AMCIS 2004 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2004

On the Elements of a Pluralist Systemic Approach for the Evaluation of Factors Affecting Software Development Productivity

O. Petkova
Central Connecticut State University

On the Elements of a Pluralist Systemic Approach for the Evaluation of Factors Affecting Software Development Productivity

Olga Petkova
Central Connecticut State University
petkovao@ccsu.edu

ABSTRACT

The elements of a pluralist systemic framework for the evaluation of factors affecting software development productivity are discussed. The framework is based on the Multiple Perspectives concept by Linstone, Multimethodology by Mingers and Critical Systems Thinking. The proposed mix of methods complements existing cost estimation techniques by providing deeper understanding of the role of the various factors affecting a software project

Keywords

Software development productivity, Multiple perspectives, Critical Systems Thinking, Multimethodology.

INTRODUCTION

Escaping the tyranny of crunch mode of work by software developers requires greater control over the development process, and reasonably accurate project estimates provide essential support for acquiring this control (Johnson et al., 2000:51). There have been a number of publications on methods for software cost and effort estimation. A broad overview of the status and prospects of software economics, including cost and effort estimation is presented in Kemerer (1998), Boehm and Sullivan (1999) and elsewhere. The multiple dimensions of the factors influencing cost and effort estimation and software productivity in general represent a challenge for an interdisciplinary research into their nature.

In theory, productivity is defined as the amount of output produced per unit of input. Due to the many intangibles associated with software, the outputs and inputs in software development are very difficult to measure. Hence, a better understanding of these factors and the relationships between them is essential to the improvement of the management of the software development process and the increase of value for the consumers.

After pointing that the traditional literature in software cost estimation leads practitioners to believe that if they do not use algorithmic models or data-intensive estimation processes their estimates will be inaccurate, Kitchenham et al. (2002:58) claim that there is some evidence for the value of the human input to the procedures or tools advocated by researchers and vendors. A characteristic feature for them is the greater role of human judgment in making estimates. It must be mentioned however that well established algorithmic cost estimation methods like COCOMOII and Function Point Analysis do require also some subjective input in the form of the relevant cost factors that might influence a particular project. Therefore, human subjectivity cannot be avoided in cost estimation. A logical conclusion is that we need to find ways in which to limit the potential negative impact of uncontrolled subjectivity and uncertainty in the estimation process.

In his taxonomy of the state of the art, Reifer (2003:80) lists two research thrust areas regarding software economics: reasoning models and tools that deal with uncertainty and multi attribute decision models. Some like Stamelos and Angelis (2001) have used a simulation approach to manage uncertainty in project portfolio cost estimation. Others have focused on the use of Systems Dynamics as a tool for organizational learning about management of the software project (see Abdel-Hamid (1996)). This paper focuses on the justification of an alternative approach to manage uncertainty in software cost and effort estimation through a mix of methods from systems thinking and multi attribute decision analysis. There has been no published justification of a similar combination of soft systems methods and multi attribute decision analysis, applied to the evaluation of factors affecting software development productivity in a particular project environment.

The paper proceeds with an overview of the research methodology in this investigation, of the justification for the inclusion of the various elements in the proposed framework for evaluation of factors affecting software development productivity and a discussion of how its elements interact, based on its experimental validation, followed by conclusions.

WHY DO WE NEED A PLURALIST FRAMEWORK FOR EVALUATION OF FACTORS AFFECTING SOFTWARE DEVELOPMENT PRODUCTIVITY

A question that needs to be answered is whether a single approach could be applied or a combination of several approaches or parts of them. Understanding holistically the factors affecting software development productivity is a very complex issue. One may advocate a pluralist approach for a complex problem like this. The nature of the proposed pluralism is an open question, which was one of the motivations behind this research.

The relevance of pluralism to the field of Information Systems is subject to debates (see Benbasat and Weber, 1996; Robey, 1996, Hirschheim and Klein, 2003). While there is general agreement about its applicability in the fields of management science and systems thinking, there are diverse views about its nature and philosophical justification (see Mingers and Gill, 1997; Jackson, 2000).

