Developing Situationally Specific Methods Through Stakeholder Collaboration.

Helen M. Edwards, J Barrie Thompson and Colin J Hardy
Software Engineering Group, University of Sunderland, UK
{helen.edwards, barrie.thompson, colin.hardy}@sunderland.ac.uk

Abstract

Software development methods can only be used effectively where there is a close match between the method being used and the situational application. There are two key features that need to be considered by those concerned with formalising the development of situational methods: (i) the stakeholder input and (ii) the method engineering process. The method presented, MEWSIC (Method Engineering With Stakeholder Input and Collaboration), formalises the development of situational methods so that links to quality assurance processes is retained. MEWSIC accounts for the number of stakeholders who have a legitimate interest in the success of the project but distinguishes between those who provide input that informs the method engineering process and those who carry out this process. A description of MEWSIC is given bringing out the collaborative nature of the approach. We then discuss MEWSIC's place within software engineering (particularly in relation to method engineering approaches and quality assurance mechanisms).

Keywords

Collaborative Systems Development, Method Engineering, Stakeholders

1. Introduction

There is widespread agreement about the importance of the principles of software engineering and historically companies have looked for ways in which to embed these principles into their development process. However, there is often a mismatch between the methods chosen by an organisation and the type of systems that they develop. Attempts by method developers to second-guess the needs of the system developers have resulted in the marketplace being swamped with both domain specific and more generic methods [1]. The studies of industrial practice in the development of software in the UK, which the Software Engineering Group at the University of Sunderland has made over the last decade, have shown that where organisations have invested in methods these have proved less than ideal (for instance [2,3,4,5,6]), these results are corroborated by others, for instance [7,8]. However, in many cases organisations have built up development infrastructures around these methods and have invested significant resources in trying to make them work. This activity and discontent with current methods has highlighted a desire for situationally specific methods. By this we mean methods which are specifically tailored to fit the organisation and particular project for which they are to be used. In such organisations software developers need to understand the context of their software development projects in order to ensure that the methods used are appropriate. This often leads to those involved in software development also becoming involved in method adaptation activities. In the academic field such method development, or change, has been classified as method engineering (see for example [9]): this is an area of software engineering that is practised rigorously by specialists, but informally by practitioners. To make the research developments in the field of method engineering accessible to practitioners we believe there is a need to specify a process that can be followed. This process can then be used in context to develop situationally specific methods. To ensure that the method developed is appropriate we also believe that such a process must identify the stakeholders in the project whose concerns will impact upon the type of method required for development.

It should also be noted that if organisations wish their software development departments to move up the levels of the SEI's software Capability Maturity Model (CMM) [10] it is not sufficient to simply be using prescribed methods. To progress they need to show that they are optimising their operations and continually improving their processes. Part of such an activity obviously involves ensuring that there is as close a fit as possible between an individual development and the method supporting it and that there is a rigorous process to ensure this synergy.

This paper outlines an approach which addresses these issues, MEWSIC (**M**ethod **E**ngineering **W**ith **S**takeholder **I**nput and **C**ollaboration). In Section 2 the need for such an approach is expounded and the rationale for the proposed process is explained; this is followed in Section 3 by an overview of MEWSIC. In Sections 4 and 5 we define the two stages of MEWSIC in some detail. In Section 6 we discuss the approach in terms of its fit within the method engineering and quality assurance fields of software engineering and finally we draw out the major features of the MEWSIC.

2.Situationally specific development methods (SSDMs)

2.1 The need for SSDMs

Current information systems methods are failing to fulfil the expectations of their developers and the needs of their users: such claims are made by many academics (for instance Baskerville [11] and Hardy [4]) based on their analysis of the use of methods in practice. Consequently, there has been a movement away from earlier pre-conceptions about methods towards one that more closely fits the nature of current systems development where the whole development process is seen as dynamic, and practitioners look (perhaps unconsciously) for a good fit between the type of system they are asked to develop and the method they use to do so. However, once developers decide that complete adherence to a specific method is not necessary then the advantages that practitioners and project managers see in methods are in danger of being eroded. For instance, it is common for organisations with formal quality assurance procedures and certification to have the use of a particular method written into its QA system. Once the link with the "approved" method is broken then the claim that work is carried out to approved standards is more difficult to sustain. Therefore, in order to retain an acceptable argument that formal procedures are used and work can be quality assured, it is necessary to provide some mechanism by which the break from generic, inappropriate methods, to situationally relevant methods can be justified, monitored and assured.

