask severe project threatening problems. For these reasons importance is placed in requirements engineering on the verification and validation of requirements; see (Penheiro 2002; Keil et al 2003; Nuseibeh & Easterbrook 2000; Avritzer & Weyuker 1998). However validation may be obfuscated by a lack of coherence in the SRS. Whereas we can confidently attempt to validate coherent requirements we cannot reasonably do the same with incoherent ones since they are unlikely to be understood consistently by all the stakeholders. This brings the argument back to the need to make clear what is known about what is wanted (prerequisite 3).

This third prerequisite extends Pinheiro's truthfulness principle and is at the heart of our work. It represents a change in thinking about the SRS and is discussed further in the following section defining a requirements profile.

4 REQUIREMENTS PROFILE

From the above reasoning we consider that there is a need for a quantitative means of revealing the coherence of an SRS as it evolves from a very early draft into a mature representation. We refer to this as a *requirements profile*.

The IEEE standard (IEEE 1998) tolerates incompleteness in the SRS by allowing TBD (to be determined) to be inserted with accompanying explanation of how the TBD can be fulfilled. This is the start of defining what is known about requirements (prerequisite 3) rather than simply defining the requirements; if specific requirements (functional or non-functional) are known they are included and in this way the traditional concept of the SRS is preserved. Our work aims to extend this idea by the development of a requirements profile that can be used both to annotate the SRS and provide indicative measures.

There is a range of grammatical and syntactic methods and tools that could contribute to a requirements profile; (Berry et al 2003; Ambriola & Gervassi 1997; Wilson et al 1997; Lami et al 2000) and others. The emphases of our work are assumptions and omissions detectable by the goal satisficing methods of (Mylopolous et al 1998;

[1] i.e. Stories as used in the Extreme development

Letier & Lamsweerde 2002; Lamsweerde 2001) and the use of semantic normal form derived from the semiotic methods in MEASUR (Stamper et al 1988; Stamper 1993; Liu 2000).

The use of our requirements profile goes beyond our premise 1. For example any exercise that involves reading and interpreting a SRS may benefit. It could also provide a quality measure to be used before and after re-drafting, or re-generating, a SRS. General examples include providing:-

1. A means, as a quality control metric (or set of metrics), to guide the rapid creation of high quality requirements.
2. A way of measuring improvement in the process of preparing specifications.
3. A means for accelerating the verification and validation of requirements by inspections; both formal (e.g. Fagan) and informal (e.g. readings by stakeholders).
4. A quality control to prepare requirements that are written in natural language for translation to another format (including formal representation).

5 CONCLUSIONS & VALIDATION

In presenting prerequisites for the practical use of the SRS we have been drawn to the need for emphasising *what is known about the* requirements. We are addressing this by developing a requirements profile that concerns itself with the coherence of the SRS and its capacity to avert misunderstanding.

We are using problem exemplars to help in developing the profile. Initially this is in the context of *inspections* since their use is well documented (e.g. (Porter et al 1995)). Our main validation will use case studies from industry and academia (Feather et al 1997; Kitchenham et al 1995; Fenton & Pfleeger 1996).

REFERENCES

Ambriola V, Gervassi V, 1997. “Processing Natural Language Requirements”, Proceedings of the 1997 International Conference on Automated Software Engineering (formerly: KBSE) , Page: 36.

Arlow J, Emmerich W and Quinn. J, 1999. “Literate Modelling - Capturing Business Semantics with the UML”. In: J. Bezivin and P.-A. Muller (eds) The Unified Modeling Language: <<UML '98>>: Beyond the Notation, Mulhouse, France, Lecture Notes in Computer Science. 1618, pp. 189-199. Springer Verlag.

Avritzer A, Weyuker E.J, 1998. “Investigating Metrics for Architectural Assessment”, IEEE Metrics, pp 4-10.

Baskerville R, Ramesh B, Levine L, 2003. Pres-Heje J, Slaughter S, “Is Internet Speed Software Development Different” , IEEE Software, November, pp70-77. Vol 20, Number 6.

Berry D, 2002. “The Inevitable Pain of Software Development Including Extreme Programming, Caused by Requirements Volatility”, International Workshop on Time-Constrained Requirements Engineering , Essen, Germany, September 9.

Berry D.M, Kamsties E, Krieger M, 2003. “From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity - A Handbook”, Ver. 1.0, http://se.uwaterloo.ca/~dberry/#Handbook,.

Eberlein A and Sampaio do Prado Leite, 2002. “Agile Requirements Definition: A View from Requirements Engineering”, International Workshop on Time-Constrained Requirements Engineering, Essen, Germany, September 9.

Feather M.S, Fickas S, Finkelstein A, Lamsweerde A van, 1997. “Requirements & Specification Exemplars”, Automated Software Engineering, October, vol. 4, no. 4, pp. 419-438(20), Publisher: Kluwer Academic Publishers

Fenton N, Pfleeger S.L, 1996 "Software Metrics: A Rigorous & Practical Approach", 2nd edition, International Thomson Computer Press

Finkelstein A, Emmerich W, 2000. “The Future of Requirements Management Tools” In G. Quirchmayr, R. Wagner and M. Wimmer (eds), Information Systems in Public Administration and Law. Oesterreichische Computer. Gesellschaft.

