The South African Institute for Computer Scientists and
Information Technologists

ANNUAL RESEARCH AND DEVELOPMENT
SYMPOSIUM

23-24 NOVEMBER 1998
CAPE TOWN
Van Riebeeck Hotel in Gordons Bay

Hosted by the University of Cape Town in association with the CSSA,
Foche/Sharoon University for CHE and
The University of Natal

PROCEEDINGS

EDITED BY
D. PETKOV AND L. VENTER

SPONSORED BY

&ABSAGroup
The South African Institute for Computer Scientists and Information Technologists

ANNUAL RESEARCH AND DEVELOPMENT SYMPOSIUM

23-24 NOVEMBER 1998
CAPE TOWN
Van Riebeeck hotel in Gordons Bay

Hosted by the University of Cape Town in association with the CSSA,
Potchefstroom University for CHE and
The University of Natal

GENERAL CHAIR : PROF G. HATTINGH, PU CHE

PROGRAMME CO-CHAIRS:
PROF. L VENTER, PU CHE (Vaal Triangle), PROF. D. PETKOV, UN-PMB

LOCAL ORGANISING CHAIR: PROF. P. LICKER, UCT - IS

PROCEEDINGS

EDITED BY
D. PETKOV AND L. VENTER

SYMPOSIUM THEME:
Development of a quality academic CS/IS infrastructure in South Africa

SPONSORED BY

ABSA Group
FOREWORD

The South African Institute for Computer Scientists and Information Technologists (SAICSIT) promotes the cooperation of academics and industry in the area of research and development in Computer Science, Information Systems and Technology and Software Engineering. The culmination of its activities throughout the year is the annual research symposium. This book is a collection of papers presented at the 1998 such event taking place on the 23rd and 24th of November in Gordons Bay, Cape Town. The Conference is hosted by the Department of Information Systems, University of Cape Town in cooperation with the Department of Computer Science, Potchefstroom University for CHE and and Department of Computer Science and Information Systems of the University of Natal, Pietermaritzburg.

There are a total of 46 papers. The speakers represent practitioners and academics from all the major Universities and Technikons in the country. The number of industry based authors has increased compared to previous years.

We would like to express our gratitude to the referees and the paper contributors for their hard work on the papers included in this volume. The Organising and Programme Committees would like to thank the keynote speaker, Prof M.C Jackson, Dean, University of Lincolnshire and Humberside, United Kingdom, President of the International Federation for Systems Research as well as the Computer Society of South Africa and The University of Cape Town for the cooperation as well as the management and staff of the Potchefstroom University for CHE and the University of Natal for their support and for making this event a success.

Giel Hattingh, Paul Licker, Lucas Venter and Don Petkov
**Table of Contents**

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lynette Drevin</td>
<td>Activities of IFIP wg 11.8 (computer security education) &amp; IT related ethics education in Southern Africa</td>
<td>1</td>
</tr>
<tr>
<td>Reinhardt A. Botha and Jan H.P. ElofT</td>
<td>exA Security Interpretation of the Workflow Reference Model</td>
<td>3</td>
</tr>
<tr>
<td>Willem Krige and Rossouw von Solms</td>
<td>Effective information security monitoring using data logs</td>
<td>9</td>
</tr>
<tr>
<td>Carl Papenfus and Reinhardt A. Botha</td>
<td>A shell-based approach to information security</td>
<td>15</td>
</tr>
<tr>
<td>Walter Smuts</td>
<td>A 6-Dimensional Security Classification for Information</td>
<td>20</td>
</tr>
<tr>
<td>Philip Machanick and Pierre Salverda</td>
<td>Implications of emerging DRAM technologies for the RAM page Memory hierarchy</td>
<td>27</td>
</tr>
<tr>
<td>Susan Brown</td>
<td>Practical Experience in Running a Virtual Class to Facilitate On-Campus Under Graduate Teaching</td>
<td>41</td>
</tr>
<tr>
<td>H.D. Masethe, T.A Dandadzi</td>
<td>Quality Academic Development of CS/IS Infrastructure in South Africa</td>
<td>49</td>
</tr>
<tr>
<td>Philip Machanick</td>
<td>The Skills Hierarchy and Curriculum</td>
<td>54</td>
</tr>
<tr>
<td>Theda Thomas</td>
<td>Handling diversity in Information Systems and Computer Science Students: A social Constructivist Perspective</td>
<td>63</td>
</tr>
<tr>
<td>Udo Averweg and G J Erwin</td>
<td>Critical success factors for implementation of Decision support systems</td>
<td>70</td>
</tr>
<tr>
<td>Magda Huisman</td>
<td>A conceptual model for the adoption and use of case technology</td>
<td>78</td>
</tr>
<tr>
<td>Paul S. Licker</td>
<td>A Framework for Information Systems and National Development Research</td>
<td>79</td>
</tr>
<tr>
<td>K. Niki Kunene and Don Petkov</td>
<td>On problem structuring in an Electronic Brainstorming (EBS) environment</td>
<td>89</td>
</tr>
</tbody>
</table>
Derek Smith: Characteristics of high-performing Information Systems Project Managers and Project Teams