2.2 The rationale driving SSDMs

A mechanism which facilitates the designing of situational methods for development must ensure that the contingency factors that would affect customisation of a method are isolated and that these can be mapped against the risks involved in modifying a method. It might be posited that a comprehensive and generic method for developing situational methods should be specified. However, our empirical data, based on survey and in-depth interviews, (some of which are reported in [3]) has highlighted that the major problem with such an approach is the existence of so many situation specific variables that any approach thus derived would be too complex to be used. Again our data and the work of, for instance, Harmsen [9], has shown that the variables involved are not just limited to the physical characteristics of an individual project on which a method might be used, but also the perceptions and attitudes of the individuals involved towards the methods and tools that might be used. Therefore the manner in which the range of potential variables can be accounted for must be tempered against the need for a practical, useful process for contextualising software development methods.

Discussions with developers, and feedback from our surveys [3,4], suggest that whilst other parties may be interested in the process of method change [12,13] it should really be performed by and for the systems development team. The argument is, that the systems development team: (i) stands between the client (who sponsors the project) and the user (who ultimately has to interact with the system), (ii) has the necessary understanding of what is possible, and (iii) should rigorously document the requirements, liaise with the relevant parties and produce the required system.

Therefore the process involved in deciding upon and using the method should be left to the experts. Other participants roles within the system development process are often perceived by developers as being interested in the conclusions gained from the deliverables of a method, and not in the nature of the deliverables themselves. Systems developers then, not only need tools to model the system, but become responsible for identifying which development approaches will also provide the appropriate conclusions for the stakeholders.

One of the key themes of the MEWSIC process is that the systems development team should not be the only arbiter in the method engineering process, since other stakeholders may have an equal interest in determining what is suitable for their needs. However, it is important that the tasks undertaken by the different stakeholders are clearly delineated. Given this perspective the focus taken in the MEWSIC process is to decompose the approach into two stages. In the initial phase (to define the areas of concern in the specific project) those who have a concern in the project and the systems development process (that is, the stakeholders) are involved. Then, once the stakeholders and their areas of concern have been defined and prioritised, they are used as inputs by the software development team in the method engineering activity phase of the process.

2.3 Our position

To summarise, there are two key features that need to be considered by anyone concerned with formalising the development of situational methods: these are the stakeholder input and the method engineering process. Situational methods satisfy a real world desire to move from "cookbook" approaches to appropriate methods. However, there is often a link in organisations between their prescribed development methods and their quality assurance processes and it important that this link is not lost or weakened. Uncontrolled method tailoring runs the risk of allowing this to happen, but a formalised approach to the development of situational method allows this link to quality processes to be retained. The second key feature to be addressed in answering the need for situational methods is in adequately accounting for the number of stakeholders who have a legitimate interest in the success of the project. Since these stakeholders have different interests, concerns and skills it is necessary to distinguish between those who provide input that informs the method engineering process and those who carry out this process. In Section 3 the overview of MEWSIC shows how these issues are dealt with.

3. The MEWSIC process

The purpose of MEWSIC is to provide software develops with a practical approach which will ensure that decisions taken in the method engineering process are recognised, justified and are capable of being audited. The approach has two main stages: (1) Derive Situationally Relevant Factors and (2) Define the Method for Use.

Within these stages activities and deliverables are specified that provide a mechanism by which the required development method can be defined and justified. This approach is adopted to facilitate quality assurance, since the formalised manner which is adopted is amenable to auditing and objective analysis. Such an approach also enhances the ability of an organisation to replicate activities, be aware that a specific strategy has been developed and identify those decisions that have led to high or poor quality results. In both stages of MEWSIC collaborative work is undertaken: in the first stage the collaboration is with all stakeholders who are affected by, or who affect, the planned systems development project. In the second stage the collaboration is at a different level since it is concerned with the engineering of the required method. Therefore it relies on the technical skills and expertise of those who are involved in the development and planning of software projects (typically, project managers, team leaders and software engineers).

