

1-1-1995

Experiences in the use of the structured systems analysis and design methodology (SSADM) in a service organisation

Andre Bavington
Edith Cowan University

Follow this and additional works at: <https://ro.ecu.edu.au/theses>



Part of the [Business Commons](#)

Recommended Citation

Bavington, A. (1995). *Experiences in the use of the structured systems analysis and design methodology (SSADM) in a service organisation*. Edith Cowan University. Retrieved from <https://ro.ecu.edu.au/theses/1175>

This Thesis is posted at Research Online.
<https://ro.ecu.edu.au/theses/1175>

Edith Cowan University

Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study.

The University does not authorize you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following:

- Copyright owners are entitled to take legal action against persons who infringe their copyright.
- A reproduction of material that is protected by copyright may be a copyright infringement. Where the reproduction of such material is done without attribution of authorship, with false attribution of authorship or the authorship is treated in a derogatory manner, this may be a breach of the author's moral rights contained in Part IX of the Copyright Act 1968 (Cth).
- Courts have the power to impose a wide range of civil and criminal sanctions for infringement of copyright, infringement of moral rights and other offences under the Copyright Act 1968 (Cth). Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

**EXPERIENCES IN THE USE OF THE STRUCTURED
SYSTEMS ANALYSIS AND DESIGN METHODOLOGY (SSADM)
IN A SERVICE ORGANISATION**

BY

Andre Bavington B.AppSc (Info Sc).

**A Thesis Submitted in Partial Fulfilment of the
Requirements for the Award of**

Masters of Business (Information Systems)

at the Faculty of Business, Edith Cowan University

Date of Submission: 25.8.95

Abstract

The purpose of this research is to examine the historical development of an information systems methodology in a large West Australian organisation and to detail the experiences and satisfaction levels of information systems practitioners' in using the methodology. The methodology under examination is SSADM (Structured Systems Analysis and Design Methodology) in a service organisation in which I am an employee.

The participants of this research are involved in the systems development process as Systems Analysts, Project Leaders, Programmers, Project Manager, and development review and consulting personnel.

There were 26 participants involved in this research study ranging in age from 25 to 55 years old.

This research is an ethnographic, genealogical study. As most of the objectives of the research relate to the behaviour, world views and values of individuals and the culture and politics of the organisation, a qualitative approach is most suitable. The study has drawn upon interviews conducted with staff, personal observations during the study, informal communication as a participant and a range of formal interactions and communications. An on-line diary was used to record appointment times with the interviewees, and field notes were made to document my thoughts, impressions and information as it came to hand. When permission was granted by the interviewees, a tape recorder was used to record the interview.

The result of the research is a description of the development, use and success of the information systems development methodology (SSADM) in one organisation. The research is of interest to the following groups: Information Systems practitioners in the organisation under study, other Information Systems professionals, and academics.

Declaration

I certify that this thesis does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any institution of higher education; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Signature

Date 31-8-1995

Acknowledgments

There have been several people who have offered their support and assistance during the preparation of this thesis. With gratitude I acknowledge their help.

The enthusiasm of Mr Craig Standing, the academic supervisor for the research, provided me with the inspiration to carry out the research. His advice, encouragement and thoroughness were greatly appreciated.

I thank the staff in the organisation under study who participated in the interviews, that gave their time although it was clear that they were really extremely busy people. Without them willingly giving their time and sharing their experiences, this research would not have been possible.

To my family, all of whom deserve special thanks. They always believed that I would make it and provided me with the emotional support and encouragement during every stage of this project.

Table of Contents

| | Page |
|--|------|
| Abstract | ii |
| Declaration | iv |
| Acknowledgments | v |
| List Of Tables | ix |
| List Of Figures | x |
| Chapter | |
| 1 INTRODUCTION | 1 |
| Background To The Study | 1 |
| Significance Of The Study | 2 |
| Purpose Of The Study | 3 |
| Research Questions | 4 |
| Subsidiary Questions | 4 |
| Definition Of Terms | 6 |
| 2 LITERATURE REVIEW | 10 |
| Studies Similar To The Current Study | 10 |
| Literature On Previous Findings | 11 |
| Metaphor | 23 |
| Metaphors and Methodologies | 26 |
| Magic | 32 |
| Recognising The Politics of MIS | 33 |
| Productivity Tools | 40 |
| CASE Tools | 43 |
| Identifying Organisational Culture Clash In MIS Implementation | 51 |
| Literature On The Research Methodology | 79 |
| 3 THEORETICAL FRAMEWORK | 81 |

| | | |
|---|--|-----|
| 4 | METHOD | 83 |
| | Sample / Subjects | 83 |
| | Research Design | 84 |
| | Instruments / Equipment | 85 |
| | Procedure | 87 |
| | Data Analysis | 89 |
| | Limitations Of The Study | 92 |
| 5 | ETHICAL CONSIDERATIONS | 93 |
| 6 | RESULTS | 94 |
| | Introduction | 94 |
| | An Historical Perspective | 95 |
| | Work and Educational Background of Employees | 99 |
| | The Selection Of The Methodology | 101 |
| | Systematic Forms of Control: Socialisation | 107 |
| | The Use Of Prototyping | 108 |
| | Satisfaction With SSADM | 111 |
| | Employee Career Paths | 113 |
| | Key Factors To Career Progression | 115 |
| | SSADM And Future Careers | 117 |
| | The Role Of SSADM In Career Progression | 119 |
| | SSADM and Its Influences On Opportunities | 122 |
| | Work Preferences | 124 |
| | Interpersonal Skills | 126 |
| | Satisfaction With CASE | 130 |
| | Problems With Using SSADM | 132 |
| | Making SSADM Successful | 135 |
| | Tools and Techniques Used | 138 |
| | Tailoring SSADM | 140 |

| | | |
|----|--|-----|
| 7 | DISCUSSION | 142 |
| | Implications For Education and Training | 143 |
| | Implications For Methodology Selection | 147 |
| | Implications For Prototyping | 149 |
| | Implications For Satisfaction With SSADM | 151 |
| | Implications For Employee Career Paths | 152 |
| | Implication For Career Progression | 154 |
| | Implications For SSADM and Future Careers | 155 |
| | Implication Of The Role Of SSADM In Career Progression | 157 |
| | Implication Of SSADM and Its Influences On Opportunity | 158 |
| | Implications For Work Preferences | 159 |
| | Implication Of Interpersonal Skills | 161 |
| | Implications Of The Satisfaction With CASE | 162 |
| | Implications Regarding Problems With Using SSADM | 164 |
| | Implications For Making SSADM Successful | 166 |
| | Tools and Techniques Used | 167 |
| | Tailoring SSADM | 168 |
| | SSADM's Impact On Staff and The Organisation | 169 |
| 8 | CONCLUSION | 172 |
| 9 | APPENDICES | 176 |
| | Interview Questions | 176 |
| 10 | REFERENCES | 179 |

List of Tables

| | | |
|---------|--|----|
| Table 1 | Widely Used System Life Cycle Methods | 17 |
| Table 2 | Metaphors and Their Corresponding Methodologies | 29 |
| Table 3 | The Games | 36 |
| Table 4 | Seven Dimensions Of Cultural Differences | 54 |
| Table 5 | The Risk Grid | 57 |
| Table 6 | Theories of Resistance: Underlying Assumptions | 59 |
| Table 7 | Theories of Resistance: Recommendations For Implementation | 63 |
| Table 8 | Scenario Writing | 64 |
| Table 9 | The Four Stage Model Of Professional Careers | 70 |

x

List of Figures

Figure 1.

A rich picture of the selection of the SSADM methodology and its impact on careers, developers, end users and organisation culture.

171

INTRODUCTION

Background To The Study

The process by which systems development staff develop computer-based information systems has changed significantly over the last fifteen to twenty years. During the early 1970's, ad-hoc methods were used to analyse and design computer based information systems. According to Hawryszkiewicz (1988) during the 1970's:

many new ideas were introduced to overcome problems associated with ad-hoc methods. One of these ideas was the use of modelling techniques instead of natural language for describing systems. Another idea was the distinction which was made between logical and physical analysis and design. A third idea was the introduction of a structured way of moving from a description of user needs to a working system. (p. xi)

With increased emphasis being placed by management and owners of computer systems on productivity and performance in the marketplace, use of and satisfaction with, systems analysis and design tools becomes increasingly important.

During this period many different methodologies and tools have been used to develop computer-based information systems. While the amount of improvement in hardware technology has resulted in lower cost and increased capabilities, Necco, Gordon, and Tsai (1987) found that this has highlighted the need to improve the software development process.

SSADM over the last ten years has become the standard systems development methodology used by central government bodies in the United Kingdom. Over that time it has experienced several revisions with version 4 being released in 1990. The reason behind its development was to have a standard systems development methodology that could be used across all government departments and by external organisations that were developing computer systems for the various departments. Ashworth (1988, p. 153) found that "SSADM has also been adopted as a standard by public utilities, local government, health authorities, foreign governments and several large private sector organisations".

Significance Of The Study

The research will be of interest to the following groups:

- a) Information Systems practitioners in the organisation under study. They can see the evolutionary process of introducing and embedding a methodology in a more holistic sense and can see their roles more clearly and how their views and practices relate to those of their colleagues;
- b) Information systems practitioners from other organisations who are using SSADM or who are considering introducing it. They can look for the problems and successes associated with the methodology and can draw their own conclusions as to the relevance of aspects of the study and its findings to their own world. It will give

these organisations a chance to learn from the experiences of Information Systems (IS) practitioners in relation to the strengths, weaknesses, benefit and problems associated with implementing and using such a methodology.

- c) Academics will find the study forms a useful insight into the influence of the social aspects of an organisation upon the introduction and development of a hard systems methodology.

Purpose Of The Study

The purpose of this research is to examine the detailed historical development of an information systems methodology in a large Western Australian organisation and to detail the experiences and satisfaction levels of information systems practitioners in using the methodology. The methodology under examination is SSADM (Structured Systems Analysis and Design Methodology) in a service organisation in which I am an employee. The study will identify what the current practices are so that the degree of usage of the various tools can be determined and their associated level of satisfaction. Within these aims the research will explore how successful the methodology is from a practitioner's perspective in relation to a project:

- . being on time,
- . being within budget,

- meeting the user's requirements,
- maintainability, and
- ease of use.

Research Questions

This research thesis will attempt to answer two major research questions that may include the following:

- Which SSADM systems analysis and design tools and techniques do Information Systems (IS) professionals use in the organisation?
- How do the culture, the politics and the myths and metaphors of the organisation affect the development and success of the SSADM methodology?

Subsidiary Questions

Besides identifying the interviewees' experiences with the formal techniques and the impact that culture, politics, myths and metaphors have had on the organisation (social aspects), this research may also explore the following subsidiary research questions to discover their significance in relation to the major research questions:

- . How satisfied are they with the tools and techniques they are using?
- . Does their use or satisfaction vary from small systems to large systems?
- . What impact are automated tools (CASE - Computer Aided Software Engineering) having on SSADM?
- . Which tools and techniques are being used on the projects that have been successful?
- . Are certain tools and techniques within SSADM more applicable to a particular type of computer system under development?
- . What do the practitioners consider being the strengths and weaknesses of the methodology and techniques?
- . Do the world views, philosophies and characteristics of the staff affect the development and success of the methodology and influence which techniques are used in this organisation, and if so, how?
- . What do the IS personnel in this organisation consider the key factors in developing improved computer-based information systems in relation to SSADM to be?

Definition Of Terms

- Artefacts:** "Are those visible and familiar aspects of culture that a person hears, sees, or feels upon entering an organisation." (Romm, Pliskin, Weber & Lee, 1991, p. 102).
- Characteristics:** Distinguishing traits or qualities that make one person different from another.
- Culture:** The total intellectual and institutional heritage of an organisation.
- Ethnography:** "Asks the researcher, as far as possible, to share first-hand the environment, problems, background, language, rituals and social relations of a more-or-less bounded and specified group of people. The belief is that by means of such sharing, a rich, concrete, complex and hence truthful account of the social world being studied is possible". (Van Maanen (1988) is quoted by Preston (1991, p. 46)).
- Games:** "Ongoing series of complementary ulterior transactions, progressing to a well defined, predictable outcome". (Berne (1981) is quoted by Grover, Lederer & Sabherwal (1988, p. 147)).

- Genealogy:** "A questioning of our contemporarily received notions by a demonstration of their historical emergence. The point of history in this sense is to make intelligible ways in which we think today by reminding us of its conditions of formation". (Miller & O'Leary (1987) is quoted by Preston (1991, p. 44)).
- Magic:** "The superstitious or religious, as opposed to the scientific method used to control nature for a definite practical end, particularly to aid the functioning, binding, and survival of a society". (Earl (1983) is quoted by Hirschheim & Newman (1991, p. 39)).
- Metaphors:** "A way of thinking and a way of seeing that pervade how we understand our world generally". (Morgan (1986) is quoted by Hirschheim et al. (1991, p. 37)).
- Methodology:** "Is an organised and systematic approach for handling the system life cycle or its parts. It will specify the individual tasks and their sequence." (Palvia & Nosek (1993, p. 74)).
- Myths:** "A dramatic narrative of imagined events, usually used to explain origins or transformations of something. Also, an unquestioned belief about the practical benefits of certain techniques and behaviours that

is not supported by demonstrated facts". (Trice & Beyer (1984) is quoted by Hirschheim et al. (1991, p. 34)).

- Philosophies:** The way people think, reason and interpret knowledge.
- Prototype:** "A software prototype is a model of a system, or part of a system, created to illustrate how a proposed system will work, demonstrate the developers understanding of the user's requirements and to solicit feedback from the users." (Doke, 1990, p. 170).
- Resistance:** "Behaviours intended to prevent the implementation or use of a system or to prevent system designers from achieving their objectives. Where one individual's use of a system is not critical to the operation of a system, that individual's choice not to use the system cannot be considered resistance". (Markus, 1983, p. 433).
- Symbol:** A sign that "denotes something greater than itself". (Morgan, Frost, Pondy and Dandridge (1983) is quoted by Hirschheim et al. (1991, p. 33)).
- Technique:** "Is a means of accomplishing a task in the system life cycle (SLC). Sometimes, it may become synonymous with the task." (Palvia et al.

(1993, p. 74)).

Tool: "Is a computer software package to support one or more techniques."
(Palvia et al. (1993, p. 74)).

World views: Also, called Weltanschauung. "In CATWOE the (unquestioned) image or model of the world which makes this particular human activity system (with its particular transformation process) a meaningful one to consider." (Checkland (1981, p. 319)).

LITERATURE REVIEW

Studies Similar To The Current Study

There has been little previous research similar to this study done on the experiences in the use of SSADM by organisations. The only previous research found by the author, was conducted by Edwards, Thompson and Smith (1989a, 1989b). They carried out a series of case studies on both first time and experienced users in separate case studies. These studies were done on both government and commercial organisations using version 3 of SSADM. Since the release of version 4 of SSADM there has not been any research on its use. The two previous case studies differed from this proposed study in that they were conducted on multiple sites, where as this study will be on only one organisation and cover the use of the SSADM methodology in greater detail.

The previous case studies were based on a series of in-depth interviews with practitioners, both first time and experienced users. They described the interview cases using their own words rather than quoting the words of the interviewees. This could have resulted in a certain amount of personal bias being introduced into the explanation by the interviewers since they did not use quotes from the interviewees.

The case studies on the first time users were carried out on six organisations (Edwards, Thompson & Smith (1989a)), while the case studies on the experienced SSADM users were carried out on four organisations. These case studies presented their findings based on a framework of headings that included:

- . background,
- . analysis,
- . logical design,
- . physical design,
- . program specification and program design,
- . automation, and
- . assessment of SSADM.

They were not looking at the human-technical interface to the same extent as my study.

Literature On Previous Findings

According to Edwards, Thompson and Smith (1989b), the two previous case studies conducted on the use of SSADM found:

that views expressed by the first-time users did not differ significantly from those of the experienced users. The main difference between the two groups in the use of the method was the confidence with which the users felt able to tailor the method and evaluate its techniques and steps for usefulness for

each particular project. (p. 426)

These two case studies found that for SSADM to be successfully used the following were required:

- must have commitment by senior management to ensure adequate resources are available for the practitioners,
- while the logical design should contain the necessary features for the system, the design team should not aim to have perfection,
- designers should consider logical data structure, report formats and screen formats and not just the entity life histories when developing the process outlines,
- physical designers should have some knowledge of the business,
- access to specialist staff such as the data administrator is necessary for producing good physical designs,
- according to Edwards, Thompson and Smith (1989b, p. 426), "the physical design stage is the area that creates most problems for the systems designers", this was due mainly to a lack of detail in the guidelines,
- program specifications are important to the development of programs and system testing,
- sites that used both SSADM and JSP (Jackson Structured Programming) found JSP useful for both the on-line and batch program/modules,
- automated aids were considered as a necessary resource for the successful use of SSADM, especially during the design process,

- . the use of a mainframe data dictionary linked to the analysis and design workbench was found beneficial to projects,
- . the major weakness of the method was the physical stage,
- . systems development teams should use their own judgement when modifying the method and the reordering and omission of tasks should be carried out in a planned and authorised manner.

Edwards, Thompson and Smith (1989c, p. 21) also presented results of a survey conducted in 1987 on the use of SSADM in commercial and government sectors in the United Kingdom. The purpose of this survey was:

- . to find out the degree to which the users of SSADM felt that the requirements and principles of the methodology had been met,
- . "to determine specifically whether practitioners would welcome the development of a formalised interface between the SSADM and JSP methodologies" (Edwards, Thompson and Smith (1989c, p. 21)),
- . to find out if SSADM will be a highly used methodology in the future,
- . to find out areas for further research.

Of the 310 organisations contacted only 72 or 23% returned forms that contained responses suitable for analysis. Of those organisations that responded, 68.1% used SSADM, 26.4% used LSDM (Learmonth Structured Design Methodology) and 5.6% used other variations of SSADM. Most of the organisations were civil service organisations (58%) using ICL computers, COBOL (64%) and 4GL (51%) programming languages and the IDMS database management system (53%). The automated tool that received the greatest use was Automate (Plus) (76%). SSADM was used mainly on projects that involved Commercial (DP) (70%), and Database (31%). SSADM techniques such as Data Flow Diagrams, Logical Data Structures, Relational Data Analysis were found by 59, 54 and 42 respondents respectively to be effective.

According to Edwards, Thompson, & Smith (1989c, p. 28) the major findings to this research survey include the following:

- . the SSADM methodology provides rules and guidelines for systems development, it does not however eliminate the need for good systems analysts,
- . adequate logical designs are produced by the analysis and logical design stages,
- . further improvements to the techniques in the physical design control stage are needed to produce standard products for systems implementation,
- . "the use of automated aids with SSADM is beneficial and is to be recommended",
- . more extensive support in the production and maintenance of SSADM products are needed from automated tools,
- . "the implementation of systems using third generation languages (3GLs), in

particular COBOL, is still very common, despite the current popularity of 4GLs and application generators",

"the development of a practical interface between SSADM and systems implementation using JSP/SDM is a sensible goal".

Several researchers have attempted to compare a few systems development methods.

These include Palvia and Nosek (1993).

Palvia et al. (1993) collected field data from MIS professionals working in North Eastern and Mid South metropolitan areas of the United States in different types of organisations. Their research attempted to:

- . Identify a comprehensive set of available methodologies and techniques for developing Information Systems.
- . Identify what techniques and methodologies are in use and their level of use.
- . For those techniques and methodologies in use, identify the perceived value of their attributes.
- . Evaluate the applicability and utility of the techniques and methodologies during the various phases of the system life cycle.
- . Evaluate the utility and applicability of the techniques and methodologies for different types of systems.
- . Assess the applicability and utility of the methodology and techniques for both structured and unstructured system problems.

Of the 300 questionnaires mailed to IS professionals, only 65 were returned, a 21.7% rate of return. The 65 returned responses cast some doubt about whether any useful conclusions could be drawn from such a small sample. The study found that there were seven methods used by more than half of the IS professionals. Many methods were rarely used. Half the methods extracted from the literature, ie. Twenty-six methods, were used by less than 10% of the respondents; thirteen were not used at all (ie. ADS System (by NCR), BIAIT (Bus. Info. Anal. and Integration Technique), BICS (Business Info. Characterisation Study), SREM (Sys. Req. Engg.) Methodology (by Mack Alford), SOP (IBM's Study Organisation Plan), automated ADS, Information Algebra, Young/Kent methodology, Langefors methodology, Information Engineering, SADT, PLEXSYS (Workbench Approach), and ISDOS (University of Michigan Approach)).

The interesting point for this study was that while SSADM may be widely used in the UK, it did not rate a mention on the list of methodologies. While Jacksons Systems Development did receive a mention only 1.5% occasionally used it.

Table 1
Widely used system life cycle methods.

| Method | Percent Using |
|------------------------|---------------|
| System flow chart | 70.8 |
| HIPO chart | 18.5 |
| Data flow diagram | 53.9 |
| Decision table | 29.2 |
| Functional description | 61.6 |
| Structure chart | 15.4 |
| Data base dictionary | 53.9 |
| Program flow diagram | 63.1 |
| Pseudocode | 58.5 |
| Structured English | 29.2 |
| Structured programming | 69.2 |
| Test data generator | 16.9 |
| Structured testing | 27.6 |
| Structured walkthrough | 38.5 |
| Prototyping | 32.3 |
| ASDM | 21.5 |

(Palvia et al., 1993, p. 78)

Based on the results of the study Palvia et al. (1993) offered the following normative model:

- a) First select the overall methodology using mainly the system type and problem structure. This may be influenced, to a slight extent, by the needed attributes.

- b) At each phase of system development, select appropriate techniques based on a contingency analysis. There does appear however, that an essential collection of four to five techniques may be satisfactory for most organisations. Eg.

ANALYSIS

- System flow chart
- HIPO
- Data flow diagram
- Decision table
- Functional description
- Structured English

DESIGN

- Structure chart
- HIPO
- Pseudocode
- Program flow diagram
- Decision table

Hirschheim et al. (1991) examined the influence of myth, metaphor, and magic as a means to facilitate a much richer understanding of systems development. Mowshowitz (1976) is quoted by Hirschheim et al. (1991, p. 29) as saying that "many, if not most, information systems are failures in one sense or another." Gladden (1982) also quoted by Hirschheim et al. (1991, p. 30), reported some worrisome figures, saying that "75% of all the cases of systems development he surveyed, either the development is never completed or the product of completed development is not used". It is believed that a major part of the problem has been the lack of recognition that Information Systems development is largely a social process. This was recognised by researchers such as Checkland (1981), Keen (1981) and others. It should come as no surprise to systems developers that social interaction such

as: prototyping, obtaining requirements, conducting structured walkthroughs, discussing design options, are all intensely social activities, important during building systems. These activities on their own however are not considered sufficient to guarantee success.