Lucas Venêter: INST AP: Experiences in building a multimedia application

Scott Hazelhurst, Anton Fatti, and Andrew Henwood: Binary Decision Diagram Representations of Firewall and Router Access Lists

Andre Joubert and Annelie Jordaan: Hardware System interfacing with Delphi 3 to achieve quality academic integration between the fields of Computer Systems and Software Engineering

Borislav Roussev: Experience with Java in an Advanced Operating Systems Module

Conrad Mueller: A Static Programming Paradigm

Sipho Langa: Management Aspects of Client/Server Computing

T. Nepal and T. Andrew: An Integrated Research Programme in AI applied to Telecommunications at ML Sultan Technikon

Yuri Velinov: Electronic lectures for the mathematical subjects in Computer Science

Philip Machanick: Disk delay lines

D. Petkov and O. Petkova: One way to make better decisions related to IT Outsourcing

Jay van Zyl: Quality Learning, Learning Quality

Matthew O Adigun: A Case for Reuse Technology as a CS/IS Training Infrastructure

Andy Bytheway and Grant Hearn: Academic CS/IS Infrastructure in South Africa: An exploratory stakeholder perspective

Chantel van Niekerk: The Academic Institution and Software Vendor Partnership

Christopher Chalmers: Quality aspects of the development of a rule-based architecture

Rudi Harmse: Managing large programming classes using computer mediated communication and cognitive modelling techniques
Michael Muller: How to gain Quality when developing a Repository Driven User Interface

Elsabe Cloete and Lucas Venter: Reducing Fractal Encoding Complexities

Jean Bilbrough and Ian Sanders: Partial Edge Visibility in Linear Time

Philip Machanick: Design of a scalable Video on Demand architecture

Freddie Janssen: Quality considerations of Real Time access to Multidimensional Matrices

Machiel Kruger and Giel Hattingh: A Partitioning Scheme for Solving the Exact k-item 0-1 Knapsack Problem

Ian Sanders: Non-orthogonal Ray Guarding

Fanie Terblanche and Giel Hattingh: Response surface analysis as a technique for the visualization of linear models and data

Olga Petkova and Dewald Roode: A pluralist systemic framework for the evaluation of factors affecting software development productivity

Peter Warren and Marcel Viljoen: Design patterns for user interfaces

Andre de Waal and Giel Hattingh: Refuting conjectures in first order theories

Edna Randiki: Error analysis in Selected Medical Devices and Information Systems
A PLURALIST SYSTEMIC FRAMEWORK FOR THE EVALUATION OF FACTORS AFFECTING SOFTWARE DEVELOPMENT PRODUCTIVITY

O. Petkova¹, J. D. Roode²
¹Information Systems and Technology - University of Durban Westville
²School of Information Technology - University of Pretoria
opetkova@is.udw.ac.za

Introduction

Past software development productivity research has taken two main directions: in the first, research has concentrated on determining the factors which have a significant effect on productivity; and in the second, the emphasis has been on determining the best way to measure productivity (Maxwell et al, 1996:706). Historically research was aiming mainly to find solutions in the second area while work in the first area was just a complementary one. The belief was that an analytical model may help resolve all problems associated with estimating software productivity. Recently there have been attempts for a new, holistic understanding of the problem. Such an example is the systems dynamics modelling approach for controlling software projects (Abdel-Hamid, 1990). Other studies that stress the importance of better understanding of various factors affecting programmer productivity without considering their role in a particular software cost estimation model are those by Kemayel et al. (1991), Finnie et al. (1993), Maxwell et al. (1996).