The first stage is iterative in nature and uses the techniques and concepts associated with soft systems approaches and participative work groups (as used for instance by Checkland [14] and Mumford [15]). The identification of stakeholders can assist in the derivation of areas of concern, or conversely acknowledgement of areas of concern can lead to the detection of previously omitted stakeholders. The output from this stage is an definition of stakeholders for the specific project (and their relative importance) and a mapping of the prioritised concerns to these stakeholders. This first stage is discussed in more detail in section 4.

Once this primary stage has been completed the second stage can begin. It is in this stage that the systems development method is engineered by a technical team (whom we term the *"Method Design Team"*) based on the concerns that have been identified by the stakeholders. For many organisations this stage will begin with an examination of their existing methods to determine if any is suitable and whether changes need to be made. However, this stage is flexible enough to allow the method design team to construct a method from individual components - if they have the competencies so to do. (It is worth noting that this is an example of an "area of concern" which should have been addressed in the first stage of the MEWSIC process). Again, during this second (method engineering) stage the decisions which are made and the reasoning supporting them would be explicitly recorded.

4. Stage 1: Derive situationally relevant factors

This stage has five main tasks within it as shown diagrammatically in Figure 1. In Step 1-1 "Identify Initial Stakeholders" the method design team (MDT) use the project initiation document (or project scoping document) and any existing template of stakeholder definitions to define those stakeholders (not individuals) who are expected to be affected by or affect the project: and thus the influence the design approach used. Typical stakeholders are, for instance: the project sponsor, the project manager, the team leader for the development team, the end user, the operations manager. Once these stakeholders have been identified a representative is nominated to contribute to the "Collaborative Stakeholder Group", CSG, this group includes the method design team as they are also stakeholders in the process. Step 1-2 is carried out in parallel with Step 1-1, in this step the document that initiates a project (such as a project initiation document or project scoping report) is examined to identify the general feature that are of relevance to the project and it development approach. Such features would include, for instance, project size, complexity and importance to the business.

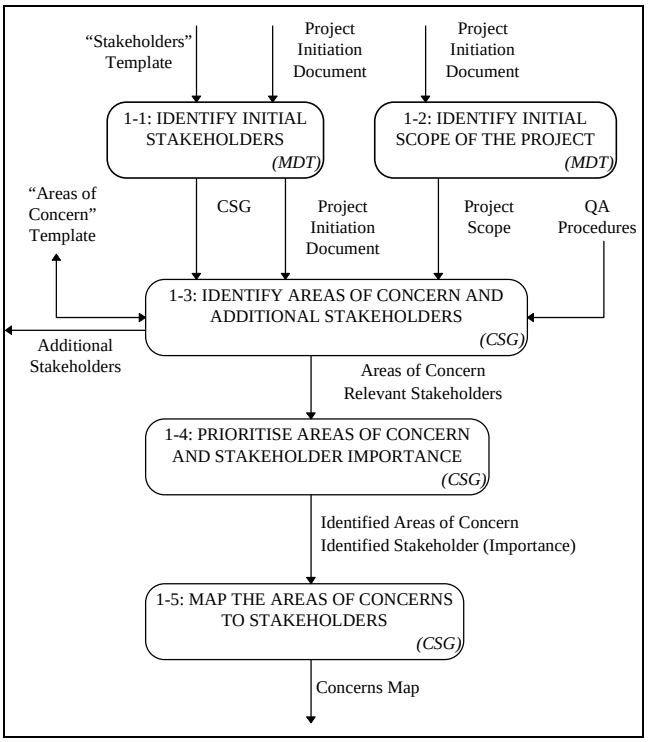

Figure 1: The Steps of Stage 1

Once these initial two steps have been carried out the remainder of the stage is undertaken by those who make up the Collaborative Stakeholder Group. Although the activities are broken into steps and the basic sequential flow is shown in Figure 1 it would be expected that these would be carried out iteratively. In Step 1-3 the areas of concern within the project are identified using approaches such as participative work groups to identify the areas of concern that exist for each set of stakeholders and any additional stakeholders that are revealed as a result of this process. The outcome of the step is a list of areas of concern for specific stakeholders: these areas of concern are analysed to ensure that equivalent concerns from different groups are matched up. In Step 1-4 the CSG then move on to prioritise the areas of concern, order the stakeholders in terms of importance for the project and produce a mapping of areas of concern to stakeholders whose topology reveals the range of areas and level of concern. It is this mapping that is used as input into the second stage of the method, Figure 2 shows the form of the intermediate results.

