

Table of Contents

1	Dissertation Overview	7
1.1	Title	7
1.2	Aim	7
1.3	Objectives	7
2	Background	8
3	Methodology.....	9
3.1	Research Method.....	9
3.2	Mapping	11
3.3	Literature Review	11
3.4	Critical Analysis	11
3.5	Gap Analysis	12
3.6	Prototype.....	12
3.7	Review of Prototype	12
3.8	Interviews/Peer Review.....	13
3.9	Data Analysis	13
3.10	Field	14
4	Justification of Methodology	15
4.1	Methods Considered	15
4.2	Method Selected	17
4.3	Data Analysis	17
4.4	Anticipated Constraints and Outcomes	20
5	Literature Review	22
5.1	Search Approach.....	23
5.2	Design And Implementation Of Records Management Systems.....	25
5.3	Design And Implementation Of Information Systems.....	26
5.4	Data Analysis Methodology.....	26
5.5	Definitions	27
6	Failures In Information Systems.....	33
6.1	Rates Of Failures	33
6.2	Requirements.....	36
7	Information Systems Development	39
7.1	Information Systems Models	39

7.2	Information Systems Methodologies	44
8	Critical Analysis.....	49
8.1	Criteria.....	51
8.2	Analysis.....	53
8.3	Summary.....	61
9	Gap Analysis	62
9.1	Strengths.....	62
9.2	Weaknesses.....	62
10	Prototype Model	63
10.1	Description	63
10.2	Addressing The Gaps.....	65
10.3	Prototype Model Diagram.....	67
11	Data Analysis	68
11.1	Questionnaires	68
11.2	Analysis of Interviews.....	70
12	Conclusions.....	71
12.1	Recommendations	72
12.2	Reflections	74
12.3	Future Application of Approach and Methods	78
13	References.....	79
14	Bibliography	85
15	Appendix A. Pre-Interview Briefing Notes	89
16	Appendix B. Pre Focus Group Questionnaire	90
17	Appendix C. Reviewers.....	91
18	Appendix D. Schedule.....	92

Table of Figures

Figure 1. Systems Development Research Method	9
Figure 2. Project Research Method.....	10
Figure 3. Errors, Faults And Failures.....	30
Figure 4. Software Project Outcomes By Size Of Project.....	34
Figure 5. Probability Of On-Time Software Delivery In Six Sub Industries.....	35
Figure 6. Economics Of Faults	36
Figure 7. Requirements Errors	37
Figure 8. Progressive Distortion	38
Figure 9. The Waterfall Model	40
Figure 10. The Spiral Model	41
Figure 11. The V Model.....	42
Figure 12. The W Model.....	43
Figure 13. ISO 15489 Model	50
Figure 14. The DIRKS Model	50
Figure 15. The Control Cycle.....	57
Figure 16. Summary of Analysis.....	61
Figure 17. The W Model for Records Management Systems.....	67
Figure 18. Summary of Questionnaire Responses.....	68

Acknowledgements

Thanks to all who acted as reviewers of either the prototype model constructed as a result of this paper or of the paper itself.

David Bowen

David Cox

Anita Dray

Nathan Friend

Gabriel Hodges

Barbara Lammers

Jon Moss

David Ryan

Gary Payne

Liz Smith

Clive Whitton

Declaration and Plagiarism Disclaimer

The opinions expressed in this dissertation are solely those of the author and acceptance of the dissertation as a contribution to the award of a degree cannot be regarded as constituting approval of all its contents by the Division of Information and Communication Studies.

I certify that all material in this dissertation which is not my own work has been identified and properly attributed.

Permission to Copy

I grant permission for reproduction at the discretion of the Division, to allow the dissertation to be copied in whole, or in part, without further reference to myself [the author]. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Signed:

Gary Johnston

Date: 20 December 2004

Wordcount, excluding contents, declarations, abstract, references, bibliography and appendices: 15,998

Abstract

This paper is concerned with identifying and analysing existing models used in the design and implementation of records management systems. These models are then compared and contrasted with models that are commonly used in the design and implementation of information systems. The purpose of carrying out this exercise was to ascertain whether or not models used in the design and implementation of records management systems would be able to improve the quality and reduce the number of defects built into a records management system during its design and implementation.

The paper contains a description of the methodology used and a literature review. It contains sections that define the terms used throughout the paper and also a potted history of information systems development models, whose purpose was to provide guidance on how a records management systems development model should work.

Criteria have been identified against which the existing models are assessed. Following this exercise of comparing and contrasting, the paper identifies strengths and weaknesses in the existing models and then uses these weaknesses as the basis of a gap analysis. The gap analysis was then used to identify areas in the existing models that need to be strengthened and played a significant part in the development of a new model to replace the existing ones. The new model is referred to as the 'prototype model' throughout the paper.

The prototype model has been reviewed by a selected group of expert reviewers and their feedback analysed to produce a set of recommendations for the next version of the model.

1 Dissertation Overview

1.1 Title

The application of information systems development techniques to improve quality in the design and implementation of records management systems.

1.2 Aim

To develop a model to assist with the design and implementation of records management systems.

1.3 Objectives

The objectives of this project can be summarised as:

- To critically evaluate existing models and methodologies.
- To compare and contrast these models with accepted information systems development models and techniques.
- To identify any gaps in records management models.
- To identify information systems development techniques to address those gaps identified.
- To reduce the number of defects in the design and implementation of records management systems.

2 Background

The purpose of this section is to describe the background to the project and provide context.

Records management, and especially electronic records management, is a discipline that is becoming evermore prevalent in government, in industry and in society in general. As a result, records management and records management systems are being used in ever increasing numbers. They might be used to manage paper records, electronic records or both, often known as hybrid systems. Much of the literature surrounding the design and implementation of records management systems claims to have roots founded in the discipline of information systems development.

However, this dissertation will assess whether the design and implementation of records management systems is indeed in line with these models, by looking at various information systems development models and methodologies and critically comparing and contrasting these models with records management systems development methods. This analysis will highlight strengths and weaknesses in existing models. The analysis should also consider whether the existing models are in line with models that are still used in information systems development, as opposed to ones that were used in the past.

The over riding purpose of the project is to take the information garnered through the analysis of these models, specifically their strengths and solutions to their weaknesses, and to design and present a new model (initially a prototype) that will allow records management systems to be implemented in such a way as to reduce, as much as possible, the number of defects in the system.

The dissertation will not examine The National Archives Functional Requirements or MoReq or US DoD 5015.2 STD as these are concerned with what the electronic records management system should be able to do, but not specifically with how it should be designed, developed, tested, implemented and maintained.

3 Methodology

The purpose of this section is to describe the methodology to be undertaken during the project and to provide a justification for adopting that methodology.

3.1 Research Method

The nature of the work to be undertaken most closely matches the Systems Development research method.

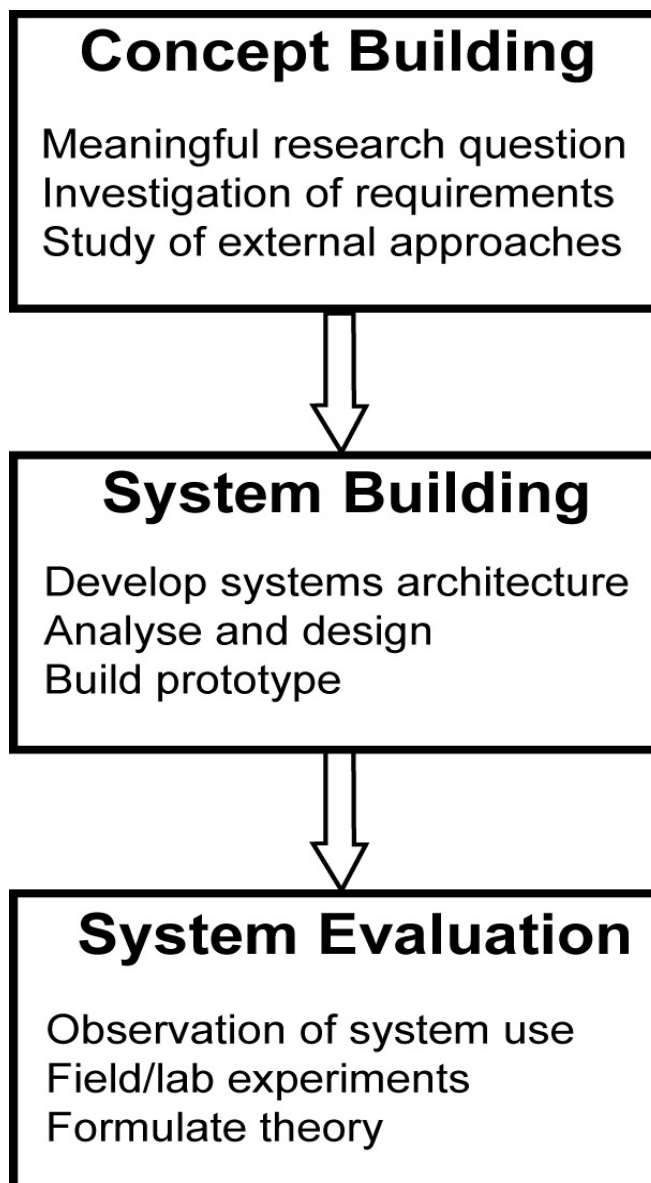


Figure 1. Systems Development Research Method

The steps shown above, those expected in a Systems Development research project, will be modified so that the project will follow the steps as shown below.

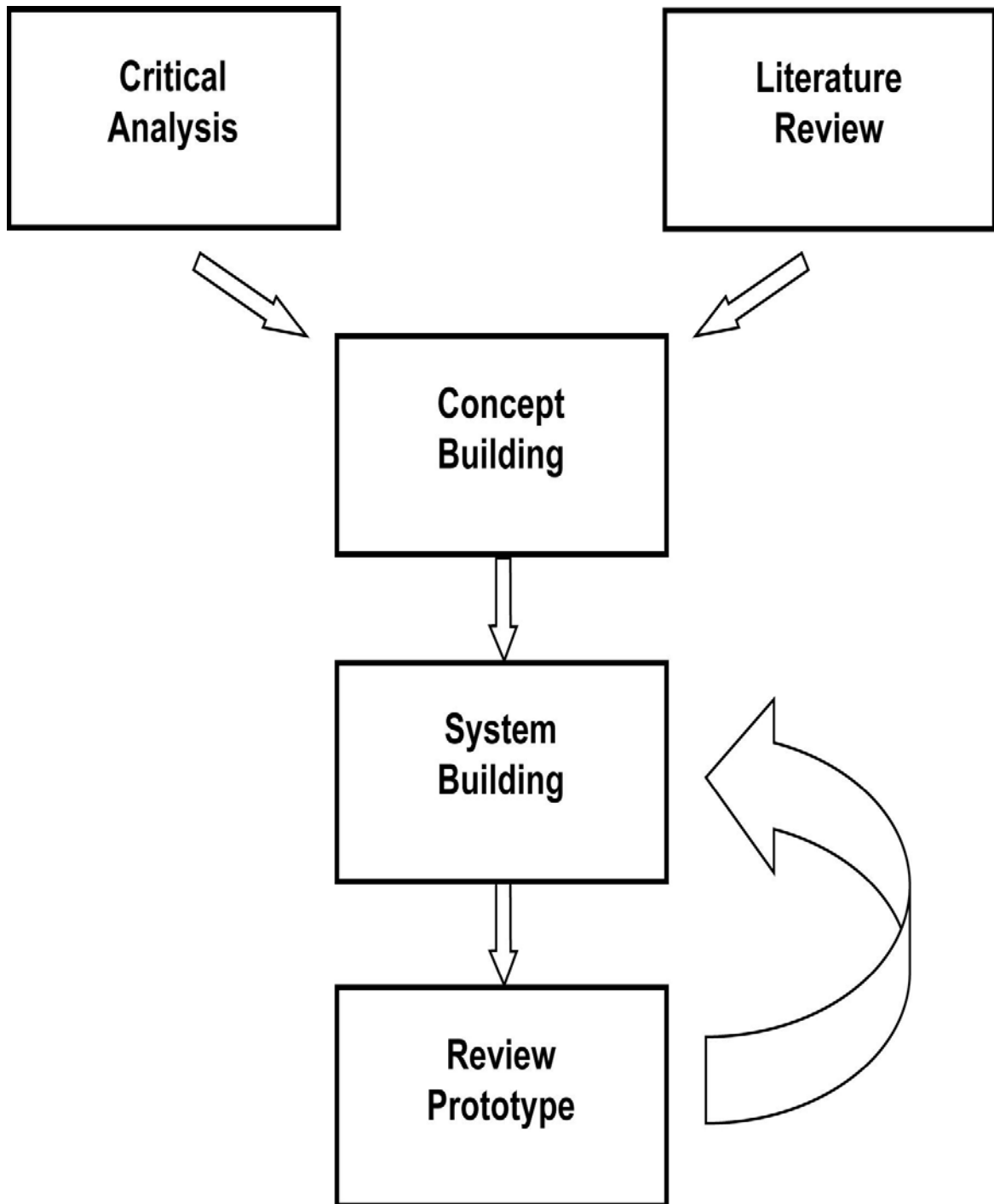


Figure 2. Project Research Method

3.2 Mapping

Below follows a description of how the steps shown in Figures 1 and 2 relate to each other.

