Why Is It So Difficult to Introduce RE Research Results into Mainstream RE Practice?

Hermann Kaindl [1] and John Mylopoulos [2]

[1] Siemens AG Österreich, Geusaugasse 17
A-1030 Vienna, Austria
hermann.kaindl@siemens.at
[2] Department of Computer Science, University of Toronto, 6 King's College Road
Toronto, Ontario, Canada M5S 1A4
jm@cs.toronto.edu

1 Motivation

For quite a long time research results in requirements engineering (RE) were developed without much interaction with, or impact on, practice. In 1993 Hsia *et al.* [2] made an honest evaluation of the requirements engineering practice. Since then, some improvements have been achieved, e.g., by some applications of *usage scenarios*. Nevertheless, mainstream practice is still to use one's favourite word processor to write down mostly unstructured text in natural language, even though there exists some influential work and good experience with approaches that have used RE results in real projects (see, e.g., [1, 3]). In fact, mainstream practice deals only rudimentarily with requirements, and what is done might not even deserve to be called "engineering".

2 Goal Statement

The goal of this panel is to really understand the issue of *why* it is so difficult to introduce research results from requirements engineering into mainstream practice. The result should be a (research) agenda that helps us to bridge the gap between theory and practice and, finally, to reach the people in the trenches so that they improve the way they deal with requirements.

3 Questions Addressed

In order to achieve the stated goal, the following questions (among others) will be addressed in the panel:

B. Wangler, L. Bergman (Eds.): CAiSE 2000, LNCS 1789, pp. 7-12, 2000

- What are the obstacles to applying RE research results in practice?
- Does schedule pressure in real-world industrial projects have an influence?
- Is it the fault of customers and users that they do not know (much) about RE?
- Are the people working on requirements from the software perspective aware of RE research results?
- Does installing and enforcing rigorous software-development processes help?
- How can we (as the requirements engineering community) reach people in the trenches?
- Does RE research address the real issues arising in practice?
- Are more practical methods needed?
- Is better tool support needed?
- What are the social problems involved in changing practice?
- Are practitioners satisfied with their current requirements process?
- Are practitioners searching for better ways to do requirements?
- How do the practitioners view academic research and its results?

4 Position Statements of Panel Members

Sjaak Brinkkemper (sbrinkkemper@baan.nl) Baan Company R&D

The title of this panel seems to assume that there exists an abundance of research results of the RE research community being worthwhile to be considered for implementation in the RE practice. Speaking in the position of being responsible for the world-wide implementation of requirements engineering in one of the larger packaged software companies, I have to admit that there are few results that really make a difference and should be considered. Most RE research focuses on Yet Another Specification Technique, or contains the traces of simple desktop research with toy examples and invented problems.

This leads me to rephrase of the title: “Why is it so difficult to have critical issues of the RE practice investigated in mainstream RE research?”. Any requirements listing of an arbitrary IT project will contain a complex myriad of statements expressing wishes of users/customers on a hardware/software artefact to be engineered by a team. Hence structuring, understandability, and formulation to name a few, are far more important to be improved. Engineering requirements is relatively simple, their management is considerably more difficult. As high level critical issues that deserve research projects right away I mention:

- Composition and communication of large volumes of requirements,
- Base-lining and scoping for release based software development,
- Empirical investigation of real-life projects,
- Tracking and tracing from requirements to designs and code.

Janis A. Bubenko Jr (janis@dsv.su.se) Royal Institute of Technology and Stockholm University

Problems encountered in transferring academic research results in Requirements Engineering (RE) to practical applications and "colleagues in the trenches" are similar to those in transferring research results from most Informatics disciplines, such as Information Systems Development Methodologies, CASE Technologies, and Software Engineering. All these areas experience difficulties in adopting research from academia. The case for RE is perhaps even a bit harder as RE is performed not only by systems analysts and engineers, but also, increasingly, by different "stakeholders" without a professional background in Informatics.

There are, in my opinion, two basic issues:

1. We are not in agreement regarding what requirements, or approaches to requirements specifications really stand for. Some of us are heavily engaged in development of more or less formal models and languages for representing requirements. We believe that there is only one way of specifying requirements: they should be formal, consistent, and complete. Others are more engaged in developing conceptual frameworks for functional as well as non-functional requirements. Still others are more concerned with the "process" of acquiring requirements and how to guide it. In Scandinavia some researchers are interested in participatory aspects of requirements development – how to get most out of the stakeholders in the process of acquiring requirements. Business oriented requirements engineering, on the other hand, focuses on achieving a best possible fit between business goals and processes and the supporting information system.
2. We cannot tell the people in the trenches what the benefit is of applying research results (whatever they are) to practical cases. We know from literature that 50% or more of all errors are made in the requirements stage, and that it costs many times more to correct these errors in later stages of systems development. We also know that many software projects never lead to systems that are applied as originally designed. Some are even never used (but paid for). We try to tell our practitioner colleagues these figures, but it does not seem to give any effect. If we want so "sell" something to practitioners then we must be able to tell them what they will *gain* by making heavy investments in skill development and in spending more resources in the requirements phase of systems development. And we must be able to illustrate this in monetary terms.