It must be noted that until recently pluralism was understood as the need to apply diverse whole methodologies in Information Systems research projects. Mingers (2001a) showed theoretically the relevance of mixing methods (parts of methodologies) in IS research. Analogically, we considered mixing methods from different methodologies in the framework for evaluation of the factors affecting software development productivity. Following Landry and Banville (1992) a disciplined diversity will be advocated in this research. This implies also that special attention should be paid to the justification of the selected approaches.

The evaluation of factors affecting software development is a complex problem. Hence, probable candidate approaches suitable for it might be from the areas of systems thinking and multi attribute decision analysis. Both of these claim to be applicable to complex, “messy” problems (see Ackoff, 1981 and Stewart, 1992). There have been a lot of changes over the past 15 years within both areas leading to their greater maturity. These are associated with significant debates of a methodological nature.

The discussions within the systems thinking field were predominantly around the feasibility of pluralism (see e.g. Jackson (1997a), Mingers and Gill (1997), Mingers (2001b)). Another issue has been the methodological foundation of pluralism, with the contenders being critical systems thinking (Jackson, 2000), critical realism (Mingers, 2000) and postmodernism (reflected in the work of Taket and White (2000)). Thus, the question of providing a philosophical justification for the potential pluralist approach presented another challenge for this research.

The field of Multi-criteria Decision Making (MCDM), which includes Multi Attribute Decision Analysis (MCDA), is still at the stage of school formation, according to Lootsma (1996). That explains the lack of a coherent paradigm within the field. The latter is characterised by heated debates on the correctness of various schools of thought within the field. There is no evidence of reconciliation of the different views of the proponents of these schools.

Since methodologically pluralism is best supported through the Multiple Perspectives idea of Linstone (1984) and Critical Systems Thinking (see Jackson, 2003), we have chosen these two theories in combination to be the methodological foundation of this research. Linstone’s multiple perspectives idea calls for examining problems through the Technical, Organizational and Personal perspectives (see for details Linstone, 1984:39-84). That is somewhat similar, though independent, to the ideas of the three worlds by Habermas (1984): the material world, the social world and the personal world. The latter concept is a corner stone in Critical Systems Thinking (CST) and one of its strands: the Multimethodology for mixing methods proposed by Mingers (1997, 2001b). The contemporary view of CST is not associated with the emancipatory paradigm but with the promotion of pluralism at the methodological level (Jackson, 2003).

An analysis of the IS literature shows that with the exception of Hirschheim and Klein (1989), Hirschheim, Klein and Lyytinen (1995) and Iivari et al. (1998) relatively little work has been done on comparisons between interpretivist and neohumanist approaches. The latter are not widely reflected in the IS literature. Apart from the work of Ngwenyama(1991), Jackson (1992), Hirschheim and Klein (1994), Ngwenyama and Lee (1997) and O. Lee (2002), a few efforts have been made to bring together Critical Social Theory and Critical Systems thinking and IS research. On the other hand a significant body of knowledge in the IS literature is linked to Soft Systems Methodology (see Checkland and Scholes, 1990 and other sources), a powerful interpretive systems approach. Very little has been published on mixing different paradigmatic approaches (for some references see Mingers, 2001).

Strong pluralism is advocated by Mingers (1997a), who argues that most, if not all, intervention situations would be dealt with more effectively with a blend of methodologies from different paradigms. This paper provides justification of a framework for evaluation of factors affecting software development productivity that can be classified as an example of strong pluralism. The postmodernist pragmatic systems approach of Taket and White (2000) can be seen as an example of strong pluralism too, as can some case studies in Flood (1995) and Mingers and Gill (1997). All of them assume that there are grounds to believe that cross-paradigm research is philosophically feasible (Mingers, 1997a:14). Mingers (2001a) considers Multimethodology to be a paradigm on its own. However, the various systems methodologies known today (see Jackson, 2000 for their analysis) are different in their philosophical assumptions. Flood (1995) and Jackson (2000) have based their later work on assumptions similar to those of critical systems thinking with some influence of postmodernism. Mingers (1997a; 2000; 2001a) is founding his work on Critical Systems Thinking and Critical Realism.