While some conventional systems development techniques have recognised the importance of the social component of Information Systems Development, a greater concentration is still given to the technical process of systems development. Several books, Hawryskiewicz (1988), Hodge and Clements (1986) have concentrated on the technical aspects to systems development. Boland (1985) is quoted by Hirschheim et al. (1991, p. 30) as claiming that "Systems development proceeds through the social interplay of multiple actors who attempt to interpret or "make sense" of their and others' actions, largely through the medium of language."

Astley (1984) cited by Hirschheim et al. (1991), felt that there was value in using symbolism in its fundamental ability to allow people to act.

Symbols ... do not just fulfil expressive, sense-making functions for managers. ... By fixating individual perception on common beliefs and values that de-emphasise differences between organisational participants they bridge across idiosyncratic cognitions to produce a basis for coherence that justifies and facilitates the mobilisation of collective action. In this respect, theories, world views, goals, visions, expectations, plans, myths, stories, rituals and terminology affect practice not directly, but indirectly, through ideologies which fuse organisations into a unitary body and thereby lubricate the process of change. (p. 32).

Hirschheim et al. (1991, p. 33) tried to view their role as "one of documenting the behaviour of ISD participants that are trying to make sense of their ISD situations, and classifying the symbols they use for sense making." They classified these into three types: myths, metaphors and magic.

Cohen (1969) is cited by Hirschheim et al. (1991, p. 34) as claiming that there are several functions that myths can serve. These include:

- . myths explain,
- . myths express,
- . myths maintain cohesion and solidarity,
- . myths legitimise,
- . myths communicate unconscious wishes and conflict,
- . myths mediate contradictions, and
- . myths provide narratives to anchor the present to the past.

It should be remembered however that myths are also ambiguous. Hirschheim et al. (1991) identified six myths that systems developers use as guides to design.

- . User involvement is beneficial and should be encouraged. This myth comes from the widely held opinion that we should be increasing the participation level of users during systems design. This myth can appear in the traditional rationality where the designer collects facts from the users and produces the design independently, to

another extreme where the design team is lead by the user who is continually involved. The designer however may still interpret involvement very narrowly, whatever the approach, not allowing users to radically challenge the design they have proposed.

Mumford (1983) is cited by Hirschheim et al. (1991) as claiming that user involvement is considered beneficial by most people. User involvement is also promoted because it is believed that users should be allowed to influence the system they are going to use.

Several writers suggest that user involvement in reality can create more problems than it solves. (Keen 1981).

Resistance to systems development is dysfunctional and should be eradicated. This myth comes from the belief that resistance to Information Systems Development is nearly unavoidable when large scale systems development is undertaken. According to Marcus (1983)

The analyst should recognise that the goal of the exercise is not to "overcome" resistance, but to avoid it, if possible, and to confront it constructively if not. Sometimes, this indicates that the implementor may have to lose the battle and sacrifice a pet system project to win the war. Resistance is not a problem to be solved so a system can be installed as intended; it is a useful clue to what went wrong and how the situation can be righted. (p. 441)

Information Systems should be integrated wherever possible. There are many advantages offered by integration motives but it also acts as a source of much conflict within organisations. This is due to the undermining effect it has on power structures and existing commitments. Newman and Rosenberg (1985) are quoted by Hirschheim et al. (1991) as claiming that

MIS undercuts the existing "rules of the game" as these are tied to the variety of weak or strong political systems in organisations that unify the various layers of management. Such a supposedly neutral act as one manager supplying data from his division to a common database ... finds ... that the data becomes the property of all units in the organisation, increasing the potential of senior managers to control middle level managers directly. (p. 35)

The systems developer is generally the best person for making decisions about the system. According to Hirschheim et al. (1991, p. 35) "armed with an arsenal of technical wonders, the analyst, is often only too willing to make policy decisions when managers abdicate their responsibility." Hedley (1970) is quoted by Hirschheim et al. (1991) as claiming that

What tended to happen in the absence of definition was that the people responsible for systems analysis made policy decisions about the various controlling factors almost without being aware of doing so. There was therefore the danger of long-term decisions being made in terms of their appropriateness to systems design rather than in relation to more fundamental criteria. Thus real control tended to shift to the systems designers. (p. 35)

Politics should not be the concern of the systems developer. Many designers view politics as part of the problem of organisations, it is treated as an irrational aberration that needs to be removed or terminated. Mumford (1972) is quoted by Hirschheim et al. (1991) as claiming that

Human relations and the ability to manage the human part of change are skill areas which are dealt with intuitively and unsystematically and this is in striking contrast to the technical side of computer systems where the programmers and systems analysts continually stress the need for a logical approach. (p. 36)

The key to successful design is the use of a top down approach. Some Systems Development staff have a belief that there is a mythical quality contributing to success in a systems design when a top down approach is used.

Metaphor

Boland (1987) is cited by Hirschheim et al. (1991, p.37) as having identified five metaphors that he believed guides systems development. He further points out that these metaphors are in fact dangerous fantasies. "They reify the human actor in ISD, and inevitably lead to dysfunctional consequences."

- Information is structured data. This is one of the more basic metaphors that view's information as an object or entity. The reason this metaphor is embraced is to get around the hermeneutic problem of having to translate information. With this metaphor information is treated as some publicly observable, immutable object rather than being concerned with the meaning of information to a free, intentional human actor.
- Organisation is information. With this metaphor an image of organisational control as the differential distribution of decision parameters and information is allowed to exist. Boland (1987) is cited by Hirschheim et al. (1991, p. 38) as claiming that "this metaphor emphasises a highly realistic way of characterising organisational life as goal-driven and purposive which further legitimates ignoring the individual actor's need to interpret and make sense of organisational situations."
- Information is power. With this metaphor the vision that information is power, in that it allows control over an individual is created. The role of the systems developer as one who, by system design reallocates and creates power.
- Information is intelligence. Here the metaphor unites human intelligence and information. Boland (1987) is cited by Hirschheim et al (1991, p. 38) as claiming that the "result is the complete removal of human beings and their problems of action and sense making from the domain of information systems discourse".

- Information is perfectible. According to Hirschheim et al. (1991, p. 38) "This last metaphor is the culmination of the other four. Here, information is fantasised as capable of being perfect and true." It discusses systems design as a context-free, timeless and ideal future. This however is reliant on complete and error-free knowledge.

Two metaphors are used to discuss pessimistic and optimistic situations for the human-technology relationship.

- Technology as tool and man as craftsman. In an optimistic situation, technology is believed to be a tool in the hands of the workers. This relationship that exists between craftsman and tool is that of master and slave.
- Technology as governor and man as machine. In a pessimistic situation, the technology is used to control the operation of some task(s). This relationship that exists between machine and governor is also master-slave, with the exception that now the technology is the master while man is the slave.

Hirschheim (1991) also pointed out, the existence of the Battle Metaphor. This metaphor sees system implementation as battles played out between users and developers. Users and developers would adopt a particular defensive and offensive strategy to overcome the other party.

Metaphors and Methodologies

The work of Kendall and Kendall (1993) has an interesting contribution to make in that they linked the concept of the metaphor with development methodologies. During interviews nine metaphors were identified. These metaphors were then linked to various development methodologies that are currently used. The nine metaphors included the following:

- . Game.

This metaphor treats the company as a team and each player has a specific position to play. There are winners and losers in the organisation and considerable pressure to perform.

- . Machine.

Here the machine has a goal or purpose, and its performance can be calculated. A machine metaphor can be enacted by users and is often described as the system's trouble-free processing, e.g. 'running like a well-oiled machine'. When problems are encountered by the machine, it is the worker's inadequacy, not the machine's, e.g. 'He threw a wrench into the works'.

Journey.

With a journey metaphor the key entailments of the 'journey' are the leader; his or her crew or team; the possibility of risk and danger due to unpredictability; and, potentially, adventure. The journey is most often considered to be a sea voyage. Depending on the charting of the captain the organisation ship may run aground or sink in rough or choppy economic waters.

Jungle.

Here the key entailment is unpredictability. The concept of 'every man for himself' is in operation. The inhabitants of the jungle actively negotiate with the environment to hold danger at bay.

Family.

The idea of a 'family' features the bringing together of people to share each other's company. The term used to describe, this metaphor is 'we're all one big happy family here.'

Zoo.

This metaphor is one that in reality is often born of frustration. Here the organisation is viewed as a zoo where people have a chaotic existence and are not fully in charge of the unnatural, hectic habitat imposed on them.

Society.

Kendall et al. (1993, p. 155) claimed that "the 'society' metaphor means that the organisation using it is alternative oriented. The society is not dedicated to any one goal but stresses the political interaction of its members who engage in negotiations to achieve their own individual goals."

War.

With this metaphor having a goal is a primary entailment for war. To fight and win against the enemy you require the best leaders and troops, material and strategy to fight and win. There are considerable risk and unpredictability.

Organism.

Kendall et al. (1993, p. 155) points out that "the 'organism' metaphor stands in animated contrast to the lifeless machine metaphor recounted earlier. The mechanical organisation is replaced with the living one, meaning the organisation can now be born, grow, become diseased, and even die."

Kendall et al. (1993) attempted to link these metaphors to the development methodologies currently used in the Information Systems development industry (see table 2).

Table 2
Metaphors and Their Corresponding Methodologies.

| Metaphor | Methodology |
|-----------------|---|
| Game | Traditional SDLC |
| Machine | Structured Methodologies/ CASE Tools |
| Journey | Prototyping |
| Jungle | Project Champion |
| Family | ETHICS |
| Zoo | Soft Systems Methodology |
| Society | Multiview |
| War | None identified |
| Organism | None identified |

(Kendall et al., 1993, p. 157)

The traditional systems development life cycle was considered to have corresponded to the game metaphor. In this situation, we have systems analysts chosen for their compatibility and competency along with other team members, all of whom have shared team goals.

Structured Methodologies / CASE tools are considered by Kendall et al. (1993) to correspond closest to the machine metaphor. The reason for this is that these types of methodologies and tools are believed to require user participation mainly during the early phases and final stages of the methodology.

The development of a prototype system usually occurs when there is a need to rapidly develop some system feature. In a way we are perhaps making a crude map suggesting that which is to come. The eventual outcome of the journey is to reach an agreed upon goal, a bit like the captain and crew safely reaching their destination.

The idea behind a project champion is one borrowed from the management and marketing literature, but here it is applied to information technology. The project champion is seen as a user chosen by the organisation who is considered as pivotal in the drive for acceptance of a new technology. Kendall et al. (1993, p. 158) believed that "the project champion possess particular personality characteristics, leadership behaviour, and career experience." Personality characteristics considered special include self confidence, high energy within organisational relationships, persistence in the face of doubt and negative reaction, the capability to express a captivating vision to the technologically unaware in the organisation, and a willingness to pursue unconventional action plans.

The ETHICS methodology is based on the socio-technical approach. This approach is believed to improve the relationship that exists between the groups and its work. Kendall et al. (1993, p. 159) claimed that "group work is to be improved in the context of the organisation's life, not just the individual's."

The soft systems methodology is concerned with assisting the organisation to learn by using general problem formulation approaches instead of techniques and tools. Its core product is the 'root definition' that explains the critical parts of the organisation. The soft systems methodology examines the chaos of the organisational situation in which analysts and users find themselves. The situation that exists here is described using the zoo metaphor.

The multiview sees analysts as people that may perform several different societal roles in a society or social system. This methodology exists in a society in which people have their own role. There are numerous rules that have to be followed and multiple alternatives need to be decided upon by negotiation.

The study by Kendall et al. (1993) was unable to identify a methodology for the war and organism metaphor.

This research provided a useful means of linking metaphors used by systems developers and end users to Information Systems development methodologies. The use of focus group interviews however may not provide the best means of gathering the data for the study as some respondents' views may dominate the discussion while others may just go along with points raised without being willing to state their own beliefs in front of other group members.

Magic

Hirschheim (1991) pointed out that there are many areas in which magic manifests itself. These include the following areas:

The systems developer as high priest. Systems Developers are on occasions exhibited as the individual (or group of individuals) that can harness the power of information technology to benefit the organisation. The developers, as the high priest of technology, possesses the apparent magical quality of making the computer productive, there by transforming a highly unintelligible piece of technology (to the lay person) into a key tool for the organisation.

Expert systems as the embodiment of the human expert. According to Hirschheim et al. (1991, p. 41) "Expert systems have been portrayed as vehicles for substituting a computer for a human expert; they can reproduce the knowledge possessed by experts or professionals." To the lay person the expert system gives the impression of intelligent behaviour, while to the computing professional the limitations of expert systems are generally recognised. This level of intelligence has a particular magical quality about it since it's an inanimate object that possesses the intelligence.

Recognising The Politics of MIS

According to Grover et al. (1988, P. 145) "user resistance to management information systems is common." While there are several different reasons why systems either succeed or fail an important part to understanding user resistance to management information systems is to recognise the role politics has to play. In some circumstances user resistance may take the form of obstruction due in part to a perceived threat to their parochial interests by reducing their autonomy, increasing their workload or invading their territory. The tactics they may use range from disingenuousness to downright dissimulation. To view user resistance to a project Keen (1981) applied the concept of "games". The games metaphor and the typology developed by Bardach (1977) and cited by Grover et al. (1988) may serve as the basis for a discussion on the existence of political activities during MIS development and implementation. Keen (1981) believed that the typology might assist MIS professionals, users, user management, and MIS managers recognise political maneuverings as they try to develop high quality systems and with substantial contribution to their organisation.

Grover et al. (1988) say that MIS practitioners and researchers might recognise a successful MIS as a system that makes a contribution to their job or as a system that is easy to use. An unsuccessful system might be seen as difficult to use and provides information that is worthless. Despite this intuitive evaluation MIS practitioners and researchers use three means to detect the success or failure of an MIS.

Project success can be shown by a positive financial return while project failure can be shown by a negative return.

A heavily utilised system can suggest project success while under use can show failure.

Success can be shown by a high level of user satisfaction while failure can be shown by substantial user dissatisfaction; this is especially appropriate where the systems use is mandatory.

During the development and implementation process there is a lack of assistance from measures to quantify excessive delays. Because of excessive delays there could be wasted resources that could lead to exorbitant costs and project failure. A project with potential would have failed. The three measures described above, ie financial return, use and user satisfaction would not have recognised the failure.

A critical determinant of information systems success is effectiveness, however, an implementation requires significant organisational commitment. If we fail to gain this commitment then it could lead to resistance to the project via political maneuvering. Because this maneuvering can change the fate of an MIS implementation, politics is an important factor to understand.

Ginzberg (1981) is cited by Grover et al. (1988) as claiming that there are three issues that are critical for successful MIS implementation.

- Commitment to a project must be developed between users and management.

- Organisational commitment to change must occur.

Grover et al. (1988, p. 147) put forward the view that "The lack of commitment and lack of willingness to change are the precursors of indifference and deliberate resistance to implementation." Markus et al. (1983) found that those who expect to lose power resist implementation while those who expect to gain power support it. Markus et al. (1983) is cited by Grover et al. (1988, p. 147) as claiming that "To design systems that will not be resisted or to devise ways to modify resisted systems, the technical system analysis must be augmented with a social or political analysis ..."

The politics that occurs as part of a systems development can be considered a series of games. These games have winners, scoring methods, rules, players and losers. The players who support the completed implementation of a successful project are classified as the winners or players who resisted a failed project. While those players that resisted a successfully implemented project or supported a failed project would be classified as losers.

Table 3
The Games

| Game Played | Game Description |
|--------------------|--|
| Up for grabs | Various parties try to take charge of a project if the decision that started the project was ambiguous or half-hearted. |
| Pilling on | Initial success of a system implementation leads onlookers to throw their own goals and objectives onto the heap. As a result the scope of the project itself might change and it might eventually fail to meet its original objectives. |
| Massive resistance | Parties that see the MIS project as a threat to their power and influence might obstruct its implementation by withholding certain critical elements. |
| Easy life | A party that is in an advantageous position in comparison to the other parties, resists changes to maintain its comfortable position. |
| Tokenism | The parties attempt to appear to be contributing to the project publicly while actually conceding only a small ("token") contribution. |
| Territory | When an MIS project has significant overlaps with the jurisdictions of several different parties, these parties might attempt to extend their control and power by taking it over. |
| Reputation | The actors seek out means of persuading an audience that they are doing more or doing better than they are really, or seek a reputation for boldness, toughness, or innovativeness. |
| Tenacity | The parties delay the progress of the project until their own particular terms are satisfied. |

| Game Played | Game Description |
|-------------------------------|---|
| Not our problem / Their Fault | If a party sees the project as imposing a heavy workload or raising controversies and thus considers it unattractive, it may try to shift certain operating responsibilities to other parties. This "Not Our Problem" game might be followed by a more aggressive version "Their Fault", in which actors seek to deflect any prospective blame for project failure. |
| Odd man out | In this game, the player creates an option to withdraw if the project gets into trouble and then the chance to say "I told you so". |
| Easy money | Taking advantage of the unclear goals or ambiguous specifications of an MIS project, some parties support the project because it can be used to finance some needed activity within their own sphere of interest. |
| The budget game | The players "maximise" their budget, the stated work force requirement, or hardware/ software demands etc., to protect or extend control. |

(Grover et al. 1988, p. 148)

Bardach (1977) is cited by Grover et al. (1988) as finding that implementation games fell into four categories depending on the major adverse effects of the games'.

Deflection of Goals.

Sometimes the stated goals of an MIS project might be ambiguous. In these circumstances there may be a need for further definition. Since the parties opposed to the project may have decided to remain quiet when a consensus was reached to initiate it, this consensus may be temporary and fragile. During the implementation process however, the opposing parties might attempt to redefine these goals. This

process of redefinition could result in a trimming down of the goals, changing them, or even adding to them in a way that could lead to implementation failure due to the overburdening of the project. This category includes two games, "Piling On" and "Up for Grabs".

Dilemmas of Administration.

The implementation process of an MIS requires the gathering together of a wide range of resources. Some may be tangibles such as software, personnel, hardware, or funds etc. They may also be intangibles such as commitment, skills, willingness to participate etc. This dependence may be used by one party to take advantage of the other party, by threatening to withhold their support. By doing this they are forcing the managers to either proceed without their resources or to meet their demands. This category includes "Easy Life", "Massive Resistance" and "Tokenism".

Dissipation of Energies.

Here considerable energy is wasted by groups and individuals either by attempting to gain control and power or by avoiding responsibility. These groups or people try to organise situations that help their own game playing strategies and that protects them from similar attempts by other parties. The result is poor performance and project delays. This category includes "Not Our Problem / Their Fault", "Reputation", "Territory", "Tenacity", and "Odd Man Out".

• Diversion of Resources.

Many organisational resources may be involved in an MIS project. The games under this category result in the diversion of these resources away from useful causes and towards the achievement of the narrow objectives of the individual parties involved.

This category includes "Budget Game" and "Easy Money".

According to Grover et al. (1988) the study was designed to determine, using a qualitative approach, whether MIS professionals can confirm the existence of the games within MIS development and implementation. Interviews were conducted on 18 MIS professionals that were either enrolled in or recent graduates of an MBA program. They had on average about 7.6 years experience in MIS.

The study had three major findings:

- MIS professionals can perceive the existence of games,
- The games typology can serve as a way of identifying situations where political maneuvering is taking place.
- The research serves as a stimulus to investigators to attempt to understand its implications and to examine political maneuvering in MIS development and implementation.

The need for a more robust typology was suggested using new or existing theory and the need for the typology to be verifiable was also recommended by Grover et al. (1988).

The sample used to determine if MIS professionals could recognise politics was restricted to MIS professionals enrolled in or recent graduates of an MBA program. A sample from a group with a wider background than just MBA graduates may produce different results.

Productivity Tools

The work of Ferguson (1983, p. 58) has an interesting contribution to make in that he states "systems development has not been a fun job lately". While productivity tools offer a tempting solution to information processing management, the tools on their own however cannot improve productivity. There is a need for a change in attitude towards systems development. The ingredient that is sorely lacking in improving systems development productivity is careful planning. The developers' needs are not being considered with the same care given to users' needs.

There are four pitfalls that need particular careful consideration. These include:

- Tools require adequate machine resources.

The idea is that the cheaper machine resources are traded for the expensive personnel time for developing systems. If however the machine resources are already bottleneck's, adding programmer tools will only worsen turnaround and response times. By transferring additional work steps from a manual approach over which

they have some control to an environment over which they have less control can lead to lower morale and a decrease in productivity.

Tools must work together.

Attempting to integrate tools that were never designed to work together can create an environment with redundant tasks. If for example we were to purchase support tools such as screen painting, code generation, and data dictionary functions, it is common for these tools to require their own descriptions of the data elements and aggregates on which they are to work. This then requires the developers to perform the data definition task several times and somehow keep all the redundant data definitions synchronised. Training for new personnel to effectively use these tools also becomes an increasing problem. There is a significantly steeper learning curve for new employees than before the tool's acquisition.

Tools must address real needs of developers.

Productivity may be hampered unless the developer's most critical needs are determined. Tools are often dropped on developers without a study into their real needs. In comparison, the user environment is examined in some detail to determine the real needs of the users.

- Tools must be supported by training and follow-up.

Without proper training and follow-up, the tools installed very quickly become inactive members in the IT program libraries.

Ferguson (1983) found that to maximise the benefits gained from productivity tools you need to:

- Obtain management commitment.

Management need to view developers as they do users. Gains in productivity will be very slow in coming if the development is treated as a production job.

- Organise developers into two groups: tool-makers and tool-users.

Tool makers are responsible for integrating the tools to present a single-system image. The tool makers perform a role similar to that carried out by a programmer / analyst in the user departments. They analyse the needs of the developers and then develop or select systems that meet those needs.

- Select and install a Systems Development Methodology.

By selecting a proven methodology there is a reduction in the development risks. The selection and use of a formal methodology for developing systems will provide an overall structure through which the use of tools can be controlled and encouraged.

Develop and document the shop's technical architecture and development standards.

