

# Systems' Requirements: Once Captured, are Slaughtered

Ban Al-Ani,

*Dept. of Software Engineering, Faculty of IT, University of Technology Sydney*

[alani@it.uts.edu.au](mailto:alani@it.uts.edu.au)

## Abstract

Despite the diverse assortment of artefacts produced to support systems development processes practitioners have survived attempts to overcome the problems they face when developing requirements. These problems arise at almost every stage of systems development and are investigated with varying degrees with success by different research projects.

This paper unceremoniously focuses on some of the problematic activities at the front end of systems development, namely: requirements capture and analysis. A multi-layered evaluation process that is composed of several sets (layers) of criteria is outlined. A new method/tool can then be put through these multiple layers. If successful it would be 'ticked' as being a viable option and adopted by practitioners. The argument made is that a move towards regulating the evaluation of methods/tools usefulness can be one way out of the ivory tower.

**Keywords:** system requirements, research methods, research products, evaluation framework.

## Introduction

It is widely accepted that establishing system requirements is considered the most important stage of systems development. No other stage affords the greatest potential for significant improvement in time, cost and the quality of the system (Robinson and Pawlowski, 1998).

Several research artefact (e.g. tools, methods) have been put forward in order to overcome the confusion that arises from inadequately documented requirements. While these proposals have succeeded when tested by its creators and its sponsors few have had widespread acceptance by practitioners (e.g. Edwards et al, 1995, Smith, 1992). This has been attributed to one or more reasons. For example: the need for training that practitioners are unwilling to invest in, the method/tool is unable to process non-textual requirements, or the performance degenerates when executed on different genres of text (Al-Ani, 2001). Practitioners often consider these artefacts infeasible for these reasons and others (Mead, 2000). Whatever the reason the end result is the same, a great deal of time, effort, and money is spent - yet the artefacts proposed are of little use.

This papers aims to highlight the importance of establishing research usefulness. It focuses on the front-end activities within the requirements engineering process and presents an unceremonious discussion of some of these activities, namely: capture and analysis. The importance of these activities lies in the conviction that faults are often introduced in these

stages of requirements development- a conviction supported by the findings of other studies (e.g. Ashry and Taylor, 2000).

The following section outlines some of the challenges of capturing requirements and the second focuses on requirements analysis. The paper also presents an outline of a possible evaluation framework to assist researchers develop more effective artefacts.

## **The Challenges of Requirements in the Raw**

Research conducted into the field of requirements revealed that the problem often lies in the initial stages of the requirements development process (Al-Ani, 2001). Jeffrey and Putman (1994) state that the difficulty of understanding the description of a proposed system is a major impediment to developing effective systems. Jeffrey and Putman (1994) and Hughes et al (1996) state the root of the problem is that stakeholders have a picture of the desired system, but probably do not have the concepts, language or skills (technical or otherwise) to communicate that picture. The analyst on the other hand, has the concepts, language and technical skills, but does not have a complete picture of the desired system or the organization, either as it exists or as the clients want it to exist. The following sections take a quick look at requirements capture and analysis.

### **Requirements Elicitation (The Capture: How?)**

Generally requirements sources vary during the elicitation process. They can be derived gradually from a succession of informal notes, recorded interviews, questionnaires, and/or through other information gathering techniques and documentation media that collectively attempt to describe the system (Al-Ani, 2001). Information can be acquired through one or more elicitation techniques (Kotonya and Sommerville, 1998).

Studies of systems requirements cite inadequacy of the available representation languages in addition to inadequacy of communication as one of the main challenges of developing requirements (Patel, 1999, Leveson, 1998, Jackson, 1997). The challenges that practitioners face are not limited to these; they can be attributed to one factor or a combination of additional factors (e.g. Alderson, 1999, Leveson, 1998, Potts, 1995).

Fantechi et al (1994) state that although the process of defining requirements is to some extent systematic, the identification of requirements is usually informal. It is still ad hoc to some extent despite the work conducted in this area (Potts, 1999).

The argument that is made in this paper is that it would still be possible to save the day if the elicited requirements were better understood before attempting to develop the specification document.

### **Requirements Analysis (The Slaughter: What Happens?)**

At the end of the elicitation process developers are often faced with the challenging task of wading through a bulky document to reach an initial understanding of the proposed system's general features.

A deeper understanding of system features must be achieved before developers can go any further. There exists a need for an approach to analysing raw requirements and developing an initial understanding. While it is widely accepted that developers rely on this initial understanding to negotiate the tender there is no evidence that there they are not relying primarily on an extemporized approach to reach this understanding.

Poorly understood requirements can lead to incorrect interpretation of these statements. Developers can make assumption during these the early stages of requirements capture. These assumptions often become imbedded in the requirements document. While this is unavoidable to some extent it becomes dangerous when they fail to substantiate assumptions. These assumptions are often not detected until the testing phase incurring higher costs to correct.