Rationalisation of the Areas of concern would reveal PS-AoC2 and Eu-AoC11 are equivalent (and can be renamed as AoC3, say). Therefore, once the complete set of documents have been collated and rationalised the "concerns" map can be developed as shown in Figure 3.

Stakeholder: Project Sponsor		
Area of Concern Id	**Area of Concern**	**Priority Level**
PS-AoC1	*6 months time scale for complete project*	4
PS-AoC2	*highly secure system*	5
.....	.....	....
.....	.....	....
PS-AoC25	*portable to different platforms within the organisation*	2
Stakeholder: End User		
Area of Concern Id	**Area of Concern**	**Priority Level**
EU-AoC1	*Core functions available within two months, other available by end of complete project*	5
EU-AoC2	*On-line help system - tailorable to different levels of expertise*	2
.....	.....	....
EU-AoC11	*security of system important: only authorised personnel in department should have access.*	3

Figure 2: Prioritised areas of concern by stakeholder

		Ao C1	**Ao C2**	**Ao C3**	**Ao C4**	**....**	**Ao Cn-1**	**Ao Cn**
	Project Sponsor	**5**		**3**	**2**			
	End User		**5**	**5**	**2**		**2**	**2**
Ordered	**Project Manager**	**4**		**2**	**2**		**4**	**3**
Nominated	**.....**							
Stakeholders	**.....**							
	QA Manager	**5**		**3**	**2**			**5**
		Areas of Concern						

Figure 3: "Concerns" map (1= low, 5 = top priority)

Using this approach an organisation will find that, over time, they will generate, and add to, templates which define typical stakeholders and other templates that record reported areas of concern. To make best use of such information a well developed information retrieval system and associated repository will be needed. Such templates are used as starting point for any project and may be refined/extended for the particular circumstances that occur.

5. Stage 2: Define the method

In this stage the participants (stakeholders) involved are limited to the method design team (MDT). This team is typically composed of:

- the project manager: who has responsibility for the budget, the allocation of resources, ensuring conformance to QA standards, the final deliverable.
- the project team leader: who has responsibility for the specific project, the management of the development process, the organisation of the team, the operational issues affecting the team and the conformance to QA standards.
- the software engineer: who has responsibility for the development of elements of the specific project, possesses expertise in certain development approaches, is responsible for achieving his/her set task deliverables. The stage consists of two steps: 2-1 Design the situational method and 2-2 Validate and Verify the Situational Method.

5.1 Design the situational method

The main activity is carried out within the Step 2-1. The method design team have as inputs to this step:

- The prioritised areas of concern and list of stakeholders for this project
- The "Concerns" map for this project
- The definitions of the methods, techniques and development tools available to the development team.
- Historical "Concerns" map for previous projects and the associated situational method.

With these inputs the team can begin the method engineering process. This process is particularly dependent upon the relationships between these stakeholders and the method aspects, as represented in the grid shown in Figure 4.

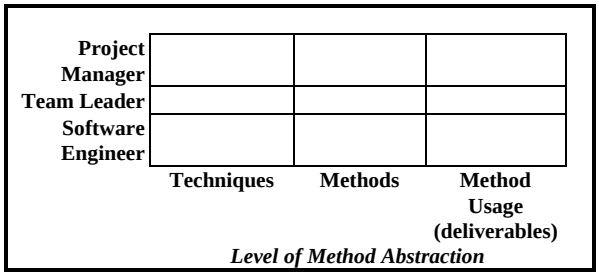

Figure 4: Relationship of MDT and method abstractions

The method engineering activity can be a complex activity and is based upon the approach discussed in [10]. Two examples are discussed here to illustrate the activities undertaken.

5.1.1 Example 1. Here, the outcome of Stage 1 has identified the importance of using systems development deliverable as milestones to track progress (since development within the allocated timescale is of overriding importance). In this case the starting point is to define the deliverables that make suitable milestones, then to evaluate available methods to determine the fit of these deliverables to the methods. Finally, if necessary, elements that belong to other methods (or exist as stand alone techniques) can be examined to and added to the development approach if these provide the deliverables that fit project management requirement. Figure 5 shows the direction of the method engineering process for this case.