In most cases these factors have been analysed with respect to the tuning or validation of a particular metric for cost/effort estimation. These factors are referred to in a different way in different models: in Boehm's model (1981) they are called cost drivers; in Bailey and Basili (1981) -input to the model. Due to their large number, a certain classification of these factors is appropriate. Thus Kemayel et al. (1991) focus their attention on the controllable factors affecting software development productivity. According to them controllable factors are those factors pertaining to the software process, that a typical software manager has the latitude to determine. They investigate 33 controllable factors placed in three groups: factors pertaining to personnel, factors pertaining to the process; and factors pertaining to the user community. Finnie et al. (1993) investigate 18 factors grouped in four groups: technical attributes, project attributes, developer attributes and user attributes. Benbasat and Vessey (1980) discuss 19 factors grouped in 7 classes: organisational operations characteristics, computer hardware characteristics, source language characteristics, programmer characteristics, programming problem characteristics.

The brief look at the lists of the factors in various publications shows that a small number of them evolve with time and the development of technology. Thus it seems inappropriate these days to consider the effect of memory size or storage devices as have been done by Benbasat and Vessey (1980). Another conclusion that comes to mind is the fact that researchers focused their attention on quite different sets of factors and thus it is very difficult to compare results from various surveys.

The purpose of this research is to define a holistic model of the interaction between factors affecting software development productivity. First the relationships between these factors will be discussed and then the methodological foundations and the steps of a framework for the evaluation of these factors within the environment of a particular software project will be examined.

2. Research on the relationships between the factors affecting software development productivity

The factors affecting software development productivity play an important role in the management of the software development process. They influence the way in which the work of software development teams is organised and controlled (Abdel-Hamid, 1996; Boehm & Papaccio, 1988 and others), the evaluation of risk associated with software projects (Madachi, 1996), the quality of software (Herbsleb et al., 1997; Kitchenham & Pfleeger, 1996) and play an important role in the broader field of software measurement (Pfleeger et al., 1997; Baker et al., 1990). Each of these topics is a separate stream of research in the Information Systems literature with specific epistemology and therefore are outside the scope of this research. On the other hand the wide use of factors affecting software development in so many research streams justifies the deeper analysis of how they interact. The research reported

-243
in the nineteenth eighties was dealing with interaction between limited number of factors. The following paragraphs
investigate attempts to modelling the interaction between larger number of factors affecting software development
productivity. Some of them are not narrowly related to software cost estimation models but have wider implications
for software development management.

Most of the IS research reported in the literature is limited to the statistical analysis of cause-effect type of
relationships between two or three factors only. An attempt to capture the complexity of these relationships is
the suggestion by Kok and Kitchenham (1989) to supplement estimation practices with risk analysis, monitoring
and sensitivity analysis. An interesting approach for analysis of certain aspects of software productivity is the
software product value chain model developed in Boehm (1987) on the basis of the value chain model proposed by
Porter. The strategies for improvement of software development productivity suggested by Boehm (1987) and
the value chain model are a very significant contribution to our understanding of software development
productivity. However, it may be noted that the model is a static one and does not reflect the interaction
between such components as infrastructure, human resource management, technology development,
management, operations, etc.

Further attempts to capture the complexity of the relationships between factors affecting software productivity is
the work by Rasch and Tosi (1992). Their research develops an integrated model that permits the assessment of the
relative effects of the level of effort, goal characteristics, and individual characteristics on performance (Rasch and
Tosi, 1992:396). Their model is based on expectancy theory, goal setting theory and on individual characteristics
and in the words of the authors, due to the inherent complexity of integrating three different, yet related lines of
research, their model is limited only to addressing several key elements from each of these concepts. That is the
reason why it does not reflect the broad picture of software development.