3.2.1 Concept Building

The aims and objectives of the project in themselves provide a meaningful research question. The investigation of requirements will be achieved through a literature review and will result in a set of criteria against which to assess existing models. The study of external approaches will be achieved through a literature review and also as part of the critical review of existing methodologies.

3.2.2 System Building

This will be based upon the critical review and gap analysis. Any positive elements of existing methodologies identified will be retained and new elements introduced to address any gaps identified, as described further below in Section 3.4.

3.2.3 System Evaluation

It is unlikely that this final stage of the method will be conducted during the project. This is purely due to time constraints, making it unlikely that the model could be observed in use. However, the prototype will be reviewed as described in Section 3.7.

3.3 Literature Review

The literature review is described in Section 5 of this document.

3.4 Critical Analysis

The critical analysis will be concerned with evaluating records management systems development models against proven information systems development models. The criteria to carry out this analysis will have to be developed as part of the project.

3.5 Gap Analysis

A gap analysis will be carried out, based on the critical analysis, to identify those areas of existing methodologies that need to be addressed, with input from other information discovered during the literature search. Gaps will be identified and suitable steps, activities or techniques will be suggested to mitigate those gaps.

3.6 Prototype

The prototype model will be constructed based on the best elements of existing models and the results of the gap analysis. It will consist of a diagram, likely to be some kind of flow chart, and possibly a set of descriptive texts explaining what is happening at each step of the model.

3.7 Review of Prototype

The prototype will be sent for review to people who have a high degree of expertise in either the field of records management or information systems or both. Those identified and confirmed so far are:

- David Bowen, Audata Ltd, who has 30 years experience with computers and 15 with records management,
- David Ryan, formerly of The National Archives, who was Head of Digital Preservation at TNA and is now Director of Records for the Royal Household,
- Clive Whitton, Pfizer Ltd, responsible for electronic document management,
- David Cox, Audata Ltd, who has 35 years experience with computers and systems development,

Other potential reviewers have been contacted and confirmation of their participation is pending.

The reviewers will be sent a guide to explain how the model works and a questionnaire designed to seek feedback. This questionnaire will have to be developed as part of the project.

Data gathered as part of this review process will be analysed by using the constant comparative method.

3.8 Interviews/Peer Review

There are currently two options that seem appropriate for this stage of the project, these are:

- A focus group or
- One to one interviews.

It is not clear at this stage which method will be the most appropriate. Deciding factors will include:

- Availability of the reviewers,
- Geographical location of the reviewers,
- Suitability of the option in the overall scheme of the research method.

One to one interviews or focus groups will both involve the face-to-face questioning of people selected for their particular knowledge and subject expertise.

Whichever method is used, there will be a requirement to send out a directional questionnaire prior to the meetings taking place, along with a brief description of the prototype model. These can be found in Appendices A and B of this document.

3.9 Data Analysis

The data analysis will be based upon developing a grounded theory by using inductive analysis, specifically the constant comparative method.

The feedback elucidated from the reviewers will be analysed according to the constant comparative method as defined by Strauss (1990).

For this project the process to be followed is as described below:

- Read the transcripts of the interviews.
- Identify indicators of categories, name them and code them on the interviews.
- Compare the codes to find consistencies and also any differences.
- Note the comparisons and any emerging categories.
- Eventually, when no new codes are formed, the category is probably finished with.
- Some of the categories may become more of a central focus.

The steps outlined above do not have an exact match with the five steps defined by Strauss, but conform to the concept, as further described in section 5.3.

This feedback will form the basis of the recommendations derived from the project. It is at this point the project will finish. In practise, the prototype model will require another iteration of reviewing, but this is not going to be possible to complete within the timescales available.

3.10 Field

There is also another potential option for assessing the model. This involves using it on a real project, that is using it for the design and implementation of a records management system. This type of assessment will only be possible once the model has been through a few iterations of review to ensure that it satisfies the needs of a records management system implementation project. It should be possible to measure the number of defects present in an ERMS Requirements Specification that are subsequently carried into the design, build and configuration of the system and to compare this against the number of defects identified and carried through each stage of the design and implementation of a records management system using the new model.

This is only a potential evaluation method as it is unlikely that a records management system could be designed and implemented within the timescales of this project. If this is feasible at a later date, the data analysis methods will be much different and are likely to be quantitative in nature.

4 Justification of Methodology

This section is concerned with justifying the research methodology to be used. It first briefly examines some of the options available but not pursued before discussing the reasoning behind the use of the systems development methodology.

The research area clearly does not lend itself to survey research. There is no actual case to study, thus ruling out case study research. There is a case for using the action research method, however, due to the iterative nature of this approach it will not be used. This is because time constraints will not allow for any iterations to take place.

4.1 Methods Considered

4.1.1 Surveys

According to Kerlinger (1986) there are two major types of survey research, exploratory and explanatory. Exploratory research is concerned with the researcher making themselves familiar with a given subject or topic. This type of survey might be appropriate for a research project designed to understand the benefits of implementing a records management system. Descriptive surveys are exploratory.

Dubin (1978) believes that descriptive surveys are vital during the initial stages of a research study to aid in theory development.

Explanatory surveys seek causal relationships. Using the example above, that of researching the benefits derived from the implementation of records management systems, explanatory research would be seeking to relate a feature of a records management system with a particular benefit or set of benefits.

It is not envisaged that this project will use surveys as a data collection tool. It could be argued that surveys are too linear and that they are used to discover what people do. Furthermore, the types of questions that would have to be asked in a survey for this research topic would be complex and potentially not easily understandable to the respondent. However, the project will use questionnaires as a primer to set the scene prior to the focus group or interviews taking place.

4.1.2 Case Study

A case study can be described as an in-depth analysis of one particular organisation or situation, such as a business or company, a community or society or part of a community or society, or a specific situation, occurrence or event. Although a case study will potentially allow the thorough examination of a particular situation, the results of such a study cannot be generalised beyond that single case with any degree of certainty.

There are many advantages to using the case study approach, including its ability to provide a detailed description of a specific example and, importantly, that it might provide the starting point for the development of an hypotheses for later testing by other methods.

However, not unexpectedly, the case study approach also has some disadvantages. These include not being able to draw general conclusions, only those of the case in question as previously mentioned, that the hypotheses cannot be tested (because testing must be repeatable) and that cause-effect relationships cannot be established.

This project will not use the case study approach, primarily because there is no actual case to study as opposed to the disadvantages mentioned above. Further work, beyond the scope of this project, will almost certainly include using the model to design and implement a records management system in a real life situation. It is outside the scope of this project because the design and implementation of a records management system is a lengthy process and would not be possible to complete within the timescales of the project.

4.1.3 Action Research

Action research can be described as a combination of a case study and a field experiment (Galliers 1992). Action research shares some characteristics with the case study approach in that it is concerned with groups or situations. The major difference between the two is that in action research the researcher takes an active role in the study. This means the researcher must undertake an evaluation of their participation at

the same time as evaluating the research. This project will not follow the action research method.

4.2 Method Selected

As described in Section 3, the systems development research method has been selected as it has the closest fit to the purpose of the project. Whilst the project is not actually concerned with developing a system per se, it is concerned with developing a model to be used in developing systems.

4.3 Data Analysis

The project will use grounded theory as its data analysis method. The data will be gathered through either a focus group or series of one to one interviews.

Kerlinger (1986) defines theory as

"a set of interrelated constructs (concepts), definitions and propositions that present a systematic view of phenomena by specifying relationships among variables, with the purpose of explaining and predicting the phenomena".

Haig (1995) states

“that grounded theory is best regarded as a general theory of scientific method concerned with the detection and explanation of social phenomena. To this end, grounded theory is reconstructed as a problem oriented endeavor in which theories are abductively generated from robust data patterns...”

The project is not concerned with social phenomena, but it is most certainly concerned with a problem oriented endeavour. Whilst the above quotation provides a relatively illuminating description, it leaves the concept of abduction open for explanation. Josephson (1994) states that

“[Abduction] is a form of inference that follows a pattern like this:

- D is a collection of data (facts, observations, givens),
- H explains D (would, if true, explain D),
- No other hypothesis explains D as well as H does.

Therefore, H is probably correct.”

This could be viewed as a variation of Occam’s Razor. Josephson also raises several key considerations about ‘H’. These include:

- How good H is by itself, independently of considering the alternatives,
- How decisively H surpasses the alternatives,
- How thorough the search was for alternative explanations, and
- Pragmatic considerations, including:
 - The costs of being wrong and the benefits of being right,
 - How strong the need is to come to a conclusion at all, especially considering the possibility of seeking further evidence before deciding.

Grounded theory can be further justified as an appropriate research approach. At least two studies into information systems development have used the grounded theory approach. The first study (Calloway, 1988) assesses the use of a software engineering tools and dialog charts by systems designers. The second study (Knapp, 1995) investigates organisations that use Integrated Computer-Aided Software Engineering (ICASE) tools.

The data was generated in different ways in the two studies. In the first study an experimental approach was taken, producing a controlled environment, whereas in the second study a case study approach was taken, producing an uncontrolled environment.

In a paper analysing these two studies (Calloway and Knapp, 1995), the authors state that

"Grounded theory can be used regardless of the way the data is generated."

They also state that

"The emergence of theory appears to transcend the specifics of methodological difference, since theory can be detected regardless of methodological differences. The methodology therefore appears to be transparent."

The constant comparative method provides a means to develop a grounded theory.

"The constant comparative method of data analysis was developed by Glaser and Strauss (1967) as the means of developing grounded theory. A grounded theory consists of categories, properties, and hypotheses that are the conceptual links between and among the categories and properties. Because the basic strategy of the constant comparative method is compatible with the inductive, concept-building orientation of all qualitative research, the constant comparative method of data analysis has been adopted by many researchers who are not seeking to build substantive theory" (Merriam, 1998, p. 159).

Glaser and Strauss (cited in Lincoln & Guba, 1985, p. 339) described the constant comparison method as following four distinct stages:

1. Comparing incidents applicable to each category,
2. Integrating categories and their properties,
3. Delimiting the theory, and
4. Writing the theory.

For this project, the constant comparative method will be used to identify recommendations for the improvement of the prototype model.

According to the Dictionary.com online dictionary, a model is

“a schematic description of a system, theory, or phenomenon that accounts for its known or inferred properties and may be used for further study of its characteristics”.

Furthermore,

"a model helps us visualize how something might work and what variables should be taken into account' (Slife & Williams, 1995, p. 220).

It has been said that the development and refinement of models is in fact a type of 'theory building' (Reynolds, 1971).

4.4 Anticipated Constraints and Outcomes

4.4.1 Constraints

The following constraints are anticipated:

- The development of a new model for the design and implementation of records management systems has the potential to go through several iterations to refine it. There is a self-imposed constraint on the project in that the model will only be reviewed once. Further iterations are likely to take place but these will be a matter of personal interest only.
- It is anticipated that searching for critical reviews of ISO 15489 and DIRKS is unlikely to return large quantities of information. This will affect the analysis phase of the project. In contrast, searching for literature around information systems development is likely to return an abundance of relevant findings. It might well be problematic sorting the wheat from the chaff.

- Much of the work to be completed during the project will be carried out by the researcher. Therefore constraints are based around time. This will be mitigated through robust planning and time management.
- There will be an external dependency around the identification of suitable reviewers who are both willing and able to participate. To mitigate this risk, potential reviewers will be identified and approached at a very early stage of the project. Their availability will be re-affirmed on a regular basis; this should identify the need to approach replacement reviewers at the earliest possible opportunity. It is anticipated that regular updates will be sent to reviewers so as to keep them informed as to the status of the project.

4.4.2 Outcomes

The overriding purpose of the study is to develop a model to ensure that records management systems are implemented so that they contain as few defects as is possible. Therefore the desired outcome of the project is a model to guide records managers in the design and implementation of records management systems. The model should contain a diagram showing the steps to take, complimented by a textual description of those steps, although at this stage it is unclear as to whether the descriptive elements will be completed.

5 Literature Review

The purpose of this stage of the project is to conduct a literature review and to record the findings of that review. Literature reviews are undertaken to identify existing informational resources, by accredited scholars and researchers, that are relevant to the research being carried out. The literature review will act as a survey of research done in the project area, but also in related areas. It will attempt to also evaluate the methods and results of previous research and to describe the "big picture". It will provide definitions of terms to be used in the paper.

It is not the purpose of a literature review to identify every existing resource related to the topic of the research article, but to identify and evaluate the most relevant resources.

A literature review should be a piece of discursive prose, that is, it is organised by themes, not by author.

According to Cooper (1988)

“... a literature review uses as its database reports of primary or original scholarship, and does not report new primary scholarship itself. The primary reports used in the literature may be verbal, but in the vast majority of cases reports are written documents. The types of scholarship may be empirical, theoretical, critical/analytic, or methodological in nature. Second a literature review seeks to describe, summarise, evaluate, clarify and/or integrate the content of primary reports.”