Practitioners often resist systems that support requirements analysis despite their ability to reduce cost through automated development e.g. Palmer and Liang, 1992, Samson, 1991 (Kim et al, 1993). While this reality is widely known little has been done to overcome or acknowledge it. As such, practitioners still rely on their own abilities that can be flawed.

While it is seems quite easy to identify problems, flaws and causes in proposed methods and tools; identifying a process in which these approaches can be improved remains illusive. The following section outlines one way to improve effectiveness.

## **How can proposed approaches be made more effective?**

The purpose of this paper is not to inflict pessimism with regards to work done in the early stages of requirements development but rather to explore a means to provide a more optimistic future for the *products* of research by making them more useful or 'marketable' to practitioners.

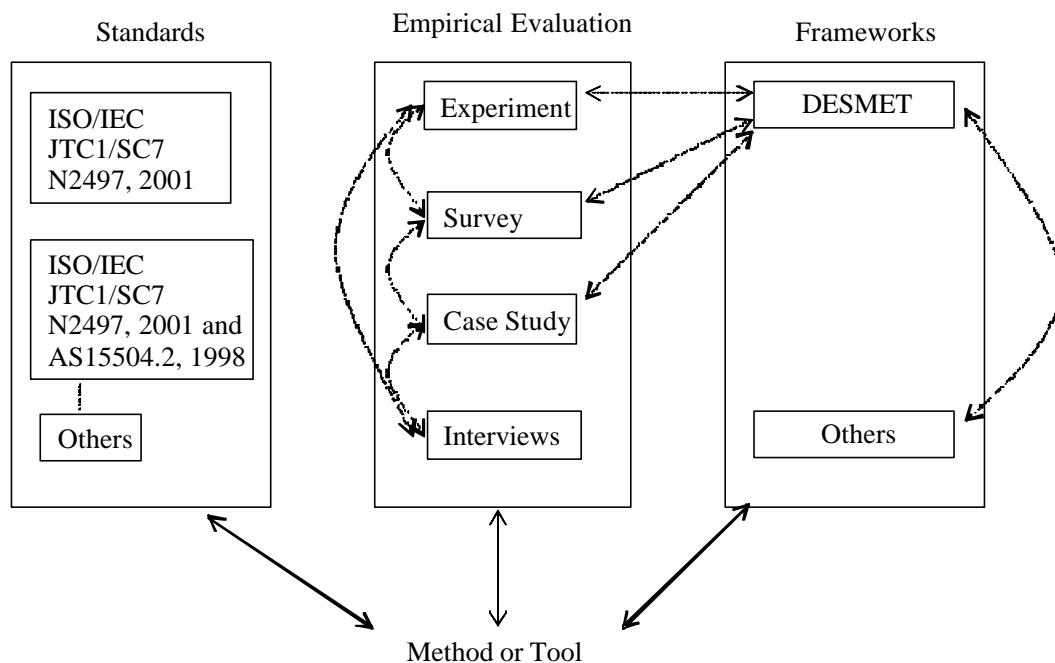
Practitioners are the clients that researchers should target. Consequently, research should be based on their 'clients' needs and not personal beliefs isolated from client participation.

Client participation is essential when evaluating the early drafts of approaches derived from research. This can assist researchers evaluate both method/tool outputs and process.

Additional layers of evaluation will ensure that the emerging approach will be feasible. These layers can include the following:

- Evaluation through one or more traditional empirical methods like experiments, case studies amongst others (Yin, 1994. The evaluation through these methods should strictly follow the guidelines provided in relevant literature (Yin, 1994, Kitchenham et al, 1995, Pfleeger, 1995).
- Evaluation through standards developed to measure the effectiveness of derived approaches (e.g. ISO/IEC JTC1/SC7 N2497, 2001 and AS15504.2, 1998 amongst others)
- Evaluation using a purpose built evaluation framework like DESMET (Kitchenham et al, 1995). This framework is designed to evaluate software methods and tools and separates evaluation exercise into two main types, namely: evaluation aimed at establishing measurable effects of using a method or tool and its appropriateness. DESMET provides a set of technical selection criteria that can help the researcher decide on evaluation approach based on the context. Other frameworks can be derived from industry practices and included within this layer as an alternative to DESMET or something that can be combined with it to achieve optimum results.

An overview of the proposed evaluation process is presented in figure 1. An artefact that has been subjected to this process would be 'ticked' as having the level of quality necessary to make it viable for industry use.



*Figure 1: An overview of achievable evaluations. A viable multi-layered evaluation process is demonstrated. The dotted lines represent the evaluation path that the proposed method/tool would take when being evaluated. At the end of this process a proposed method would achieve a 'tick' of approval.*

## Concluding Remarks

Despite the great leaps and jumps forward in technology humans have survived attempts to overcome some of the imperfections of the thinking process.

An outline of the early stages of requirements gathering and understanding was presented in this paper. In addition to a run through some of the challenges practitioners still face.

