Medical Infobahn for Europe
A. Hasman et al. (Eds.)
IOS Press, 2000

User Requirements in a System Development & Evaluation Context

Jytte BRENDER[1], Peter McNAIR[2]
1) Aalborg University, Dept. of Dev. & Planning and Virtual-Centre for Health Informatics, Fredrik Bajers Vej 7D, DK-9220 Aalborg E, Denmark; jytte.brender@v-chi.dk.
2) Hvidovre Hospital, Dept. of Clinical Biochemistry, Kettegaard Allé 30, DK-2650 Hvidovre, Denmark; peter.mcnair@dadlnet.dk.

Abstract: The case study presented investigates how the interaction between users and developers changed when introducing a three level User Requirements Specification with a coarse granularity of the lowest (operational) requirements.

1. Introduction

Our starting point is a previous study, which analysed a Laboratory Information System (LIS) in daily operation versus the development project and the organisational and contractual relations that led to it, [1]. The system was tailor-made based on a formal system development approach with extensive user-participation and involvement of consultants on the user site. The question explored was "Did the users get what they wanted, - and why not?" Despite the fact that the contract and the warranty were fulfilled, it is implicitly clear from the question investigated that something did go wrong. Essentially, a lot of the problems identified goes around the approach and role of the User Requirements Specification /Document (URD).

The present study is a long-term review and follow-up on the conclusions and recommendations from the case, based on a sequence of cases.

2. The User Requirements Specification

The foundation for the development of the LIS was visionary and evolved from strategic aims, to procedural ideas and to very detailed operational requirements [1], of which only the latter level of requirements was included in the URD and the subsequent contract. Essentially, the problems related with the system in operation could be traced back to the following main causes: a) the user representatives couldn't bridge fully from formal specifications to real functionality and vice versa; b) the developers had not fully understood the domain; c) the vendor patched up identified problems in stead of correcting them; d) the user representatives couldn't fully envisage the future system; e) the need for maintenance and further development was viciously underestimated; and f) the management was not fully aware of their responsibility as managers and couldn't see through the consequences of their decision-making. Further, the data collection of the study demonstrated that the users were unable to describe in detail how they accomplish their work tasks, and that the managers had little insight into details of the work procedures. The recommendation following the analysis was to extend the URD (and consequently the contract) with a description of strategic aims and ideas and subsequently extend the warranty to include a functionality assessment rather than only a technical verification of functions, see also [2].

Within the ISAR (A 2052) AIM Project, Beuscart-Zéphir and co-workers also experienced a divergence between the expressed and observed work processes [3]. The case was concerned with a dominantly cognitive task, and this problem of explicating knowledge is known from a number of domains, see e.g. [4]. The (extreme) logical consequence is the paradox that the assumption for application of traditional system development approaches is not fulfilled [4]. Traditional system development approaches are based on a specification of the future IT-based system with a detailed set of functional requirements that are used for a call for tender and subsequently used as the superior frame of reference for a contractual relation. This assumes:

1) that the users are capable of specifying their requirements – they are not, see e.g. [5],
2) that the users are capable of explicating how they (really) accomplish their work – they are not, and there is even a natural element of variation or even irregularities (see e.g. [3,6,7]) within the work processes,
3) that the users are capable of using formal specification techniques with confidence – they are not, see e.g. [8-12], see also the extensive discussions on the topic in [2,4].

This points in the same direction as the LIS case, but both cases further indicates that a model of the domain (an Enterprise model) is needed within the URD, as suggested by [3]. Moreover, both points at an incremental and iterative approach for the system development.

The Babel (TE 2002) Telematics Engineering Project is dealing with the implication of culture on the implementation and application of IT-based systems and the means to handle this implication, see (http://babel.upv.es/users/~babel/babel.htm), i.e. an innovative issue in systems development. For innovative projects like the Babel Project, for which a pre-condition is that no one knows the solution at the project start, it seems obvious that even extensive backtracking may become necessary if and when lessons learned from the assessment recommend such a decision, i.e. advocating an evolutionary prototyping. In such a case it is obvious that a traditional User Requirements Specification with specifications of each detail of the functionality of the model solution – irrespective of modelling or description approach - is obsolete.