WHAT TECHNIQUES MIGHT BE CONSIDERED IN THE FRAMEWORK FOR EVALUATION OF FACTORS AFFECTING SOFTWARE DEVELOPMENT PRODUCTIVITY?

A holistic systemic framework for evaluation of the factors affecting software development productivity should be comprehensive and well justified regarding its origins within the body of knowledge of Information Systems and Systems Thinking. In addition, the framework for the evaluation of factors affecting software development productivity should be easy to use and adaptable to any software development environment. The outcome of such a framework leads to a better understanding of the constraints affecting a particular project and a deeper appreciation of the factors that may lead to improvement of software development productivity.

Strategic Assumptions Surfacing and Testing (SAST) by Mason and Mitroff (1981) is an operationalization of some of the ideas of Churchman, such as incorporation of dialectical debate involving all stakeholders when dealing with wicked problems. A software project involves often many stakeholders. They may have in common a desire to produce a better working system. They may still have differences in their views. It might be considered that the stakeholder groups may not be necessarily representative of adversarial viewpoints. Therefore, the debate itself may not be dialectical. On that basis one can then employ only the first two stages of the SAST methodology. The most crucial stage of SAST to be considered for the proposed framework here is the assumption surfacing stage (see Mason and Mitroff, 1981). It incorporates stakeholder analysis, during which groups of stakeholders with common vested interests are identified. We used for formulation of stakeholder assumptions the Nominal Group Technique (NGT).

The framework for evaluation of factors affecting software development productivity has to ensure full participation of all stakeholders, something that is supported by most soft systems approaches. This issue is related to another element of Churchman's approach, his insistence that the systems designer's first obligation in carrying out a systems study is not to the decision makers, rather it is to the "clients", customers, or beneficiaries of the system (Churchman, 1971). It can be claimed that the above characteristic, together with his dialectical approach and his desire to strive for "whole systems improvement", can be seen as a vehicle promoting improvement in the problem situation, in this case increased software development productivity.

Soft Systems Methodology (SSM) was developed by Checkland as a strategy for handling complex problems involving socio-technical systems (Checkland and Scholes, 1990). Instead of being based upon the paradigm of "optimization", it is rather founded on the paradigm of "learning". The strong aspect of SSM is its tendency to support judgment rather than represent it. It is interpretive in nature, and embraces subjectivity. Since the nature of the problem of evaluation of factors affecting software productivity is associated predominantly with the identification of the "mess", just two elements from SSM are considered here, namely: rich pictures and CATWOE analysis with their corresponding root definitions (see Checkland and Scholes,1990). The assumptions raised in the SAST stage can be classified along the three dimensions of a systemic intervention suggested in the revised second mode of SSM: analysis of the intervention, social analysis and political analysis (Checkland and Scholes, 1990). There are a number of criticisms leveled at SSM, and an important one is the lack of support given by SSM in the Choice and Implementation stages of Simon's decision- making process (Jayaratna,1994). These criticisms are important for understanding of why we mixed different methods in our framework. There are several other criticisms of soft systems thinking methodologies summarized in Jackson (2000), omitted here for space reasons.

It is appropriate to provide some comparisons between MCDA and Soft Systems Methodology, which might be of use when considering their potential combined application in the envisaged methodology. The first comparison refers to the type of variables in both approaches. In both cases qualitative variables are considered, while in MCDA quantitative ones are also included. The second one refers to the expression of strength of association between the variables involved. While AHP is using a ratio scale to express the comparisons between the variables, SSM does not provide a means for measuring the intensities of relationships between elements of a problem.

One of Churchman's greatest contributions to systems thinking was establishing the fundamental idea that the drawing of boundaries is crucial to determining how improvement is to be defined and how action should be taken (Jackson, 2003:214). Ulrich, a student of Churchman, extended in 1983 his idea of boundary judgments under the influence of Habermas, who argued that rationality emerges through dialogue between those that are involved and those likely to be affected by a systems design (see Jackson, 2003:214). Since some of the parameters of the messy problem of evaluation of the factors affecting software productivity are already revealed through the SSM techniques mentioned above, we used only eight of the twelve critically heuristic categories in Ulrich's boundary judgment questions (Ulrich, 1998). These covered issues concerning what is a measure of improvement, what resources are needed, identification of the decision environment, expert, expertise, guarantor for success, witness, world view of the improvement process (see Ulrich, 1998).