While this paper does not propose evaluation criteria it does highlight viable options that are currently available. The multi-layered evaluation process proposed in the previous section includes different sets of criteria that can be used to evaluate different types of methods and tools. The paper also recommends involving the industry in developing at least one layer of evaluation. Collectively these measures could help researchers come a step closer to getting out of their ivory towers.

## References

- Al-Ani, B., (2001). "A Framework for the Detection and Evaluation of Partially Complete Requirements Documents", Submitted Doctor of Philosophy Dissertation in Computer Systems Engineering, Faculty of Engineering University of Technology Sydney.
- Alderson, A., (1999). "False Requirements Express Real Needs", Requirements Engineering Journal, Springer-Verlag London Limited, 4:60-61.

- AS15504.2, (1998). Information Technology- Software Process Assessment: Part 2: A Reference model for Processes and Process Capability, Interim Australian Standard™, Standards Australia.
- Ashry, N., Taylor, W., (2000). Requirements Analysis as Innovation Diffusion: A Proposed Requirements Analysis Strategy for the Description of an Integrated Hospital Information Support System, Proceedings of the 33<sup>rd</sup> Hawaii International Conference on Systems Sciences, pp 1-9.
- Edwards, M. L., Flanzer, M., Terry, M., Landa, J., (1995). "RECAP: A Requirements Elicitation, Capture, and Analysis Process Tool for Large Complex Systems", *Proceedings of the 1st International Conference on Engineering of Complex Computer Systems, Institute of Electrical and Electronics Engineers, Aug.*, pp. 278-281.
- Hughes, J., O'Brien, J., Rodden, T., Rouncefield, M., Sommerville, I., (1995). "Presenting Ethnography in the Requirements Process", Proc. 1<sup>st</sup> Int. Conf. On Requirements Engineering, York, UK, March, pp. 27-34.
- ISO/IEC JTC1/SC7 N2497, (2001). Resolution 600, , Quality Engineering and Research, Software Engineering Secretariate: Canada (SCC).
- Jackson, M., (1997). "The Meaning of Requirements", *Annals of Software Engineering*, 3, J. C. Baltzer AG, Science Publishers, pp. 5-21.
- Jeffrey, H., Putman, A., (1994). "Relationship Definition and Management: Tools for Requirements Analysis", *The Journal of Systems and Software*, March 01, V. 24, N. 3, pp. 277-294
- Kim, H., Ko, Y., Park, S., Seo, J., (1999). Informal Requirements Analysis Supporting System for Human Engineer, ????, pp. 1013-1018.
- Kitchenham, B., Pickard, L., Pfleeger, S., (1995). "Case Studies for Method and Tool Evaluation", *IEEE Software*, Vol. 12, Issue 4, July, pp. 52-62.
- Kotonya, G., Sommerville, I., (1998). *Requirements Engineering: Process and Techniques*. John Wiley and Sons, pp. 30-38.
- Leveson, N., (1998). "Intent Specifications: An Approach to Building Human-Centred Specifications", *Proceedings of the 3<sup>rd</sup> International Conference on Requirements Engineering*, April 6-10, Colorado Springs, Colorado, IEEE CS Press, Los Alamitos, California, pp.204-213.
- Mead, N. (2000). Why is it so difficult to introduce Requirements Engineering Research Results into Mainstream Requirements Engineering Practice?, ????
- Palmer, J., Liang, Y., (1992). Indexing and clustering of Software Requirements Specifications, *Informal and Decision Technologies*, Vol. 18, pp. 283 – 299.
- Patel, N., (1999). "The Spiral of Change Model for Coping with changing and Ongoing Requirements", *Requirements Engineering Journal*, Springer-Verlag London Limited, 4:77-84.
- Pfleeger, S., (1995). "Experimental Design and Analysis in Software Engineering", *Annals of Software Engineering*, Jan., Vol. 1, Number 1, pp. 219-253.

- Potts, C., (1995). "Invented Requirements and Imagined Customers: Requirements Engineering for Off-the-Shelf Software", Proceedings of the 1<sup>st</sup> International Conference on Requirements Engineering, April, Colorado Springs, Colorado, IEEE CS Press, Los Alamitos, California, pp. 128-130.
- Robinson, W., Pawlowski, S., (1998). "Surfacing Root Requirements Interaction from Inquiry Cycle Requirements Documents", Proceedings of the 3<sup>rd</sup> International Conference on Requirements Engineering, April 6-10, Colorado Springs, Colorado, IEEE CS Press, Los Alamitos, California, pp.82-89.
- Samson, D., (1991). A Knowledge Based Requirements System, Productivity: Process, Prospects and Payoffs, 27<sup>th</sup> Annual Technical Symposium, D. C. Chapter of the ACM and NBS, Washington.
- Smith, T. J., (1992). "READS: A Requirements Engineering Tool", IEEE CS Press, Los Alamitos, California, Jan., pp. 94-281.
- Yin, R., (1994). Case Study Research: Design and Methods, 2<sup>nd</sup> edition, Sage Publications.