Conducting a literature review is an essential first step in any research project. According to Degelman (2003), conducting a literature review will identify:

- What research has already been conducted that is relevant to the research question,
- What theories have guided the research that has already been conducted,

- What hypotheses have been tested,
- How other researchers have defined their variables,
- What kinds of research designs other researchers have used,
- How participants were tested in these previous research studies,
- What participant populations have been studied,
- What mistakes other researchers have made, and
- What suggestions have been made for future research.

The literature review to be undertaken for the entire project can be broken down into separate pieces of work. These are described below:

- A literature review to support the methodology.
- A literature review to identify information available on the design and implementation of records management systems.
- A literature review to identify information available on the design and implementation of information systems.
- A literature review to:
 - Define quality,
 - Define defects, when they occur and the associated costs,
 - Define software engineering,
 - Define information systems,
 - Define records management systems.

This literature review will not specifically examine the content of ISO 15489, DIRKS or any IT/IS development models as these models will be critically evaluated in the critical analysis stage of the project, but will instead concentrate on literature concerning them.

5.1 Search Approach

5.1.1 Identify And Confirm Timescales

Identify the target finish date for the search, any days that are already taken up on other work or projects, holidays etc to understand the available time and allow suitable planning to take place.

5.1.2 Identify Desired Output

For the purposes of this project, the desired output is a list of 50 or so references.

5.1.3 Identify Informational Resources Available

The search will use the following informational resources:

- The World Wide Web using Google and Yahoo. Although subject directories, such as Yahoo, cover only a small proportion of the pages available on the World Wide Web, they can often be effective for finding general information on popular or academic subjects. This will be followed by using a search engine, Google, to broaden the search out, followed by the use of title searching using the same subject directory and search engine to gain a more manageable number of relevant hits.
- Conference proceedings,
- Searching for journal articles and other full text materials available online,
- Locating a particular book through the University of Northumbria library using Opac, Emerald etc,
- Audata's technical library.

5.1.4 Other Factors

Below are detailed other factors to be considered.

- Identify the main concepts in the topic and examine any synonyms, alternate spellings, or other word forms for the topic, for example records management and records keeping.
- Search for only those informational resources that are in English.
- The search will focus on information that is under twenty years old, but will not automatically exclude information that exceeds that limit. Over twenty years old in relation to information systems probably means the information is out of date or superseded, whilst information over twenty years old about records management might still be relevant.

5.1.5 Search Terms

Terms to be searched are listed below:

- Qualitative Research Methods,
- Qualitative Research & Information Systems,
- IT Projects & Success & Failure,
- Information Technology Projects & Success & Failure,
- ISO 15489,
- ISO 15489 & Review,
- DIRKS,
- DIRKS & Review,
- Records & Keeping & Systems,
- Records & Management & Systems,
- IS & Development & Methods,
- Information Systems & Development & Methods.

5.1.6 Rationale For Search Strategy

The topic being researched has to an extent dictated the strategy. It is both a commercial and academic subject. This means that to find the maximum relevant information on the search subject, it will be necessary to use a variety of sources where both commercial and academic information is likely to be found.

5.2 Design And Implementation Of Records Management Systems

There is very little literature available on the design and implementation of records management systems. The two major pieces of literature are ISO 15489 and DIRKS. It could be argued that the ISO is too generic, possibly due to the very fact it is an international standard and therefore has to at least partially satisfy many stakeholders, whilst DIRKS is possibly too prescriptive, not allowing for localised differences.

The report 'Records Management: A Literature Review' from the ICA (1996) promises much but is perhaps hindered by the lack of practical experience in the design and

implementation of records keeping systems available when it was written. Consequently, the paper does not offer much.

Fanning (2002) does provide an overview of ISO 15 489, but the paper focuses on MoReq and DOD 5015.2.

5.3 Design And Implementation Of Information Systems

Not only is there a wealth of information available on this subject (Beynon-Davies, 1998; Weaver, 1993; Yeates, 1991 & 1994; Yourdon, 1968), but a significant amount of research has been conducted in this field (Cash, 1989; Fidel, 1993; Galliers, 1992; Gorman, 1997; Kaplan, 1998; Mumford, 1985; Myers, 2002; Orlikowski, 1991).

Much of this information has a basis in time. That is to say, the development models and methodologies described in the work identified reflects when that work was written. This can largely be attributed to the fact that the evolution of development methods for information systems has an almost symbiotic link to the evolution of technology. As the technology has evolved, developers have been presented with new challenges. Often a new model or methodology has been developed to address these new opportunities and challenges.

Information systems models and methodologies have also evolved organically. Weaknesses in particular models and methodologies have often led to the development of a new variation designed to address those weaknesses.

5.4 Data Analysis Methodology

Mason (2002) states that

“Qualitative research is capable of producing very well founded cross-contextual generalities” (p1).

This would seem to suit the project, which is concerned with applying techniques from one discipline to another.

The project will follow grounded theory as proposed by Glaser and Strauss (1967). Martin and Turner (1986, p.141) describe grounded theory as

“an inductive, theory discovery methodology that allows the researcher to develop a theoretical account of the general features of a topic while simultaneously grounding the account in empirical observations or data”.

This approach seems particularly appropriate, as no research (at least none has been found so far) concerning the aims and objectives of this project has been undertaken to date.

Grounded theory is concerned with producing useful results. It also aids

“the generation of theories of process” (Glaser and Strauss, 1967, p.114),

both of which are prime objectives of this project.

5.5 Definitions

There are a number of terms that must be defined to ensure that a common, understood language is used throughout the paper.

5.5.1 System

Before defining a records keeping system or an information system, it is worth considering what a system is. According to dictionary.com, a system is

“A group of interacting, interrelated, or interdependent elements forming a complex whole.” and “A network of related computer software, hardware, and data transmission devices.”

According to the Cambridge Dictionary of Philosophy, systems theory is

“the transdisciplinary study of the abstract organization of phenomena, independent of their substance, type, or spatial or temporal scale of existence. It investigates both the principles common to all complex entities, and the (usually mathematical) models which can be used to describe them.

5.5.2 Information System

Webopedia defines an information system as

“a system that collects and stores data”.

However, this definition does not explicitly describe the need to retrieve data from the system in such a way as to make that data useful. By combining the definition of a system with that of an information system and recognising the need to be able to retrieve data it is possible to say that an information system is

“a network of related computer software, hardware and data transmission devices that collects and stores data, and is further more able to retrieve and present that data in a human readable format.”

5.5.3 Records Management System

According to Horsman (1999) there is a distinction between a records management system and a records keeping system. He argues that records management is what records managers do, whilst records management is what organisations and indeed society in general does. Horsman defines a record keeping system as

“the whole of records, methods, procedures, tools, [meta]data, knowledge, means and persons with which an organisation

fulfils its requirements to preserve evidence of its activities, maintain its memory, and preserve its knowledge.”

This is a useful definition in that it emphasises the fact that the design and implementation of records management systems is about more than just the software; it highlights that it is about a complete system that involves people carrying out their jobs, but otherwise any differentiation serves little practical purpose and could cause unnecessary confusion.

DIRKS states that

“all business systems used by an organisation must be capable of capturing, maintaining and providing evidence of its business activities over time to satisfy the organisation's record keeping needs.”

It also states that

“many business systems in your organisation may currently keep information as records. These systems include your organisation's registry or main correspondence system, personnel system and financial management system.”

DIRKS goes on to discuss the fact that

“in order to be full and accurate, records must be authentic, reliable, complete, unaltered and useable and the systems that support them must be able to protect their integrity over time.”

DIRKS describes the characteristics that an information system should possess to allow it to carry out record keeping functions. Many of these characteristics, if not all, would be expected to be found in a records management system, thus negating the need to differentiate between the two.

By taking the definition for an information management system and extending that definition, it is possible to conclude, for the purposes of this project, that a records management system is

“a network of related computer software, hardware and data transmission devices that collects and stores data, and is further more able to retrieve and present that data in a human readable format ensuring that the data is authentic, reliable, complete, unaltered and useable, and that the integrity of the system is protected over time.”

Where ‘readable’ also includes the interpretation of audio, video and other digital formats.

5.5.4 Defects

In software testing theory, bugs do not exist, only errors, faults or failures, which combined are often referred to as defects. The differences can be seen in the diagram below.

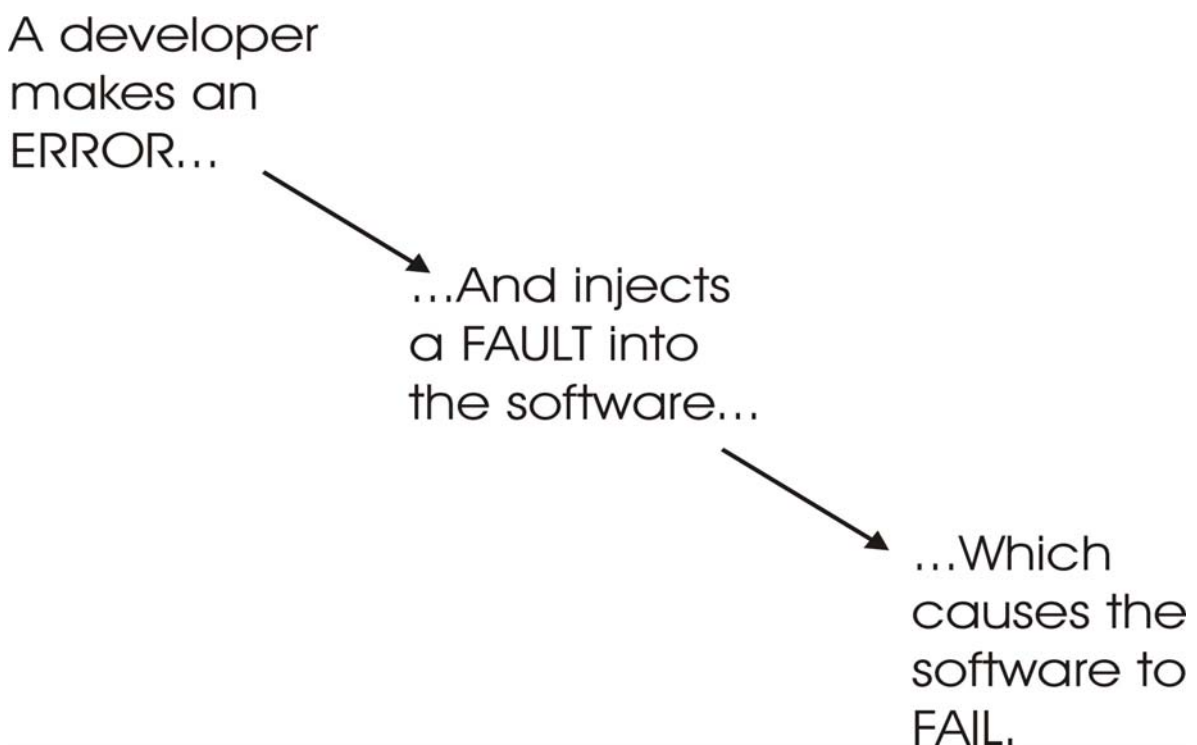


Figure 3. Errors, Faults And Failures

Examples of errors, faults and failures are:

- An analyst who does not adequately document all the requirements in a specification causes an error to occur.
- A programmer who forgets to write the validation code for a numeric value creates a fault.
- A screen that rejects “12/01/99” as an “invalid date” is a failure.

5.5.5 Quality

Quality is often difficult to define. The Webster's dictionary definition is

“degree of excellence”

which, it could be argued, is not a particularly useful definition. Indeed this dictionary definition closely relates to what Crosby (2001) criticises as “measures of goodness”.

Pirsig (1984) has observed that

“Quality creates the subjects and objects of the world. The facts do not exist until quality has created them.”

Which, whilst being undeniably philosophical in nature, closely matches the thoughts of Shewart (1931), who believes that quality has two aspects, one objective and the other subjective. Shewart maintains that an item possesses objective quality independently of any observer, whilst at the same time possessing subjective quality which results from what that observer sees or feels.

Pirsig (Op cit) further argues that quality itself is not a method. An example to illustrate this thinking might be to envisage the design and implementation of a records management system according to ISO 15 489. Following the method prescribed in this standard would not assure quality, but it would ensure that the system complies with the standard.

To be able to utilise quality, a more pragmatic approach and definition must be used. Juran states that:

“Quality is fitness for use.”

Further guidance can be gained from ISO 9000

“Quality is the totality of features and characteristics of a product or services that bear on its ability to satisfy stated or implied needs.”

The implementation of a records management system must be undertaken to meet a specific business need. That need should be presented as a Requirements Specification document.

By combining Juran’s definition and that presented by ISO 9000, and for the purposes of this project, a quality system is therefore

“a system that is fit for use and is able to satisfy an organisations stated or implied needs”.

6 Failures In Information Systems

This section is intended to investigate why information systems projects go wrong. This information might prove useful later in the project when constructing the prototype model.

Whilst the IT industry is littered with many high profile failed projects (with government seeming to excel in this respect at least), there are also many more that succeed. Statistics outlining the rates of successes and failures are detailed later in this section.

However, even the failures can prove useful in that they can sometimes highlight where mistakes were made. The information systems industry is one that has been subjected to much study, both academically and practically, and lessons have often been learnt. These lessons have been implemented and this has, on occasion, led to the development of new methodologies.