The need for multidimensional outputs in measuring software productivity like SLOC supplemented by the number
of routines and lines of comments as well as inputs like hours spent on projects was indicated also in a report on the
use of Data Envelopment Analysis (DEA) for the analysis of productivity of software projects (Mahmood et al.,
1996:70). The results of following that approach can be used to identify efficient and inefficient software projects
and to identify factors that affect software productivity in a positive or negative manner.

An alternative way for modelling the relationships between factors affecting software development productivity was
the application of the Analytic Hierarchy Process by Finnie et al. (1993). It is based on the assumption that these
relationships can be viewed as static at any moment of time and hence can be modelled using the traditional
conventional AHP hierarchy (Saaty, 1994). Though the factors at a particular level of the hierarchy are considered
to be independent of each other, it is possible according to that model that a certain factor is viewed simultaneously
as related to two or more groups of factors from the higher level in the hierarchy. Thus in view of the interaction
between the users and the developers’ team, the factor user computer literacy is viewed both as a user factor and
in another perspective, as a developer attribute. The method that they used involves psychometric measurements
of preferences. At each stage the factors are compared in a pairwise fashion with respect to the root of the cluster
to which they belong in the hierarchy. The above approach can be extended further in the direction of capturing
the richness of the interrelationships between the elements in a particular level of the hierarchy using an extension
of AHP for systems with interdependencies. However both of these approaches suffer from the inherent weaknesses
of traditional multicriteria decision modelling, related to the assumption that it is through the process of working
with the model that the decision makers will gain the deeper understanding into the nature of the factors involved
and their relationships. This leads to the reasons for the search of a better framework for assessment of these factors
attempted in this research.

Other examples of the inclusion of psychometric measurements in analysing programmer productivity are in
Lee (1993), Hanson & Rosinski (1985) and in the following account of simulations of the software development
process using Systems Dynamics pioneered first by Abdel-Hamid (1990). A recent accounts can be found in
Madachy (1996). This approach acknowledges that software development, a dynamic and complex process,
requires new ways of thinking in order to improve the current software environment. However, Systems Dynamics
is only one strand of Systems Thinking and those other strands should not be ignored in view of the tradition of
Critical Systems Thinking (Jackson, 1995). If we acknowledge the complexity of modelling the relationships
between factors affecting software development we will come to the conclusion that this field comes under the area
of "messy problems" as defined first in Ackoff (1979). Then it is a challenge to apply some of the techniques for
handling "messy problems" to the problem of analysing the factors affecting software development productivity.
It is well known that the ability to reproduce results is a significant feature of the scientific approach (Checkland, 1976). On the other hand some researchers like Kemayel et al (1991) correctly note that their results are inevitably affected by the socio-cultural background and the computing tradition in a particular country. This fact relates to another one, that Management Science deals with socio-technical systems. That fact was particularly underlined by Rosenzweig (1994) who pointed out that this feature prevents the internationalization of results in Management Science. In view of the above argument it might be appropriate to stop attempting to replicate results using the existing methods for estimation of software productivity in different environments and to accept that this weakness is inevitable.

Another point to note are the limitations of statistical variance theories in explaining social processes, including the process of systems development. As Robey concludes "...by conceiving of processes as systems of variables, the variance strategy affords little insight into the dynamics of the social processes it purports to explain. While some percentage of variance in one variable may be "explained" by the variation in another, little "explanation" of how and why social events occur is possible (Robey, 1994:443). The same author points out that to compensate for this weakness, process theories have been prescribed as a means of treating social processes as sequences of events that occur over time. He indicates that normal science is not the only, or necessarily best, method of conducting research. According to him, as valuable tools for research as the traditional science approaches are also interpretive research methods that allow novel theoretical insights to be induced from both qualitative and quantitative data (Robey, 1994:442), Galliers, 1992). The community of researchers in information systems have moved beyond the narrow confines of normal science to employ a diversity of disciplinary perspectives and research methods (King, 1993). This is another motivation for the approach in this research.