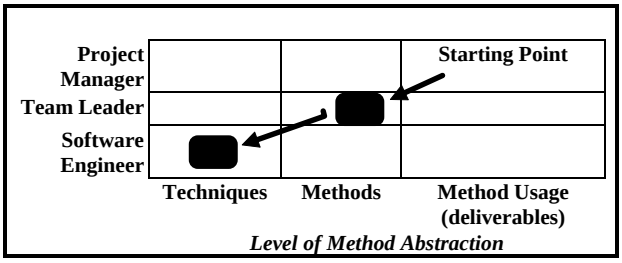

Figure 5: Direction of method engineering process for Example 1

5.1.2 Example 2. The outcome of Stage 1 has identified the system that is to be developed has complex data with numerous update events that need to be effectively scheduled, the organisation have a standard systems development method that is well understood by their developers. The systems is also mission critical system for the company. In this case the starting point is to look at the method elements in the standard method to ensure that the consistency required between them is present, and that they support the required type of modelling. If limitations are found with the method then need to make sure the method is expanded to add the missing elements and ensure that these assist the consistency of design and are understood by the software engineers. The mission critical aspect implies a need for good project management and therefore also need to ensure good event modelling at the method usage level. Figure 6 shows the direction of the method engineering process for this case.

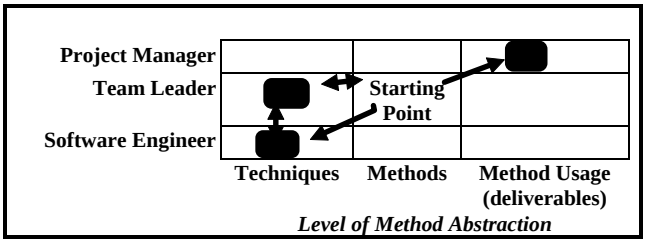

Figure 6: Direction of method engineering process for Example2

5.2 Validate & verify the situational method.

The second step in the stage is needed to ensure that the proposed situational method actually matches the "concerns" map from Stage 1.If this is acceptable then the process is complete and the system development activity can begin in earnest. If , however, there is a mismatch then the MDT need to provide an analysis of the mismatch and identify to which part of the MEWSIC process they need to return.

6. Discussion

6.1 Where does MEWSIC fit ?

Method engineering is a complex field and there are numerous issues to be considered in it. Much of the focus today has been on the technological issues and this has been reflected in work that is primarily linked to metaCASE technologies. However, like many technological developments and solutions these are only truly evaluated when put into practice. Once such tools are trialled the issues of how people interact with, and use, them must be addressed. Therefore we believe our approach, which provides a detailed framework within which software engineers can address method engineering activities, offers a practical mechanism to aid the technology transfer of research and specialist solutions into the software developers' domain . Moreover, the approach ensures that the project stakeholders are involved in the process, but in clearly delineated roles and tasks .

6.2 What does MEWSIC offer ?

The advantages of using a formalised process such as MEWSIC are many. Some of these are highlighted here.

6.2.1 An Explicit Audit Trail The use of a formalised method with specified tasks and identified deliverables ensures that an audit trail exists. This trail reveals the decisions that have been taken (with the supporting justification), the concerns that have been articulated and defended and thus reveals how the method used for a project was derived and judged to fit the required system and its development. The verification and validation step in the method provides confidence is the fit of the method to the project and its areas of concern. Therefore, this mechanism also increases the likelihood that such decisions could be justified and repeated for another similar exercise.

6.2.2 Knowledge Sharing. In these days of impermanence and change in working patterns knowledge sharing and technologically assisted methods of collaborative working are essential for the corporate being, if not the individual. However, there are advantages for the individual also in that novices can be trained more effectively and that knowledge can be transferred even among experienced staff (since it is rare for one person to hold all the skills required within an organisation's software department).The articulation and justification of decisions that has to be made during the MEWSIC process enables software development teams and their stakeholders to share knowledge. In the technical area this is particularly important as it facilitates the learning and acquisition of skills from experienced personnel (such as project team leaders) to more junior systems developers. This addresses the problem that has been reported to us, in our empirical data, by software developers whose ability to customise methods is gained through personal experience. The MEWSIC approach of explicitly acknowledging the issues that are used to design a method helps to unearth the tacit knowledge that is often held by experienced software engineers.

6.2.3 Ownership. The two stage approach of MEWSIC encourages all stakeholders to "buy into" the system design process and feel ownership for the project. Therefore, the progress and success of the project is a common concern. This reflects the advantages of any participative approach, many of which are used in systems development, but it broadens the participation and ownership from a focus on the product include also the development of that product.