After an investigation of software development, systems thinking and multiple criteria decision analysis, and the methods and techniques used in them, their strengths and weaknesses, we could move to the selection of appropriate methods. The reasons for the selection of the candidate techniques for this mix-method multi-methodology are based on the previous analysis of their features. They complement each other. The following section will attempt to formulate a pluralist framework for the evaluation of factors affecting software development productivity.

A PLURALIST FRAMEWORK FOR EVALUATION OF FACTORS AFFECTING SOFTWARE PRODUCTIVITY

The foundation of the proposed framework is based on Critical Systems Thinking which implies a framework that is able to take into account the strengths of the individual techniques involved and the inclusion of all relevant stakeholders(see Jackson, 2000). It is inspired also by the Multiple Perspectives Approach, defined originally by Linstone (1984).

The techniques in the framework for evaluation of the factors affecting software development productivity are: rich pictures and CATWOE analysis as elements from SSM (Checkland and Scholes, 1990); stakeholder identification from SAST (Mason and Mitroff, 1981); stakeholder assumptions surfacing through idea generation and their rating using NGT and their classification as technical, cultural or political issues in a SSM analysis; Boundary Judgment Questions within CSH (Ulrich, 1998); and prioritisation of the factors using AHP (Saaty, 1990). The selected techniques were applied according to their relevance to Habermas's three worlds (see Habermas, 1984) as is shown in Figure 1.

Social world	**SSM (Rich pictures, CATWOE),**
	Critical Systems Heuristics
	Stakeholder analysis
Personal world	**Rich pictures, idea generation (NGT), MCDA**
Material world	**Statistical project data if available, MCDA**

Figure 1. The techniques included in a framework for the evaluation of factors affecting software development productivity serving the three worlds of Habermas (1984), providing multiple perspective evaluation following Linstone (1984).

Linstone (1984) does not use similar techniques to ours for serving the purposes of a multiple perspectives approach as several of those were not known at the time when he developed his ideas. Mingers(2001b) points the applicability of techniques from Checkland's SSM and Ulrich's Critical Systems Heuristics to some aspects of the three worlds by Habermas in his Multimethodology approach, but he does not mention the possibility of using SAST, NGT or MCDA, as they are used in our mix-method evaluation of software development productivity factors. The foundations of our mix-method framework are different somewhat from those of Mingers' Multimethodology (Mingers, 2001a) as we do not consider like him that pluralism belongs to a new paradigm, recognizing the plurality and diversity of the world. Rather we accept Jackson's idea behind Critical Systems Practice, according to whom the specifics of the different existing paradigms need to be recognized when applying different techniques developed originally within particular paradigms. The linked methods in our framework inform each other with their results in the course of the evaluation of the factors affecting software development. The participants in the evaluation are lead by a facilitator who continually assesses those results through the concerns of the different paradigms of the techniques involved (see Jackson, 2003:306).

The potential issue of paradigm incommensurability when dealing with methods from different paradigms in the evaluation process is resolved on the assumption of the possible dialogue between approaches from different paradigms on the basis of Witgenstein's theory of language games (see Hassard, 1990). This communication between methods from different paradigms is seen as a vehicle to enhance the effectiveness of the proposed framework as a vehicle to develop a multiple perspective understanding of the factors affecting software development productivity in a particular project.

The first task associated with the application of the proposed framework is to identify those who have a role to play in improving software development productivity. As a suitable tool for this purpose stakeholder identification from the Strategic Assumptions Surfacing and Testing (SAST) approach by Mason and Mitroff (1981) is conducted.