6.1 Rates Of Failures

According to the Standish Group, in the United States in 1995 government and businesses spent approximately \$81 billion on cancelled software projects, and another \$59 billion were spent on budget overruns. In this survey the claim was made that in the U.S. only about a sixth (16%) of all projects were completed on time and within budget; that nearly a third of all projects were cancelled and that well over half were considered 'challenged'. Of the challenged or cancelled projects, the average project came in at 189% over budget, 222% over schedule, and contained only 61% of the originally specified features.

In 1998, in another survey undertaken by the Standish Group, the number of successful projects had risen to 26%. However, neither of the reports contain a clear definition of what constitutes a successful project or a failed project. Whilst the differences between success and failure might seem self evident, that is not always the case. For instance, is a 5% over run on the budget considered a failure or a success? It is likely that different organisations would answer that question in different ways, according to their

own pre-defined tolerances and expectations, or whether they are in the public or private sector.

It is an interesting point of note that the report gives projects with a budget exceeding \$10 million a probability of successful completion of 0%. This is attributed to the project, at this size, becoming too complex to be fully understood. This is supported by Brooks (1997), who said that the conceptual integrity of the [system] architecture is the most important factor in obtaining a robust software system. Brooks further observed that this could only be achieved by one mind, or a very small number of resonant minds.

The tables shown below are based on research also carried out in 1998 in the United States. A typical system sized software development project would be expected to contain approximately 10,000 function points (FP). A function point is generally defined as one end-user business function, such as a query for an input.

The results shown in Figure 4 seem to support the research conducted by the Standish Group, with only 28% of projects of 10,000 FP being delivered on time.

	Early	On Time	Delayed	Cancelled	Sum
1FP	14.68%	83.16%	1.92%	0.25%	100.00%
10FP	11.08%	81.25%	5.67%	2.00%	100.00%
100FP	6.06%	74.77%	11.83%	7.33%	100.00%
1,000FP	1.24%	60.76%	17.67%	20.33%	100.00%
10,000FP	0.14%	28.03%	23.83%	48.00%	100.00%
100,000FP	0.00%	13.67%	21.33%	65.00%	100.00%
Average	5.53%	56.94%	13.71%	23.82%	100.00%

Figure 4. Software Project Outcomes By Size Of Project.

However, it is worth noting in the table below, which breaks down projects into industry sectors or types, that a project of 10,000 FP for a systems software project is likely to be delivered on time in 54% of cases, which suggests that systems software developers have refined their development techniques to lower the risks of failure.

	Systems Software	Military Software	MIS Software	Outsource Software	Comms Software	End User Software	Average
1FP	99.00%	98.00%	98.00%	98.00%	99.00%	95.00%	97.83%
10FP	96.00%	93.00%	95.00%	97.00%	98.00%	75.00%	92.33%
100FP	88.00%	84.00%	86.00%	88.00%	89.00%	50.00%	80.83%
1,000FP	75.00%	65.00%	68.00%	74.00%	75.00%	5.00%	60.33%
10,000FP	54.00%	38.00%	30.00%	47.00%	35.00%	0.00%	40.80%
100,000FP	28.00%	15.00%	5.00%	24.00%	10.00%	0.00%	18.40%
Average	73.33%	65.50%	63.67%	71.33%	67.67%	37.50%	65.09%

Figure 5. Probability Of On-Time Software Delivery In Six Sub Industries.

Source: Jones (1998)

The tables above should be treated with trepidation, as there are inconsistencies in the data presented. As previously mentioned, Figure 4 suggests that 28% of projects of 10,000 FP are delivered on time, yet Figure 5 suggests a 40% average probability of a 10,000 FP project being delivered on time. The article does not address these inconsistencies suitably.

These figures are obviously dated, six years in information technology is a long time, but no more recent figures have been identified. The figures shown above seem to indicate that in a three year period the successful conclusion of projects increased by 100%. If this is extrapolated to today, about 46% might conclude successfully, but this is a purely speculative figure.

The lesson for the design and implementation of records management systems seems clear, keep it simple and increase the probability of delivering a system that meets the requirements and expectations of the organisation. It could also be argued that records management systems should be implemented in a staged approach if they are going to be complex, increasing the likelihood of successful delivery of each stage.

6.2 Requirements

Within the information systems development and software engineering communities, it is generally accepted that faults in a systems development project become more complex, and therefore more time consuming and expensive to rectify, the later in the project it is that they are discovered. This is shown anecdotally in the diagram below.

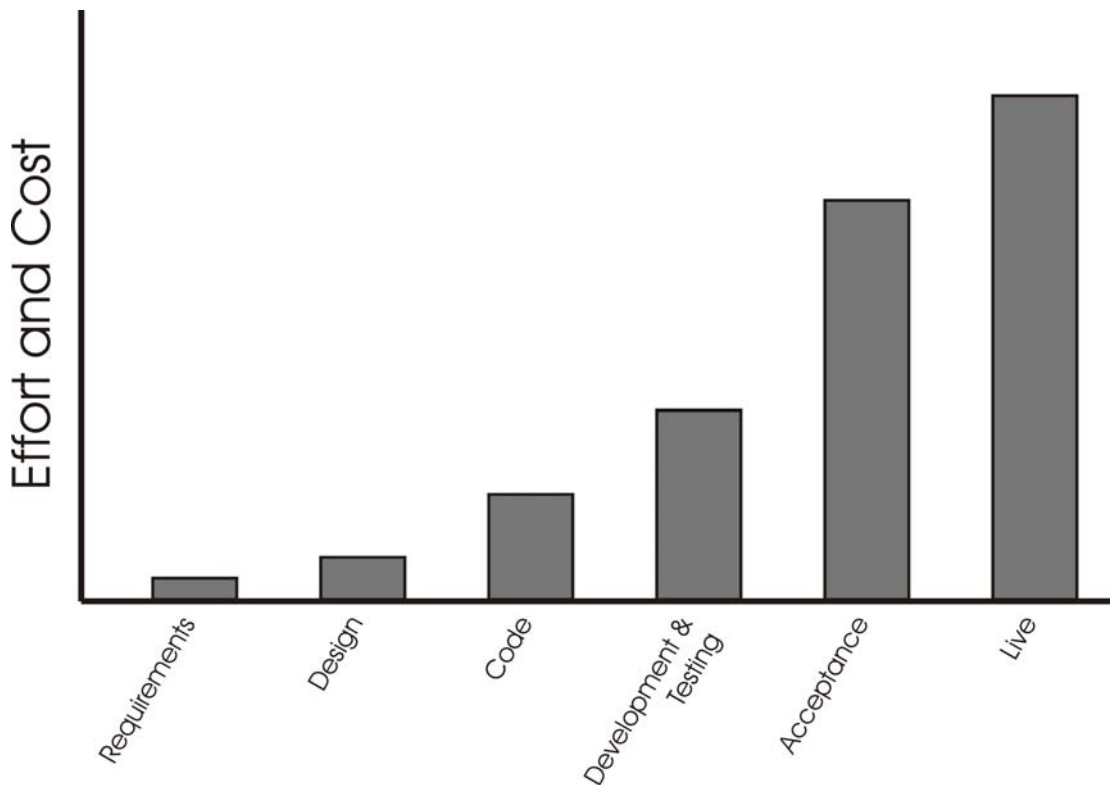


Figure 6. Economics Of Faults

Young (2001) estimates that 85% of defects in software can be traced back to the requirements. Hook et al (2001) list the most common types of requirement errors as:

- Incorrect assumptions,
- Omitted requirements,
- Inconsistent requirement,
- Ambiguities.

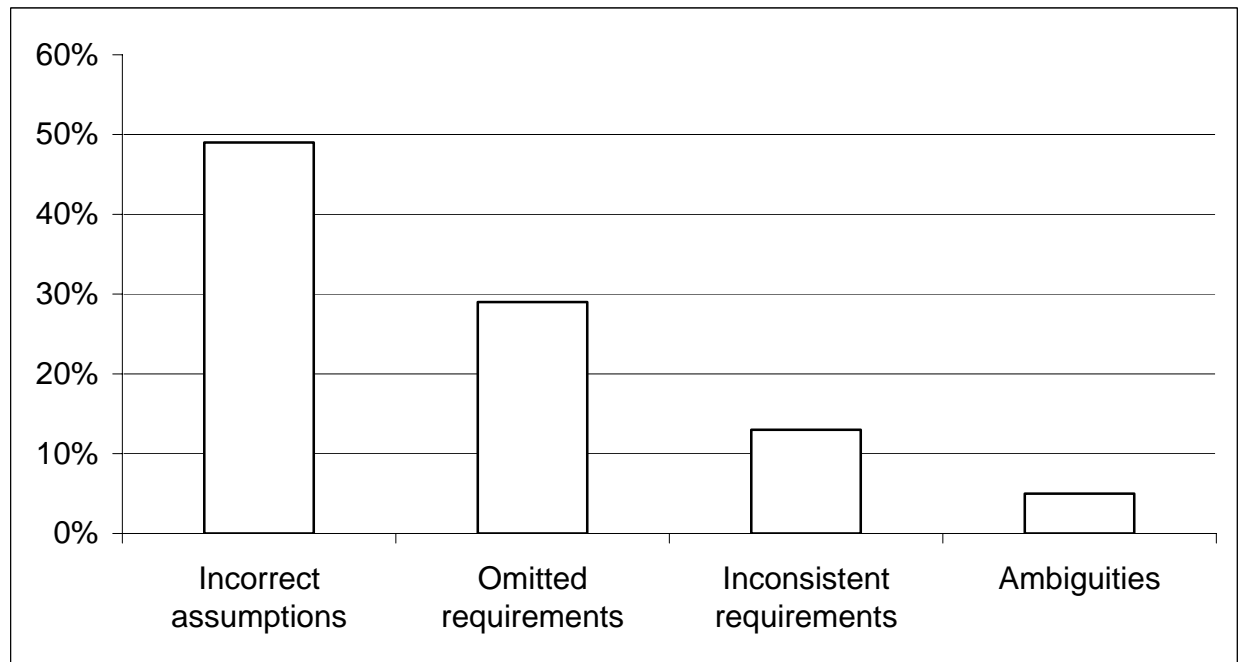


Figure 7. Requirements Errors

Accepting the figures presented by Brooks highlights the need to produce clear, concise and robust requirements when developing or implementing a records management system. This can be difficult, requirements gathering is not an easy task; combined with management often wanting to ‘see’ results early in a project which can often lead to pressure being placed to start the development of the system without fully understanding and documenting what it is that the system is going to do.

Robust requirements also go some way towards addressing and preventing scope creep, that is where the project grows due to a lack of understanding of what is to be achieved at the outset of the project and metamorphoses into something completely different.

The picture shown below, a well known and often used analogy within software engineering, takes a light-hearted look at what is known as ‘progressive distortion’, but does show the importance of gathering well defined and robust requirements.



As management requested it



As the project leader defined it



As systems designed it



As programming developed it



As operations installed it



What the user wanted

Figure 8. Progressive Distortion

Kaner et al (2001) list 6 test points to be covered when reviewing requirements and specifications, described here briefly:

- Are these the "right" requirements?
- Are they complete?
- Are they compatible?
- Are they achievable?
- Are they reasonable?
- Are they testable?

7 Information Systems Development

The following section will describe some of the major development models and methodologies that are currently in use or have been in use in the past.

During the early days of information systems development, usually using Cobol on mainframe computers, there was frequently an ad hoc approach to systems design and implementation, often known as 'code and fix', with very little documentation. This often led to systems that did not meet either user requirements or expectations. Consequently, the industry identified the need for a more structured approach. There seems to have been two main areas of study, gathering and defining user requirements in a more logical and precise way, and defining robust, consistent techniques for the analysis and design of information systems.

7.1 Information Systems Models

7.1.1 The Waterfall Model

The Waterfall model was originally developed by Royce in 1970 and is possibly the earliest method of structured systems development. Although it has come under attack in recent years for being too rigid and unrealistic when it comes to quickly meeting customer's needs, the Waterfall Model is still widely used. It is often attributed with providing the theoretical basis for other process models, because it provides a "generic" model for software development.

The demise of the waterfall model has been predicted for many years. However, a recent survey (Laplante and Neill, 2004) undertaken found that it is still very much alive and in use. Laplante and Neill state that

“in a survey of almost 200 practitioners, accounting for several thousands of projects over the past five years, the dominant process model reported was the Waterfall, with more than a third claiming its use.”

The model consists of the following major steps:

- System conceptualisation,
- Systems analysis,
- System design,
- Coding,
- Testing.

The major criticism of the Waterfall Model is its assumption that the requirements will not change during the lifecycle of the project. The reality is that requirements often do change in many, if not all, projects. The failure of the Waterfall Model to recognise this could be argued to be a fundamental flaw. A mistake in the requirements phase can not be detected in the Waterfall Model until near the end, when the customer gets to see the product. This leads to a huge cost in correcting any defects. (Economics of faults).