Issue one illustrates my opinion that there is no "standard model" for requirements specifications and requirements work. However, to some extent, all aspects, formal/technical, economical, conceptual, organisational, and social must always be considered. But the degree to which specifications should be developed may vary considerably depending on a number of situational factors. We can, for instance, imagine two extreme cases, one where there is no software development support in the company (all software development is out-sourced) and one where there is a "receptive and competent" software unit next door in the corridor. In case one, if the user company wants bids for tender from alternative suppliers then this calls for detailed

and complete requirements specifications. If, on the other hand, the company has a good in-house software developer, then a realistic alternative is to have an iterative process and incrementally develop requirements and a working system. Perhaps we can, in this case, even do without formal models of requirements. Clearly, there are many additional alternative combinations of situational factors such as stakeholder maturity and skills, type of system to be built, maximum cost permitted, maximum time given, etc. Each combination will give more or less different guidelines how to develop requirements.

Issue two really says that we should do more research on the "economics of requirements specification". To my knowledge no directly useful research is reported on this issue, but I may be wrong. In conclusion, our lack of concrete knowledge of what organisations can gain from applying skill and time consuming requirements approaches is one of the major obstacles of getting research results adopted by practitioners. The only way to gain more knowledge about the economics of RE is to go out in the trenches and work together with the professionals.

Barbara Farbey[1] (b.farbey@cs.ucl.ac.uk) University College London

The position adopted is that a) RE research results will only be introduced into practice if they are shown to be aligned with, and sympathetic to, the goals of the business and b) the introduction of new thinking should be seen as an organisational innovation and managed as an exercise in change management.

The problem

Organisations face a highly competitive environment with intense pressure on costs, time to market and flexibility in the face of rapidly changing requirements. The range of development choices has opened up a brave new world of kernels, components and COTS as well as outsourcing. Contemporary sourcing policy demands new skills – commercial and process management, including requirements management – but RE and requirements management are not usually institutionalised.

Software providers are under the same pressures as their customers. Incorporating new methods into a contract means justifying the extra cost and time to their customers. And to their hard-pressed staff.

Universities could form a spearhead. But they do not have the resources to tackle large systems, nor do they necessarily have the domain expertise which companies see as essential. So they lack credibility.

A way forward

… is to treat the introduction of new results as a problem in innovation and strategic change and manage the change.

To do that we need to marshal the strategic as well as the operational arguments. And we need to:

[1] This statement is based on experience as the industrial liaison co-ordinator of RENOIR, an EU Research Network of Excellence, Project no. 20.800.

- get senior management commitment [4],
- identify and justify costs and risks, as well as promote benefits,
- design effective reward structures for good practice,
- advertise success [4],
- leverage and optimise skills across the supply network,
- make better use of universities' core competencies – teaching and the critical imagination,
- use universities for projects that are high-risk, high-gain problems, two to five years away.

Ivar Jacobsson (ivar@rational.com) Rational Software Corp.

Requirements engineering is really not a discipline in its own but part of something larger - that larger being software engineering or systems engineering. Finding the right requirements is just one set of activities in the whole chain of activities that need to work seamlessly together: initiated by business design, and followed by design, implementation, test, etc. And all these activities need to be integrated in an iterative development process that focuses at first creating the architecture in parallel with finding most of the requirements. This is why use cases have been so successful and so universally adopted. Use cases model the business to be designed, they capture the requirements on the software (for e-business that software is the business), they drive the design of architecture and components, and they drive the testing of the developed system. They bind together the whole development work in each iteration and in the whole development project.

5 Panel Format

In order to leave sufficient time for discussions, each panel member is limited to a short position statement presentation. The written statements above already show a diversity of views, so that the panel format should enable spontaneous, interactive discussion, involving both the panel members and the audience.

Acknowledgment

We would like to thank Pei Hsia for email discussions about the topic of the proposed panel.

References

1. S. Greenspan, J. Mylopoulos, and A. Borgida. Capturing more world knowledge in the requirements specification. In *Proceedings of the Sixth International Conference on Soft-*

ware Engineering (ICSE'82), 1982, also published in Freeman and Wasserman (eds.) *Software Design Techniques*, IEEE Computer Society Press, 1983.
2. P. Hsia, A. Davis, and D. Kung. Status report: Requirements engineering. *IEEE Software*, 10(6):75–79, 1993.
3. H. Kaindl. A practical approach to combining requirements definition and object-oriented analysis. *Annals of Software Engineering*, 3:319–343, 1997.
4. Pugh, D., (1993), Understanding and Managing Organisational Change, in Maby, C. and Mayon-White, W. (eds.), *Managing Change* (2nd Edition), Paul Chapman, London.

Information about the Moderators

Hermann Kaindl is a senior consultant at Siemens AG Österreich. His research interests include software engineering, with a focus on requirements engineering; human-computer interaction as it relates to scenario-based design and hypertext; and artificial intelligence, including heuristic search and knowledge-based systems.

Dr. Kaindl received the Dipl.-Ing. in computer science in 1979 and his doctoral degree in technical science in 1981, both from the Technical University of Vienna, where he has given lecture courses since 1984 and served as an adjunct professor since 1989. He is a senior member of the IEEE, a member of AAAI, the ACM, and the International Computer Chess Association, and is on the executive board of the Austrian Society for Artificial Intelligence.

John Mylopoulos is professor of Computer Science at the University of Toronto. His research interests include conceptual modelling, requirements engineering and knowledge management.

Prof. Mylopoulos has published more than 160 refereed journal and conference proceedings papers and five edited books. He is the recipient of the first ever Outstanding Services Award given out by the Canadian AI Society (CSCSI), a co-recipient of the most influential paper award of the 1994 International Conference on Software Engineering, a fellow of the American Association for AI (AAAI) and is currently serving as elected president of the VLDB Endowment Board. The most influential paper award was earned for the paper of reference [1]. The award is given out by the ICSE PC to the paper judged ten ICSE conferences later "... to be the most outstanding and influential on subsequent work ...".