According to Subramanian and Breslawski(1995:140) there is definitely a need for an estimation approach based on life cycle stage-based estimate break-up, analysis and support that specifically address the requirements, design, coding and testing stages. This stage-based estimation approach would help managers spot some of the reasons or problems that cause delays earlier than usual. This idea can be further extended towards the need for a relatively simple to use framework for evaluation of the factors affecting software development productivity at any stage of the software process. Such a framework should be able to capture the expertise of staff directly involved in a particular project, the richness of the relationships between the factors affecting software development and the different types of quantitative and qualitative variables interacting in the actual circumstances of the project. It should cope with the real constraint of lack of sufficient historical information on software development in many organisations. Such a framework should reflect the particular process, structural, cultural and political aspects of any organisational environment. Such an approach is in line with the conclusion by Bailey and Basili (1981) mentioned earlier that "one software development environment ... [cannot] use the algorithms developed at another environment to predict resource consumption".

On the basis of the analysis of the research in software metrics we can conclude that an investigation of factors affecting software development productivity is a complex problem, with many intertwined subproblems and as such can be classified as a “messy” problem. It is an important and worthwhile problem both from a theoretical and a practical point of view. A framework is needed for its analysis that is easy to use, allows for the incorporation of all factors in a holistic way and incorporates available quantitative data on a particular project environment as well as the expertise of those involved in its management and who have to make the estimates of productivity / effort / cost. It is necessary therefore, to investigate what approaches are suitable to handle this kind of complex problems.

3. A Review of Attempts to Use Emancipatory Ideas in Information Systems Development

Klein et al. (1991:1) make a plea for methodological pluralism in Information Systems research, which in their opinion is a better strategy than the traditional alternatives of supremacy, contingency, eclecticism and dialectism. After discussing pluralism and these four research strategies, the same authors conclude that in practice the differences between moderate advocates of pluralism, radical contingency and eclecticism tend not to be very large and they use pluralism in a broad sense to include eclectism and radical contingency as special forms of pluralism (Klein et al., 1991:9). This view is less coherent than that of Jackson (1991) with respect to pluralism, which is understood by him as a complementarist application of a variety of systems methods within the framework of the
System of Systems Methodologies. Historically, critical social theory was applied to Information Systems development before Critical Systems Thinking and therefore it will be discussed here first.

A summary of the application of Critical Social Theory to ISR prior to 1990 is presented in made by Ngwenyama (1991:273). Apart from the concise, but articulate presentation of the ideas of Critical Social Theory, Ngwenyama presents an Action Science Research Strategy, based on the ideas of Argyris. He considers it an exemplar of critical social science because it shares similar core assumptions: (1) critique of the status quo and a search for alternatives to it; (2) collaborative action for learning and fundamental change; (3) free and open participation by individuals in the creation of their social world; (4) critical self reflection as a methodology for improving self awareness and transformation (Ngwenyama, 1991:274). Action science is one strand of Action research(Flood and Romm, 1996:141). Its similarities to the emancipatory ideas of critical thinking were also pointed by Levin (1994) who concludes that Action Research and Critical Systems Thinking are two icons carved out of the same log, in spite of their independent development. While Ngwenyama reveals the potential of the Action Science approach to capture a number of parameters of IS development that cannot be reflected easily in a technical specification, he points out that more work is needed on implementing an action science methodology. In addition he suggests that the critical theory information systems research programme could learn a great deal from dialogues with the Soft Systems Approach and Trade Union Perspective (Ngwenyama, 1991:277).

In his seminal paper on the links between Critical Systems Thinking and IS research, Jackson (1992) provides a thorough survey of the critical literature related to ISR. He demonstrates that IS researchers have, at one time or another, shown an interest in each of the issues under consideration in that paper (Jackson, 1992:84). The contribution of that publication is that it demonstrates the power of an integrated approach in the ISR field and indicates that the five commitments of critical systems thinking, as discussed earlier, can assist the development of an integrated programme for critical thinking in IS research.