Based on the discussions in [1,2,4], the recommendation is that the contents of a URD should contain at least the following elements, but not necessarily a Functional Specification of the functionality required to constitute the (future) IT-based solution: Objectives and vision, Intentions (including the perspective and philosophy, i.e. guiding principles for the solution), Strategy (including an analysis of the premises and recommendations for the development & assessment processes and their interaction), Basic requirements for the design of a solution, and an Enterprise model of the application domain.

Within the Babel Project this was further elaborated for the projects internal purposes in terms of a framework for the URD. The above elements are structured into the following three levels: the strategic level, the procedural level and the operational level, in agreement with the suggestion by [13]. The idea is gradually to develop the frame(s) of reference for assessment in a three-step approach from the organisational objectives (the strategic level), to the middle (procedural) level of system objectives, and to the measurable requirements at the operational level. This tree of objectives serves as the frame of reference for assessment at different levels or stages throughout the project design and evaluation. Note that transitions between the three levels naturally mirror a delegation of responsibility and tasks from a Steering Committee (the strategic management level) to the Project Management Team and individual Working Teams (the production level). The usefulness of this recommendation is tested with the CANTOR (HC 4003) Healthcare Telematics Project as a case.

3. Case Study

A study was implemented as case-based Action Research with the CANTOR (HC 4003) Healthcare Telematics Project as the case. At this type of 'research', the researcher cannot and

are not meant to keep a distance from the object of study, but is actively engaged within the practice situation in close collaboration with the practice team and interacts with this (see the review in [14]), thereby bringing into play an intrinsic relation between research and practice. Within the CANTOR case, the author's role was of a non-interfering type as consultant on quality management & technology assessment, observing the process and providing recommendations for actions, approaches and interventions on the project process and its progress.

The CANTOR Project is concerned with IT-supported classification of images – or objects in images – within the domains of histo-pathology and auto-immune serology. Such a classification is a subjective activity, even when criteria for classification exist. As to improve the quality of care it is necessary to have tools and mechanisms that support the training in classification of images as well as the consensus formation and the later quality assurance of the interpretations provided. The objective of the CANTOR project is to develop software that supports a focussed effort on decreasing the inter- and intra-personnel variation at the mentioned type of image recognition, based on the adaptation and extension of an existing software tool developed for image recognition within another medical domain, i.e. a small and relatively simple IT-development project; however, concerned with the support for cognitive tasks. The involved users (from a total of 7 institutions European-wide) are practicing specialists within the application domains, i.e. in general at the level of medical consultants and chief physicians.

A third party within the CANTOR Project, independent of both the developers and the users, prepared the URD. The approach was based on a synthesis of approaches recommended within the literature, such as Work Analysis, cognitive engineering approach and Checkland's idea of relating systems thinking to systems practice, all based on past experiences with the task analysis approach of the research group on cognitive systems at Risø National Laboratories. This URD was implemented in agreement with the three layered structure developed for Babel, yet with a summary description of the application domain rather than a formal Enterprise model.

The Strategic Level is concerned with the problem definition, the purpose and a global specification of the task as well as a circumscription of the application areas. The Procedural Level is implemented in terms of a number of textual scenarios for the application tasks. The Operational Level is implemented in terms of a large set of specific requirements, like "The users must have possibilities to compare ..." and "...there must be a high degree of freedom in ...". However, note that these requirements are not prescriptive regarding the actual design of the solution. Instead emphasis was put on the needs of the users within their work situation and urged for an iterative development and for discussions between users and developers on the specific solution. Also note that the requirements thus are not fully operational, i.e. in contrast to the recommendations on requirements specification by a couple of standards, [15,16]. The requirements at the operational level were iterated with the user groups and with the developers to achieve a common understanding of the written text and an agreement on the wholeness, including a prioritisation of individual requirements. Also emphasised was that this requirements elicitation process should seek a complete set of requirements fulfilling the strategic and procedural objectives, while the end-result would have to be a trade-off between the total set of needs and requirements and the project's pre-conditions in terms of available resources.