A second aspect of the problem is how to develop a deeper understanding on the part of each stakeholder. On the basis of the preceding analysis, some elements of Soft Systems Methodology (SSM) are included in the framework. It promotes organisational learning through encouraging employee participation in problem solving (Checkland and Scholes, 1990). An influence of SSM on our framework is observable in the way in which various issues raised by the stakeholders are grouped. The participants are asked to classify the ideas raised during the NGT session of stakeholder assumption surfacing into three categories: technical, cultural and political. It is in line with the different streams of analysis in SSM (mode two) (Checkland and Scholes, 1990). The other elements of SSM that are considered explicitly here are rich pictures and CATWOE analysis.

By supporting the choice phase of Simon's model of decision making, AHP allows stakeholders to evaluate the various factors affecting productivity and determine the one most important. They can explore the influence of various management strategies on the priorities of various factors affecting software development productivity (see Saaty, 1990).

The boundary judgment ideas of Critical Systems Heuristics (Ulrich, 1998) together with the value issues that emerge through the application of MCDA, help implicitly to formulate action plans for improvement of software development productivity which emerge as a result of the evaluation process.

We need to stress the intertwined relationships between the framework elements of SSM, Nominal Group Technique, MCDA and the Boundary Judgment Questions within CSH. The cycling between the techniques in the framework contributes to the emerging systemic understanding of the factors affecting software development productivity. As a result of the incorporation of an MCDA technique, the suggested framework is focused on the most important factors within a particular project, making it more action oriented than the traditional soft systems approaches applied alone.

The framework for evaluation of factors affecting software development productivity was experimentally validated on a complex software project. It is not possible to include details for space reasons and because this paper is rather about the justification of the elements of the framework. The implementation involved a large metal processing plant working with two external software providers on a new production planning information system that aimed to improve its flexibility in reacting to the needs of the market. The framework was used for a systemic evaluation of issues affecting software development productivity and their prioritization according to the circumstances of the particular software project. The findings of the brainstorming sessions and stakeholder identification, as well as the rich picture, the CATWOE analysis and the classification of the stakeholder assumptions into technical, cultural and political issues helped in the problem structuring step (see Rosenhead and Mingers, 2001). It was followed by the application of an AHP model for prioritization of the factors. The boundary judgment questions provide additional information expanding the knowledge about the messy problem of evaluation of the factors affecting software development productivity.

The framework was applied through three half-day sessions with representatives of the client company and of the two outside software companies involved in this IS project. The knowledge of the three teams about the various issues surrounding the factors influencing the productivity in that complex software project grew through the dialogue that took place during the sessions. The teams found the framework easy to use. More importantly, the post-implementation interviews showed that they considered it relevant for improvement of the management of the project. It helped them understand better each others' values and build trust for improved relationships. It allowed formulation of an understanding for the reasons why the project was falling behind schedule. It outlined also the most important factors for that particular project from the points of view of the users and of the developers. The application of the framework was a first step towards improvement of communication and the management of the project by both sides.

CONCLUSION

The proposed framework promotes a view of software development not just as a software engineering activity, but as an Information Systems development activity, thus placing a greater weight on the role of the users and the software developers in the development process, including their values and culture.

To the best knowledge of the authors, it is the first systemic analysis on the relationships between the factors affecting software development productivity in a particular project and organization. This framework contains a unique synthesis of elements of techniques from six different methodologies. It is true that its components are parts of well-known approaches. However, the combination of the techniques concerned and the way they are used has not been reported before in the literature. As Bowen (1998) remarks, "in the jigsaws that are methodologies, the pieces are…much the same, but the pictures produced are individual creations".

It is assumed that it is the process for evaluation of the factors affecting software development productivity that is transferable from one organisation to another, and not the actual form and strength of these relationships. The latter will inevitably vary from one environment to another as a result of the unique features of any organization. The proposed framework is aimed at gaining a better insight into the importance of the specific factors affecting software development productivity. Management can concentrate on them as a result of that. It must be stressed, however, that this research is not on the software process per se. It is also not aimed at producing estimates of software development effort and productivity, but indirectly it contributes to the improvement of their precision.