In spite of the effort by Jackson (1992), there have been relatively few publications subsequently on the practical application of Critical Systems Thinking in Information Systems Research. A thorough operationalisation of the ideas of Total Systems Intervention to IS development is presented in Clarke and Lehaney (1996). Its contribution is in the identification of relevant methodologies that may be used in ISD: Systems Design/Systems Engineering, Systems Development Life Cycle Approaches, Strategic Options Development and Analysis, Strategic Assumptions Surfacing and Testing, Interactive planning, Soft Systems Methodology and Critical Systems Heuristics (Clarke and Lehaney, 1996:361). Another contribution of that paper is the report on the successful practical implementation of a TSI based intervention through the identification of a clear perception of the user needs in the design of a new student records management information system. However it does not question the foundations of TSI in light of the critique to it in the past four years from certain circles, the most vocal of which has been the postmodernist trend (see Taket and White,1996). It seems therefore appropriate for the IS community to explore postmodernism on a broader scale in future research.

4. Recent ideas on pluralism in Management Science

The analysis of the literature indicated that at least in two influential developments in Operations Research and Systems Thinking, viz, Total Systems Intervention by Flood and Jackson and Pragmatic Pluralism by Taket and White the issue of methodological pluralism played a central role. The general nature and history of pluralism in Management Science are presented in a very systematic way in Jackson(1997) and for space reasons will not be discussed here. The first question that arises is why there is a tendency to embrace pluralism in Management Science. According to Jackson (1997:1) there are three reasons which explain why the climate is right. One is the critique of many of the traditional approaches; a second reason is the prevailing fashion for "relativism", preceding postmodern thinking but now usually associated with it; and the third reason, crucial for practitioners, is that pluralism seems to be necessary. One recent approach, named Multimethodology, was recently suggested by Mingers (Mingers,1996:7). It is based on ideas from the Critical Discourse as described by Deetz(1996). It stresses the importance of the historical constitution of situations. It pays attention to the discourse - power/knowledge and relies on agents in implementing the intervention. The relationships between these three groups of factors are reflected in a checklist of questions. Certain similarities can be identified between these questions and the Critically Heuristic Boundary Questions of Ulrich(1996), since they provide the foundation for understanding of the values
of those involved in the decision making process.

Mingers and Brockelsby (1996) distinguish three possibilities in pluralism: the first is methodology selection - that methodology, and its associated methods, models and techniques, which best corresponds to the demand of the problem situation will be selected. The second alternative is called “Whole methodology management”. Methodologies are used together, in the same intervention. The third version of pluralism is using parts of different methodologies in the same intervention. Various techniques and methods are combined in accordance to the particular problem situation. Mingers (1996) calls this approach “Multimethodology”. It involves three types of systems which are used in combination as a platform for successful problem solving (Mingers, 1996:7):

- the problem context system, representing the real world situation of concern and associated with the historical constitution of situation;
- the intellectual resource system, representing the available theories and methodologies and associated with the constitution of discourse-power/knowledge;
- the intervention system, involving agents undertaking the intervention and associated with the constitution of the subject - technologies of self.

The three systems described above are in constant interaction with each other, and there are a number of recursive relationships within them. The Multimethodology suggested by Mingers has its roots in Critical Systems Thinking, and as such can be viewed as the third major effort to develop a methodology in the area (the other two being Critical Systems Heuristics (Ulrych, 1996) and Total Systems Intervention (Flood and Jackson, 1991)). However it has certain positive features like the consideration of the historical constitution of the organisation and the use of the notion of agents. The theory of agency provides to the critical discourse an additional activist tone (Deetz, 1996:202). The agents are both self conscious, evaluative subjects, and the result of historical processes of constitution (Mingers, 1996:11). According to Mingers (1996) the criterion for choice has three dimensions that are along the Habermas tradition: pragmatic, ethical and moral. While it is not possible to provide answers to satisfy each dimension directly, the goal is to try to set a framework within which they can be rationally debated (Mingers, 1996).

There are other differences between Multimethodology and TSI. Multimethodology still pursues the ideals of Critical Social Theory but without declaring that so openly, probably in line with the postmodernist rejection of grand narratives. For similar reasons it seems to take a much more practical approach regarding the suitability of particular techniques or methods, recognising that parts of a methodology can be combined with other methodologies in an intervention. As mentioned earlier, TSI assumes that we can apply whole methodologies only. However recently Jackson points that “in TSI an explicit choice of dominant methodology is made to run an intervention, with dependent methodologies, reflecting alternative paradigms, in the background. The relationship between dominant and dependent methodologies is, however, allowed to change as the intervention proceeds in order to maintain flexibility at the methodology level” (Jackson, 1997:6).