"The technical architecture describes major components of the vehicle that will deliver function to the user". (Ferguson, 1983, p. 60). The development standards provide specific guidelines for things such as naming and coding conventions, report design, performance requirements, and screen design.

The tools selected should help enforce standards and follow the organisation's technical architecture. To improve the chances of success, the tools selected should be used immediately preceding and then by a major business development project. This enables the developers to put into practice what they have learned during training. Ferguson (1983) points out that:

Tools by themselves are of little value. But when experienced analysts design an integrated environment that recognises and satisfies the unique needs of a specific shop, productivity will naturally and dramatically improve. (p. 60)

CASE Tools

McClure (1988) found that Extended Intelligence Inc. in Chicago, Illinois, has done extensive research on the CASE tool's market. They found that in 1984 there were only a few vendors selling only a few hundred products. These were mainly PC-based, front-end, life-cycle tools. By 1992, worldwide sales of all CASE tools were expected to be about \$1 billion.

CASE tools are usually characterised according to the stage or stages of the software development life cycle on which they focus. These basic steps of software development are:

- . planning,
- . analysis,
- . design,
- . implementation,
- . maintenance.

According to McClure (1988, p. 52) "Success with CASE most likely will occur when developers and managers choose tools based on methodologies similar to those already in place within the organisation".

During the 1970's structured methodologies emerged to provide an organised approach to software development. These methodologies consisted of sets of rules, methods, and assumptions used to organise the problem solving approach by listing, documenting all steps, and diagramming. They helped standardise and systemise software development and maintenance by approaching it using an engineering discipline rather than by whatever individual software developers fancied. These diagramming techniques, until CASE, were produced using manual techniques, which were slow and tedious.

CASE tools usually support one or more structured methodologies. One approach to classify a structured methodology is by orientation. Structured methodologies may fall into the following categories:

- . procedure-oriented methodologies,
- . data-oriented approach, and
- . information-centred approach.

CASE tools fall into the following general categories:

- . diagramming tools for graphically representing system specifications,
- . report and screen painters for creating simple prototypes and system specifications,
- . information management systems, dictionaries, and facilities to query, store and report technical and project management system information,
- . specification-checking tools to detect syntactically incorrect, incomplete, and inconsistent system specifications,
- . code generators to generate executable code from graphical system specifications,
- . documentation generators to produce user and technical documentation required by the structured methodology.

McClure (1988) believed that

Organisations that benefit the most from CASE tools are those that already have in place a structured methodology for planning (or whose managers are willing to support the one chosen) and those that select well-integrated tools. Because a structured development methodology provides the overall frame for defining and linking software process steps, understanding structured

methodologies is the key to making CASE work in an organisation. (p. 67)

Before implementing a CASE tool, developers should

- . establish a structured methodology,
- . if the organisation already has a methodology in place then the methodology can narrow the eligible products to those that support their techniques,
- . the CASE tool should be checked to determine whether the tools support existing hardware,
- . which life-cycle phase they automate,
- . how they integrate with other tools,
- . sophistication of features eg. documentation, repository, code generation, analysis and design checking, report generation and graphic support, which the tool needs to support, and
- . whether they support enough methodologies to be useful when considering the diverse nature of the organisation.

Hayley and Lyman (1990) state that the Chicago offices of Delloite & Touche conducted a survey to address the use of CASE in US organisations. CIO's were asked to assess the use and impact of CASE tools in their organisation, which they believed yielded the following two compelling conclusions:

- . The use of CASE tools does not necessarily lead to an improvement in systems development's productivity. (System quality did improve, but productivity did not

improve usually).

- . There are considerable barriers to the implementation of CASE tools. (This is about the level of investment required and the logistics of effectively managing CASE tool implementation).

The survey revealed the current most common tools are:

- . 4th generation languages,
- . Prototyping,
- . SDM (Systems Development Methodology),
- . Relational DBMS,
- . Application generators,
- . CASE

CIO's currently view CASE as a technology that holds the greatest promise of improving productivity in the future.

Only one third of the respondents had used CASE tools and half of these had been using CASE for less than one year. (CASE is typically used in large IS shops with annual budgets of \$47.7M U.S.)

CIO's rated the potential benefits of CASE:

- . High quality systems,
- . Less or easier maintenance,
- . Better documentation,
- . Clearer communication,
- . Accelerated SDLC,
- . Improved user satisfaction,
- . Reduced system development costs

(Better documentation while rated 3rd, may be CASE's most tangible benefit).

There may be a considerable learning curve with CASE, with actual benefits not appearing until it has been in use for more than three years.

The application of CASE tools does not automatically lead to increased productivity, which is perhaps the greatest myth concerning CASE's potential benefits.

The time required for implementation is very slow. Typically a company can expect to spend more than \$25,000 U.S per employee to implement a fully integrated CASE environment.

The authors suggest cultivating a few specialists in the department and using a few pilot projects as part of the training program, (but to expect that productivity may go down

in this learning period).

There is also a problem of higher staff turnover in shops with CASE than those without. This may be due to a resistance to change and differences in basic philosophy among team members trying to implement CASE tools and technologies.

At the very least, the importance of this staffs turnover intimates the need for creating an appropriate learning environment.

Norman and Nunamaker (1989) found that as we enter the decade of the 1990's more organisations are making use of CASE technology in both business and scientific application areas. These automated tools mainly support structured methodologies and their variants. CASE tools are the products of many years of research on integrated development environments.

They state that some advertisements for CASE tools claim that users can receive from 30 to 300 percent improvements in productivity however in reality the improvement in productivity is considerably less.

The authors of this paper attempted to measure using a psychometric scaling method called Multi Dimensional Scaling (MDS), the ordering and underlying relationships of CASE technology.

During this study the subjects were required to rank pairs of CASE functions under how they thought the similarity of each affected their productivity. Two factors that are very important during systems development were:

- . Communication among project team members.
- . Adherence to the organisations system development standards.

There were 91 subjects who used CASE technology in this study. Of those 67% of the targeted systems being analysed by these subjects were scheduled to use either a 4 GL or COBOL, while 23% used other languages. All subjects used the Excelerator CASE tool. The results of the survey showed that tools such as data flow diagrams and data dictionary support were the highest rated stimuli. The third highest rating went to project standardisation, which is quite significant as the authors considered that the main reason that CASE products were purchased was their ability to allow software engineers to produce models of the user's requirements.

The fourth ranked stimulus was screen / report design, which would suggest a high use of prototyping by the subjects.

There were eight major recommendations that came from the observations:

- . Software engineers believe that CASE technology improved their productivity.
- . It identified what parts of a CASE tool were seen to provide the most productivity.

- . Productivity improvements could be attributed to adherence to the enterprise's systems development standards when CASE technology was used.
- . This study provided the steps towards a more rigorous validation of the effects of CASE technology on software engineers' productivity.
- . The use of CASE technology with adherence to enterprise systems development standards needs further attention and evaluation.
- . The CASE tool vendors should continue to enhance their products to improve the software engineers' productivity.
- . The need for robustness of functionality in CASE tools need not be great to improve productivity.
- . Research should continue to push to automate more of the software engineer's jobs.

While the research by McClure (1988) found that CASE improved quality rather than productivity Norman et al. (1989) research contradicted these findings by stating productivity was improved. More research is needed to determine who is right.

Identifying Organisational Culture Clash In MIS Implementation

The works of Romm et al. (1991, p. 99) found that Management Information Systems (MIS), implementation researchers "have emphasised that the analysis of MIS projects must not only consider technical validity but also organisational validity". Technical validity includes factors such as screen layout, response time, system design, or complexity

of supported task. While organisational validity includes factors such as the interaction between MIS and the organisation, Micro aspects of organisational validity include:

- . user involvement,
- . user familiarity with the task,
- . user cognitive style, and
- . user motivation.

Macro aspects of organisational validity include:

- . deployment of information systems as strategic weapons within the organisational environment,
- . organisational hierarchy and channels of communication, and
- . organisational distribution of power.

There are two principal types of important basic assumptions that organisational members may share. These include:

- . beliefs - these include facts about the world, cause/effect relationships, and how the world actually works, and
- . values - represent preferences for desirable ideals, ones worth striving for or end states.

Romm et al. (1991) believed that if a culture clash was to go unnoticed, the likely result would be implementation failure. Early detection of a culture conflict can help towards reducing the damage. There are two ways by which we can avoid culture clash. These include:

- . withdrawing from the implementation altogether, and
- . by early identification the potential conflict-related losses can be reduced by taking corrective action that alters the MIS design.

Romm et al. (1991), put forward the view that seven measures of cultural differences have been used with a high level of reliability and validity. These include the following:

Table 4
Seven Dimensions of Cultural Differences

| | |
|----|---|
| 1. | Innovation and Action Orientation: This dimension of cultural difference relates to the beliefs concerning the importance of encouraging innovations, the urgency of taking actions and rapid responses to changes in their environment. |
| 2. | Risk Taking: Here the beliefs relate to the justification for taking risky decisions such as handling employee pension funds, investing in new ventures and purchasing manufacturing equipment. |
| 3. | Integration-Lateral Interdependence: This relates to the belief that cooperation (instead of competition) and communications between the sub units of the organisation for obtaining overall organisational objectives as reflected by the amount of encouragement given to information sharing and to understand one another's difficulties. |
| 4. | Top Management Contact: This concerns the belief that whether subordinates should receive warmth, managerial support, consideration and whether open expressions of criticisms by subordinates are acceptable. |
| 5. | Autonomy in Decision Making: This relates to the belief about whether autonomy and responsibility for important decisions can be delegated, influences how formally or broadly management procedures are defined. |
| 6. | Performance Orientation: This dimension concerns the belief about holding subordinates accountable for their performance, about the nature of demands placed on members, the specific focus of performance appraisals, and about making performance expectations clear. |
| 7. | Reward Orientation: This dimension of cultural difference related to whether subordinates should be paid equitable and competitive salary and whether compensation and performance should be directly related. |

(Romm et al., 1991, p. 101)

Romm et al. (1991) believed that the use of a four-cell risk grid typology that mapped the expected risk for a cultural clash would assist in reducing the likelihood of implementation failure. Due to the considerable subjective nature of culture clash it is difficult to know whether problems with implementation can be categorised into just these four categories.

In cell number one high predictability of outcome and low culture content compound to produce a very low risk level. The low culture content means that:

- . culture-related losses are contained, and
- . culture conflict is above the surface.

The associated losses are thereby further contained by the low culture content interacting with the high predictability of outcome to lower the likelihood of late detection.

In cell number two high predictability of outcome and high culture content combine to produce a medium-level risk. The high predictability of outcome acts like an opposing force by shedding light over circumstances clouded by high culture content. This has the effect of reducing the time to clash detection and constrains clash-related losses.

In cell number three a low predictability of outcome compounded with low culture content result in an intermediate level of risk. The likelihood of early detection and thereby increase in associated losses is reduced by the interaction between low culture content and

low predictability of outcome.

Cell number four has the greatest risk due to the increased losses, which are usually associated with the higher levels of culture content, due to the increased chance of a culture clash going undetected, the compounding effect of high culture content and low predictability of outcome. Spending a substantial effort on detection in situations where there is a low predictability of outcome and a high culture content level is worthwhile since this combination compounds to the highest level of risk.

My opinion is that this typology, while offering a good starting point, is somewhat difficult in several areas. These include:

- . It has only been tested using systems that easily fit its structure. It needs to be tried on systems other than just personnel systems.
- . The subjective judgement of the person performing the detection could lead to the wrong combination and subsequently wrong level of risk being detected.
- . It can be difficult to determine the cost of cancelling a system implementation compared with the cost of detecting and reducing the impact of a culture clash. A company may go out of business if it needs a particular corporate system that gets cancelled, but going ahead may lead to culture clash that may also endanger the

organisation.

Table 5

The Risk Grid (the risk if an effort to identify a culture clash is not undertaken before implementation)

| Predictability of Outcome | Culture Content | | | |
|---------------------------|-----------------|-------------|------|-------------|
| | Low | | High | |
| High | 1. | Low Risk | 2. | Medium Risk |
| Low | 3. | Medium Risk | 4. | High Risk |

(Romm, 1991, p. 102)

Markus (1983) claimed that there are several familiar comments regarding resistance.

These include:

- Try to gain management support and user involvement in the design process as it will help avoid resistance;
- Systems that have poor response time and frequent down time are more likely to be resisted than technically sound systems;
- Systems that are not "User friendly" will be resisted by users (assertions by EDP equipment vendors);
- People will resist change, given that all other things being equal (received wisdom);
- When the costs for a system outweigh the benefits the users will resist the application (received wisdom).

This research paper examined three basic theories of resistance. These include the following:

- the sub unit or person may be believed to have resisted due to factors internal to the group or person;
- the group or person may be believed to have resisted due to factors inherent in the system or application being implemented;
- the groups or people resist systems because of an interaction between characteristics related to the system and the people.

Markus (1983), said that besides the interaction theory there are several variations of them that exist. One of these variations is called the socio technical variant. This variant concentrates on the distribution of responsibility on work-related communication and coordination around this division of labour and for organisational tasks across various roles. A second variation of the interaction theory is called the political version. In this version resistance is stated as the product of intra organisational distribution of power and the interaction of system design features, defined objectively, vertically, as horizontal power dimensions or subjectively, as symbolism.

Besides the usual views that people within an organisation work towards common organisation goals there is also a belief by some people about the existence of non rational views. Markus (1983) points out this view by saying:

The view of organisations that most frequently coexists with the Rational Theory of Management and with beliefs in the Rational purposes of information systems is that all organisational members share common goals for the organisation and that they will collaborate to achieve these objectives. In contrast, the non rational view assumes that different individuals or subgroups in the organisation have different objectives depending upon their location in the hierarchy and that they can be expected to try to achieve these local goals rather than global organisational goals whenever differences exist. (p. 432)

Table 6
Theories of Resistance: Underlying Assumptions

| | People-Determined | System-Determined | Interaction Theory |
|---------------------|--|--|---|
| Cause of resistance | Factors internal too people and groups | System factors such as technical excellence and ergonomics | Interaction of system and context of use |
| | Cognitive style, Personality traits, Human nature. | Lack of user-friendliness, Poor human factors, Inadequate technical design or implementation | Socio technical variant: interaction of system with division labour Political variant: interaction of system with distribution of intra organisational power |

| | People-Determined | System-Determined | Interaction Theory |
|---|---|---|---|
| Assumptions about purposes of information systems | Purposes of systems are consistent with Rational Theory of Management, can be excluded from further consideration | Purposes of systems are consistent with Rational Theory of Management, can be excluded from further consideration | Socio technical variant: Systems may have the purpose to change organisational culture, not just work flow Political variant: Systems may be intended to change the balance of power |
| Assumptions about organisations | Organisational goals shared by all participants | Organisational goals shared by all participants | Socio technical variant: Goals conditioned by history Political variant: Goals differ by organisational location; conflict is endemic |
| Assumptions about resistance | Resistance is attribute of the intended system user; undesirable behaviour | Resistance is attribute of the intended system user; undesirable behaviour | Resistance is a product of the setting, users and designers; neither desirable nor undesirable |

(Markus, 1983, p. 433)

Markus (1983) points out that there are several predictions that can be derived from the three theories. The people determined theory predicts that replacing individuals or coopting individual resistors by allowing them to suggest improvements to the system might eliminate or reduce resistance. The system-determined theory leads to the prediction that

if the human factors and technical features of a resisted system are changed then the resistance will disappear. Markus (1983) argues that with the political variant of the interaction theory, neither of these changes will have much effect on the strength of resistance, if patterned interactions among competing groups was the reason for the generation of resistance.

An implementor who believes in the people-determined theory of resistance would find tactics such as:

- . educating and training users to change their cognitive styles or opinions about computing;
- . allowing users to self-select or carefully selecting the people who will use a new system after careful explanations about the system;
- . changing the structures or reward systems of an organisation to conform to the features of the system;
- . obtaining user participation in the design process so that they will feel a greater level of commitment to the outcome;
- . obtaining the support of the users' bosses who will either demand that the recalcitrant users comply or encourage them to support the system.

An implementor who believes in systems determines person behaviour will look at tactics that are the same but for different reasons from those used in people-determined theory. These include:

- . gaining user involvement during the design process to produce a better design than that which would have been developed without user involvement;
- . making changes to the package so that it suites the ways people work, think or do business;
- . training system designers to improve the ergonomic excellence, technical efficiency and having a smoother man-machine interface.

The implementor who believes in a combination of people and system determined theories pick and choose among the tactics. The implementors who believe in the interaction theory find that no one tactic is useful in all situations.

Markus (1983) points out that

The most important implication of the interaction theory is that the best prescription for an implementation strategy and for the specific design content of a system will follow from a through diagnosis of the organisational setting in which the system will be used. Now, system builders are using methods such as structured systems analysis that allow them to describe and analyse only the technical features of a setting that is to be automated. To design systems that will not be resisted or to devise ways to modify resisted systems, this technical systems analysis must be augmented with a social or political analysis. (p. 441)

Table 7**Theories of Resistance: Recommendations for Implementation.**

| People-Determined | System-Determined | Interaction Theory |
|---|---|--|
| Educate users (training) | Educate designers (better technology) | Fix organisational problems before introducing systems |
| Coerce users (edicts, policies) | Improve human factor's | Restructure incentives for users |
| Persuade users | Modify packages to conform to organisational procedures | Restructure relationships between users and designers |
| User participation (to obtain commitment) | User participation (to obtain better design) | User participation is not always appropriate |

(Markus, 1983, p. 440)

In concluding Markus (1983) made two observations on the use of the theory:

- too successfully use the interaction theory the implementors should consider themselves as one party in the analysis. By conducting a self-examination of the pay offs, power bases, motives and interests will lend much to the implementor's ability to understand other people's reactions to the system that is currently under design or installation;
- avoidance and not to "overcome" resistance should be the analysts goal, if possible and to confront it constructively, if not.

While Markus (1983) argument in support of the interaction theory was supported by the evidence provided, further research needs to be carried out using other organisations and different types of systems. It was evident very early during the explanation of the theories that the author had a bias towards the interaction theory as it occupied the bulk of the text. The other theories were dismissed at a very early stage of the discussion.

According to Keen (1981) two strategies that can be used after initially recognising hidden motives and devious strategies and before applying techniques to prevent future games or favourably resolve current games is to use techniques such as scenario writing and fixing. According to Keen (1981, p. 26) a fixer is "An actor with the organisational resources to negotiate among interested parties and make side payments. A person or group with the prestige, visibility and legitimacy to facilitate, deter, bargain, and negotiate effectively."

Table 8
Scenario-Writing

| | | | |
|----|-----------------------------|---|--|
| A. | Basic Objectives: | - | What exactly are you trying to get done? (not what does the system look like?) |
| | | - | What resources are needed? |
| | | - | Who controls them, directly or indirectly? |
| | | - | How can you minimise the effects of social inertia? |
| B. | Dilemmas of Administration: | - | What elements are critical? |
| | | - | Are any of them subject to monopoly interests? |
| | | - | Will their owners be uncooperative? |
| | | - | Can you work around them or buy them off? |
| | | - | Will they respond with delays or tokenism? |
| | | - | How will you deal with massive resistance? |

- | | |
|---------------------|--|
| C. Games: | <ul style="list-style-type: none"> - What games are likely to <ul style="list-style-type: none"> (a) divert resources? (b) deflect goals? (c) dissipate energies? - How can you counteract or prevent them, if necessary by redesigning the project? |
| D. Delay: | <ul style="list-style-type: none"> - How much delay should you expect? - What negotiations are needed? - What resources do you have for negotiations and/or control? - Would it help to use project management, work around possible obstacles and delay or enlist intermediaries? |
| E. Fixing the Game: | <ul style="list-style-type: none"> - What senior management and staff aid do you need? - What resources do they have? - What incentives are there for them to play the fixer role? - Can you build a coalition to fix the game? |
-

(Keen, 1981, p. 30)

Keen (1981, p. 30) felt that before starting a project it is essential to ask:

- "1) are people likely to play games?
- 2) is the proposal proof against subversion?"

The basic question that should be asked is "who can foul it up". Using scenario-writing (Table 8) helps forewarn designers and partially protects them against:

- . tokenism and monopoly;
- . massive resistance;
- . accidental or deliberate delays.

Keen (1981) suggested as an outline strategy for implementation the designer should consider doing the following:

- . The information function must be headed by a senior level fixer; they must have full authority and resources so that they can negotiate between or with users and with those affected by information systems;
- . There needs to be some policy planning or steering committee that must have senior line managers; responsibility for the project will be delegated to technical staff from the policy planning or steering committee for projects that do not have significant organisational impact but will be actively involved in projects that form part of the politics of data;
- . Substantial time and effort will be required during the planning process, in the pre design stages, where evolution of the large system is defined by breaking it into clear phases and objectives are made operational;
- . Formal contracts will be needed, which contain a clear statement of commitment and make games such as Easy Life, Up for Grabs, Territory and Reputation ineffectual and illegal;
- . Systems development staff cannot dismiss political and organisational issues as not being their responsibility or irrelevant, they must be able to build credibility across

the organisation and operate in the manager's world by developing "Hybrid" skills.

Using the umbrella provided by the steering committee and the fixer's authority, the tactical approach remains an excellent guide to managing the implementation process for a given system.

The development of information systems is political and at times far more so than, technical in nature. "Unfortunately, "politics" have been equated with evil, corruption and, worst of all, blasphemy in the presence of the Rational Ideal, but politics are the process of getting commitment, or building support, or creating momentum for change; they are inevitable." (Keen, 1981, p. 31)

For many people careers and career progression is an important part of a persons working life. Ginzberg and Baroudi (1988) found that there were very few MIS directors that had moved beyond Vice President for MIS to a corporate General Management position.

Ginzberg et al. (1988, p. 587) said "Career DP managers do not typically possess the requisite communication and general managerial skills to move into general management of the organisation". They go on to point out that while the reported results from various empirical studies are not all consistent, the recommendations for managing DP personnel found in the literature is. DP managers need to:

- . Pay closer attention to formal career planning maybe by adopting such tools as skills inventory, career ladders, training programs and skill matrixes;
- . Provide both a technical and managerial career options for IS personnel by setting up a dual career ladder.

Ginzberg et al. (1988) found early researchers believed that it was important to think about three interacting perspectives to better understand organisational careers:

- . internal Vs external career,
- . careers over time,
- . the career cube.