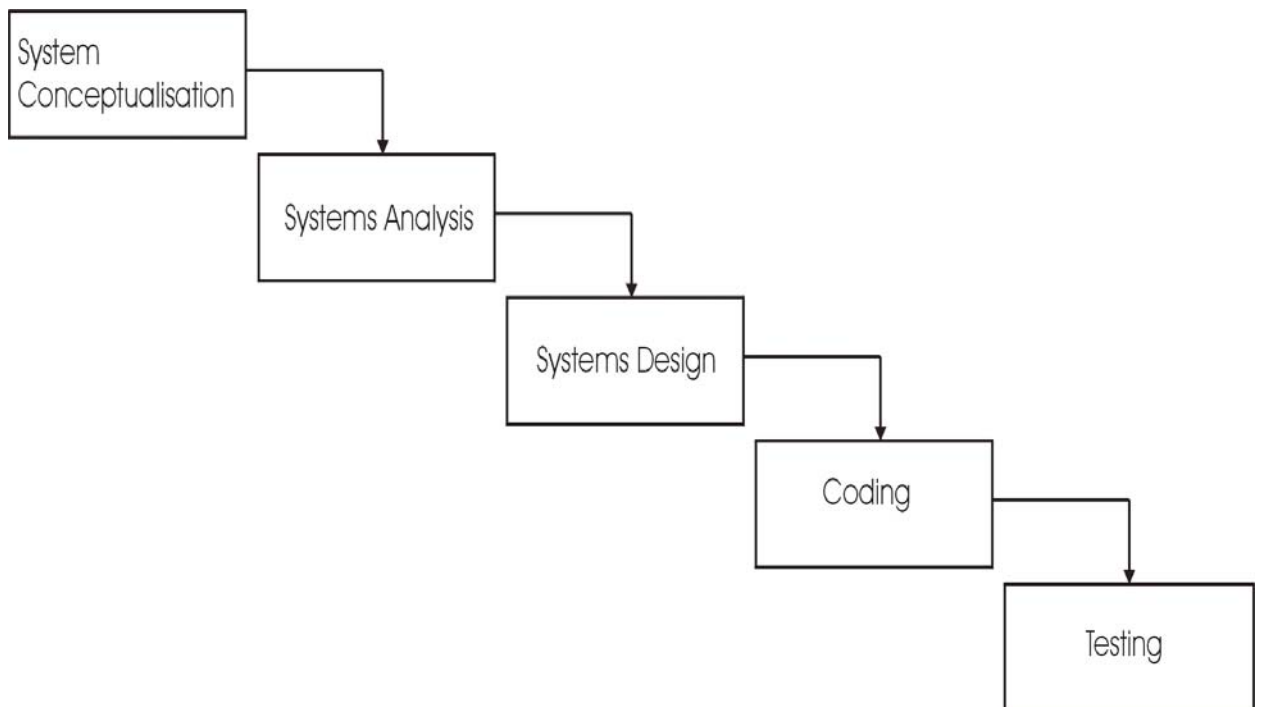


Figure 9. The Waterfall Model

7.1.2 The Spiral Model

The Spiral Model as originally designed by Boehm (1988), and reproduced below, consists of repeating the Waterfall Model in a series of rounds. Each round consists of four phases:

- Determine the objectives,
- Evaluate any alternatives,
- Develop the product,
- Plan the next series of rounds.

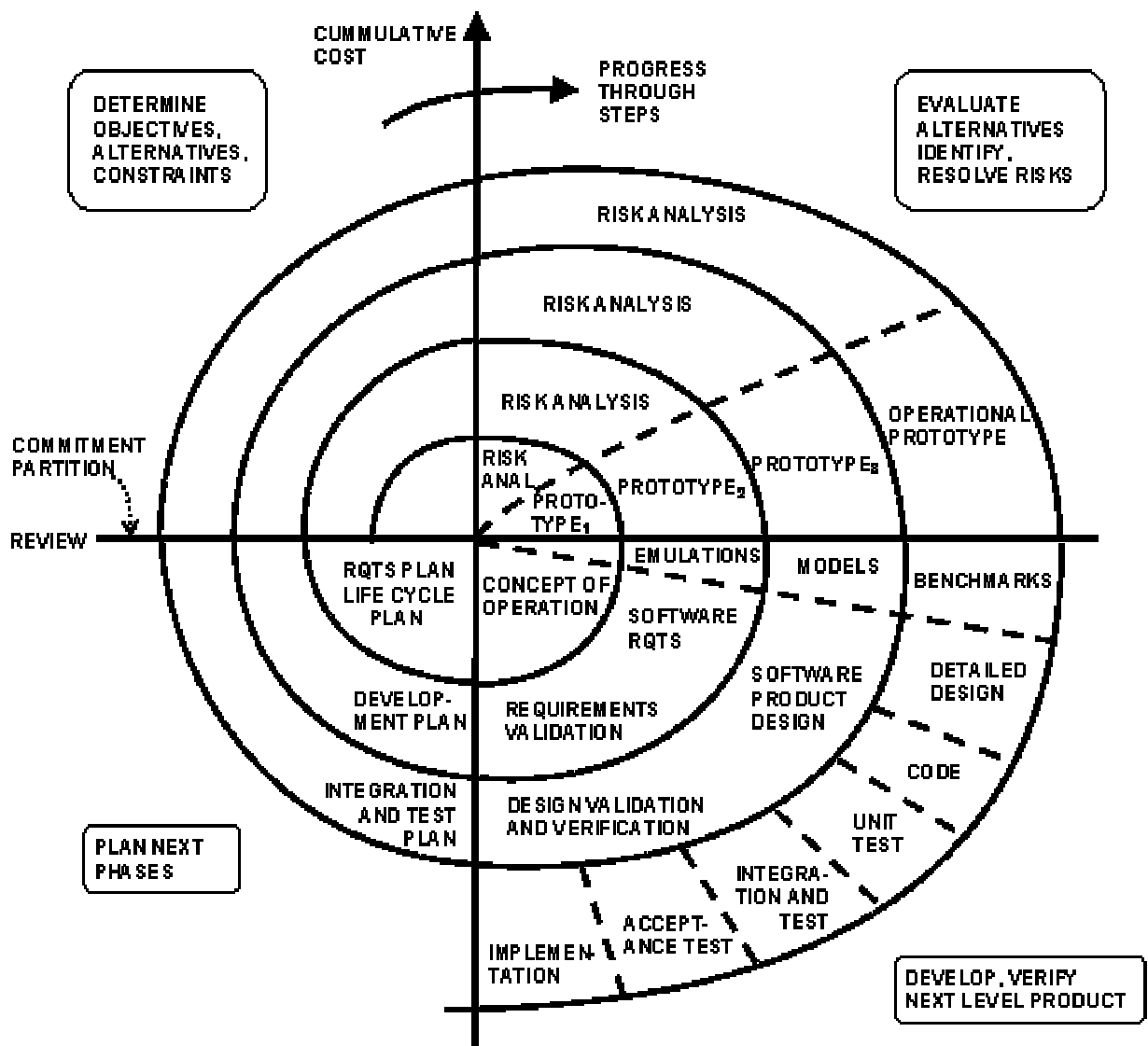


Figure 10. The Spiral Model

7.1.3 The V-Model

As originally described by Ould and Unwin (1986) the V Model was known as the U Model and was laid on its side. It is an extension of the Waterfall Model used in software development and testing. The V model associates each development activity with a test or validation at the same level of abstraction. Each development activity builds a more detailed model of the system than the one before it. Each validation tests a higher abstraction than its predecessor.

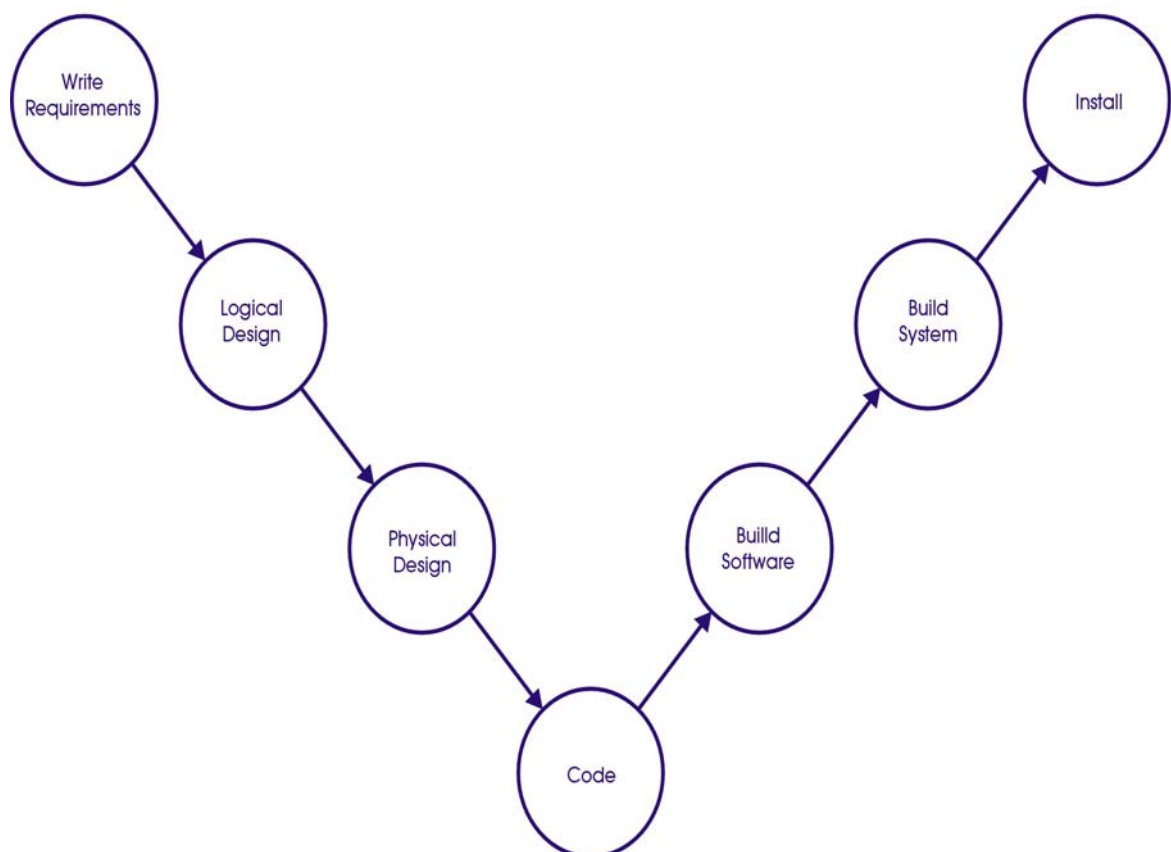


Figure 11. The V Model

7.1.4 The W Model

The W Model can trace its history back to the V Model, and prior to that the Waterfall Model.

Both the Waterfall and V Model, whilst encouraging process and method, are not explicitly quality mechanisms. However, by extending the V Model to include an explicit quality element, the W Model is its natural output. This quality element is expressed in the model by the mandatory use of testing throughout the project lifecycle. Whilst, as previously stated, originally developed for software engineering, the model can be adopted for any project as it dictates the testing of specifications, designs and any other project deliverable through a process of peer review.

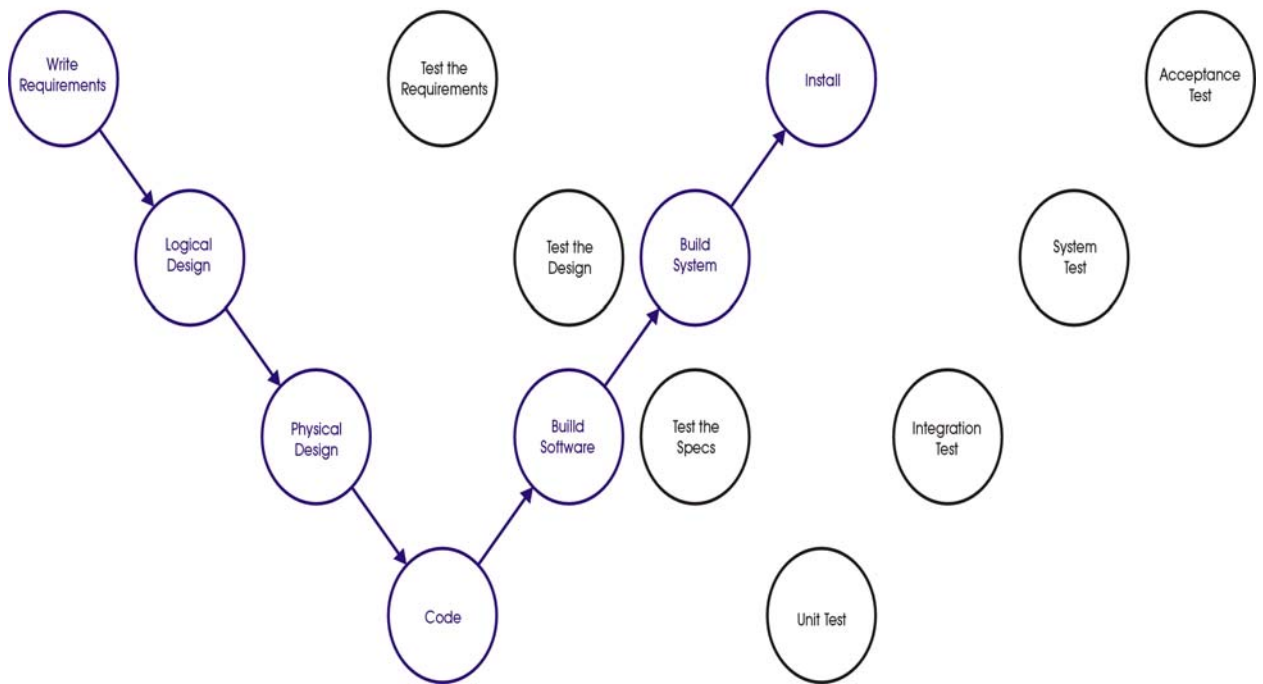


Figure 12. The W Model

7.2 Information Systems Methodologies

There are a number of characteristics which can be expected to be present when the term methodology is applied. These include the ability to accurately record requirements and to provide a systematic method of development. A methodology is a collection of methods and therefore a methodology is applicable to a wider range of information systems development than a single method.