Initially, the developers felt deeply frustrated, because they couldn't discuss directly with the users until the URD was synthesised and approved and hence, couldn't get started with the implementation. They had a prior fine knowledge of the gross application domain from the existing prototype and a trial experiment, and they already felt a clear perception of what was needed and wanted to discuss this face-to-face with the users. However, when the user-developer dialogue started communication problems arose, and the developers asked for interfacing technical user representatives. Moreover, the developers didn't understand that users are generally incapable of grasping the diagramming techniques with the same ease as themselves, c.f. [1,4]. The main reason for the precaution (in terms of the inclusion of a third party for re-

quirements elicitation) was to avoid the risk of bias towards the existing system and to prevent the experienced problems, both of which would hamper an assessment of the real usefulness of the system for the planned purpose.

The impression definitely was that the developers' attitude initially was that of 'adjusting the user to the system', and 'the system is well-designed and well-working, - it is straightforward to use'. The balancing between 'technology push' and 'market pull' was experienced several times at meetings, although fading at the end of the developments. This is perceived only as a natural emotional reaction from a system provider when extending the application range to adjacent domains. Nevertheless, it is a bias (prejudice), which is potentially harmful for all parties in case one seriously wants to assess the usability of a prototype functionality. Therefore, the detachment of the developers in the UR elicitation process was justified. An example is 'kappa-statistics', which the users demanded as a prerequisite for their willingness to use the system after the project, but which was initially and repeatedly abandoned by the developers as not important. The impression was that this was merely a matter of lack of deep knowledge of specific details within the application domains rather than incapability or unwillingness to implement it. Other similar examples indicate misunderstandings at the level below the text of the URD, i.e. interpreted as a lack of deep understanding of the user domains.

Throughout the technical implementation the assessment was primarily accomplished in terms of a sequence of demo sessions with hands-on providing a verbal exchange of feedback and opinions. Only at the point where the prototype was mature enough for the users to judge the usefulness of the new system's function it was feasible to provide ample feedback in stead of just reporting problems, and feasible to plan and initiate more formal assessment activities. Specific shortcomings correlated with cognitive aspects of the prototype's functionality completely blocked the planning of the formal assessment rather than functional aspects like access control and set-up facilities, which were also missing.

Noteworthy was that the discussions between users and developers mainly took place at the strategic and procedural level of the requirements at demos on the functionality, while the operational level seemed mainly to be used for keeping track of progress and completeness and occasionally was used to verify that 'this or that' indeed was part of the agreement (falling back to the traditional UR role). This was opposite within the LIS case.

At least twice, the users strongly refused needs that had previously been elicited, and subsequently agreed upon them again when confronted with the concrete use scenario. Besides this, only a couple of the specific (operational) requirements within the original URD was deleted as irrelevant at a later instance. No user requirement changed radically during the development, but several were given more detail or explained or iterated to find a solution. Some requirements were later 'added' (at the operational requirements level) as a result of discussions on the strategic and procedural level requirements.

4. Discussion

The case presented gives a rough idea of how the interaction between users and developers changed when introducing a three level URD with a coarse granularity of the lowest level, i.e. the operational requirements, and apparently for the better. However, there is – given the premises of this type of research – no frame of reference by which to judge the absolute truth of the latter statement.

Further, the study confirmed that users have difficulties in explicating how they (really) perform their work, - one of the premises leading to the approach taken. The case also demonstrated that misunderstandings arise below the specified level, irrespective of the level of granularity of the requirements, but that the URD evolved primarily as a result of discussions on the strategic and procedural level requirements. In spite of the developers' fine knowledge

of the functions within the application domain, it seemed that the essence of the cognitive processes was much more difficult to explicate and convey and handle than the functional ones, both at the planning of the development and at the assessment of usability and usefulness. Consequently, the Alpha and Omega is that system development is consistently based on strategic aims, procedural ideas, a deep insight into the application domain and with collaboration on specific details of the implementation of the IT-based solution.