This paper is indirectly linked to research in the area of software cost and effort estimation, software process management, and software quality assurance, and in broader terms to the entire field of Information Systems development. It does not aim, however, to substitute for any of the existing approaches related to the investigation of relationships between factors affecting software development productivity, but rather to complement them through enabling organizational learning about the factors affecting software development productivity in a particular organization. Future work may focus comparing of the proposed framework with other research on the impact of organizational learning on improvement of software project management. Another possible direction is measuring the impact of the application of a similar framework on the improvement of software development productivity.

The framework for evaluation of factors affecting software development productivity, whose components were discussed above, addresses some of the disconnects of the IS research community as discussed by Hirschheim and Klein (2003:14). From the point of view of the needs of the software industry, there is a gap between the claims of research in the area of software measurement about its usefulness, and the actual degree to which such approaches are used in practice (Pfleeger et al. 1997). One of the ways to reduce that gap is to provide better methods for assessment of factors affecting software development productivity that are intuitive and easy to use by managers, and at the same time are rigorous from the prospective of the current state of decision making as a scientific field. The proposed framework is transferable to any environment and does not depend on the collection of large amounts of information in specialized project databases, as is typical for other approaches to software measurement.

REFERENCES

1. Abdel-Hamid, T.K. (1996) The Slippery Path to Productivity Improvement, *IEEE Software*, 13, 4, 44-52.
2. Ackoff R. (1981) The art and science of mess management, *Interfaces*, 11, 1, 20.
3. Benbasat, I. and Weber, R. (1996) Research Commentary: Rethinking "Diversity" in Information Systems Research, *Information Systems Research*, 7, 4, 389-399.
4. Boehm, B. and Sullivan, K. (1999) Software Economics: Status and Prospects, *Information and Software technology*, 41, 14, 937-946.
5. Bowen, K. (1998) Some Thoughts on Multimethodology, *Systems Practice and Action Research*, 11, 2, 169-177.
6. Checkland P B, and Scholes, J. (l990) Soft Systems Methodology in Action, Wiley, Chichester.
7. Churchman, C. W. (1971) The Design of Inquiring Systems, Basic Books, New York.
8. Deetz, S. (1996) Describing Differences in Approaches to Organizational Science : Rethinking Burrell and Morgan and their Legacy, *Organization Science*, 7, 2, 190-207.
9. Flood, R. (1995) Solving Problem Solving, Wiley, Chichester.
10. Habermas, J. (1984) The Theory of Communicative Action Vol.1: Reason and Rationalisation of Society, Heinemann, London.