A methodology is

“a recommended collection of philosophies, phases, procedures, tools, documentation, management, and training for developers of information systems”. Maddison (1983)

Or according to Checkland (1980)

“a set of principles of a method which in any particular situation has to be reduced to a method uniquely suited to that particular situation”.

Whilst Avison & Wood-Harper (1991) state that a

“methodology is a coherent collection of concepts, beliefs, values and principles supported by resources to help problem-solving groups to perceive, generate, assess and carry out, in a non-random, changes to an information situation.”

A methodology must show what steps to take, how these steps are to be performed and describe why they are being performed.

The relationship between a methodology and a model, according to Avison and Fitzgerald (1995), is that a model is the basis of the methodology's view of the world.

The information systems industry has long realised that a standard approach using proven analysis and design techniques should improve the 'quality' of systems. There are an almost innumerable number of information systems development methodologies in existence. These range from commonly used and widely accept methodologies that have almost become de facto standards, to methodologies designed within an individual organisation. This project will confine itself to those methodologies that are in the public domain and have at least a degree of usage. The descriptions given below are in a rough chronological order of development and adoption.

7.2.1 Jackson Systems Development

JSD was developed in the 1970's by Michael A. Jackson. The methodology has three major stages:

- Modelling Stage
- Network Stage
- Implementation Stage

The modelling stage is concerned with understanding the organisation, especially in reference to the areas of the organisation in which the system will be working. The description is usually represented as a series of diagrams.

The network stage is concerned with defining the things that the system has to do, including inputs, outputs and interfaces. The system is described as a network of programs.

The implementation stage is concerned with the actual code production, but also considers the technical environment in which the system must operate. This stage also defines data structures.

7.2.2 STRADIS

Structured Analysis, Design and Implementation of Information Systems is a methodology developed by Gane and Sarson (1979). The methodology is based on the

idea of top down functional decomposition and relies on the use of Data Flow Diagrams, a common technique still in use.

7.2.3 SSADM

The Structured Systems Analysis and Design Methodology, was originally developed in 1980 and selected by the Central Computer and Telecommunications Agency (CCTA, now part of the Office of Government Commerce) in 1981 as the methodology for UK government projects. It adopts a prescriptive approach to information systems development, that is to say, it specifies in advance the modules, stages and tasks which have to be carried out, the deliverables to be produced as part of the project and furthermore it specifies the techniques used to produce the deliverables.

SSADM adopts the Waterfall Model of systems development, where each phase has to be completed and signed off before subsequent phases can begin. It is now on version 4.2.

As with all methodologies that follow the Waterfall Model, it has been criticised as being inflexible to changes in requirements, however, its prescriptive approach also ensures that all necessary tasks are undertaken to assist with the successful delivery of a system.

7.2.4 RAD

RAD (Martin, 1991) approaches began to be adopted in the late 80's and are based on a number of fundamental premises, the most important being the acceptance that business processing requirements will inevitably change during the development cycle of a system. RAD is a programming methodology that enables software developers to build working programs in relatively short timescales.

Common examples of RAD tools are Visual Basic, Visual C++ and Delphi, which are visual programming languages. The introduction of this type of language has caused a blurring between RAD and more traditional development methodologies.

The RAD approach has risks in that it is only suitable for certain kinds of applications, i.e. projects which do not have well defined and rigid requirements, do not require complex programming, are interactive with clear functionality at the user interface and have a clearly defined user group.

RAD is not suitable for real-time or safety-critical applications or for applications where functional requirements have to be fully specified before any programs are written.

7.2.5 MERISE

Merise, developed by Quang and Chartier-Kastler in 1991, is a methodology that is widely used in France, Spain and Switzerland. It consists of three 'cycles', the decision cycle, the life cycle and the abstraction cycle. The abstraction cycle is the most important. In this cycle data and processes are viewed firstly at the conceptual level, then the logical or organisational level and finally at the physical or operational level. This methodology has evident shared characteristics with JSD.

7.2.6 YSM

The Yourdon Systems Method was developed in 1993 by Edward Yourdon. YSM is similar to STRADIS in its use of functional decomposition. It differs in that it uses a middle-out approach and places more emphasis on the importance of data structures.

7.2.7 Euromethod

The Euromethod came about in 1994 and could best be described as a providing a framework for the integration of existing European methodologies rather than as a methodology in its own right.

7.2.8 DSDM

The Dynamic Systems Development Methodology came into being in 1995. It is essentially a Rapid Application Development (see earlier section) methodology that is ideally suited to the development of web-based applications. It makes the starting

assumption that requirements will change throughout the project and adopts an iterative approach.

7.2.9 Object Orientation

The Object Orientation (OO) approach to information systems development can be viewed as networks of objects that communicate with each other via messages. An object consists of data attributes and a set of methods that enable objects to be created, updated, queried and deleted. One of the primary objectives of OO is the re-use of software. The concept behind OO is that all the processing complexities are encapsulated within the object itself. Using OO allows information systems to be constructed from existing objects.

7.2.10 Unified Modelling Language

UML has evolved through an open community, with version 1 being released in 1997. It is a methodology that allows software designers to visualise and design software through twelve standardised drawing types. It is a methodology that specifically lends itself to OO development but can be used to complement more traditional methodologies.

7.2.11 Other Methodologies

Other methodologies uncovered but not described include:

- *ETHICS*. Effective Technical and Human Implementation of Computer-Based Systems (Mumford, 1995),
- *Participatory Development*. (Floyd et al 1989),
- *SSM*. Soft Systems Methodology, (Checkland, 1981, Checkland and Scholes, 1990),
- *ISAC*. Information Systems Work and Analysis of Change (Lundeberg et al, 1982),
- *MULTIVIEW*. (Avison and Wood-Harper, 1990).

8 Critical Analysis

The purpose of this section is to analyse existing models for the design and implementation of records management systems and to compare and contrast them against accepted practises in the design and implementation of information systems.

Terry Cook (1997) describes the differences between information systems and records management systems by saying that

"information systems (which is what we have) contain data that are timely; efficient from a technical perspective ... manipulable; and non-redundant -- old data are bad data, and are therefore replaced by new, updated, correct data. Record-keeping systems (which is what we need, and largely do not have) are just the opposite: they contain records that are time-bound and context stamped; inefficient technically ... inviolable and unchangeable once created; and redundant -- old data are not condemned as outdated and therefore deleted, but are viewed as being just as valuable as new data."

This is largely a true statement, but somewhat simplistic (unlike the punctuation). These systems behave in the way that they do because that is how they have been programmed. By explicitly defining within the requirements document the definition of a records management system as given in Section 5.5.3 of this document, these weaknesses are effectively made redundant,

There is, in reality, one model in common usage for the design and implementation of records management systems, although it can be found in two places. These are in "Design and Implementation of Record Keeping Systems" (DIRKS) and ISO 15 489. The models are identical in function although they have a slight difference in their graphical representation. ISO 15 489 essentially 'borrowed' the model from DIRKS.

Both variations are shown below.

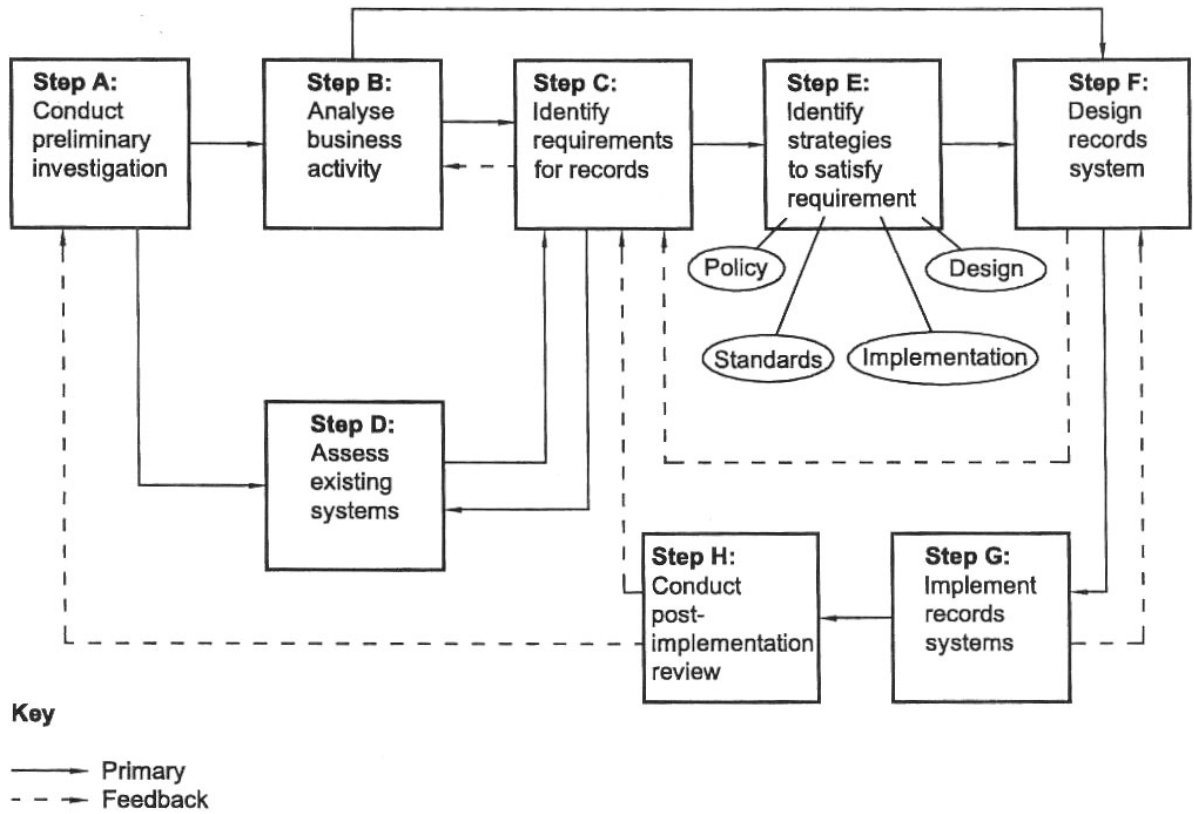


Figure 13. ISO 15489 Model

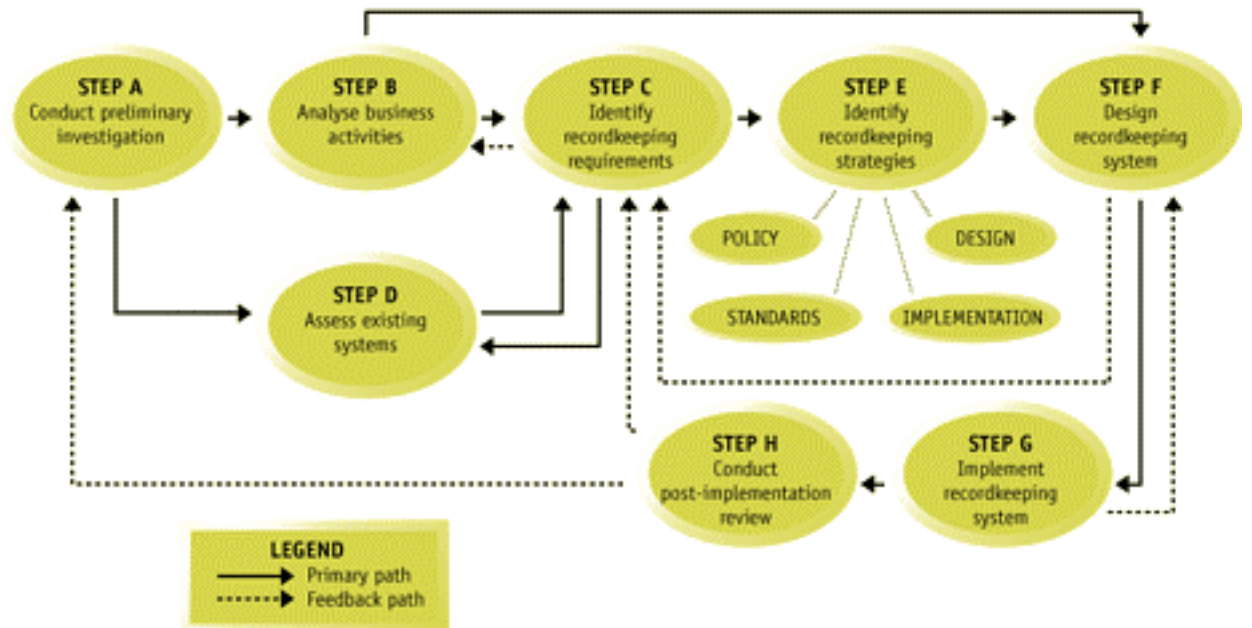


Figure 14. The DIRKS Model

8.1 Criteria

To adequately analyse the existing models, it is necessary to identify criteria against which to compare and contrast them. Criteria that are likely to prove useful for this purpose are described below.