The external career concerns the categories that describe the progress that a person goes through during an occupation. These categories normally consist of promotions and job titles but may include changes in status, working conditions and salary. Ginzberg et al. (1988, p. 588) points out that "The internal career refers to the individual's concept of the steps or stages that will mark the progress through his or her career." For an employee to

stay motivated and satisfied, there has to be a correspondence between the external opportunity structure and the employee's criteria for success. A career anchor is one way of conceptualising the internal career. A career anchor refers to an individual's motives, personal talents and gives shape to career decisions.

Five major career anchors, which Ginzberg et al. (1988) identified in early research include:

- . managerial competence,
- . functional / technical competence,
- . stability / security,
- . entrepreneurship / creativity,
- . independence / autonomy.

Additional career anchors that are also believed to have a relationship to the above five career anchors are:

- . identity,
- . service, and
- . variety.

Careers are not fixed or *cast in stone* when an employee first joins an organisation. The career anchors change, but this occurs after the individual is well settled in an organisation. This change in career anchors helps to direct some persons career choices that in turn will have an impact on the persons career anchor.

Ginzberg et al. (1988) found that one way that is particularly useful for representing some major stages of some persons career is to use the four stage model. (See table 8).

Table 9
The Four Stage Model of Professional Careers

| | Stage 1 | Stage 2 | Stage 3 | Stage 4 |
|----------------------------------|--|----------------------------|--|---|
| CENTRAL ACTIVITY | Helping, learning, following directions | Independent contributor | Training, interfacing | Shaping direction of organisation |
| PRIMARY RELATIONS | Apprentice | Colleague | Mentor | Sponsor |
| MAJOR PSYCHOLOGICAL ISSUES | Dependence | Independence | Assuming responsibility for others | Exercising power |

(Ginzberg et al., 1988, p. 589)

Besides the idea of career stages there is also a related concept called the three-dimensional career mobility, career movement and career transitions. An individual's career movement within an organisation can occur in three dimensions:

- . circumferential,
- . radical, and
- . vertical.

Circumferential movement occurs when an employee transfers to a new function or department. Vertical movement implies that a decrease or increase in organisational rank. Radical movement relates to a decrease or increase in criticality or importance. Changes of organisation and a change of profession have been becoming an increasingly common occurrence. According to Ginzberg et al. (1988). The occupational community also shapes the person's career besides the employing organisation. Evidently "the frequency with which promotions are expected changes over time and depends on the person's education and age. The promotions received earliest in a career are the most important in shaping the individual's career". (Ginzberg, 1988, p. 590).

To understand an individual's movement through a career you need to consider three interacting life cycles, all of which may affect people's careers:

- . the bio social life cycle,
- . family life cycle, and
- . career / work cycle.

A persons career and work is only one part of a person's life. Some people will be heavily involved in their careers, with their families or other parts of their personal lives. All these involvements may change over time, too.

According to Ginzberg et al. (1988) almost all studies involving IS personnel have concentrated on external careers and have completely ignored the concepts of internal careers held by these personnel. Longitudinal studies of IS personnel have not been informed by models such as the four-stage model. If the four-stage model is valid for MIS / DP careers then personnel remaining in a technical role and avoiding management duties may result in an individual's career stalling at stage two or three, never progressing to stage four. Most of the career research on DP personnel has mainly concentrated on vertical movement, the other equally important dimensions have not been considered to the same extent.

If we look at DP as an occupational community then things such as the rapid jumps from one company to another and the high turnover that have been so common in DP may actually be predictable, rational behaviour. One possible explanation for individuals rapidly moving from one organisation to another is the need to remain central in terms valued by the occupational community. The career ladder as favoured by individual organisations employing MIS / DP personnel and the MIS occupational community is possibly very different which could lead to employee turnover. The job demand placed on IS personnel such as fixing problems too late at night, weekends or early mornings can place strains on

an IS professional's life; therefore there needs to be a greater recognition that careers' exist in a broader context, as this may have been overlooked by researchers.

While the research conducted by Ginzberg et al. (1988) raised several issues that it states have not been given adequate attention by other researchers, it failed to provide any answers itself about whether these issues were important. Many issues were just listed as needing further research.

According to Igbaria and Siegel (1993) two significant factors behind the increased concern over the career issues of information Systems (IS) people are:

- an increase in the gap between supply and demand of IS personnel;
- the high turnover rates reported.

A decision that eventually plagues IS personnel and creates internal conflict is also one that occurs in most technical oriented fields. That is, the decision on whether to move into management or remain in the technical arena.

According to Igbaria et al. (1993) the available research points to the need for a dual career path for IS employees. One for technical employees and a separate but equal ladder for management. The concern raised however was that good technical personnel may opt for the more lucrative and prestigious management jobs. The idea of creating additional career paths that consider the diversity of career desires and provides incentives for IS

personnel to remain within a technical area. These career ladders allow for horizontal movement and are wider at each rung. An alternate approach that some organisations are using to retain technical expertise is by creating senior-level in-house consultants with the prestige traditionally associated with managerial positions.

A survey was mailed out to 1152 Data Processing Management Association (DPMA) members. Of these a final sample of 348 responses, from the Mid Atlantic Region were received. The demographic characteristics of the response sample are fairly biased towards males. There were three times as many males' responses to female responses. Many employees in the survey were MIS managers (56.7%) while technical specialists such as Programmers (14.3%) and Analysts (9.8%) were not well represented with only a total of 24.1%.

The results of the survey showed that there were 266 IS employees (76.4%) that had a particular job position in mind that they would like to have within the next three years. While only 82 (23.6%) of those respondents had not made a decision. Of the respondents that had made a decision about there career over the next few years 23 (8.6%) indicated that their plans were to hold technical positions (Programmers), 33 (12.4%) wanted to be consultants, 26 (9.8%) were planning on doing non-IS jobs.

Most of the respondents, 184 (69.2%) showed a preference to hold IS management positions. These figures may support the view that the respondents preferred to stay working in IS performing either managerial or technical roles (77.8%). Few respondents showed a desire to move across to other business areas. As indicated by the results, there are primarily four categories that careers fall into. These include consultancy (outside the organisation), Non-IS general organisational areas and two that are within the IS area. Namely managerial and technical. Quiet a few respondents, would also like to be considered for management positions in these functional areas.

Igbaria et al. (1993, p. 28) points out that "It is apparent that organisations need to place greater emphasis on dual career paths within the IS department." Igbaria et al. (1993) views differ with those of Ginzberg et al. (1988) in that while Igbaria suggests that organisations should place a greater emphasis on dual career paths' Ginzberg et al. (1988, p. 591) points out that "the technical specialist may receive rank and remuneration, but not the type of power and influence that comes from centrality."

The results of the survey shows that there is no relationship between career directions and job titles for those decisive IS employees who have indicated a specific job goal. Analysis conducted by Igbaria et al. (1993) to determine the relationship between career decisions and gender found that these factors were unrelated. Education and organisational tenure were related, while age, job tenure and number of years in the IS field were unrelated to career decisions.

The major finding to come out of the study by Igbaria et al. (1993) were that most of the respondents had already decided which particular jobs they would like to do within the next three years. These IS employees could state four directions in which they sought future careers. The four areas were:

- . consultant,
- . IS technical,
- . Non-IS (business), and
- . IS management.

To maintain a high level of performance and retain IS personnel, Igbaria et al. (1993) believed organisations need to:

- . provide multiple career paths,
- . rewards and incentives must be attainable, recognisable and relevant,
- . frequent monitoring of progress for IS employees by management to provide them with opportunities for advancement within the organisation,
- . incorporate dual career paths for IS employees who wish to stay within the IS department,
- . wider rung career ladders need to be designed for each step of the career paths,
- . IS employees that show an interest in moving up to senior management outside the IS department should receive encouragement from the organisation.

According to Couger (1988) growing concerns for IS executives, as the computer industry matures is the need to maintain motivation. In the past when people who had computing skills were scarce, they earned rapid promotions and high salaries. With moderating salary increases, and promotions becoming more infrequent. "The results of my most recent study of 1, 800 Analysts and Programmers, however, hold good news for managers: the number one motivating factor for IS personnel is the work itself." Couger (1988, p. 62)

There are five job variables that are most sensitive to motivation. These include:

- . autonomy,
- . task identity,
- . skill variety,
- . feedback from the job itself, and
- . task significance.

The presence of the above five key job variables contributes to an employee's:

- . feeling of meaningfulness from their work,
- . knowledge of the results of their work, and
- . responsibility for its outcome.

"Maintenance work may represent the most difficult job enhancement challenge facing IS managers." Cougar (1988, p. 62)

By using the five core job dimensions, we can analyse maintenance work for finding ways of enhancing it:

- . **Autonomy.** By jointly setting goals with personnel doing the maintenance and then allowing them to complete the work without close supervision.
- . **Task Identity.** Identify how the modules currently under maintenance relate to the entire system and how the system relates to the organisations set of systems. Task identity can also be enhanced by making it possible for personnel to complete the entire maintenance task-from user interaction to producing workable code.
- . **Skill Variety.** A variety of tasks should be assigned where the skills used are constrained by the design of the system being maintained.
- . **Feedback From The Job.** Mechanisms that allow employees to track their progress should be established.
- . **Task Significance.** Enabling maintenance personnel to have the opportunity to work directly with users so that they can recognise the importance of their work.

Different employees will have different levels of growth and challenge. "Employees' growth needs can be determined by evaluating the degree to which they are goal-oriented, ambitious, capable of perspective (able to separate the important from the less important), interested in further education / training, self-starters, internally motivated, confident, in need of recognition, assertive, inquisitive, and systematic." Couger (1988, p. 62)

Another significant characteristic of IS personnel is the need for social interaction. IS employees such as Programmers have a lower need for social interaction than most other occupations (Couger (1988)). Large organisations may find it harder to motivate their IS personnel in the five job dimensions. This can be due to factors such as the increased levels of management, more policies and procedures, specialised jobs reducing skill variety. Analysing the five job dimensions with each employee once every six months or so, or when a different employee is assigned or whenever the job changes can help improve the productivity levels of IS personnel.

Literature On The Research Methodology

A case study approach is one of many strategies to examine the experience of IS practitioners who use SSADM. It allows the researcher to answer questions such as, "why do you use data flow diagrams"? "How is SSADM helping you to develop computer systems on time, budget and meeting your user's requirements"?

Patton (1990, p. 54) found that case studies can be "particularly useful where one needs to understand some special people, particular problem, or unique situation in great depth, and where one can identify cases rich in information". By using a qualitative, case study approach it is hoped that the researcher can "describe that unit in depth and detail, in context, and holistically".

Researchers such as Preston, Orlikowski and O'Connor found the case study approach to be a good method to gather data from Information Systems professionals. Preston used a genealogical, ethnographic case study approach as it allowed the subjects to tell their story rather than have their responses possibly misinterpreted by the researcher. In this way there is a reduced chance of having the researchers personal bias introduced into the responses.

The use of a genealogical, ethnographic case study approach also helps make the findings of the research easier to discuss since this research methodology allows the researcher to use quotes from the interviewees to back the researcher's findings. The approach helps reduce the limitation that exists with questions asked in a questionnaire, in that the respondent in an interview can ask the researcher to explain the question asked if they do not fully understand what the researcher is asking. The approach also allows the researcher to expand the open questions asked and gather further information as the need arises. The main disadvantage of using this approach is the large amount of time needed to gather the data.

THEORETICAL FRAMEWORK

There are four major areas that this thesis will cover:

- . How and where (what type of projects) the method is used, including its future use.
- . How the Information Systems methodology was selected.
- . The perceived strengths and weaknesses of the methodology (ie its success).
- . The factors that influence the perceived successes and failures of the methodology (i.e. whether they are technical eg. CASE, prototyping, types of projects etc.; or political eg. careers, culture of organisation).

The SSADM methodology is the central theme of this thesis. It will be viewed in a holistic way, i.e. from the technical to the political.

This study will explore how the SSADM methodology is currently used and could be used in the future.

A detailed examination of why and how the SSADM methodology was selected, and the political factors that could have influenced its selection will be undertaken.

This study will explore the issue of the availability (i.e. progress) of CASE (Computer Aided Software Engineering) tools and their frequency of use. It will attempt to determine whether for CASE tools to be used, the following are significant:

- . adequate computing resources,
- . integration of tools,
- . the developers needs,
- . training and follow-up support,
- . management commitment,
- . a systems development methodology,
- . technical architecture and development standards.

The case study will explore if it is important to have the necessary infrastructure, training in the information systems methodology, management commitment, and appropriate technical standards for the information systems methodology to be successful.

This case study will look at whether the selection of the systems development methodology was based on the rational text book type selection method. This typically identifies the costs and benefits of using a particular information systems methodology and determines which methodology is best suited to the organisation and type of systems being developed (Modha, Gwinnett and Bruce (1990)), (Dos Santos (1988)), (Richter (1988)). If this method was not used then the other methods used will be examined (Modha et al. (1990)).

METHOD

Sample / Subjects

This case study was conducted in a large service organisation since most large organisations are the ones that will employ staff with the widest development experiences in the use of structured systems development tools.

The organisation targeted for this case study is one that develops computer systems that vary from small PC based systems to large corporate wide mainframe systems. This enabled the researcher to compare the responses from developers of both small and large systems. The main employees targeted within this organisation were computer systems development staff employed as Programmer/Analysts (8), Project Leaders (3), Programmers (9), Information Technology Officer (1), Project Manager (1), and development review and consulting personal such as the Chief Analyst (1), Quality Assurance Administrator (1), Data Administrator (1) and Chief Programmer (1).

There were 26 participants involved in this research case study. Some staff who have not had any experience or very little experience in using SSADM were still selected for an interview so they could express their opinions and the reasons for them.

Research Design

The research methodology is an ethnographic, genealogical study. As most of the objectives of the research relate to the behaviour, world views and values of individuals and the culture and politics of the organisation, a qualitative approach was most suitable. The study has drawn upon informal communication as a participant, personal observation during the study, interviews conducted with staff and a range of formal interactions and communications. Field notes were made to document information, my thoughts, and impressions as they occurred, while an on-line diary was used to record appointment times with the interviewees. A tape recorder was used to record the interview only if the interviewees granted their permission. There was considerable difficulty in finding any background and historical data on the selection of SSADM as the actual documents used to evaluate and select the I.S methodology from the other methodologies being evaluated have been lost. Two of the five member panel responsible for evaluating the methodologies and both the Information Technology and Systems Development managers who were involved in the decision making processes have left the organisation.

Instruments / Equipment

The only equipment used to conduct this research was a tape recorder and a lap top personal computer for the recording of the interviewee's responses to the interviewer's questions.

Yin (1989, p. 40) identified four criteria for judging the quality of a case study research design. These include construct, internal and external validity and reliability. There are several tactics that will be used to increase the validity and reliability of the research design.

Construct validity can be improved by using multiple sources of evidence, by establishing a chain of evidence and by having the draft case study reports reviewed by key informants.

According to Nachmias (1987, p. 208) "in order to minimise the risk of erroneous conclusions, a researcher can use two or more methods of data collection to test possible hypotheses and measure variables; this is the essence of triangulation". Triangulation also helps improve reliability.

Davis & Cosenza (1988) identified four procedures that can be used to increase the content validity. These include:

- . Conduct an extensive literature review for all possible items to be included.
- . Ask experts in the field for suggestions on the inclusion of items.
- . Try the interview questions out on a set of respondents similar to the sample to be studied.
- . Using the suggestions from (2) and (3) modify the interview questions to determine if the wording of the questions is providing the responses required.

For this study the content validity can be considered as achieved if the information received falls within the theoretical framework.

"External validity: establishing the domain to which a study's findings can be generalised"; Yin (1989, p. 41). This case study does not attempt to generalise the findings to other organisations. It is up to the reader of this research to interpret the research findings and determine its applicability to their organisation.

There are two tactics that can be used to improve the reliability of the research design. Reliability can be improved by using a case study protocol and by developing a case study data base.

Procedure

Data to address the research questions were collected through interviews. The initial support from the selected organisation was gathered by the researcher contacting the manager responsible for the systems development section in the organisation to be studied. The manager was briefed on the purpose of the research and was notified of how the research would benefit his / her organisation.

Interviews were conducted using interview questions based on the research questions (see appendices for interview questions). Results were made available to all respondents who participated in this case study. Some words used in the interview questions were defined clearly so that all interviewees knew what the researcher was referring to.

The interviews for this case study were carried out using an opportunistic approach starting with staff who were most readily available and proceeding to the next available staff member. Interviews with the Chief Analyst, Quality Assurance Administrator, and Data Administrator provided the researcher with background details on how and why SSADM was selected. The Project Leaders, Systems Analysts, Programmers and review personnel, such as the Chief Programmer, could provide feedback on the strengths and weaknesses of using SSADM. It took two months to complete all the interviews. Approximately four interviews were completed on average per week.

An interview plan was developed which specified:

- the personnel to be interviewed;
- the order in which they were to be interviewed;
- the interview plans for each respondent.

A combination of open and closed questions were used during structured and unstructured interviews. According to Emory (1985, p. 161) there are three broad conditions that must be met to have a successful personal interview. "They are:

- (1) availability of the needed information from the respondent;
- (2) an understanding by the respondent of his or her role;
- (3) adequate motivation by the respondent to cooperate".

Davis and Cosenza (1988) suggest that one way of improving the strategy for data collection is to use a telephone prescreen followed by the personal interview. This will help overcome some weaknesses inherent in using personal interviews on their own.

A range of formal interactions and communications, personal observations during the study, and informal communication as a participant were some techniques used to conduct this study. Appointment times with the interviewees was recorded in an on-line diary, and field notes were made to document my impressions, thoughts, and the information as it came to hand. If the permission of the interviewees was granted, a tape recorder was used to record the interview.

As I am a participant in the organisation I already hold a significant amount of information and understanding of the organisational myths and metaphors. (Preston (1991), Orlikowski (1991))

Data Analysis

The NUDIST (Non-Numerical Unstructured Data Indexing, Searching and Theorising) software system for managing, organising and supporting researching in qualitative data analysis projects was used to assist the researcher in carrying out the data analysis. Data or information gathered was written up in a genealogical manner drawing upon the events and decisions that played a role in shaping the development of the I.S methodology within the organisation. It is important to let the actors tell their own story and so there are many quotes from them. The study will also answer the research questions within an ethnographic framework. In essence, much of the analysis will be performed by the readers of the research.

The opinions of the research supervisor and research consultants will be used to ensure that the data is collected, analysed and reported with the minimum of bias.

All the interview notes were transcribed onto a word processor either from an audio taped recording or typed in directly. After gathering all the data during the interviews and observations I then identified the major themes in the research questions and developed a

bubble diagram based on a hierarchical tree structure. There were nine major themes identified in the research and interview questions. These included the following:

- . Employee backgrounds - course experience, work experience.
- . Methodology - reasons for its selection, prototyping, system applicability, standardisation, satisfaction, tools and techniques used.
- . CASE - satisfaction, use of tools.
- . Quality accreditation.
- . Problems.
- . Customer feedback.
- . Gaining success.
- . Employees' future - career path, career progression, SSADM and future career, opportunities and its relationship to SSADM and career progress, the roles that SSADM plays in career progression.

- Work preferences - types of work, influencing decisions, group speaking, presentation skills.

Each individual interview was transcribed into a separate document. These documents were then converted to a Microsoft DOS ASCII (American Standard Code for Information Interchange) text file. These twenty-six text files were introduced into the NUDIST software system. Index nodes were defined using the main themes identified in the research and interview questions.

Each response in the document was then linked to a particular node on the index tree. The NUDIST software was then used to allow me to gather the responses from each interview case and bring together responses related, into one document. This allowed me to examine the responses to a particular interview question and to determine whether there was agreement or disagreement and how strongly staff felt about a particular issue.

Considerable effort was required by me to initially learn and use NUDIST. Several terms used, are concepts borrowed from Computer Science and Mathematics. Researchers who don't have this background may find some terminology difficult to understand. Setting up a research project on the software requires considerable effort. While this project only used some capabilities of NUDIST, it highlighted to me the importance of proper planning before putting the information into the NUDIST software.

Limitations Of The Study

The study is limited to one organisation as the intention is to provide detail, rather than a superficial study of the complexities involved. Therefore, the research findings will not be generalised for other organisations. Of course, the study will be influenced by my own experiences and views as a participant in the organisation. However by setting out my views and biases as openly as possible and by letting the other participants tell their stories it is hoped to reduce the impact of any personal bias. (Preston (1991), Orlikowski (1991))

ETHICAL CONSIDERATIONS

Participation in this case study by potential respondents was totally voluntary. The names of the respondents were only known to me, my research supervisor and a university research consultant. No mention of the actual names of respondents has been made and only the researcher and the supervisor will be aware of the organisation under study. Only alias names have been used to identify a person or organisation. Any respondent who decided to withdraw from this research was free to do so any time without being penalised in any way in the organisation. Consent to go ahead with this research was received from the Systems Development manager and interviews were carried out only with those employees willing to participate in the study. A standard letter explaining the study in simple terms along with any details that might be expected to influence their decision to participate in the research was distributed to potential respondents. The responses will be kept strictly confidential and respondents were notified of this. The Systems Development manager and each respondent signed a form of consent before any interviews took place. Since an organisation wide on-line diary was used to book appointment times and rooms with each respondent, the universities' standard referencing format for interviews was altered slightly to avoid respondents being identified by the date and time of the interview.

RESULTS

Introduction

The findings of the research conducted focus on the changes that accompanied one organisation's implementation of a structured systems development methodology (SSADM). The organisation under study is a large organisation that will be called Customer Oriented Services (COS), with offices throughout the state of Western Australia providing several vital services to the public. The section within this organisation that is the focus is the Systems Development Section. The Systems Development Section is responsible for building customised software applications for clients throughout the organisation. These applications vary greatly. They include personal, work management, asset registering, customer information, payroll, and stores systems. The software developed by Systems Development typically consists of large, Mainframe CICS / COBOL / DB2 based transaction-processing application systems that the organisation's clients use to support their major administrative activities. Some Personal Computer systems are also developed but there are fewer PC system developments.

The discussion of the results involves four major areas. These include the following:

- How and where (what type of projects) the method is used, including its future use.
- How the IS methodology was selected.
- The perceived strengths and weakness of the IS methodology (i.e. its success).

The factors that influence the perceived successes and failures of the IS methodology (i.e. whether they are technical eg. CASE, prototyping, types of projects etc.; or political eg. careers, culture of organisation).