6.3 What support does MEWSIC need ?

If the MEWSIC approach is to well in practice then the collection and analysis of data at all stages will need computer based support. There is a need for a well developed repository which also has effective information retrieval and presentation capabilities. Such a repository would hold information of the following types:

- stakeholder templates
- templates of areas of concern
- "concerns" maps (results from previous projects)
- templates of applicable methods
- specifications of method elements (techniques) and methods, or
- mechanisms for linking with existing metaCASE technologies.

The use of the method also needs support from senior managers/CEOs as it is an additional investment over the

usual investment in methods, and therefore the benefits of the approach need to clear.

6.4 Summary

In summary, the main features of MEWSIC are that it formalises the development of situational methods and explicitly focuses on the method engineering process and stakeholder input and collaboration. Since these stakeholders have different interests, concerns and skills MEWSIC distinguished between those who provide input that informs the method engineering process and those who carry out this process. A primary feature for those concerned with QA is that the formalised approach ensures that the link between development methods and quality assurance processes is retained.

The future work on the MEWSIC method will concentrate on the further definition of the detailed tasks within the method and the testing of the method at case study sites. Some work has already been undertaken to develop the repository that is needed to support the method and this will continue, particularly focusing on the information retrieval and presentation issues.

Finally, for those organisations concerned with progress along the CMM levels, the MEWSIC approach offers one tool to help in continually improving their development processes by providing synergy between an individual development and the method supporting it.

7. References

[1] Jayaratna, N. Understanding and Evaluating Methodologies NIMSAD: A Systemic Framework, McGraw Hill, 1994.

[2] Stobart, S.C., Thompson, J.B., Smith, P. Use, problems, benefits and the future direction of computer-aided software engineering in United Kingdom, Information and Software Technology, Vol 33, No 9., November 1991.

[3] Hardy, C.J., Thompson, J.B., Edwards, H.M. The Use, Limitations and Customisation of Structured Systems Development Methods in the United Kingdom, Information and Software Technology, Vol 37, No. 9, p.467

[4] Hardy, C.J., Thompson, B.J., Edwards, H.M. Problems associated with the customisation of structured methods in the UK, BCS Conference on Information Systems Methodologies, Wrexham, 1995.

[5] Hardy, C.J, Stobart, S.C, Thompson, J.B, Edwards, H.M. A Comparison of the Results of Two Surveys on Software Development and the role of CASE in the UK, CASE '95, Toronto 1995.

[6] Davis, C.J., Gillies, A.J, Smith, P and Thompson, J.B. Current practice in software quality and impact of Certification schemes. Software Quality Journal 2,145-161, 1993.

[7] Dekleva, S.M. The Influence of the Information Systems Development Approach on Maintenance, MIS Quarterly, September, 1992 pp355-372.

[8] Palvia, P. Nosek, J.T. A Field Examination of System Life Cycle Techniques and Methodologies, Information and Management, 25,1993 pp73-84

[9] Harmsen,F, Brinkkemper,S and Oei,H. Situational Method Engineering for Information System Projects, In: Olle, T.W., A.A. Verijn-Stuart (Eds.) Proceedings of the IFIP WG8.1 Working Conference CRIS'94 Maastricht, 1994, pp. 169-194

[10] Humphrey, W.S. Managing the Software Process, Addison-Wesley, Reading, Massachusetts, 1989.

[11] Baskerville, R, Travis, J, and Truex, D. Amethodical Systems Development: The Deferred Meaning of Systems Development Methods, Working Paper, Copenhagen Business School, Denmark, 1996.

[12] Avison, D, Fitzgerald, G. Information Systems Development: Methodologies, Techniques and Tools, McGraw-Hill, 1995.

[13] Olle, T.W., Hagelstein, J., MacDonald, I.G., Rolland, C., Sol, H.G., Van Assche, F.J.M., Verrijn-Stuart, A.A. Information Systems Methodologies: A framework for understanding, Addison Wesley, 1994.

[14] Checkland, P., Scholes, J., Soft Systems Methodology in action, Wiley, 1990

[15] Mumford.E. Effective Requirements Analysis and Systems Design: The ETHICS Method, MacMillan, Basingstoke, 1995.