The proposed Multimethodology by Mingers (1996) seems to be an attractive vehicle for research in Systems Thinking as well as in the field of Information Systems Research. It has to be considered however in conjunction with the safeguards suggested by Jackson to prevent relapse from pluralism into pragmatism or methodological imperialism (Jackson, 1997:2). These are defined in the same source as the following requirements towards pluralism in management science:

- the first requirement is that pluralism must encourage flexibility in use of the widest variety of methods, models, tools and techniques in any intervention;
- a second requirement is that methodologies owing allegiance to different paradigms should be employed in the same intervention unless good reasons are given for temporary relapse into methodological imperialism;
- a third requirement of pluralism follows from the need for paradigm diversity. According to it a pluralist must learn to live with and manage a degree of paradigm incompatibility.

Analysing Jackson (1997) one concludes that there is a considerable difference between its spirit and the one advocated by Flood and Jackson (1991) in their first version of TSI, as well as by Flood (1995) in his extension of TSI. The first difference is in the rejection of the idea of “grand narratives”. On this issue Jackson (1997) differs from Mingers (1997) as the latter replaces Habermas’s theory of constitutive interests, which was seen by Flood and Jackson (1991) as the basis of TSI, with a more recent framework based on Habermas’s three worlds: the
material world, our social world and my personal world (Habermas (1984, 1987) and also Mingers (1997:9-11)). The second major difference is that the metamethodology for managing pluralism in management science should not operate above the specific methodologies as TSI but rather as a facilitator of a reflective conversation between methodologies. The operationalisation of the latter idea is still an open question according to Jackson (1997). One possibility for this is indicated by Jackson in terms of one feature of TSI that might be applicable separated from the other characteristics of TSI that are seen as no longer acceptable after its criticisms on the basis of its practicality, ease of use and some postmodernist assumptions. This is the idea that in TSI an explicit choice of a “dominant” methodology is made to run an intervention with “dependent” methodologies, reflecting alternative paradigms, in the background with the possibility to change the relationship between dominant and dependent methodologies (Jackson, 1997:6).

Following the preceding analysis of the problem under concern in this research, and of the different strands of Systems Thinking one can identify as potential candidates for inclusion in the required framework for evaluation of factors affecting software development productivity certain elements from the Analytic Hierarchy process (AHP), Soft Systems Methodology, Churchman's Social Systems Design and Strategic Assumptions Surfacing and Testing.

5. A Framework for the evaluation of factors affecting software development productivity

The first task associated with the problem is to identify those who have a role to play. As a suitable tool for this purpose the Strategic Assumptions Surfacing and Testing (SAST) approach by Mason and Mitroff is usually considered (Jackson, 1991). One has to take into account the large number of stakeholders involved in case of a complex software project. This raises the practical difficulty of applying the SAST procedure consistently in all its stages since all parties may meet on only a few occasions during the whole process, while the participating stakeholders may not be classified as adversaries and their debate may not always be a dialectic one. In such cases the brainstorming component as a creativity-supporting technique can be employed. Then the two dimensional rating of issues according to their likelihood and urgency as suggested originally by Mason and Mitroff can be applied to evaluate the seriousness of the particular issue.

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) by Checkland are included in the framework. SSM was chosen as it promotes organisational learning through encouraging employee participation in problem solving (Checkland and Scholes, 1980). The elements of SSM that are considered here are rich pictures and CATWOE analysis. These are used in the three forms suggested in the revised form of SSM, technical analysis of the management intervention, social or cultural analysis and political analysis of the problem area (Checkland and Scholes, 1991).