An Historical Perspective

In the mid eighties Information Technology (IT) was beginning to have a major influence on the way organisations went about doing business. COS had gone through a major restructuring process that had taken several years, and was now looking at ways of improving the process by which information could be communicated between its many and wide spread branch offices. The most favoured information systems platform at the time within the organisation was the mainframe since it provided an adequate centralised repository of data and remote access from the regions.

Information Systems (IS) planning started as early as 1986. While the plan was, in the main, generally considered a failure, it provided valuable experience to the newly created IT branch on developing IS plans. It highlighted a major difficulty in planning for IS in that technological change was happening so rapidly, that making plans longer than three years was very difficult.

The IT branch had also been through several restructures over the period 1987-94. During the 1987 / 1988 financial year the branch comprised four sections. These included:

- . Systems Development,
- . Computer Services,
- . Information Planning and Support and,
- . Library Information Services.

During 1989 / 1990 a new section was created called Communications, while the Information Planning and Support section was renamed to Technology and User Services. By 1990 / 1991 the Library Information Services section was no longer a section of IT while the Communications section was renamed to Communications Technology.

Over the period 1988-94, the types of tasks each of the IT sections performed have changed. The Computer Services section used to be responsible for networking but by 1989 the Communications Technology section was set up to handle networking services, support, development and design. Systems Development has gained responsibility for Quality Assurance from Computer Services and Data Administration from Technology and User Services. Technology and User Services gained Systems Integration.

Before SSADM's selection the methodology that was partially being followed was SDM70. According to Peggy (personal communication):

we attempted to use SDM70 in its complete form. The methodology was so verbose that it became very difficult to use so we rationalised its use and used selected parts of the forms and used that for some years with considerably more success. There appeared some initial use of SDM70 but that waned very quickly. There were training courses conducted but very little follow up and very little enforcement.

My first experiences with the SSADM methodology came in November 1990 when I was given the responsibility of developing a small mainframe computer system. At the time there were not many other systems that were using the methodology. I can only remember one system that had started to use it but that system was cancelled by April, 1990. There were very few people that knew anything about SSADM. While we were sent for five day training courses in SSADM it is my opinion that only when you actually start using the tools and techniques do you become comfortable with them. It is not really a methodology that you can attend a five-day course in, wait for six months to a year while you complete the programming on another system and then come back to developing projects using SSADM. An analyst should not be expected to remember how to use all the tools and techniques straight away without reading the manuals to find out what the next step, task or phase is.

One reason that the SSADM methodology was not used to the extent it is now is that its use was optional since senior staff wanted to get an idea of how applicable the methodology was to the organisation. Another reason was that many major systems developed were in too advanced a stage to make use of it. It is difficult to start using the methodology in the middle of its phases, for example, starting to use it during the Physical Design Phase without producing the products needed for this Phase during the Analysis Phase.

The Systems Development section had several young Project Leaders and Analyst/Programmers at the time so rather than fix a new, largely untried methodology on the organisation the senior staff decided to wait a while to determine the type of culture that actually existed in the organisation before making SSADM the standard methodology. Though only structured methodologies were considered during the selection process there was a small group of staff that actually wanted to use Prototyping. I believe, this group tended to be made up of the more technically minded and PC development staff. They preferred to use their technical skills to prototype a system and show the user what the technology could do. This group was however, out numbered. The tools available for prototyping were very basic especially since most of the development was done on a mainframe. It meant that Prototyping was very time consuming and frustrating.

Why no attempt has been made to introduce Prototyping tools into the organisation, is still not clear. The reasons given are a lack of cheap tools, and that Prototyping can lead to developments that keep on going indefinitely.

Work and Educational Background of Employees

An employees work and educational background may have some impact on the type of methodologies, tools and techniques that they are likely to want to use.

Employees of the organisation under study mainly had training in four of the five major tertiary institutions. Eleven did a Bachelor of Business or Applied Science degree at Curtin University. Seven did a Bachelor of Business or Applied Science degree at Edith Cowan. Four did Science degrees in Maths or Information Technology and Chemistry. Four out of the 26 staff interviewed did their first computing courses at a TAFE college. Only one of the 26 graduated from Murdoch University. Several staff had more than one qualification.

The courses run by these institutions are based largely on providing students with an exposure to tools and techniques of various methodologies. These methodologies tend to be largely based on the Water Fall or structured approach to developing systems. These methodologies generally have an initial investigation, feasibility study, analysis, design, programming and implementation phase.

Students from Curtin University's Faculty of Business were, during the early and mid 80's, exposed to a large extent to a methodology developed by Bernie Glasson. According to Lionel (personal communication) "It was not a structured methodology such as SSADM. What he did was, to combine various products from different methodologies and their steps and created his own methodology. Based on the water fall approach. This included techniques such as DFD's."

Students from UWA and Curtin who studied Computer Science or Maths had very little or no exposure to structured methods before starting work. Students who attended courses at TAFE had some experience with structured methods, namely the Systems Life Cycle. The TAFE course concentrates mainly on programming and only touches on structured methodologies, and their tools and techniques briefly as part of the third year industry project. Edith Cowan University students however, especially those studying Computer Science, are exposed to structured tools and techniques during the second semester of the first year.

UWA, Curtin and Edith Cowan University students were also exposed to non structured approaches (eg. Prototyping) for developing systems while other institutions such as TAFE only covered structured approaches.

Prior experience in using structured methods varied greatly. The more experienced staff, i.e. those working in the Information Systems industry for more than seven years have used other structured methodologies such as SDM70, Method 1, Yourdon Structured analysis and design, APT, Spectrum, Jackson, and Information Engineering. Structured methods have been used more extensively than any other type of methodology (eg. Prototyping, Object Oriented Analysis). A likely reason for this could be that the last fifteen years has seen considerable work put into the further development of structured methodologies since their first introduction during the late 70's under developers such as De Marco, Gane and Sarson, Yourdon, Jackson, Warnier and Orr.

Other methodologies such as Prototyping and Object Oriented Analysis have been gaining in popularity due in part to the greater level of software support through Fourth Generation Languages and the quest to find and develop better methodologies. As new graduate trainees come through the university system the number of staff with backgrounds in these alternate methodologies will increase.

The Selection Of The Methodology

My intent in this thesis is to explore how the introduction of a structured development methodology in a systems development section of a large service organisation, changes the way its staff go about developing systems, and what the implications-intended and unintended-for organisational software development are.

There are several beliefs that exist on the reasons behind why the SSADM methodology was chosen above all the other methodologies available.

The real reason was according to Karen (personal communication):

We initially identified the requirements for our group and the way we worked and we assessed the methodologies on these criteria. This included criteria such as use, small systems, tailorable, easy to use without lots of training, and we weighted these. We gave all the methodologies under consideration to the Project Leaders and got them to vote about what they wanted to use. The Project Leaders selected SSADM, I believe it was because it was the smallest at the time. The third thing we did was contact other organisations as referees to the methodologies and asked them about what the strengths and weaknesses to the methodologies were. All the methodologies considered were structured because we felt that all the projects were well defined and the planning group were using an SSM (Soft Systems Methodology) approach to planning.

According to Stanley (personal communication):

SSADM was selected because we were very inexperienced in the development of systems and it was felt that a highly structured methodology with documented techniques was highly appropriate. The fact that it was a government based methodology, was also considered an advantage. There was some in house experience that also gave SSADM an advantage. It was a how to methodology rather than what you need to do.

According to Patrick (personal communication):

We decided it had a very good analysis and design structure and processes. It did not cover the full cycle but was considered very good in its processes. We felt that we could use what we have to cover the parts not covered by SSADM. We also felt that there was going to be a tool to support it. It was more rigorous than the other methodologies we were evaluating. The others were aimed more at the management level.

According to Clara (personal communication):

I have no idea, probably marketing I'd imagine, they probably just marketed it well, and you find many people, senior people in IT in Australia anyway are of English decent, it's an English package. It seems very strange to me considering when all the programmers are Australian. Its marketing, marketing to ex pats.

Others such as James (personal communication) said:

I don't know why it was selected. I think they are wrong because it's old technology, it's a 1970's methodology played with to pretend it lives in the 1990's.

The SSADM methodology was one of the three structured methodologies selected for evaluation by the Systems Development manager at the time. The other two were APT and one by DMR. The Systems Development manager set up a small team of five senior staff from Systems Development. Their responsibility was to meet regularly and evaluate all three structured methodologies. To evaluate the methodologies the team identified and agreed upon a set of criteria, which they believed that the methodologies should meet. Over the last seven years or so the documentation outlining the selection criteria and the final report submitted to the Systems Development and Information Technology managers have been lost. Even soft copies of the documents have gone missing.

As part of the selection process other organisations that used the methodologies were contacted as references, to get their opinions. Of the three methodologies, SSADM, had several advantages. One of the advantages was, it was already used widely by government departments in the United Kingdom. Organisations that decided to use the methodology

only had to purchase the manuals. So there was no on going cost involved in using the methodology, as with the other methodologies. Being a widely used methodology in the United Kingdom, it was believed that CASE tools would be developed shortly to support the method, which was seen as another advantage.

The Systems Development staff that worked in the organisation during the years when the SSADM methodology was being evaluated and were involved in the selection process or were consulted for their opinions on its suitability were more pro SSADM. This was further supported by the belief that if there is greater involvement of staff in the selection process there will be a higher level of ownership of the methodology selected and a greater desire to use it.

Training in the use of SSADM was also available from a former university lecturer who had gone to the United Kingdom and had become accredited to teach SSADM to Systems Development practitioners. He ran several early courses in structured systems analysis and design methodologies for the organisation and would have been a major influential member for the support of SSADM. Most of the training in SSADM was organised through an education consultancy company that organised the courses. These courses usually ran for five days. They included an overview of SSADM, and the analysis phase of SSADM. The courses ran during the late 80's and early 90's and covered SSADM version 3. Almost everyone went on the SSADM course at different times, usually in groups of ten. In 1991, some staff received further follow up training in using the design

components of SSADM. Some also attended courses that helped them convert over from SSADM version 3 to version 4.

Towards the end of 1993, almost all staff were placed on SSADM version 4 courses for five days at a time. This course included an overview, requirements' analysis, requirements' specification and design and was run in house. The course tutor had however changed from the original tutor. The feasibility stage of SSADM was never included in any of the courses as the organisation uses its own standards for feasibility.

The organisation under study has adopted a variety of development tools and techniques to ensure that development tasks are performed according to established quality plans, standards, Australian Standards AS3563 quality criteria, and in conformance with organisational goals and values. Development methodologies are understood to be both enabling and constraining: enabling in that they help project leaders and analysts by providing them with a set of tools and techniques to develop a system that meets the business requirements and constraining as they restrict the manner and method by which you can develop the system.

Systems Development is a quintessential "knowledge-based" section providing consulting services to its clients, which are staff employed by other sections of the organisation. It is organised along hierarchical lines with a pool of development staff. This pool of staff consists of four levels: Programmers, Programmer/Analysts,

Analyst/Programmers and Project Leaders, going up the hierarchy. This pool of staff can be allocated to different projects at any time depending on priority and workload constraints. Staff allocated to project teams are usually not given maintenance work while working on development projects.

In the last four years, since the decision to make SSADM the standard methodology came into force, there has been a long period during which projects have been tackled by individual developers rather than teams. Besides this career path, there is also one that exists for the data administration path. This path is more limited. There is however, no career path for those staff interested in progressing down a purely programming or technical path, to a position of Chief Programmer.

Standards and procedures have been developed in Systems Development for several years now. These standards are not reviewed or updated as frequently as they need to be which has resulted in some staff following undocumented standards and procedures. With projects becoming bigger and more complex, and the stakes higher, there is a potential for problems to become more visible and severe. The methodology consists of a sequence of stages. These stages consist of steps that in turn are built up from a set of tasks and deliverable for each stage. The method also specifies quality criteria and process milestones.

Orlikowski (1991) found during interviews that:

In SCC we believe that you should be able to field a team at a moment's notice, and that the team members should be effective in two hours. And we can do that because all our people speak the same language, have the same training, and share the same experiences. They've even heard the same war stories passed on from the trainers, our senior people in SCC. So we have a small army that can do things the same way, and they all have highly predictable expectations about SCC and the work. (p. 16)

However, this level of uniformity did not exist to any such extent in the organisation under study. Though almost all development staff were sent on SSADM Version 4 courses and the method being the standard over the last five years, very few have followed it completely.

Systematic Forms of Control: Socialisation

The use of SSADM's version 3 and 4 varied depending on the person who was in charge of a project team. Systems Development had for several years been separated in as many as four separate locations in the same building and on three different floors. These different areas each had their own project leader or acting project managers. The staff working in these different areas each followed SSADM to varying degrees. Some decided to follow Version 3 while others followed Version 4. With the change in versions, and the changes to the product names and contents, came the problems for reviewers having to review and be knowledgeable in a methodology that they did not have a working knowledge of. The location of staff has also inhibited socialisation and knowledge transfer between the

different project teams.

The Use Of Prototyping

In exploring the success / failure of the methodology in the organisation many expressed views about prototyping.

There was widespread support by almost all employees to use either a combination of prototyping and structured methodologies or prototyping on its own. Only one employee wanted to use only structured methodologies. The idea on what a prototype is varied from being a screen prototype during the design stage to full blown evolutionary prototypes that become the final system. Most felt that prototyping would be a good tool to use as part of a structured methodology.

According to Joseph (personal communication):

Preferably a prototype to me seems the best way of absorbing a user's requirements. Users mainly associate with screens and identify processes via screens. This gives them an added advantage because they see a system related to their initial idea therefore it allows change without major implication, i.e. cost, it can be thrown away, and requires minimal effort to develop. I have developed one or two systems using a prototyping technique and the results have been encouraging. Users tend to provide more information and the analyst can communicate in better terms, thus delivering a product that is more useable by the user.

Those who felt very strongly about the greater use of prototyping felt frustrated at not being able to use it due to the restricted tools available to them to conduct prototyping.

Thomas (personal communication) said that the current environment did not support the development of systems using prototyping.

In our current environment prototyping is not user friendly enough. There is no coordination between users and developers to implement prototyping. There is a lot of money wasted in the interactive business rule approach which results in many changes to programs.

The desire to use prototyping involved the belief that by showing the users what was possible, or what the developer understood their requirements to be, they would be more likely to get the system they wanted.

Clive (personal communication) felt that through prototyping "the user gets to see things earlier. It's more involved than reading lots of manuals and paper work. They get to look and feel the system. They can participate in the development." Another group did not see the use of prototyping as the only methodology or tool that should always be used.

Clara (personal communication) did not believe that prototyping could be used in all situations.

Again that is a case by case thing, there will be users where you can probably use prototyping and there will be users that have not got a clue. Prototyping is not going to make it very helpful either, because there are users who ask for systems who have no idea on what their business processes are and then you have to go right back to the methodology, back to the ground roots of not letting the users touch your software, not letting them see stuff on the screen because it's far too premature for them, for the mature user whose experienced in this yes I would prefer to use this, but for the inexperienced user I would not.

Those who had not had extensive experience in prototyping still believed that prototyping was an important tool to use.

According to Carl (personal communication):

There is no doubt about that. I have not actually used it extensively in my working life but having done the computing course I am convinced that is the way to go. The course exposed us too many case studies and people have documented them and given their reason prototyping is a far superior method. Both advantages and disadvantages were brought up but I still feel that the advantages far out weighed the disadvantages. The main advantage with prototyping is the user could see something, it's like seeing a building being drawn on paper up front and when it's being done interactively they have a feel of what the system is going to do for them. So that immediately gives them a clear idea about what the ultimate product will be like and they can also immediately propose changes interactively. Where as the other method you put all your analysis down on paper and to get one thing changed you got to change it on so many pieces of paper and that takes a long time and users generally cannot understand what you put down on paper.

Satisfaction With SSADM

While most staff supported the greater use of prototyping, more often than not they wanted to use it as a tool of a structured methodology, since prototyping on its own does not produce any documentation. There was no universally accepted idea of what prototyping was as some staff considered it screen design while others believed it to be building entire systems and gathering requirements by building quick system mock ups through a 4 GL or graphics software. There was wide spread disagreement about whether the prototype should be evolutionary and become the final product or be a model on what the final product would look like.

There were many mixed answers about whether SSADM or any one structured methodology was in use. This is surprising when it is considered that SSADM has been the organisation's standard methodology for the last five to six years. The responses received varied greatly from one respondent to another. Some strongly disliked or liked it, while others stated they would like to use it but were not getting the opportunity to use it, due mainly to the type of work they were currently doing.

There were only a few respondents who actually used most of the methodology. Most used certain techniques and left out the rest, in effect, tailoring the methodology for their own use.

Lionel (personal communication) said:

I would use some techniques. I find Entity Life Histories a very useful way at looking at entities. I can't remember a circumstance where I would say that by using the various tools and techniques they assisted me in developing systems that were on time, on budget, which meet my user's requirements, and were easy to maintain and easy to use. But I have not used any other development methodologies. I would like to use SSADM. It's the best I have seen. It's flexible since it does not force you into doing steps that are unnecessary, the main focus is producing the product. It also allows you to refine your various products as you go along. It's more iterative than the water fall methodologies.

Peggy (personal communication) felt that using SSADM would help her develop systems on time, on budget, that meet users requirements, were easy to maintain and easy to use:

If I could use them properly, but I am frustrated by the lack of commitment to SSADM and lack of commitment to the project itself. I'm sure if it could be used properly it could bring the project on budget because you could more accurately estimate. SSADM is only used as a tool rather than a basic approach. When it's used, it does assist in identifying requirements. It also assists the user understand their business better and to have a better appreciation of what has to be changed and of the requirements. I would prefer to use SSADM because it allows for the involvement of the user. Users are an integral part of the use of SSADM.

Others such as James (personal communication) felt:

that it is the complete opposite. The deliverables of SSADM are not true business deliverables, to understand the business, to do maintenance and the technical ones are not complete enough to change a view that the only way to maintain the program is to read the program.

Clara (personal communication) summed the situation up by saying:

Does anybody use SSADM, I mean really use it, no because no body has a CASE tool for it. As far as I'm concerned, although I am following or supposedly following a simplistic version of it, there is not the structure within our organisation at the moment nor the CASE tools in place to really use it. It's used in name as a skeleton of SSADM is used that is taught in all institutions, its not really SSADM. If a CASE tool was available I would like to use it, it depends on the way I would be allowed to use it. I would like to use it in a more flexibly way than it's presented and I would like my CASE tool to be able to respect that flexibility that I would want. Since there would be some places when I would not need certain aspects of SSADM, or I would like to focus my energies on other aspects such as the data base designs that may need more going through than the Entity Relationship design, but I would like to use it with a CASE tool. I would like to use any methodology as long as it had a CASE tool, since they all usually end on the same plane and follow the same methodology. Actually I do not believe SSADM is a methodology, it is just a group of techniques, and those so-called methodologies are in fact the same methodology under a different name, different marketing and different grouping.

Employee Career Paths

The research investigated the relevance of the methodology to the career paths available within the organisation. Most staff in Systems Development, feel that their career will go down an analytical path, this is despite their preference for the technical path. They felt that it would be an analytical career because "there are limited promotional opportunities down the technical path", according to Sharon, (personal communication).

Some staff such as Clara (personal communication) felt that the career structure was too heavily directed down the analytical path to project leading and towards management.

There is no career path down the technical way because you have the Chief programmer sitting up there and no body between but the analyst types and that is really ridiculous, I do not know how that evolved. Probably because management types believe that if your not a good manager then you are not a good worker and management types tend to believe that if you can't manage then you are not worth more than \$30,000 a year. I would prefer a technical career path. Honestly you get very frustrated with the analytical side because the analysis is usually the same as the technical path, staff are supposed to go from technical to analysis to management and if the analysis always leads to management I am not interested. I have never been able to see why an analyst or a technical programmer cannot be more senior than a manager because they are different jobs, the person will have a different proficiency. I would prefer a technical path. It is my experience in this business and this organisation, Analysis/Project Leading/management is just not the way to go. I don't see my career lasting here very long.

Some staff decided to select the analytical path because they felt that they were more analytically capable while others who did not like communicating with users or the politics in the organisation both from the users and Systems Development sides had a preference for the technical career path.

Clive (personal communication) said "I like the technical path. I don't like talking to users. They never know what they want and they are always changing their minds."

Thomas (personal communication) felt that his career path was going to be "definitely technical. Due to my interest in technical rather than user oriented things and because of my anti political stance on a moral ground." Unfortunately for those who are more technically oriented or dislike the extreme problems that exist with politics the only solution for them as they see it is to leave Systems Development and join an organisation that develops technical software or to leave the software development industry.

Peggy (personal communication) felt this way when she said "I see myself abandoning my career over the next few years. I am somewhat tired with the industry and there are other things that I wish to do."

Key Factors To Career Progression

One research question was, is competence or expertise in the methodology perceived as an important factor in career progression? More generally staff were asked for what they saw as the key factors to progressing in their careers. The answers reflected two differing views. Some strongly believed or wanted to believe that career progression was based on technical ability. According to Sharon (personal communication):

You need to be versatile, have a sound technical knowledge, and sometimes you need to be conniving and not entirely truthful. The interviews conducted require you to have strong technical knowledge in areas such as the programming environment, data modelling, SSADM.

During some interviews with staff who had said that their ability to do the job was a key factor too career progression, some of them, while unwilling to say it openly during the interview, commented during informal discussions that they believed that their friendship with senior staff who were responsible for allocating work or giving them the opportunities to work on good projects were important factors. Several staff felt that there were other non-software development aspects such as their personality that was an important feature of promotion consideration.

Patrick (personal communication) felt that:

They have to promote themselves using marketing techniques. Gain attention. Don't sit in the corner. Don't argue. You get promoted here by passing professional processes. It is possible to produce systems or documents quickly without regard to quality and this is what gets noticed despite the facts that there may be real problems in the subsequent systems or documents. Manage perceptions of yourself. If you sit in a corner doing good work, producing good systems this does not necessarily get noticed. The work that we do is conceptual in nature and therefore is difficult to assess. So it's how you manage the process of being seen that matters. I don't think we always promote our best people. The internal person has more advantages. Males have better opportunities than females.

The political and friendship aspects of working in groups also came through. It was especially important among staff who were involved in close friendship with their supervising or other senior staff who could influence the decision making staff.