Acknowledgement

Elements of this work were partly funded by the Babel (TE 2002) Telematics Engineering Project and the CANTOR (HC 4003) Healthcare Telematics Project under respectively the 3rd and 4th Framework Programme of the European Commission. Further, it was part of a study supported by a grant from the Danish Research Council for Technology & Science and by the Virtual Centre for Health Informatics, Aalborg University.

References

[1] Brender J. Quality Assurance and Validation of large Information Systems - as viewed from the user perspective. Master Thesis, report no 89-1-22. Copenhagen March 1989. (available from the author).
[2] Brender J. Methodology for Assessment of Medical IT-based Systems - in an Organisational Context. Amsterdam:IOS Press, Studies in Health Technology and Informatics 1997;**42**.
[3] Beuscart-Zéphir MC, Brender J, Beuscart R, Ménager-Depriester I. Cognitive Evaluation: How to Assess the Usability of Information Technology in Healthcare. Computer Methods and Programs in Biomedicine 1997;**54(1-2)**:19-28.
[4] Brender J. Methodology for Constructive Assessment of IT-based Systems in an Organisational Context. Int. J. of Med. Informatics 1999;**56**:67-86.
[5] Beuschel W. What will Impact Assessment tell us about Expert Systems Design? In: [17], 1991:63-72.
[6] Goodstein LP. Computer Aids for Decision-Making in Process Control. In: Falzon P, ed. Cognitive Ergonomics: Understanding, Learning and Designing Human-Computer Interaction. London:Academic Press. 1990:231-243.
[7] Gruding J. Groupware and Social Dynamics: Eight Challenges for Developers. Scientific American 1991;**September**:762-774.
[8] Cunningham RJ, Finkelstein A, Goldsack S, Maibaum T, Potts C. Formal Requirements Specification - The Forest Project. In: Proceedings of the Third International Workshop on Software Specification and Design. Washington DC:IEEE Computer Society Press. 1985:186-191.
[9] Falck M. Information System, Work and Organizational Design, How to do it? In: [17], 1991:307-315.
[10] Friis S. User Involvement in Requirements Analysis and Specifications: In Theory and Practice. In: [17], 1991:317-325.
[11] Ruohonen M. Contradictions of Managerial Learning System for Strategic Information Systems Planning. In: Stamper RK, Kerola P, Lee R, Lyytinen K, eds. Proceedings of the IFIP TC8 Working Conference on Collaborative Work, Social Communications and Information Systems, COSCIS '91. Amsterdam: Elsevier Science Publications B.V. 1991: 181-211.
[12] Bansler JP, Bødker K. A Reappraisal of Structured Analysis: Design in an Organisational Context. ACM Transactions on Information Systems 1993;**11(2)**:165-193.
[13] Parker BR. A Multiple Goal Methodology for Evaluating Management Information Systems. OMEGA Int J Mgmt Sci 1985;**13(4)**:313-330.
[14] Mathiassen L. Reflective Systems Development. Doctoral Thesis in computer science at the Faculty of Technology and Science, Aalborg University, ISSN 1397-8640. 1998; **R-98-5006**.
[15] ISO. Quality Management and Quality Assurance Standards - Part 3: Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software, 1st ed., corrected and reprinted 1993-05-01. Genève:ISO. 1991;**ISO 9000-3-1991 (E)**.
[16] IEEE: IEEE Recommended Practice for Software Requirements Specifications, Vol. 830-1984, IEEE, New York, 1993.
[17] van den Besselaar P, Clement A, Järvinen P, eds. Information System, Work and Organization Design. Proceedings of the IFIP TC9/WG9.1 Working Conference on Information System, Work and Organization Design. Amsterdam:Elsevier Science Publishers B.V.