There are certain criticisms leveled at SSM related to the danger of lack of clear direction for further action that sometimes is experienced after a SSM session. For this reason, it is suggested that the SSM analysis of a problem situation is supplemented with a multi-criteria decision making approach like the Analytic Hierarchy Process (AHP) by T. Saaty (Saaty, 1994). By supporting the choice phase of decision making, the AHP allows stakeholders to evaluate the appropriate alternatives and choose the one most appropriate. They can vary the priorities of the criteria in the upper levels of the value tree and thus explore the influence of various management strategies on the importance of various factors affecting software development productivity. The purpose of the framework for evaluation of factors affecting software development productivity is to provide a deeper understanding within the development team of the forces affecting a particular software development project within a particular environment. It can be summarised as follows:

- Identification of the stakeholders
- Delimitation of the goals and boundaries of the software development problem
- Generation of a systems model of the problem in terms of a rich picture
- CATWOE analysis of the problem
Some details on the stages are listed below:

- Idea generation on issues related to the technical, cultural and political analysis of the software project. Rating of the issues according to their importance for the purpose of identification of key management issues.

- Prioritization of the factors affecting software development productivity in the particular project for the purposes of identification of the those that can contribute most significantly to management improvement.

At various stages of the above process the facilitator may go back to a particular step for the purpose of feedback or iteration of a previous formulation.

Some details on the stages are listed below:

- Identification of the stakeholders: Specify those groups that have a vested interest in the particular software project under concern.

- Generation of a systems model of the problem in term of a rich picture: The purpose of the rich picture is to provide information on the structure of the factors affecting the problem, and also of any processes that may be related to it.

- CATWOE analysis of the problem: CATWOE is a mnemonic associated with an analysis technique in Soft Systems Methodology of P.Checkland (1991). For the particular project the group of experts needs to specify:
  - Customers: the victims/beneficiaries of the purposeful activity.
  - Actors: those who are involved in the activities.
  - Transformation process: the purposeful activity which transforms an input into an output.
  - Weltanschauung: the view of the world that makes the root definition meaningful in context.
  - Owners: who can stop the activity.
  - Environmental constraints

- Delimitation of the root definition of the software project: Define the core purpose of this project, identifying the primary tasks and other issue based tasks.

- Idea generation on issues related to the technical, cultural and political analysis of the software project. Use brainstorming to generate:
  - Ideas on the general (technical) side of the process of development of this particular project.
  - Ideas related to the cultural analysis of the software project. These concern:
    - various roles,
    - various norms and
    - various values of the stakeholders in the project.
  - Ideas related to the political analysis of the software development project, revealing power relations and processes in which differing interests reach accommodation.

- Rate the generated ideas in the previous stage according to their importance for the purpose of identification of key management issues.

- Prioritization of the factors affecting software development productivity in the particular project for the purposes of identification of the those that can contribute most significantly to management improvement. This stage involves a pairwise comparison process between the factors using a multicriteria decision making technique called The Analytic Hierarchy Process. It is a way of measurement of the qualitative and quantitative knowledge that the members of the group have about the various factors.

The framework deals with systemic issues affecting software development productivity and their prioritisation. The findings of the brainstorming sessions and stakeholder identification, as well as the applied elements of SSM help in the problem formulation stage of AHP, which is assumed to be the least formalised stage of AHP as there are no strict rules for it published in the literature. The framework presents a synthesis of multicriteria decision making and problem structuring techniques.
Conclusion

The above framework combines elements from two systems approaches with an MCDM method. This is justified by the recently proposed pluralistic meta methodology for solving of complex, messy management problems, called Multimethodology (Mingers and Gill, 1997). This allows methods, models and techniques as parts of different methodologies, from different paradigms, to be brought together according to the requirements of a particular intervention. Early results from the application of this framework for evaluation of factors affecting software development productivity to two different, but interrelated software projects, one in the domain of manufacturing automatic control systems and the other in production scheduling, are encouraging. These are not shown here for space reasons. Following Jackson’s call for the need for a coherent pluralism in Management Science within the framework of Critical Systems Practice (Jackson, 1997), this framework aims at the full realisation of the potential of the actors involved in the software project in a way serving interests other than the status quo. This is particularly relevant for the company where this framework was applied on an experimental basis as it is undergoing a two billion rand extension accompanied by complete reengineering of its industrial processes. Such an approach is one of the preconditions for successful business transformation and reengineering as a result of the implementation of any large software project.

References