David Bearman (cited in McGovern and Samuels, 1997) has identified four tactics that can be used to successfully implement a records management system:

- *Policy.* Provides guidelines for the use of electronic systems and lays the foundation for the adequate creation, retention, and access to electronic records.
- *Design.* Ensures that adequate provisions are designed into systems so that required information is created, maintained, or destroyed as authorized; adequately documented; made legally acceptable; maintained; and kept usable over time.
- *Implementation.* Ensures that users and technical support staff understand what is required to manage electronic records.
- *Technology standards.* Encourages the creation and use of appropriate standards.

These four tactics are already included in the models as part of Step E and the source is attributed to Bearman. The models, DIRKS especially, further develop these tactics but only in a discursive form. For the purpose of the analysis these four tactics will be used as assessment criteria against the model as a whole rather than as part of one step.

Avison and Fitzgerald (1995) developed a framework for comparing methodologies. The framework consists of 7 elements:

- *Philosophy.* According to Avison and Fitzgerald, a philosophy is a principle or set of principles that underlie a methodology. In fact they define a methodology as a set of techniques underpinned by a philosophy, which is further supported by Avison and Woods-Harper (1991).

- *Model.* The model is the basis of the methodology's view of the world, e.g. the Waterfall Model, V Model or Spiral Model.
- *Techniques and Tools.* In general, a methodology will utilise a set of related techniques, such as Entity Relationship Modelling (ERM) and Data Flow Modelling (DFM) and may use Computer Aided Software Engineering (CASE) tools to support the techniques.
- *Scope.* The scope of a methodology defines its start and end points within the information system's lifecycle. For example, does the model incorporate all stages from an initial feasibility study through to maintenance of the system?
- *Outputs.* This is concerned with the definition of the deliverables to be produced during the phases of the methodology. Whether these deliverables are described in detail will be an important factor.
- *Practice.* This element looks at the use of the methodology in terms of the differences between the theory and the practice.
- *Product.* This element looks at the nature of the product itself, in terms of documentation, CASE tool support, training courses, maintenance and on going support and any user base and user groups that might exist.

The criteria described above will form the basis of the evaluation to be carried out.

Other criteria that will be used to evaluate records management systems design and implementation methodologies are noted below:

- *Feedback loops,*
- *Explicit quality assurance.*

8.2 Analysis

The DIRKS and ISO 15 489 models have many strengths, which are described below. Strengths identified must, wherever possible, be included in the prototype model. Weaknesses in the existing models are also discussed below. These weaknesses should be addressed by the prototype model.

The first four criterion discussed are those presented by Bearman. They have one common weakness; they are considered too late in the project. Policies, standards and technology should all be considered as part of the requirements gathering step.

8.2.1 Policy

As one of Bearman's tactics built into the models, there is an explicit activity within Step E. Furthermore, the models dictate the use of relevant policy in other places such as can be found in Step C of the model, that of identifying records management requirements. The requirements must define any policies and legislation and regulations that are applicable.

The concept of policy, or more importantly the identification and inclusion of relevant policy, is probably appropriate to the design and implementation of records management systems as a whole and the restriction within the model of first mentioning policy at Step E is a weakness; policy has a role to play throughout the lifecycle of the system and should be introduced earlier in the lifecycle, probably during the requirements definition step.

8.2.2 Design

The models themselves do not explicitly define any tools for the design and implementation of the system. This is a good thing in that technology changes mean that new technical architectures are coming to the fore on a regular and frequent basis. This is discussed further in the Technology Standards section.

However, the models do contain, in Step C, references to defining the records management requirements. These requirements will ensure that both Bearman and Cook's points are addressed, but as the design will be based on the requirements, it is too late in the development process to first consider the design in Step E.

The models also have an explicit design step, Step F.

8.2.3 Implementation

The implementation of a system is a major task. Within the existing models it is shown as one step, Step G; this could be described or shown to include sub-steps to ensure that, for instance, the business processes are adequately implemented.

The implementation of a new system is generally carried out in a staged approach as opposed to a 'big bang'. This could involve a pilot implementation for a small group of key users before being expanded to a department and finally organisation wide. The current models do not reflect this commonly used approach.

The current models do not, on their own, ensure that both users and technical support staff understand the records management aspects of a system. This is probably too much to expect from any model and retrospectively might not be a particularly useful criterion.

8.2.4 Technology Standards

As with the previous three criteria, the models go no further than to recommend that relevant technology standards are complied with. This is clearly a sensible thing to do and as a tactic to successfully design and implement a records management system is a useful one. The investigation of technology standards and their current usage could be viewed as fulfilling the criterion.

Whilst fully accepting that technology is a fast changing arena and therefore one of fast changing standards, it could be useful for the description of the model to contain details of the kinds of standards to be aware of.

8.2.5 Philosophy

The models have a strong philosophy. The basis of this philosophy is that it is necessary to have a structured, controlled approach to the design and implementation of records management systems. However, the philosophy does not have a strong enough bent towards quality.

8.2.6 Model

The models are simple enough to be easily understood by most people regardless of their exposure to IT. This is obviously a benefit; a simple model is more likely to be adopted and followed than a complex one. Whilst a seemingly minor point, the type face in the ISO 15 489 version of the model is somewhat clearer, as is its general layout in terms of using rectangular boxes instead of ellipses.

The models can be viewed as generic. This is a useful attribute in that it means that the models can be used in a variety of situations and tailored to match the specific needs of an individual organisation of any size.

The existing models make use of, to some extent, models and methodologies that are accepted in the information systems development world. The DIRKS and ISO 15 489 models are based on The Waterfall Model which has the inherent weakness of not reacting to changes in the requirements. Whilst the implications of this weakness should not be underestimated, it is fair to say that at some stage the requirements must reach a point whereby they are frozen. This means everyone in the project knows what is being implemented and is a powerful tool in ensuring that the delivered system meets expectations. Freezing requirements does not necessarily mean that they are unalterable; by incorporating a change control process into the project new requirements can be added or redundant ones removed, but within a controlled environment.

The existing models jump from designing the system to implementing it. This could be viewed as reasonable in that the system development, if a bespoke one, is likely to be carried out by specialist software developers. However, their omission could also be

viewed as a weakness in that these development activities must take place, and should therefore be covered by the model. Indeed these development activities are likely to represent a significant proportion of the overall project both in terms of time and cost. There are existing methods, some of which are described earlier in this paper, that will be appropriate for the development of these steps.

If the assumption is made that the system development activities are treated as being distinct from the rest of the project to design and implement a records management system, or if in fact there is no development work to be undertaken as the system is a commercial off the shelf one, it is still useful to expand on the activities that must occur between designing the system and implementing it.

It is not a good idea to have multiple lines in each direction on a model. It makes it easy for users to miss steps as the order of flow is not explicit. For example, the models are not clear as to whether it is preferable to advance from Step B to Step C or to Step F. The second of these two options would not be sensible, as discussed in Section 8.2.10.

8.2.7 Techniques and Tools

The model, at various stages, makes use of some tools and techniques. This can be of assistance in the implementation of a records management system in that it provides a degree of guidance. More tools could be specified, although there is then the potential for the models to lose some of their generic nature, so there is a balancing act to ensure that the models remain flexible whilst also providing the right level of guidance.

As with technology standards, tools for the development of systems are likely to change, but there are others that have remained in place for some time, such as the Fagan Method for reviewing documentation.

8.2.8 Scope

The existing models have clear start and finish points, ensuring that the project does not drift on, but has a defined end. The models also, as part of Step A, define the boundaries of the organisation which is also a scoping activity.

8.2.9 Outputs

The model defines the inputs and outputs expected at each stage of the project. This is an important tool for managing the project in that it allows the project to be managed. By understanding what should be accomplished at any given point in the project and comparing that to what has actually been accomplished, the project manager is able to keep the project on track. This is the only reliable way to maintain control over the project; it is often referred to as the control cycle and involves planning what should happen, measuring what has happened and finally taking corrective action where necessary. This is shown in Figure 15 below.

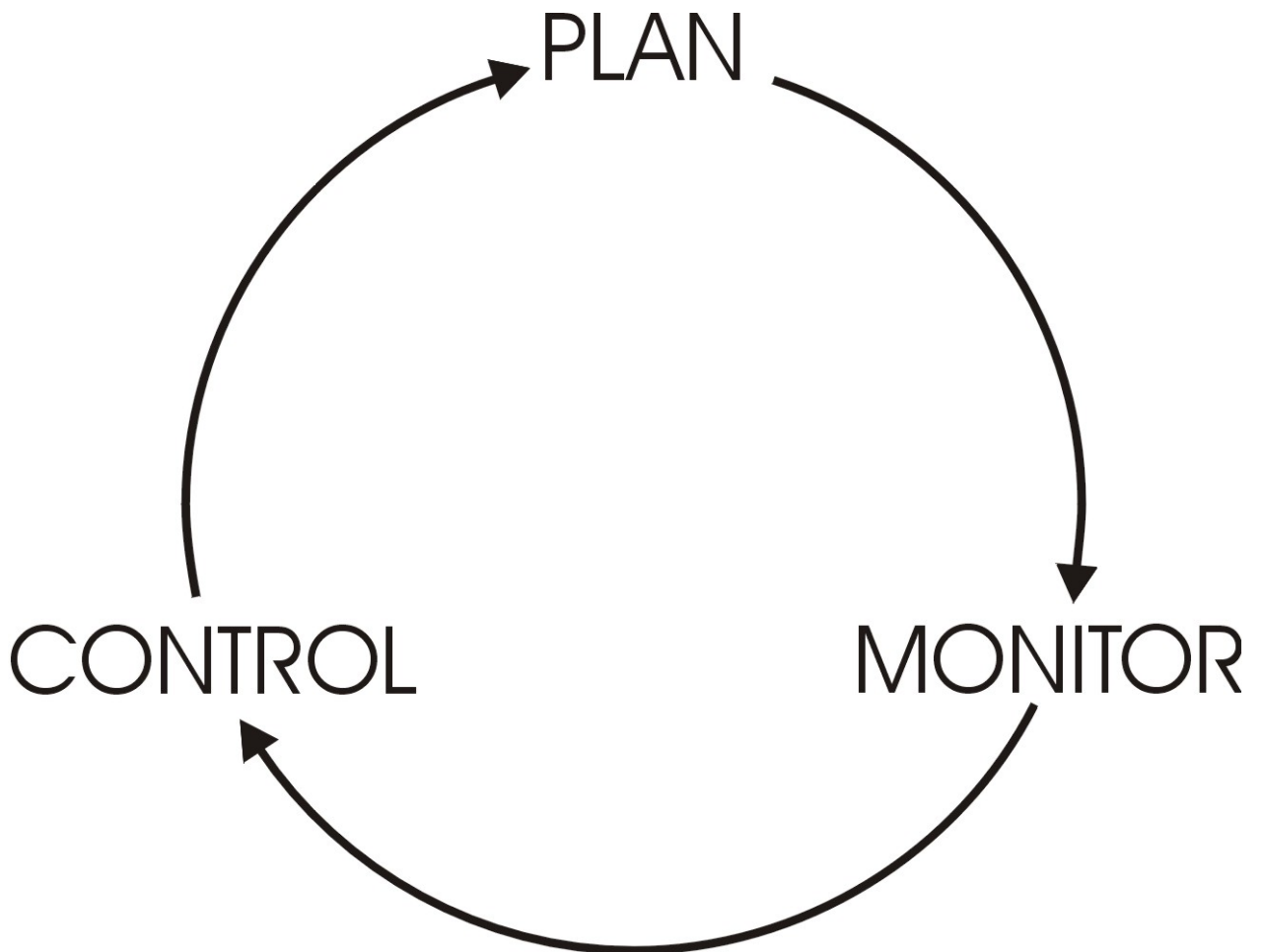


Figure 15. The Control Cycle

8.2.10 Practise

The existing models show a primary path from Step B to Step F. The inclusion of this primary path is confusing at best and plain wrong at worst. Confusing in that whilst the business processes will obviously influence the design to some extent, this should be stated in the requirements. Plain wrong in that to progress from an analysis to building the system without any specification is almost certain to cause the project to fail.

The Waterfall Model, upon which the existing models are partially based, has a defined step for testing. The DIRKS and ISO 15 489 models do not include an explicit testing step. This is a serious flaw and presents a significant weakness in the models. Testing is absolutely essential in any systems development project. A common mistake in the approach to testing is when to test. Testing should not be regarded as an activity only undertaken by technicians at the end of the project, but rather as a series of activities undertaken throughout the systems development lifecycle by project team members who understand the projects aims and objectives.