According to Thomas (personal communication) people progress in Systems Development for reasons other than ability:

Unfortunately political involvement of who likes you and who does not rather than on merit. It is obvious from the way people get promoted and the way people get allocated analysis tasks according to favouritism, prejudice, political expediency. There is no career path except people who want to specialise in analysis. Analysis is the only way up. My main experience of this is due too not towing the political line. I don't play the political game and I don't expect to be promoted.

Some staff such as Clive (personal communication) felt that you need to know:

How to 'bull-shit' best, you have to know who is pulling the strings. It's not what you know but whom you know. You can see it happening, it's blatantly obvious, they don't even try to hide it any more.

While Clara (personal communication) said that you need to determine:

How many games are you prepared to play, being a member of the club and if you don't like something then speak up and make yourself be heard because otherwise people will trample all over you. Anyone with motivation and is interested in getting their skills up and getting paid what they are worth should not be here. They won't get paid what they are worth, there is no career path, and they won't be improving their skills because the private sector is way ahead of us. There are no opportunities in this organisation.

SSADM And Future Careers

Not all the staff in Systems Development saw themselves wanting to use SSADM in the future. Most accepted that while they worked for Systems Development they would have to use it or whatever the standard was at the time.

Joseph (personal communication) felt that "since this organisation adopts the use of SSADM as a standard then I guess I, since I am here, will be using this product. I am not familiar with other methodologies therefore I cannot comment about whether I would use any other."

Some felt SSADM was a good methodology and one they saw themselves using in the future. Lionel (personal communication) felt that "if I moved to another organisation I would recommend SSADM. Another organisation not using SSADM would not be the single reason I would not join. But use of SSADM would be a positive factor to move to that organisation. I feel SSADM is a useful methodology. Anyone who used it would have a similar ideal to my own."

Clive (personal communication) said "I don't see my future career with SSADM, it's going to go Object Oriented. Object Oriented is a better discipline. You start defining detail earlier in comparison to the water fall methodology." Other staff felt that their careers were likely to lead to the use of prototyping, contracting in a technical programming career path or by not using structured techniques but by working in close co-ordination with all users.

The Role Of SSADM In Career Progression

There was a very mixed reaction as to whether SSADM has a role to play in career progression. This was evident from the responses of staff. Responses varied between believing that being an expert in a methodology would help you progress in your career to the belief that interpersonal skills and communication skills were far more important. Staff who had worked in Systems Development for a long time believed that the interpersonal/communication skills were more important than just knowing the methodology.

Karen (personal communication) believed that "personal qualities such as application, analytical abilities, following up of loose ends, ability to think about their work are much more important than specific knowledge of a method, because you can use a method and can not be good at your job. I don't think SSADM should be a key. You should be able to progress whatever the name of the methodology." This view was also supported by Cyril (personal communication) "I think that SSADM has a medium term future. The skills required to go from a programmer to a programmer analyst are different. I think that SSADM does not play a role in career progression. It is more important to have the analysis skills. Since SSADM is not a long term methodology it is likely to be replaced by something else such as Object Oriented Analysis."

Because the methodology has not been used extensively in the past most staff felt like Peggy (personal communication) that "it has no role to play. For so many years its never been used properly and there has been a lack of commitment, that side of people's career profile has been ignored."

Some staff such as Clara (personal communication) were more sceptical about SSADM's role:

It has the role of allowing you or guiding you through getting your documentation done so that it looks really nice so that people say 'oh what a good worker you are, you really know what your doing'. They are really looking at the pretty diagrams, they are not looking at the substance, and they think you know what you are doing. These people think that gee you really did well in that project, you will be looked on favourably when a job comes along. That's what role I think it has. I have known people who have not done much follow through with projects that they have started using SSADM on. So in fact they have used a very small part of SSADM. These people are looked on very favourably because they have done initiations and big long documents but they have not followed through with them. They are considered proficient in it and they are looked on favourably by people, they are seen as capable. Where as I wonder if they were given a chance what they would do with the rest of SSADM because that has not been tested out. People are not very flexible in this organisation, once you do something once and you do ok you will be doing it for the rest of your life until you speak up about it. It helps if you have a mentor or somebody looking over your shoulder. Actually I have known some people who are hopeless on their own and need someone holding their hand every step of the way. That to me is not a good worker. A good worker is someone who can find out things for themselves with a little bit of effort. You don't waste somebody else's time and therefore costing the effort twice as much.

Thomas (personal communication) felt that there was more to career progression than being an expert in analysis or SSADM:

I think its being used as a political tool. If you use the right words and do the right thing you have a better chance in career progression. Only because analysis equates to money. If the persons' ability was the only consideration then things would be different.

According to the experiences that Donald (personal communication) has had SSADM does not play a role. "I don't think that SSADM has any role in career progression. Only if you are going for the Chief Analyst position. From my experience it does not. Promotions are based on subjective views rather than objective. They assume that everyone knows about it."

SSADM and Its Influences On Opportunities

Systems Development staff had divided opinions about whether detailed knowledge of SSADM would provide them with the best opportunities. Most of the staff felt that it either should not and would not improve their opportunities, as the other factors such as having good analysis skills were more important. This feeling was echoed by Karen (personal communication):

I don't think it should work that way. Each project should take peoples' skills and train people to make the most of those skills. SSADM is a tool not a substitute for analysis skills. It should not increase their salary. It should not improve promotional prospects. It would increase the value but only to the same extent as COBOL would for a programmer. It's how good you are at doing analysis rather than SSADM that determines your ability.

Each staff member who had made extensive use of the methodology felt that it had not provided them with better opportunities.

Peggy (personal communication) said:

I don't see that it would since I have been on training courses before and it has not helped, you need to have management commitment at least. I have better knowledge of SSADM and it has not resulted in better opportunities. Its primary purpose is for developing information systems in the organisation.

Clive (personal communication) believed that having a detailed knowledge of SSADM would not improve the opportunities to work on preferred projects:

But it would increase salary through acting opportunity. That is the way management views' it. There are no promotional prospects any way. Knowledge of SSADM is of very limited value to the organisation, because people are dodging the issues. They pretend they are following the methodology to look good but if you have a closer look at what they are doing they really don't. Also, others keep coming up with techniques outside the SSADM environment. So it has not been accepted.

Betty (personal communication) supported the idea that knowing SSADM would improve your opportunities:

I think it is just a tool and obviously you have the better opportunities to apply those tools to your projects. Yes I think you have the advantage of using the tools to develop the system, which is one of the great advantages, but it does not really matter, it also depends on your capability to handle the system, it's not just whether you know the tools or not. But definitely there is an advantage that you don't need to learn how to use the techniques. I don't think it will increase my salary but it might improve my promotion opportunities. It's more competitive with other people if you have more knowledge it's not only applicable if you know the SSADM methodology although you may have other knowledge like COBOL CICS you will have better opportunities. In certain aspects I do agree by knowing SSADM you can develop systems in a more systematic way than without knowing them. The way that they structure is more procedural and give better ideas on how to develop systems than jumping up and down to determine which methods you should be applying for this area and going to the library and searching for all possible relevant references.

Gillian (personal communication) said that it did not matter whether you were knowledgeable in SSADM:

Since, it depends on what the Project Managers want to allocate you to. I don't think that it will increase my salary. I don't think it will improve my promotional prospects but it will increase my expertise. In other organisations detailed knowledge of SSADM would I feel allow me to work on the projects I wanted to work on, if hired as an expert it would increase my salary. In other organisations it would increase my promotional prospects to a certain level. After that if I wanted to become a manager, I would need other skills.

Others such as Patrick (personal communication) also did not believe it would although personal feeling suggested it would:

My personal feeling is yes but currently no because people have not totally accepted SSADM. I don't think it would improve promotional prospects, or increase your salary but good analysis skills would but being an expert in SSADM, no I don't think so at the moment since we don't have any position promoting and monitoring its use.

Work Preferences

The overall feeling amongst staff working in Systems Development was that they preferred to do a bit of everything on projects. They tended to be interested in managing projects and people and working on larger systems. Managing project budgets however was one thing several staff did not like doing.

Betty (personal communication) said:

I would like to do a bit of every thing rather than specialist tasks. You can see more in that aspect as a whole rather than concentrate on a particular area. I think small systems is good because it is small you have to monitor many things and for me small system is more appropriate. In small systems you have to learn many different techniques, many different things, manage several tasks so you learn more. I think in small systems you learn more.

There were some, like Thomas (personal communication) who wanted to have a balance between specialisation and doing a bit of everything:

I would like a balance but would like an opportunity to specialise simultaneously. I feel comfortable with the idea of managing resources in a proper environment but I don't feel this is a proper environment. The reason being the management practices here are archaic because they are based on a rigid hierarchical structure. I believe in the peer group approach.

Jenny (personal communication) believed that she did not have the skills to manage:

I would prefer to work on specialist tasks. I would not like to manage resources since I have difficulty managing. I think you are born with skills and training can help. It does not matter what the size of the system is. I find it easier to concentrate on one thing.

Stanley (personal communication) said:

I probably prefer to work on specialist tasks. I like managing people, I like managing projects, but not budgets. I like to work on small projects because of the politics.

Some staff such as Peggy (personal communication) were partially influenced by what she believed was going to take place in the computing industry and her own experiences:

I prefer to work on everything but the industry is demanding specialisation and I see that as becoming more important. I do sometimes enjoy managing resources but at the moment I get insufficient authority over these things and because of this I don't enjoy this. I don't mind working on systems of any size.

Staff such as Clara (personal communication) felt that specialising was the way to go based on her experiences:

I would like to work on specialised tasks because if you do all round tasks you usually end up chasing your tail, taking too long, if you do one thing and you enjoy doing that one thing then it's ok. I don't mind rotating, but doing all round tasks is not productive, I have been there and done that. It does not work since the different roles require different ways of thinking. Managing, I am really not interested in it. There is more scope to specialise in large systems.

Interpersonal Skills

With regard to the interpersonal skills of the staff working in Systems Development the responses given varied between not being very confident, to more confident now than before, through to being confident. Experience of staff seemed to have the main impact about whether staff were more or less confident. Several staff reported that they were more confident with regard to their interpersonal skills than before. This has been largely to do with the individual staff member's experience. While some staff were confident in chairing

a meeting this did not necessarily mean they were confident in making presentations or speaking in groups.

Sharon (personal communication) was one staff member who felt that her confidence improved with experience:

I am not as confident as I would like to be but more confident than what I used to be. You can advise users but most of the time you can't influence users. I feel I could convince project managers and other sub-ordinate staff. As project managers are working towards the same goals as me I feel I could convince them.

Being able to prepare for a presentation or any form of communication between people is an important factor to being able to convince others, negotiate or present ideas. This belief was supported by Carl (personal communication) who said "I believe I can, I would find it a lot easier if I am not asked to do it on the spot, I would find it a lot easier if I had time to prepare and I know what I am going to say. Provided these conditions are met and I could prepare I would feel more confident."

Derek (personal communication) supported the idea that experience and preparation were important factors to going into a presentation, meeting or negotiation feeling confident:

It's something you would have to build up too. If you were not prepared before hand you would definitely be in a situation that you would feel intimidated. It would depend on my knowledge of the area. This is due to a lack of experience, not having the knowledge of what is expected in a meeting mainly because you are not sure what the organisation expects.

Based on the comments of staff in Systems Development there is a general feeling that they either don't feel confident with using their interpersonal skills or don't enjoy it. James (personal communication) was one that felt this way.

I have done it but my nature does not enjoy it. I don't mind talking in groups but don't like giving presentations. I feel confident in chairing or controlling meetings. I feel confident about influencing the decision making processes of others. Because of my experience, which might be quite different to others I feel confident in doing all these things but because of my experience again I don't have patience with others who do not know things that I think they should.

There were also feelings of frustration by some staff that their interpersonal skills were of little benefit as none of the senior staff listened to them. Patrick (personal communication) felt this way when he said

I feel that I am confident in speaking in groups, making presentations, controlling or chairing a meeting, influencing the decision making process of others, both senior and subordinate to myself. I think I know what I am doing and what I am doing is correct and I feel happy to do this. However within this organisation I don't feel confident that what I have to say is valued at senior level. I feel that they are not ready for what I have to say.

Clive (personal communication) also felt the same way as Patrick (personal communication):

I feel confident about presentations and speaking in groups. Controlling or chairing a meeting, I don't feel confident. No I don't feel confident about influencing the decision making process of others, because I can't manipulate people. I don't care about point scoring, as long as it's good for the organisation. This also applies for meetings. But that depends on what type of meeting it is.

Not everyone in Systems Development saw meetings and the use of interpersonal skills as a means of manipulation or changing people's ideas for political reasons. Norman (personal communication) was one who felt this way.

Generally I have not got a problem with speaking, presentations, chairing a meeting. If I feel strongly about a point I would raise it. You can either keep quiet and hope it goes away or in the end if your view was the accurate one and you did not say anything and the project is a failure it's still on your neck. If you feel strongly about a point you say it. Open discussion is better than keeping it to yourself. People respect you more if you put in a good system in the end.

Norman's (personal communication) attitude was also supported by Betty (personal communication):

If there is something that I would like to argue, if I have sufficient reason to support my argument I would say ok yes I have confidence to provide that evidence, and to provide those reasons for discussion, if I don't have that support then I may not have that much motion to express to people what I am trying to do. I feel comfortable in meetings.

Some staff such as Clara (personal communication) had a strong dislike towards presentations and meetings:

I hate presentations and meetings. I don't see meetings as opportunities for controlling or manipulating the meeting, I see them more as sharing information and ideas.

Satisfaction With CASE

At present the CASE tools available to staff in Systems Development don't fully support the SSADM methodology. This is because the CASE tool ADW (Application Developers Workbench) which is used, supports a methodology called Information Engineering. While some techniques such as Data Flow Diagrams and Entity Relationship Diagrams are included in ADW the shapes of the diagrams and the terminology used to describe the techniques are sometimes different. The ADW CASE tool was introduced to replace another CASE tool called Excelerator. Neither tool has supported SSADM and attempts are currently under way to evaluate a tool called Excelerator for SSADM, which allows you to customise the tools to suit the methodology. In the past other CASE tools such as LBMS were also trialled and reviewed. However despite several products having been reviewed there has been no firm commitment to purchase a CASE tool that supports SSADM. The quality of the CASE tool alone was not enough to determine whether SSADM would be used or not. While some staff felt that a CASE tool that supports SSADM was important for using it others felt they would use the organisation's standard methodology regardless of whether the CASE tool supported it or not.

A third group felt that reasons other than the availability of CASE tools affected SSADM's continued use. Betty (personal communication) felt that it was important to initially select a methodology and then select a CASE tool to support it rather than the other way around.

There are several CASE tools available, such as ADW and Excelerator. I think the Excelerators version is currently being customised to fit into the SSADM design procedures. I have not had a chance to use these CASE tools. I think you have to decide whether you are selecting the methodology first and then selecting the tools or selecting the tools and trying to look for a methodology that still fits into the tools. I think the way to go is to look for a methodology that is suitable for your organisation or your system design and then find a CASE tool that supports your methodology. I think this idea is more appropriate than the other way, which is to find a tool first and then the methodology.

As Averil (personal communication) pointed out it's not enough just to have a tool:

What CASE tool? We have ADW but I never use it. I refuse to use it because it is on one machine, which you cannot access it when you want to. I like to have access to it whenever I want to. I usually use GEM or Word Perfect. If ADW was on the network I may have used it. If the CASE tools were available to support SSADM I would definitely use it all the time. The CASE tool must be accessible by multiple users.

There are problems though with CASE as Karen (personal communication) found:

They assist a little bit but they have a steep learning curve, and they are not as user friendly, and not widely available, i.e. on a network. I would use SSADM anyway but I would enjoy it a lot more with a CASE tool. I am used to the amount of documentation since I have used other structured methodologies without a CASE tool.

Most staff believed a good CASE tool that supported SSADM and provided multi user access would go a long way to convincing them to use SSADM. At present only the Entity Modelling aspects of ADW were being used and this was mainly because the staff were told to use it.

Joseph (personal communication) did not believe that ADW was suited to SSADM:

The only tool we presently have available is ADW and I only use this tool presently for the purposes of drawing data models. I use it because I have been told to use it and because it is the only tool available in my area that allows me to draw boxes and connections, i.e. data models making it easier to change than doing it on paper. Yes provided the CASE tool is integrated with SSADM giving you most deliverables mentioned in SSADM.

Only three staff did not feel that a CASE tool was not a major factor in them wanting to use SSADM.

Problems With Using SSADM

There were several concerns raised about problems with using SSADM. These problems were related to the size of the methodology, the terminology used and the fact that it does not lead in from an IT planning module. The lack of a CASE tool to assist staff in producing the various products deterred some staff from wanting to use it.

James (personal communication) was unhappy that SSADM was being used at all:

Yes I think SSADM is a load of rubbish. It's too rigid in structure, it's too long, it does not reflect the preferred sequence of events. To use prototyping or rapid forms of systems development it would have to be tailored or cut and pasted rather than a preferred option of getting rid of it. In detail the DFD's do not reflect business processes, they do not flow information in a business sense. It enables one too easily to develop a system where the entity model derives the processes and therefor the programs rather than the processes deriving the programs and the information requirements deriving the model. Any business flow is found secreted in the Entity Life Histories or the Entity Event Matrix. Due to my experience in having to perform high-level business analysis as a first stage of systems development. The tools used there bypass 75% of the requirements of SSADM in its analysis phase and the products of business analysis are understood by both the business customers and the system developer where as the deliverables of SSADM mean nothing to the business customer and I believe it is not our place to train them in these deliverables.

The amount of time to complete a project was expected to go up according to several staff due mainly to the large number of standard and cross checking products.

Joseph (personal communication) found that

The product is very long winded. It requires too much cross validation where by heaps of products are being produced when they really mean the same thing. I feel the use of reviews within the methodology is over board.

With project funding difficult to get it was felt that the volume and detail of information that you must produce when you are working under time constraints are far too great. SSADM is a methodology that some inexperienced staff felt was difficult to use. It requires the user to use it regularly to maintain familiarity with its accronymns, terminology and jargon.

Carl (personal communication) felt this way when he said:

I do see a problem in the sense that it is getting far, far too complicated. It's hard to see the whole picture, it's hard to see how one diagram leads into another and what is the purpose of all of them. I can't see why I am producing them or what for. Right now I do not know how some techniques could be useful.

It is not a methodology that you can attend a training course on and expect to be competent in it. Karen (personal communication) said

You need to be trained in it and even the training does not make you competent in using it. It's not easy for users to participate in. It is difficult to start in say the design phase without having produced the products for the design phase. It might not work if parts of projects are out-sourced. It's not popular with the people who have to use it.

Some of those who felt that there were very few problems, identified SSADM's lack of consideration with the whole of business and IS planning as a weakness. Planning was not the only part that was not addressed by SSADM. System integration is another area identified as not being covered by SSADM. SSADM treats each system as stand alone.

Making SSADM Successful

There was widespread agreement among staff about how we can make SSADM a success. They felt that to make SSADM a success we need to have widespread commitment to using the methodology. Commitment is needed from management in the user departments for whom the system is being developed, the users themselves, systems developers and their management. Systems development staff need to be trained in how to use the methodology, its products and techniques. A combination of training at a course and by using the methodologies tools and techniques on the job in a real life situation were considered important.

Joseph (personal communication) felt that:

We need to train all the staff in the use of the product. These trained individuals must be used on projects that require the use of SSADM. This expertise needs to be held within the organisation by constantly exposing these individuals to projects and to training courses to improve their SSADM understanding. We need strong management commitment stating that individuals must use the SSADM product if it is the standard. As well management needs to support development staff when estimates are provided to clients because SSADM does require some additional work and does not allow you to cut corners. Clients are a group that find SSADM long winded. If the product could be tailored or cut down some how with the essentials then it may help in being more successful. The product needs to be marketed across the organisation prior to it being used.

Quite a few staff felt that a CASE tool was important. Having a CASE tool on its own is not enough however since additional hardware support is usually also needed.

Better marketing of the SSADM methodology needs to take place so that the users are made aware of what the method and products are, why we produce them and for what purpose. Karen (personal communication) believed that "the people using SSADM must want to use it and see the benefits in using it. Management must be committed to using it and there must be some level of strong encouragement to use it. Customers must understand what we are doing and why we need to follow a methodology. The availability of the methodology must be improved and CASE tools need to be available to support it. The concept of tailoring and learning from the tailoring must be understood by everyone who uses it."

The need for having a centre of expertise or an SSADM support group was also mentioned by a few of the staff. Carl (personal communication) believed that:

we must have a centre of expertise, someone who knows what SSADM is about or a group of people whom other people can consult. It must be supported by a CASE tool. All products that come out of SSADM must be properly documented for reference by other people, and it must be enforced by senior managers.

Control mechanisms need to be in place to properly monitor whether the SSADM methodology is being used. Patrick (personal communication) felt that

Preparedness of management, of IT management to understand and promote a structured methodology. An acceptance that this is a professional way to develop professional systems. An acceptance that we are a profession and that we do have professional standards that we don't deviate from. The organisation needs to have Business Plans from which IT systems may be developed. You need a person or people (Project Managers) to provide consistency, promote and monitor its use at all times. A specific start up of projects, in the format of a management meeting with all players, which give a clear indication to the users that SSADM will be used to produce the best product. It must be supported at all levels of IT including technical consulting areas such as User Support and Computer Services. We should use it to produce our own systems.

Other factors considered by staff to be important to SSADM's success was the need for a sample system that used all the steps. This system could then be used by staff as a case example. The need to tie in SSADM with the systems developments section Quality Management system and the standards and procedures were also raised as factors important to SSADM's success.

Tools and Techniques Used

Systems development staff within COS used mainly a subset of techniques within SSADM. The number of techniques varied from one developer to another and the type of system being developed. If the system was expected to be complex or take a long time to develop staff would be more likely to use many tools and techniques within SSADM. As many staff were working on maintenance and enhancements to systems over several years there were less staff that reported using all or most of the techniques. The most commonly used tools and techniques tend to be the ones that most Computer Science and Business Information Systems graduates were trained to use during their university degree's. These include the use of Data Flow Diagrams (DFD), Entity Relationship Diagrams, Pseudocode, prototyping, Normalisation and Data Dictionary. Other techniques that are particular to SSADM used, include Entity Life Histories, Entity Event Matrix, Problem requirements' Catalogue, Effects Correspondence diagram, I/O Structures, User Catalogue, Data base access paths, Entity descriptions and elementary function descriptions.