11. Hassard, J. (1990) An Alternative to Paradigm Incommensurability in Organization Theory, in Hassard, J. and Pym, D. (Eds). (1990) The Theory and Philosophy of Organizations: Critical Issues and New Perspectives, Routledge, London.
12. Hirschheim, R. and Klein, H. (1989) Four Paradigms of Information Systems Development, *Communications of the ACM*, 32, 10, 1199-1216.
13. Hirschheim, R. and Klein, H. (1994) Realizing Emancipatory Principles in Information Systems Development: The Case for ETHICS, *MIS Quarterly*, 18, 1, 83-109.
14. Hirschheim, R., Klein, H. and Lyytinen, K. (1995) Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations, Cambridge University Press, Cambridge.
15. Hirschheim, R. and Klein, H.K (2003) Crisis in the IS Field? A Critical Reflection on the State of the Discipline, *Journal of the Association for Information Systems,* 4, 5, 237-293.
16. Iivari, J., Hirschheim, R. and Klein, H.K. (1998) A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies, *Information Systems Research*, 9, 2, 164-193.
17. Jackson, M.C. (1992) An Integrated Programme for Critical Thinking in Information Systems Research , *Journal. of Information Systems*, 2, 83-94.
18. Jackson, M.C.(1997a) Pluralism in Systems Thinking and Practice, in: Mingers, J. and Gill, A. (Eds) (1997) Multimethodology: the Theory and Practice of Integrating OR and Systems Methodologies, Wiley , Chichester, 347-378.
19. Jackson, M. C. (1997b) Towards Coherent Pluralism in Management Science, working paper No. 16, Lincoln school of Management.
20. Jackson, M. C. (2000) Systems Approaches to Management, Kluwer/Plenum, New York-Boston.
21. Jackson, M. C. (2003) Systems Thinking: Creative Holism for Managers, Wiley, Chichester.
22. Jayaratna, N. (1994) Understanding and Evaluating Methodologies, McGraw Hill, London.
23. Johnson, P., Moore, C., Dane, J. and Brewer, R. (2000) Empirically Guided Software effort Guesstimation, *IEEE Software*, November/December, 51-56.
24. Jones, C. (2003) Variations in Software Development Practices, *IEEE Software*, November/December, 22-27.
25. Kemerer, C.F. (1998) Progress, Obstacles, and Opportunities in Software Engineering Economics, *Communications of ACM*, 41, 8, 63-66 .
26. Kitchenham, B., Pfleeger, S., McColl, B. and Eagan, S. (2002) An Empirical Study of maintenance and development estimation accuracy, *The Journal of Systems and Software*, 64, 1, 57-77.
27. Landry, M. and Banville, C. (1992) A Disciplined Methodological Pluralism for MIS Research, *Accounting, Management and Information Technology*, 2, 2, 77-97.
28. Lee, O. (2002) Cultural differences in virtual teams, a critical social theory perspective, *Cyber Psychology and Behaviour,* No 3.
29. Lootsma, F.A. (1996) Comments on Roy, B. and Vanderpooten, D., The European School of MCDA: Emergence, Basic features and Current Works*, Journal of MultiCriteria Decision Analysis*, 5, 1, 37-38.
30. Mason, R. and Mitroff, I. (1981) Challenging Strategic Planning Assumptions, Wiley, New York.
31. Mingers, J. (1997a) Multi-paradigm Multimethodology, in Mingers J. and A. Gill Eds) (1997), Multimethodology: the Theory and Practice of Integrating OR and Systems Methodologies, Wiley, Chichester, 1-22.
32. Mingers, J. (1997b) Towards Critical Pluralism, in Mingers J. and A. Gill Eds) (1997), Multimethodology: the Theory and Practice of Integrating OR and Systems Methodologies, Wiley, Chichester, 407-440.
33. Mingers, J. (2001a) Combining IS Research Methods: Towards a Pluralist Methodology, *Information Systems Research*, 12, 3, 240-259.
34. Mingers, J. (2001b) Multimethodology – Mixing and Matching Methods, in Rosenhead J and Mingers J. (Eds), Rational Analysis for a problematic world revisited, Wiley Chichester.
35. Mingers, J. and Gill, A. (Eds) (1997) Multimethodology: the Theory and Practice of Integrating OR and Systems Methodologies, Wiley , Chichester.
36. Pfleeger, S.L., Jeffery, R., Curtis, B. and Kitchenham, B. (1997) Status Report on Software Measurement, *IEEE Software*, March-April, 33-43.

37. Reifer, D.J. (2003) Is the Software Engineering State of the Art Practice Getting Closer to the State of the Art?, *IEEE Software*, November/December, 78-83.
38. Robey, D. (1996) Research Commentary: Diversity in Information Systems Research: Threat, Promise and Responsibility, *Information Systems Research*, 7, 4, 400-408.
39. Rosenhead, J. and Mingers, J. (2001) Rational Analysis for a problematic world revisited, Wiley Chichester.
40. Stewart, T. (1992) A Critical Survey of the Status of Multiple Criteria Decision Making Theory and Practice, *OMEGA*, 20, 5/6, 569-586.
41. Saaty, T. (1990) Multicriteria Decision Making - The Analytic Hierarchy Process, 2nd ed. RWS Publications, Pittsburgh.
42. Stamelos, I. and Angelis, L. (2001) Managing uncertainty in project portfolio cost estimation, *Information and Software Technology*, 43, 759-768.
43. Taket, A. and White, L. (2000) Partnership and participation, Wiley, Chichester.
44. Ulrich, W. (1998) Systems Thinking as if People Mattered: Critical Systems Thinking for Citizens and Managers, Working paper No.23, Lincoln School of Management.