Finkelstein A, 1993. “Requirements Engineering: an overview”, 2nd Asis-Pacific Software Engineering Conference Tokyo, Japan.

Gervassi V, Nuseibeh B, 2000. “Lightweight Validation of Natural Language Requirements: a case study”, Lightweight validation of natural language requirements: a case study. Proc. of the 4th International Conference on Requirements Engineering, pages 140-148, June.

Goetz R, 2002. “How Agile Process Can Help in Time-Constrained Requirements Engineering”, International Workshop on Time-Constrained Requirements Engineering (TCRE'02), Essen, Germany, Sept. 9.

IEEE, 1998. “Recommended Practice for Software Requirements Specifications”, Std 830-1998, IEEE Inc 354 East 47th St. New York NY 10017, USA.

Jepsen O, 2002. “Time Constrained Requirement Engineering – the Cooperative Way”, International Workshop on Time-Constrained Requirements Engineering, Essen, Germany, September 9.

Keil M, Rai A, Mann J.E.C and Zhang G.P, 2003. “Why Software Projects Escalate: The Importance of Project management Constructs", , IEEE Transactions on Engineering Management, Vol 50, No 3, August.

Kitchenham B, Pickard L, Pfleeger S, 1995 "Case Studies for Method and Tool Evaluation", IEEE Software, Vol. 12, No. 4, pp. 52-62

Kohler K, Paech B, 2002. "Requirement Documents that Win the Race: Not Overweight or Emaciated but Powerful and in Shape", International Workshop on Time-Constrained Requirements Engineering, Essen, Germany, September 9.

Lami G, Gnesi S, Fabbrini F, Fusani M, Trentanni G, 2000. "An Automatic Tool for the Analysis of Natural Language Requirements", C.N.R. - Information Science and Technology Inst. " A. Faedo", Pisa, Italy

Lamsweerde A van, 2001. "Goal-Oriented Requirements Engineering: A Guided Tour (2001)", Axel van Lamsweerde, 2001. Proceedings 5th IEEE International Symposium on Requirements Engineering August 27 – 31.

Letier E, Lamsweerde A van, 2002. "Agent-Based Tactics for Goal Oriented Requirements Elaboration" Proceedings ICSE'2002, 24th International Conference on Software Engineering, ACM Press.

Leveson N.G., 2000. "Intent Specifications: An Approach to Building Human-Centered Specifications", IEEE Transactions on Software Engineering, Vol 26, No. 1.

Liu K, 2000. "Semiotics in Information Systems Engineering", Cambridge University Press, ISBN 0-521-59335-2.

McPhee C. and Eberlein A. 2002. "Requirements Engineering for Time-to-Market Projects", Proceedings of the 9th Annual IEEE International Conference on the Engineering of Computer Based Systems, Lund, Sweden.

Mich, Luisa and Franch, Mariangela and Novi Inverardi, Pierluigi 2002. "Market research for requirements analysis using linguistic tools", Technical Report 66, Informatica e Studi Aziendali, University of Trento.

Myolopolous J, Chung L, Yu E, 1999. "From Object-Oriented to Goal-Oriented Requirements Analysis." Communications of the ACM Vol 42 No 1 pp31-37

Nuseibeh B, Easterbrook S, 2000 "Requirements Engineering: A Roadmap", Proceedings of International Conference on Software Engineering , 4-11 June, Limerick, Ireland.

Pierce C.S, 1935. "Collected Papers 1931-35", (6 volumes), Hartshorne C & Wiess P (eds), Cambridge, Mass. Harvard U.P.

Pinheiro F.A.C, 2002. "Requirements Honesty", International Workshop on Time-Constrained Requirements Engineering, Essen, Germany, September 9, 2002.

Porter A, Votta L.G, Basili V. R, 1995. "Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment", IEEE Transactions on Software Engineering, Volume 21 , Issue 6 Pages: 563 – 575.

POST 2003. "Government IT projects", Parliamentary Office of Science and Technology, Report 200, July 2003, http://www.parliament.uk/post

Rolland C, Proix C, 1992 "A Natural Language Approach For Requirements Engineering" Proceedings of the Fourth International Conference CAiSE'92 on Advanced Information Systems Engineering.

Stamper R. K, "Signs, 1993. Norms and Information Systems", Ronal Stamper, 1993, ICL/University of Newcastle Seminar on "Information", September 6-10.

Stamper R.K, Althans K and Backhouse J, 1998. "MEASUR: Method For Eliciting, Analysing and Specifying User Requirements" Computerized Assistance During the Information Systems Life Cycle: 67-115

Taylor A, 2001. "IT projects sink or swim", BCS Review, http://www.bcs.org.uk/review .

Tomayko J.E, 2002. "Engineering of Unstable Requirements Using Agile Methods", International Workshop on Time-Constrained Requirements Engineering, Essen, Germany, September 9, 2002.

Wilson W.M., Rosenberg L.H. Hyatt L.E. 1997. "Automated Analysis of Requirement Specifications" Nineteenth International Conference on Software Engineering, Boston, MA.

Zowghi D, Gervassi V, 2003. "On the interplay between consistency, completeness, and correctness in requirements evolution.", Zowghi D, Gervasi V, Information and Software Technology, 45 (14): 993-1009, Elsevier B.V.