The main reason given by staff for using a particular tool or technique was how confident they felt with using a particular technique or how essential they found it to be. The level of confidence would come from whether they have used a certain tool in the past and have found that it worked. Those who came across techniques that they had difficulty in understanding or using or they could not see the benefits in using were the ones they were likely to leave out during the tailoring process.

The size and complexity of the system being developed appeared to have a major impact about which tools were used. Clara (personal communication) felt that financial systems were one category that SSADM would be suited to.

I try to use the basic parts, normalisation, the Entity Life Histories, data model part, the core parts, but not state indicators. It may be worth while to use state indicators in some parts but honestly when your entities are going through birth, retrieval and death it is not really an issue. Tools, ADW, a stencil, a pencil and a piece of paper. I have not seen anything satisfactory in particular for data flow diagrams. These techniques are the only ones available to me. I am more familiar with them, they are the ones considered the absolute required, the absolute deliverables one must have, the rest are just peripherals. It would depend on what I came across, there would be times when I would use other techniques that I would do on scrap pieces of paper that was not filed or reviewed, which were done for my benefit. They were not filed since if they were it means that more things would have to be reviewed and checked.

Betty (personal communication) on the other hand felt that SSADM could be used for any system:

I think DFD's is the most common technique to start with to develop your logical and physical DFD's, and context diagrams. These are the very well-known techniques to be used to develop systems, they make you aware of how the system works, you can easily trace what's missing and what is required by the users. Physical and logical ER-models are another product that I usually produce.

Only a few staff have attempted to use SSADM in its entirety. Peggy (personal communication) was one of them.

I followed the task list provided. I used it as a guide. I use most of the Data modelling, DFD's, Entity Life Histories, entity descriptions. I used it totally. Normally I don't use all techniques because there is a cost, and unless my project participants are willing to pay the cost I just use the most important ones. Most of the techniques are common to SSADM but not exclusive. The core methodology is the same but the approach to using the methodology is different. I would use it for all IT asset creation projects. I am not sure whether it has a particular relevance for planning projects.

Tailoring SSADM

There were many differing opinions about what extent and who should be involved during tailoring the methodology. Several staff felt that a template should be used along with the experience of developing similar systems in the past. In Systems Development a lot of the time projects are carried out with one or two person project teams. The roles of the staff on any particular project change during the different phases of the methodology. Chances are the developer who produces the business specification document during the analysis phase will also do the screen designs, program specifications and data base design during the design phase. A project member can be the Systems Analyst, Project Leader and Programmer. Severally staff recognised the need to have some standard since without having some essential products needed by most systems there will be no documentation to show what was required and what impact the changes have on the data model.

Clara (personal communication) said:

I think the analyst should to be able to choose. There has to be some sort of standard otherwise some people will just ride all over it. You should have some form of DFD's and ELH's and must have an ER diagrams but the state indicators maybe not. The analyst should determine it along with the Project Manager or Project Leader and whatever review group should have a look at it and say whether there is a need to have more documentation needed to fill the story. The analyst should be initiating the action about what products should be used and the Project Leader should provide some input if they feel that some techniques are needed.

Karen (personal communication) felt that:

We should have a family of tools and techniques and each of these should be tailorable but I think we should work to templates so we can standardise across similar types of projects. However tools, techniques should be tailorable using a technical review and a post implementation review to determine the successes or failures. The person responsible for a project should make the final decision but should listen to advice from people such as the Chief Programmer, Chief Analyst, and Data Administrator. The Project Leader is responsible for the quality of the system produced and every one else is just there to support the Project Leader.

Clive (personal communication) felt that there was a need to fix the tools and techniques to be used:

Preferably fixed otherwise there is no standard. It should be controlled. If they don't use the standard they should be making this decision after consulting management. The main reason is for the hand over of the project. If it's at the project level there should be a methodology person who decides. At top level they should customise it. The person who looks after the methodology and CASE tool should decide.

DISCUSSION

The previous chapters have described several key experiences of Systems Development staff within COS before the deployment of SSADM to seven years later, and examined the influence of factors both inside and outside the control of the Systems Development section. The results indicate that, contrary to certain expectations in the literature (Hawryszkiewicz (1988), Hodge et al. (1986)), a structured methodology such as SSADM, which is receiving widespread use in the United Kingdom's government departments, is not the single most important part of building successful systems. Instead, we see the emergence of factors such as politics, interpersonal skills, culture, resistance, group dynamics, symbols and world views. Keen (1981) found that

There have been few studies of the political aspects of information systems development. The topics are rarely discussed in textbooks and even the literature on tactical implementation deals with it only peripherally. Yet when one tries to reconstruct or observe the progress of any major project, this is an obvious and important feature. It is absurd to ignore it or treat it as somehow an unsuitable subject for study or for training MIS specialists. (p. 31)

These results, although from only one organisation, highlight, yet again, the fallacy of believing a methodology alone could cause the development of successful systems. Technologies are built to achieve certain goals at a given time. The productivity tool's particular form and functioning reflect the interpretation of COS Systems Development (SD) management strategic intentions in the late eighties. Within COS, the SSADM methodology was introduced into the development process to augment and automate newly developed

work practices.

Implications For Education and Training

The research results, however, do indicate a need too:

- . continuously improve the general IS knowledge of all employees,
- . improve the IS personnel's organisational knowledge,
- . end-users and IS personnel's need to be educated so that each can be made more sensitive to the other's problems,
- . end-users need to improve their IS product-related skills and their technical skills,
- . regular periodic needs assessments need to be conducted.

The need for extensive training in the methodology may not be so important. Staff development in the above areas would help staff understand the content of the methodology. Technical understanding is only one part of working with a methodology.

These findings are supported by Nelson (1991) who also found that the above listed needs were lacking among IS and end-user personnel. In the past COS has had to make some difficult decisions regarding training, due in part to the amount of money available in the training budget. This resulted in large groups of Systems Development personnel being sent on training courses at a time when they did not need those skills. Due to the limited availability of certain courses IS staff attended courses with the expectation that

skills gained would be used soon after the course. The result was that the newly learnt skills may be lost unless regularly practiced. While course notes are valuable aids, time does not always allow IS personnel to read up on particular topics.

The fact that only one employee within COS's SD section has completed a higher degree ie Masters, may highlight the need for universities to better promote the value of completing a higher qualification in the Information Systems field.

Based on the findings of this study, it appears that the organisation needs to pay more attention to IS-related education for all employees, whatever the functional area may be. It appears that IS and end-user personnel need to know more about issues such as how to use information technology (IT) and IS for competitive advantage, the potential for IS / IT within the organisation and the fit between IS and the organisation. With the onset of increased competition both in the business and IS services sector, organisations need to pay closer attention on how they can work together better to continue being competitive against similar external service providers.

The findings also suggest that there be a lack of organisational knowledge in IS personnel. IS personnel need to know more than just the objectives and goals of the organisation. IS personnel need to know about the environmental constraints that the organisation operates in. These may include competition, government regulations, relationships with customers and suppliers. IS personnel also need to know about the

organisations business processes and business rules. "The notion that it is useless to bother putting more than minimal training dollars into IS employees because they may leave often becomes a self-fulfilling prophecy." (Bartol and Martin (1982) are quoted by Nelson (1991, p. 514))

The search for skill enhancement and knowledge is also motivated by a desire to reduce the communications gap. Steps are currently under way at COS to help the SD section become more sensitive to understanding the business through external consultants.

The results of this study indicate that it is sometimes difficult for end-users to understand what SD staff are doing as part of the development process. Determining the users level of understanding can be a very subjective exercise as some may think they understand but might really misunderstand. The difficulty in understanding operates both ways. SD staff have in the past misunderstood end-user requirements that have resulted in several information systems being developed which fail to meet end-user requirements.

Argyris (1976) is quoted by Nelson (1991, p. 515) as claiming that:

Research suggests that even if people become aware of the other's views, if they became aware of new behavioural alternatives, if they accepted these alternate behaviours and even if they learned them well, they still would not be able to use them in everyday life. The unfreezing of the old and the development of new values and skills is very difficult.

With end user involvement in systems development on the increase it seems logical that the knowledge transmitted to them include the basics of what is possible, and the first steps on how it can be done. On occasions end users need to produce ad hoc reports. By providing end users with the proper training to improve their product-related skills may be considered a form of preventive maintenance. At present end user staff with basic computing backgrounds are building small systems that the SD section at COS is being asked to maintain. Improving the technical and IS product-related skills of end users may help reduce maintenance problems for SD personnel who may be expected to take over the maintenance of end user developed systems.

Nelson (1991) suggested a five-step approach to conduct periodic needs' assessments. They include the following steps:

- . carry out skills / knowledge needs assessments,
- . decide on requisite areas of significant deficiency,
- . decide upon appropriate training / education programs,
- . implement training / education programs, and
- . conduct training / post-education assessments.

O'toole and Mitroff (1989) are quoted by Nelson (1991, p. 515) as claiming that:

The distinction between learning as training on the one hand, and learning as education on the other, is crucial to an understanding of the business school's current dilemma. Training is directed towards learning how to do the same things in the same way, the "right way"-that is consistent, and thus measurable. Education is directed towards learning the analytical and conceptual skills needed to cope with different tasks-that is, the skills needed to cope with ambiguity and change. One might say that training is concerned with facts and techniques, while education is concerned with developing the habits of mind needed for understanding (and as preparation for lifelong learning).

Implications For Methodology Selection

The selection of the SSADM methodology was largely dependent on a few key personnel. These included the IT and SD managers, the methodology evaluation and selection panel and the Project Leaders. With joint software development contracts in place, game playing may also have taken place as each party attempted to influence the decision makers about which methodology should be selected.

The findings on how COS selected the SSADM methodology provide a basis by which other organisations considering the selection of a development methodology may approach this difficult task. With any decisions there may be people for and against the outcome. Using an approach that involves people who may be considered as independent with an objective evaluation procedure may reduce the likely hood of disputes between parties with opposing views. The decision to select SSADM occurred during a time of

change initially in the structure of the SD section and then a potential change to the work practices of SD from an external management consulting firm named DMR who had plans to jointly develop software using their own in house developed methodology. The methodology was one of the three selected for evaluation.

One of several weaknesses with this approach to methodology selection though is that depending on the circumstances that may exist in an organisation an apparently objective approach such as selection through weightings can easily be altered to suit the desired option. For example if the situation that exists in an organisation is one where the external organisation involved in the joint software development contract plans to take over the software development through an outsourcing process then those performing the evaluation, who may not be in favour of outsourcing, could weight the criteria in such a way as to favour one of the alternate methodologies which would reduce the likelihood of outsourcing. The situation that has just been described does not appear to be the case as far as COS is concerned as there was research carried out on training and support available, referees for the methodology and feedback from Project Leaders was sought.

Implications For Prototyping

In exploring the success / failure of the methodology in the organisation many staff expressed a desire to make greater use of prototyping in combination with SSADM. Prototyping at COS is used to some extent but the type of prototyping that can be carried out by SD personnel is largely limited by the hardware and software technology available. With mainly mainframe systems making up the bulk of systems developed, prototyping is largely restricted to what is called mock up or illustrative prototyping. At present COS does not have the software to support prototyping to be able to do simulated, functional or evolutionary prototyping.

Doke (1990) described the various forms of prototyping as follows:

- Illustrative: produces only mock ups of reports and screens.
- Simulated: simulates some system functions but does not use real data or a database, model not implemented.
- Functional: perform some actual system functions and uses real data and/or a database, model not implemented.
- Evolutionary: produce models that become part of the final operational system. (p. 172)

COS does not follow any formal prototyping methodology. Based on the respondents comments, the decision about when prototyping should be done is largely a subjective one. Prototyping is usually conducted at the system design stage at COS, with structured analysis techniques being used to gather user requirements.

At present there is widespread support for prototyping across SD personnel. There are however mixed feelings about what form prototyping should take. Some felt that they would like to produce throw away prototypes while others preferred evolutionary prototypes that would eventually become the final system.

The decision to develop a system using a prototyping approach should be arrived at through some rational process that, according to Voltmer (1989) may include the following:

- . senior management commitment that a system development effort is required,
- . a belief in your own ability that you can obtain the required resources needed to get the prototyping job done,
- . evaluate other approaches to determine the best approach, and
- . understand the corporate needs, culture, priorities and policies.

For prototyping to be successful at COS the idea must be sold by the project manager from the lowest ranked person whose endorsement can make or break the project to the top end of the corporate hierarchy. "The hierarchy can kill your effort politically, and

the 'loerarchy' can subvert the effort technically." (Voltmer, 1989, p. 25)

Implications For Satisfaction With SSADM

The SSADM methodology is currently not being used as widely as expected with a methodology regarded as an organisation standard. This may imply that the staff using it are not completely satisfied with it. The COS results, indicate that SD personnel are divided about whether they should be using SSADM or another methodology. This division exists throughout the SD hierarchy. Although officially SSADM is the standard development methodology, some senior staff who need to support it to guarantee its future use are not supporting it. This has left some SD personnel frustrated by the lack of commitment to SSADM. While others are hoping that they can influence the decision so that it can be changed.

Towards the end of 1993 all staff were offered the opportunity to attend a week long course that covered SSADM Version 4. This course included both the analysis and design stages. Despite the training there has not been any increase in support for SSADM. At present alternate approaches to developing systems through process modelling techniques are being tried out in an attempt to better understand the business processes and business rules.

With the onset of Client / Server development at COS, further evaluations may need to be carried out to determine whether a structured approach to developing systems or an alternate approach such as Object Oriented Development, Rapid Application Development (RAD), or prototyping is most suitable. The belief that using one standard methodology for all systems developed is now beginning to be rejected. Senior staff in SD are starting to recognise the need for a standard family of methodologies. At the start of a project a decision will then be made to determine which methodology is best suited for the development

Implications For Employee Career Paths

The research investigated the relevance of the methodology to the career paths available within the organisation. The COS findings show that there is considerable dissatisfaction among IS personnel regarding the career paths available to them. The majority of IS personnel felt that there should be more than one career path available to them. Preferably a technical path that helped them progress up towards a Chief Programmer position.

Of the twenty-six SD personnel interviewed for this research six have left COS to obtain promotions with other organisations in the last year. A further two more employees have joined another computing section of COS. Four of the eight have gained promotions following a more technical career path.

SD personnel have shown the same concerns about their career paths as Ginzberg et al. (1988). The COS results show that staff feel that they have to go down the analysis path towards Project Leading and management because they believe that this is the only way they can earn a higher pay rate and promotions.

COS has a layer of senior-level in-house consultants as suggested by Igbaria et al. (1993). They perform duties such as Chief Programmer, Data Administration, Chief Analyst, and Project Managers. While a path does exist down the Data Administration path it is very restricted. Besides this the only other path is the one that goes from Programmer, Programmer/Analyst, Analyst/Programmer, Project Leader to Project Manager and Management. This career path corresponds to the four stage model of professional careers described by Ginzberg et al. (1988). Using this model Programmers fall under stage one and stage two, Programmer/Analysts and Analyst/Programmers into stage two and stage three with Project Leaders in stage three and the technical consultants and management in stage three and stage four respectively. COS does create temporary in-house consultant positions depending on the present needs of the SD section. These positions may be created for jobs such as setting up Client / Server or Project Management development projects.

Implication For Career Progression

The results of the interviews with SD personnel working for COS show an alarming trend for management and one that needs to be addressed quickly to maintain the integrity of the interview, selection and promotion process. SD personnel believe that the problem started by the allocation of higher duties to certain employees. The higher duties resulted in certain selected staff receiving additional pay for doing duties above their present level. An employee could be given the opportunity to perform duties (at most usually two levels above their existing level) at a higher level, receive extra pay and gain experiences that would assist them when promotions were advertised. The decision to allocate higher duties being largely a subjective decision based on the comments of an employee's supervisor or on a decision by the managers based on the difficulty level of the work to be completed.

Not all staff have received the opportunity to perform higher duties. The reasons for the discrepancy vary depending on the employee interviewed. The main concern for management was that negative comments were given from employees from differing level in the Systems Development hierarchy, and from employees performing the higher duties. Due to a fixed hierarchy structure with low turnover because of the recession and the lack of promotion opportunities in other organisations, promotion opportunities within the SD section have been rare. So the main opportunities for staff to gain more challenging duties were by performing higher duties. An example follows: A programmer is allocated to perform some system enhancements to a system. While performing these duties they

received a higher duty allowance and were given an opportunity to act as a Programmer/Analyst. To assist them to make all the program changes an additional programmer was allocated to the project. The acting Programmer Analyst was then given an additional temporary, pay rise and their position was upgraded to Analyst/Programmer for the duration of the changes. This left the allocated Programmer feeling that he was being unfairly treated as there was only a year's difference in experience between them.

On another occasion a Programmer was allocated to a project, given a two level increase in position to Analyst/Programmer and supervised another SD employee with more experience. While in some circumstances these situations are unavoidable, examples like these can sometimes leave employees feeling that they have been unfairly treated. This can then lead to increased turnover if there is a belief of unfair treatment by management.

Implications For SSADM and Future Careers

The findings show that SD personnel at COS don't see SSADM as a vital part of their future careers. To them SSADM was a methodology, which they would use only if it was the standard methodology they were meant to be using. Only one SD employee said it would be a methodology that they would look for if they joined another organisation. Since the experience level in other development methodologies were low, some found it difficult to say if that would be the methodology they would want to use in their future careers.

Most employees had differing opinions about whether it would be the methodology they would see themselves using in their future careers. SSADM as a methodology, while extensively used in the United Kingdom's government departments has not received as widespread use outside the UK or Europe. At present there are only a few local sites that actually use the methodology. So the likelihood of staff who leave COS using SSADM in another organisation is very low. With SSADM's use under long term doubt employees need to be adaptive and support the current organisational standard rather than become too attached to using only SSADM in their future careers.

Based on present trends at COS with the advent of Client / Server technology and a push towards the use of methods and techniques that produce products that are directly verifiable by business personnel it would appear possible that SSADM may not have a long term future as the only development methodology in use at COS.

Implication Of The Role Of SSADM In Career Progression

SD personnel were divided on the issue regarding whether SSADM does have a major role in career progression at COS. Other personal qualities such as analytical abilities, interpersonal, communication, group dynamic and leadership skills were considered more important than being an expert in a methodology by some employees. The views are partly consistent with the views of text books and research on systems analysis, such as Hawryszkiewicz (1988), and Hodge and Clements (1986).

Some staff in SD were very cynical in their views on the role of SSADM. They believed it was being used as a political tool to convince, in combination with friendship with senior staff, as a means of progressing by stealth. In the opinions of these employees, SSADM was being used as a tool that made the documentation of a system graphically 'pretty' and bulky. As these systems were not continued through to completion, the personnel who produced these big reports were not expected to produce these systems using these documents as other staff were given these roles. SD personnel who believed this, used this reason for explaining why some staff of equal experience and perceived ability were given opportunities to perform higher duties and be paid more in comparison to those who had not.

Based on the way selection for interviews and the interviews themselves are carried out at COS people who are experts in the organisation have a slight advantage provided their analysis skills are of an equivalent level to those competing with them for promotions. A person who has good analysis skills can learn a methodology if they were selected for promotion, while a person who just knew the methodology but has weak analysis skills will have difficulty in producing products from the methodology that reflects what the business requires.

Implication Of SSADM and Its Influences On Opportunity

Most SD personnel felt that while knowing the organisation's standard methodology, there is still no substitute for good analysis skills. A lot of the time it is very much dependent on making the most of the opportunities that you are presented with.

Being good at SSADM does not, have any major influence in an employee gaining any additional opportunities. Other factors such as interpersonal skills, demonstrating leadership ability whether this is by leading a project team or organising a social lunch or morning cakes, sharing good relations with senior staff, having a mentor, who can influence the people who could give you better opportunities, all help.

It is one thing to be good at analysis but people need to market themselves a lot more and simultaneously be able to support their message with personal ability otherwise they will lose credibility and their career will suffer.

Implications For Work Preferences

Some findings discussed here may also have implications for related organisations whose practices and policies resemble those of COS. In particular, the SD sections of service providing organisations may have similar work processes, and may be contemplating or implementing structured development methodologies. Consequently, we may see similar situations and work practices in operation.

Personnel at COS preferred to perform different types of tasks as it gave them the opportunity to use a variety of their skills. They felt that the task variety needed to be sizeable chunks so that they gained enough experience on that task. These findings were consistent with the research conducted by Cougar (1988), as he had also pointed out that most people needed to use a variety of their skills to keep their motivation levels high.

There were many employees within Systems Development (SD) who wanted to manage people / projects / budgets. Some staff expressed a desire to have a greater level of autonomy so that they could use their own judgement and make decisions on their projects.

Gaining on the job feedback from the job itself can be difficult at times, especially if there are several political games being played by the users as described by Keen (1981). The political game playing can lead to a higher level of frustration for SD personnel who may be trying to implement a system. The increase in negative feedback from game playing can result in system failure. With game playing being a continuous threat to successful implementations senior staff need to provide feedback as regularly as possible to help maintain a high level of motivation for project team members. An increased level of feedback whether positive or negative is needed preferably from doing the job itself and if not from the job, feedback should be provided by senior staff overseeing the completion of the project. Staff in SD felt that a greater level of feedback was needed over current levels.

SD staff expressed a desire to work on systems of differing sizes. Some wanted to start of working on small systems and then as their experience levels increased to work on medium and large systems, as the task significance and the challenge were greater.

Implication Of Interpersonal Skills

One of the more important skills that systems analysts need to have is interpersonal skill. Analysts frequently find themselves attending meetings with senior management from end user departments and from SD to discuss project issues, gather data or just explain project progress. With systems sometimes crossing departmental boundaries the Systems Analyst needs to be able to negotiate, influence the decision making process or control / chair meetings. A Systems Analyst lacking these skills may end up facing an implementation failure if the games as described by Keen (1981) are played, and they are unable to negotiate, persuade, present their ideas or resolve the issues.

Many staff felt that they would lack the confidence or would be nervous if they were in a meeting in which they were required to carry out negotiations or influence the decision makers. The interpersonal skill levels of staff varied from one employee to another. The more experienced staff showed a higher level of interpersonal skills than the less experienced staff. The important point however is that staff believed their interpersonal skills were improving with experience.

The circumstances played an important part regarding whether staff were more confident or not. SD personnel felt more confident with their interpersonal skills if they could prepare before hand and if they knew the people they were communicating with.

While some staff felt that they were competent regarding their interpersonal skills it did not necessarily mean that they enjoyed using them. A person who was competent in producing presentations or chairing meetings did not mean that they enjoyed doing it.

A high level of competence in one area such as presentations did not necessarily translate into confidence in chairing meetings. Those employees from a country of origin with a non English speaking background usually felt lower levels of confidence than those from an English speaking background.

Implications Of The Satisfaction With CASE

The findings show that SD personnel feel that their existing CASE tool does not provide them with adequate support when it came to producing the SSADM products. The only CASE tools available to staff supports a methodology called Information Engineering. While this methodology supports some products produced by SSADM such as Data Flow Diagrams and Entity Relationship diagrams many SD personnel either did not use it or did not want to use ADW. Besides not supporting SSADM, staff in SD found that it has a steep learning curve. Present usage levels of the tool had more to do with it being a compulsory requirement by Data Administration that data models are recorded in ADW then it being a tool used for productivity improvements. With question marks hanging over the long term use of SSADM, management have held off making a decision to replace ADW until other related issues such as Client / Server development is decided.

At present several requirements for productivity tools such as CASE to be effective are not in place. While the personal computer (PC) that ADW resides on performed adequately, there was only one PC with ADW installed on it. It was located on a separate floor to the majority of users who would require the software. With one copy of ADW, bottlenecks sometimes occurred when more than one developer needed to use it simultaneously. Some tried to use alternate graphic software but were quickly notified of SD standard practices that required them to use ADW.

Since SD only had some modules of ADW a convoluted method was developed to convert the information stored in the ADW encyclopaedia into a form it could be used for physical design purposes.

The ADW tool while providing support to a limited extent did not address the real needs of developers at COS. No formal training was ever provided to the SD staff. Most staff learnt to use the tool on a piecemeal basis and support was usually reliant on a couple of Data Administration staff who knew the product. Delays usually resulted when problems occurred and they were not available.

The ADW tool provided reasonable support to the Data Administration group but failed to meet the developer's most critical needs as a productivity tool to support SSADM.

Ferguson (1983) raised several issues that productivity tools, such as CASE, need to address for an organisation to gain maximum benefits from it. Unfortunately for COS most of those recommendations have not been implemented resulting in an under utilisation of the tool. There has been a lack of commitment by management due partly to the great uncertainty that exists over whether an organisation should tie itself into any one methodology or CASE manufacturer.

Implications Regarding Problems With Using SSADM

The findings suggest that SSADM is a reasonably difficult methodology to use without adequate support from a CASE tool which supports the methodology. Several areas that SSADM was particularly weak on included, the missing links to business planning and IS planning, lack of good CASE tool support. The number of cross checking, products created additional work for the Systems Analysts and some felt that there was little benefit in them. SSADM has several review points and some SD staff felt there were too many reviews. The larger number of acronyms meant that those who were not regular users of the methodology would need to continuously refer to the glossary to find their meaning.

SSADM is a large methodology with many steps, methods and products most of which had to be produced by hand or by another means due to lack of support from CASE. Some staff found it difficult to see how the products could be of any benefit to them.

The amount of additional work needed to produce the products had some staff concerned as they felt following the methodology in total could lead to blow outs in project budgets and time constraints. SSADM is largely a methodology with products that some SD staff felt they would have difficulty learning or using. Its one that staff need to continuously use to maintain their knowledge and expertise in. If SD developers have had difficulty producing the products then it's quite possible that end users may have difficulty in confirming the products produced are what their business requirements are, without attending an SSADM, for end users course, beforehand.

SSADM can be considered a methodology that falls under a category that Checkland (1981) calls a hard systems methodology. It covers rational aspects such as user requirements, Entity Models and other products that may be considered important to develop a system. Researchers such as Keen (1981) point out that information systems development is sometimes more political than technical in nature. Other researchers such as Romm et al. (1991), Markus (1983), and Grover et al. (1988) have all examined the existence of the non technical aspects of developing systems. SSADM ignores some irrational events that can occur during a systems development, such as game playing and treats the process as a totally rational event. Other methodologies such as Soft Systems Methodology, Checkland (1981), deal with these social, political, human interaction issues that, if not handled correctly could lead to a failed implementation.

Implications For Making SSADM Successful

The findings show widespread agreement among staff on ways of making SSADM successful. Being sent on a training course in how to use the methodology alone will not ensure staff are competent in using the methodology. There needs to be continuous reinforcement of the method and techniques by having the opportunity to practice them in a real life system.

When SSADM was originally selected and implemented there were people that were part of the selection team that could support staff when they needed assistance or someone to review the products. This role was performed by the Chief Analyst, but over the last three years no one has performed this role.

At the end of 1993 all staff in SD who wanted to attend the SSADM V4 course were sent but since then there has been no on the job experience or additional training. This has left most staff wondering about managements commitment to SSADM. Both on the job experience and follow up training and support is needed to keep refreshing the memories of staff.

Gaining a greater level of commitment for the use of SSADM and improved marketing to end users and user management were also seen as important, since SSADM is believed to result in an increase in development cost and time according to those interviewed. SD personnel felt that their own management need to stick by them and support them even if it costs more time and money to build the system using SSADM.

The implementation of a CASE tool to support SSADM was seen as an important part of convincing more SD personnel to use SSADM. At present only one copy of SSADM Version 4 is available for all staff. For SSADM to be used more widely the manuals need to be made more available, possibly one copy for each project team.

Tools and Techniques Used

Personnel at COS's SD section used a small subset of SSADM's large set of methods, tools and techniques. The usual reasoning behind this was that they used the tools and techniques that they were most confident with. These techniques corresponded with the techniques they were taught at university or college. The standard set of techniques and products produced usually included Entity-Relationship models (standard requirement of Data Administration), data base field names, descriptions and table structures (for DB2 tables). Data Flow Diagrams was also a reasonably common product produced.

With a maturing IT branch more time is being spent by SD staff performing enhancements and maintenance that don't use many techniques from SSADM. Project Leaders who mainly developed large mainframe systems were the ones that usually used most or all the techniques in SSADM. Most staff did not believe that the techniques used, will help them identify problems at an earlier stage and there by reduce total costs. Instead they used a subset of tools because they felt to use all of them would result in increased development costs.

Tailoring SSADM

The findings for COS show that SSADM has been tailored to some extent on most projects produced by SD. Almost none of the design aspects of SSADM are followed since SD personnel use their own standards for design. Before the start of each project a Project Leader is allocated to a project and as part of their initial tasks they develop a Quality Project Plan and Quality Phase Plan. These plans document what products from the SSADM methodology will be produced, when they will be produced and who will review them. The Quality Plans are then reviewed in consultation with a quality review team, usually including a project manager, Quality Assurance Administrator, and Chief Analyst. A system Review Committee consisting of key user representatives also review the products produced.

Staff in SD were divided about who should be included in the tailoring process but the need for having some standard products for all projects came through. This was believed to assist when it came time to hand over the project to someone else.

The use of a template and a family of tools may be useful as it would help Project Leaders by giving them a starting point rather than having to perform the tailoring process on the entire SSADM methodology every time a project is started. This would be especially useful when small systems are being developed.

SSADM's Impact On Staff and The Organisation

The rich picture depicted in figure 1 summarises the historical developments involving the selection of the SSADM methodology and its impact on the SD section. The top of the diagram shows the selection of the SSADM methodology. This selection process took place at a time when the SD section had several new inexperienced staff. The decision to evaluate and select the SSADM methodology may have resulted in some conflict between the IT and SD managers since there was an existing joint software development contract with DMR. There was also some conflict between SSADM, CASE and the design phase since there was little support for these areas.

There are two basic career ladders that SD staff were most interested in. The analytical and the technical career ladders. The analytical path involves social interaction between the users and developers. This requires staff to have good interpersonal skills. Having detailed knowledge of data modelling, data architecture, design / implementation tools, SSADM and CASE. While these skills improve the value of the employee to the organisation to a limited extent, in the main it provides staff with project political power.

The technical career ladder is currently non existent. If it did exist, it would provide these staff with technical political power. While almost all staff in SD have been taught SSADM on SSADM courses there is some conflict that does exist regarding whether the training is relevant when you don't make use of the knowledge. The version of SSADM used, is customised following meetings between a Quality Assurance review group and the Project Leader. The customised methodology is then used by the systems developers to gather information from the end users on the information systems development project. SSADM is currently mainly used on large systems with more than one SD employee per system.

At present there is still some disagreement among staff whether SSADM is just suited to structured projects or whether it can be used on unstructured projects and problems.

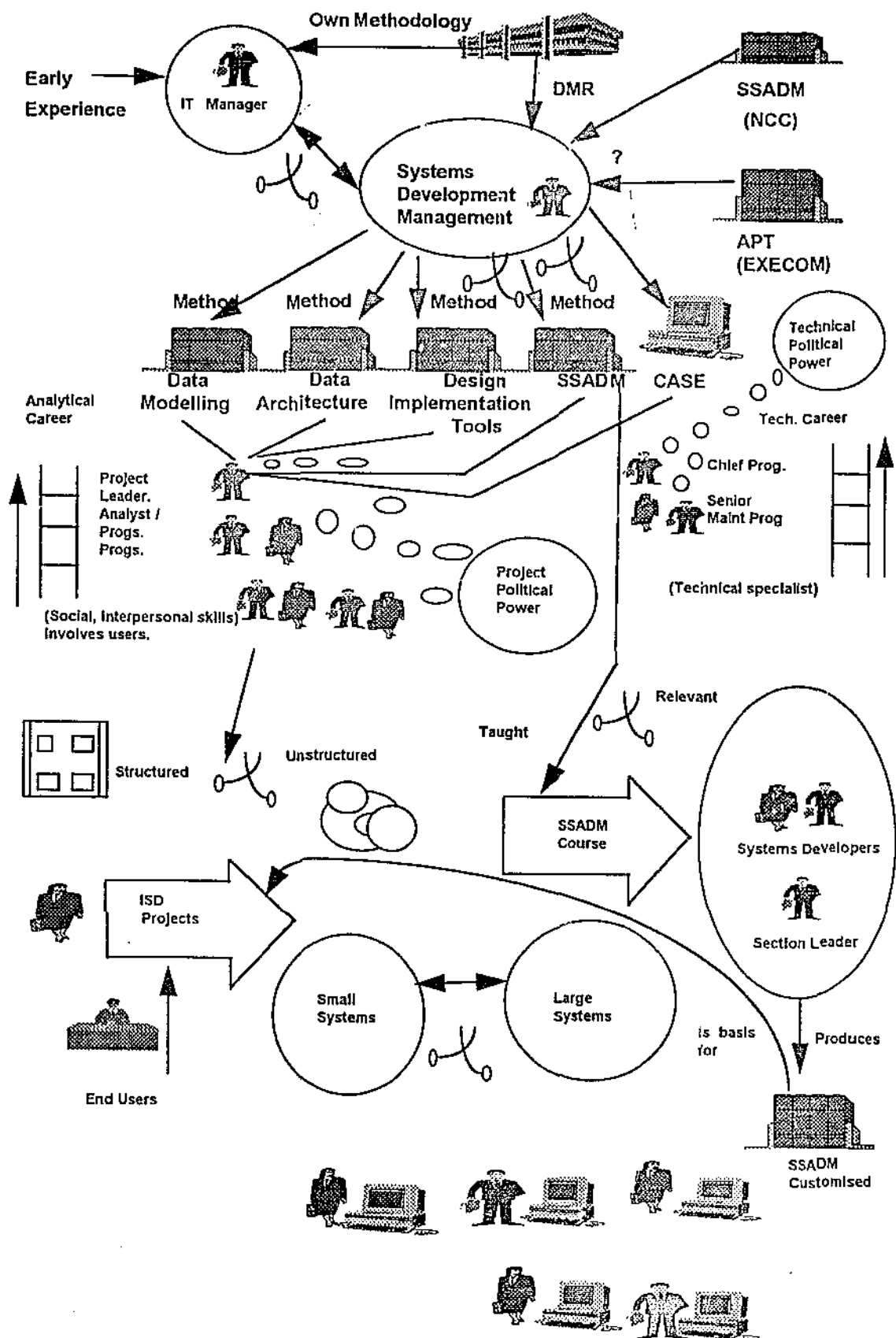


Figure 1. A rich picture of the selection of the SSADM methodology and its impact on careers, developers, end users, and organisation culture

CONCLUSION

The results reported here, while bounded by their scope, struck a balance between what SD personnel believe and what is generally perceived to happen. The following assertions are based on my own experience and perceptions as well as on an interpretive analysis of the qualitative research material. While controversial in nature, the key factors to progressing in one's career according to the findings have more to do with an employee's interpersonal skill than their technical ability in building information systems. While new, inexperienced graduates may believe that it will be their qualifications or their technical knowledge and ability that will earn them the promotions in a highly competitive IT environment, these findings show that this might not be the case. If we are to believe previous literature on factors affecting the successful implementations of information systems then IS professionals need to pay closer attention to building up skills in communicating, negotiating, presenting, handling culture clashes, and politics.

Most of the current discussions do not discuss the tremendous difficulties associated with successfully implementing information systems.

Hedberg, Nystrom and Starbuck (1977) are quoted by Orlikowski (1991, p. 39) as claiming that "Organisations typically display inertia through their established routines, institutionalised practices, and taken-for-granted assumptions that inform and reinforce the status quo."

There was no evidence in the findings to suggest that an employee's educational or work experience background or their knowledge of SSADM played a significant role in their career progression.

The SSADM methodology while considered as the SD standard is usually tailored to a point where the steps of the methodology are treated more as a check list. The products produced are those that form the essentials of most structured methodologies. SSADM has been in use for five years but still lacks the support of some senior staff who would prefer to see it replaced. The main reasons for its lack of support stems from the difficulty in end users confirming that the products produced meet their requirements. Personnel believe that the methodology does not produce business deliverables that any user can understand and confirm.

The lack of a CASE tool that supports SSADM counted as a further disadvantage. The CASE tool that was available did not fully support the methodology and was not as accessible as some staff would have liked. The fact that only one copy of the methodology exists adds further doubt about SSADM's long term future at COS. While it may remain as one of a family of methodologies, based on current trends and the move towards business process modelling and corporate data modelling, SSADM's long term future is doubtful.

The selection of the SSADM methodology took place during a controversial period in the history of the SD section. Since its selection most of the staff who participated in its selection have moved on to other organisations. The documentation produced, such as the evaluation criteria, weightings and final report with its recommendations have been lost in time, which prevented any in depth documentation reviews. The only information that could be gathered was received during in depth interviews with the three remaining members of the selection panel. Besides the selection team only the Project Leaders at the time had the opportunity to have any input into the selection of the methodology. Based on the results this has resulted in a lack of ownership or commitment to SSADM. Some staff cannot fully identify with the methodology and feel more like it has been thrust onto them, rather than being a tool to help them develop successful systems.

The selection and eventual use of a methodology usually affects both developers and users at various parts of the organisation hierarchy. With joint software development contracts involving external software developers who were using their own methodology, the selection of SSADM may have resulted in political involvements at a senior management level in the selection process. Unfortunately, since the senior management who were performing these roles have left the organisation this may never be known for sure.

This thesis has described a study, which examined the experiences that SD staff have had with the selection and implementation of a structured development methodology named SSADM. The results indicate that while methodologies and systems development are largely regarded as technical in nature, the major parts are actually non technical. We see the historical development of an information systems methodology and the levels of satisfaction that exist in using it. This study highlights the impact of other factors such as the behaviour, world views and values of individuals and the culture and politics of the organisation and its relationship with SSADM. Finally, it reaffirms the notion that for changes to take place associated with the introduction of a new methodology, it requires considerable negotiation among all players effected by the change.

APPENDICES

Interview Questions

1. Prior to starting work in systems development what kinds of work or study did you undertake? Eg. Computer Science, Information Systems, Engineering.
2. Did your past education or work require you to follow a set of procedure or steps? Eg. using structured methods?
3. Why do you think SSADM was selected over all the other methodologies available? Do you know how it was selected and for what reasons?
4. Do you prefer to develop systems interactively with the users using software? Eg. prototyping.
5. Do you use SSADM and if you do, do you find that by using the various tools and techniques they assist you in developing systems that are on time, on budget, which meet your users requirements, are easy to maintain and easy to use? If not, would you like to? Why?
6. Which tools and techniques do you use, why do you use them, when and for what types of systems?

7. Are the CASE tools available to you assisting you in developing systems in SSADM? Do you believe that a good CASE tools that supports SSADM will go a long way in convincing you to use SSADM?
8. Do you see any problems with SSADM as a method? Why?
9. Has the move to gain Australian quality standards accreditation convinced you to begin using SSADM?
10. Do your customers give you positive or negative feedback on the use of SSADM tools and techniques and the products you produce for them? Why?
11. Should the choice about what tools, techniques and products produced be fixed as standards for all systems or should the analyst have a choice? At what level should this customising take place?
12. What needs to happen in an organisation for SSADM to be successful? eg. training in the method, management commitment etc.
13. Do you see your career going down a technical path or down an analytical path. eg. Maintenance Programmer, Senior Maintenance Programmer, Chief Programmer or as a Programmer/Analyst, Project Leader etc. What made you select this path?

14. What do you consider being the key factors in career progression in IS in this organisation? Why?
15. Where do you see your future career in relation to SSADM? Why?
16. What role does SSADM, do you feel, have in career progression in this organisation? Why?
17. Do you feel that detailed knowledge of SSADM will give you the opportunities to work on projects you want to work on, increase your salary, improve your promotion prospects, increases your expertise, ie. value to the organisation? Why?
18. Would you prefer to work on specialist tasks, all round tasks (bit of everything), managing projects / people / budgets, small or large systems? Why?
19. Do you feel that you are confident in speaking in groups, making presentations, controlling or chairing a meeting, influencing the decision making process of others, both senior and subordinate to you? Why?

REFERENCES

- Ashworth, C. M. (1988). Structured systems analysis and design method (SSADM). Information and Software Technology, 30 (3), 153-163.
- Checkland, P. B. (1981). Systems thinking system practice. Chichester: Wiley.
- Couger, D. J. (1988, September). Motivating IS personnel. Datamation, pp. 59-64.
- Davis, D., & Cosenza, R. (1988). Business research for decision making. Boston: PWS-Kent.
- Doke, R. E. (1990). An industry survey of emerging prototyping methodologies. Information & Management, 18, 169-176.
- Dos Santos, B. L. (1988). MIS project management: a contingency approach. Management Decision, 26 (6), 22-28.
- Edwards, H. M., Thompson, J. B., & Smit, P. (1989a). Experiences in use of SSADM: series of case studies. Part 1: first time users. Information and Software Technology, 31 (8), 411-419.

Edwards, H. M., Thompson, J. B., & Smith, P. (1989b). Experiences in use of SSADM: series of case studies. Part 2: experienced users. Information and Software Technology, 31 (8), 420-428.

Edwards, H. M., Thompson, J. B., & Smith, P. (1989c). Results of survey of use of SSADM in commercial and government sectors in United Kingdom. Information and Software Technology, 31 (1), 21-28.

Emory, W. C. (1985). Business research methods. Illinois: Irwin.

Ferguson, G. T. (1983, December). Making productivity tools work. Infosystems, pp. 58-60.

Ginzberg, M. J. & Baroudi, J. J. (1988). Mis careers - a theoretical perspective. Communications of the ACM, 31 (5), 586-594.

Grover, V., Lederer, A. L. & Sabherwal, R. (1988). Recognising the politics of MIS. Information & Management, 14, 145-156.

Hawryszkiewicz, I.T. (1988). Introduction to systems analysis and design. Englewood Cliffs: Prentice Hall.

Hayley, K. J. & Lyman, H. T. (1990). The realities of CASE. Journal of Information Systems Management, 7 (3), 18-23.

Hirschheim, R. & Newman, M. (1991). Symbolism and information systems development: Myth, metaphor and magic. Information Systems Research, 2 (1), 29-62.

Hodge, B. & Clements, J. P. (1986). Business systems analysis. Englewood Cliffs: Prentice-Hall.

Igbaria, M. & Siegel, S. (1993). The career decision of information systems people. Information & Management, 24 (1), 23-32.

Keen, P. G. W. (1981). Information systems and organisational change. Communications of the ACM, 24 (1), 24-33.

Kendall, J. E. & Kendall K. E. (1993). Metaphors and methodologies: living beyond the systems machine. MIS Quarterly, 16 (2), 149-171.

Markus, L. M. (1983). Power, politics and MIS implementation. Communications of the ACM, 26 (6), 430-444.

McClure, C. (1988, August). The CASE for structured development. PC Tech Journal, pp. 51-67.

Modha, J., Gwinnett, A., & Bruce, M. (1990). A review of information systems development methodology (ISDM) selection techniques. Omega, 18 (5), 473-490.

Nachmias, D., Nachmias, C. (1987). Research methods in the social sciences. New York: St. Martin's.

Necco, C., Gordon, C., & Tsai, N. (1987). Systems analysis and design: current practices. MIS Quarterly, 10 (4), 461-475.

Nelson, R. R. (1991). Educational needs as perceived by IS and end-user personnel: a survey of knowledge and skills requirements. MIS Quarterly, 15 (4), 502-525.

Norman, R. J. & Nunamaker, J. F. (1989). CASE productivity perceptions of software engineering professionals. Communications of the ACM, 32 (9), 1102-1108.

O'Connor, A. D. (1992). Soft systems methodology - a case study of its use within an Australian organisation. The Australian computer journal, 24 (3), 130-138.

Orlikowski, W. J. (1991). Integrated information environment or matrix of control? The contradictory implications of information technology. Accounting, Management & Information Technology, 1 (1), 9-42.

Palvia, P. & Nosek, J.T. (1993). A field examination of system life cycle techniques and methodologies. Information & Management, 25, 73-84.

Patton, M. Q. (1990). Qualitative evaluation and research methods. Newbury Park: Sage.

Preston, A. M. (1991). The "problem" in and of management information systems. Accounting, Management & Information Technology, 1 (1), 43-69.

Richter, C. A. (1986). An assessment of structured analysis and structured design. ACM SIGSOFT Software Engineering Notes, 11 (4), 41-45.

Romm, T., Pliskin, N., Weber, Y., & Lee, A.S. (1991). Identifying organisational culture clash in MIS implementation. Information & Management, 21, 99-109.

Yin, R. K. (1989). Case study research design and methods. Newbury Park: Sage.