The existing models break down the “Systems Analysis” step found in the Waterfall Model into the three smaller steps of “Analyse Business Activities”, “Identify Requirements”, and “Identify Strategies”. This is useful direction in that it provides guidance on what is actually involved at this stage of the project at a practical level.

8.2.11 Product

The existing models are both well documented, with DIRKS especially providing a wealth of readily available explanatory documentation, although to an extent slightly prescriptive. None the less, this documentation explains in clear and precise terms what is to be done at each step and the expected inputs and outputs.

Considering that the existing models are the standard for the design and implementation of records management systems, there is very little training available on how to use them. Training on, for example, implementing an electronic records management system rarely seems to pay regard to the existing models.

The model does not cover the maintenance phase of a systems lifecycle, which can represent a significant proportion of a systems total cost over its entire life span. The existing models are not alone in this, many information systems development models likewise omit maintenance. None the less, this omission should be regarded as a weakness.

The amount of ongoing support available for the models has not been ascertained. Clearly, the support of the models must be viewed as a separate exercise to the support of DIRKS or ISO 15 489. The ISO in particular has representatives throughout the world.

8.2.12 Feedback Loops

The model is based, as previously stated, on the Waterfall Model; however, it has been modified to a small degree to provide feedback loops that allow for the model to be regarded as cyclical. To an extent, these feedback loops give the impression that the models are edging towards the spiral model of software development, but do not go far enough to complete this transition from waterfall to spiral as the feedback loops are one offs and not part of a continuous iterative process.

The last step, the Post Implementation Review, feeds back into the first, allowing for potential modifications of the next version of the system based on lessons learnt during the implementation of the first. This is likely to negate the need for a feedback path from Step H to Step C. This is not because the information making up this feedback is without use, but because it is best fed back into the start of the lifecycle rather than the middle.

Other of the feedback paths do not make as much sense. The feedback path from Step G to Step F is not useful. Whilst gathering feedback post implementation is useful to guide future versions, the feedback should be presented as a series of requirements, whilst altering the design of the system during implementation, as suggested by the models, should be discouraged.

The feedback path from Step C to Step B is not useful either. The analysis of business activities is just that, an analysis. If the business activities undertaken do not conform to the records management requirements then there might well be a case to alter the activities, but this should be included as part of the overall systems design, which meets the aims of Horsman as described in Section 5.5.3. Therefore the gaps identified between the records management requirements and the practise should be documented as non-functional requirements.

8.2.13 Explicit Quality Activities

There are too few explicit quality steps in the models. The models do suggest a review of the business processes identified as a result of Step B to validate them and also to review the design documentation produced as part of Step F. These are good things to do, but do not go far enough in themselves. Testing, as mentioned in Section 8.2.10 is a quality activity. In terms of the weaknesses identified, this lack of quality is potentially the greatest.

As discussed earlier in this paper, some 85% of software defects can trace their roots to the requirements phase of the project. This implies the need for a quality review of the requirements. This need should be expressed in the development model as an explicit activity. The more errors that are trapped and resolved at this stage of the project the better. Not only will the finished system cause fewer failures to the user, but the system is far more likely to have been delivered on time and within budget.

8.3 Summary

Criteria	Comment
Policy	This is adequately addressed, but too late in model.
Design	This is adequately addressed, but too late in model.
Implementation	Could be described in greater detail and include a pilot rollout.
Technology standards	This is adequately addressed, but too late in model.
Philosophy	This is adequately addressed
Model	Lack of quality inherent in the existing models
Techniques and tools	This is adequately addressed, helpful rather than inflexible.
Scope	This is adequately addressed.
Outputs	This is addressed well.
Practice	The practise is not always aligned with the theory.
Product	The maintenance phase is not addressed, but otherwise addressed well, especially in terms of supporting documentation.
Feedback loops	This is not addressed well.
Explicit quality steps	This is not addressed well.

Figure 16. Summary of Analysis

9 Gap Analysis

This section provides a brief summary of the strengths and weaknesses identified as part of the critical review, although it is the weaknesses that are of the most relevance as these form the gaps that need to be addressed.

9.1 Strengths

- The models are simple.
- They include references to relevant policy.
- The models are generic, allowing usage in various environments.
- They are based on an accepted development model.
- They use tools and techniques.
- There are defined start and end points.
- Inputs and outputs are defined.

9.2 Weaknesses

- There are no defined steps between design and implementation.
- Some activities occur too late in the lifecycle.
- There are too few explicit quality activities.
- There are no test activities.
- There are not enough specified tools and techniques.
- They define incorrect feedback paths.

10 Prototype Model

The purpose of this section is to provide a brief description of the prototype model and how it attempts to address the gaps identified.

10.1 Description

The models given in DIRKS and ISO 15 489 are the same model. This model is, in essence, a Waterfall Model in that each step leads to the next. Whilst, as seen previously, the Waterfall Model has been criticised in the past, it is a model that is still frequently used, perhaps because it is easy to understand.

A derivation of the Waterfall Model is the W Model and the evolution from Waterfall to W is described in Section 7. The prototype model is based on the W Model. It has taken the steps defined in DIRKS and ISO 15 489, retained the naming conventions and presented them as a W Model. To do this it has been necessary to introduce new steps. The introduction of these new steps also serves to describe the implementation phase of the project in greater detail, therefore addressing another one of the weaknesses identified in the existing models. These new steps introduced are:

- Build system,
- Configure system,
- Build and configure workflows.

Each of these additional steps correlates to a design step, which gives the first 'V' of the model, shown in blue. Each of these additional steps has a corresponding quality step; these are shown in black, which gives the second 'V' in the model. The emphasis of the prototype model is to build quality into the system from the outset as opposed to finding defects at the end which, as previously described, is more time consuming and expensive to rectify.

10.1.1 Build System

This new step is concerned with the development of the system itself. This is a specialist activity and is best undertaken by suitably experienced software developers. If the system being developed is a commercial off the shelf product (COTS), this step will be missed out, although its quality activities won't be. Whilst it might be unreasonable to expect records managers to have an in depth understanding of software development, at least one member of the project team should, which might mean co-opting a suitably experienced person onto the project team.

It would also be useful for the developers to have their development procedures assessed prior to the project starting to check that they have a suitable quality system in place, probably against the Capability Maturity Model or similar.

10.1.2 Configure System

This is a different step to building the system, not just semantic differences in its name but in what the step actually involves. During the build stage of the project, pre-defined requirements will be met. If, for instance, and is likely, the system needs to be able to incorporate retention scheduling, then during the system build step this functionality will be implemented within the system, but it would be foolish to hardwire it. It would be much more sensible and useful to build the system in such a way as to allow new retention schedules to be added as required. This is an example of what is meant by configuring the system., or in other words, the ability to customise or tailor the system to meet the specific needs of an organisation.

The same process will be applied to the classification scheme, this is something that will be implemented dynamically and although it could be hardwired, it wouldn't actually leave any flexibility to react to changes.

This configuration step will largely be governed by the requirements.

10.1.3 Workflow

Once again it would be foolish to hardwire any workflows into the system as they are based on business processes that are liable to change.

10.2 Addressing The Gaps

The prototype model addresses the gaps identified as described below.

10.2.1 Implementation

The prototype model is based on the W Model instead of the Waterfall Model. It has associated development activities with design activities and simultaneously introduced explicit quality activities. It has expanded the implementation phase and removed the primary path between Step B and Step F.

10.2.2 Tools and Techniques

The model has not specified the use of any specific tools and techniques to any large extent, but the use of quality steps does imply their use. The use of testing, for example, will require the usage of specialist testing software, whereas reviewing the requirements might use the Fagan Method. Similarly, the audit of the software developers might be against the Capability Maturity Model. Individuals and organisations might have their preferred methods, the prototype model only suggests that quality activities are a useful thing to do, not how to do carry them out.

10.2.3 Practise

The major difference between the theory and practise in the existing models is the lack of testing. The Waterfall Model has a clearly defined testing step, but this is not present in the existing models. The prototype model has introduced testing at every step, thus also addressing the lack of explicit quality activities. It is important in any discussion on testing to understand why testing is done at all. There are many reasons that could be put forward, including:

- To find bugs,
- To prove the software works,
- Because my team leader told me to,
- To assess the quality of the software,

But, arguably, a tester should have one objective in mind when undertaking testing, to prove that the software does not work. The mindset of the tester must be to find errors, faults and failures.

Testing can essentially be thought of as falling into two categories, static and dynamic. Dynamic testing is that with which most people are familiar and involves testing a system and its code, or programming. In the case of User Acceptance Testing, this involves testing that the requirements have been met, which has an evident dependency on the quality of the initial Requirements Specification. Dynamic testing will also include testing individual modules of code, testing these modules together and testing the system as a whole.

Static testing is the testing of documentation and is essentially the type of testing that will be used throughout much of the design and implementation of a records management system using the prototype model. There are a number of techniques in use to aid with the testing, or the reviewing of documentation. Two commonly used are the Fagan Method and the Delphi Method. Either would be applicable in the design and implementation of records management systems.

10.2.4 Feedback Loops

The critical analysis suggested that some of the feedback loops were either not necessary or not useful. To address this, the prototype model does not include two feedback paths that are present in the existing models. The paths from Step G to Step F and from Step F to Step C have been removed.

10.3 Prototype Model Diagram

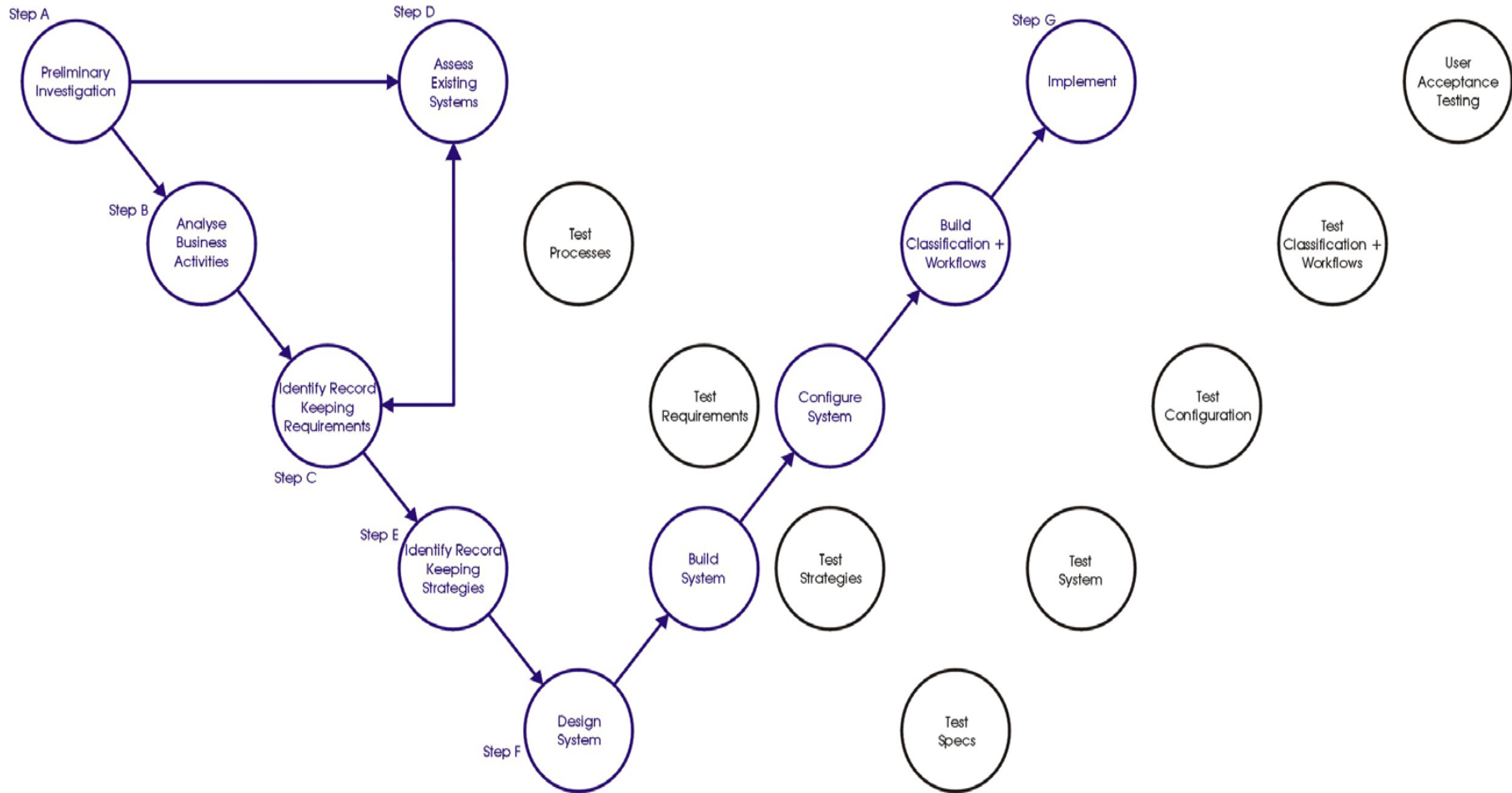


Figure 17. The W Model for Records Management Systems

11 Data Analysis

The purpose of this section is to analyse the data collected and present that analysis. During the planning stage of the project, two methods of data collection were identified, either focus groups or interview. In the event, due to the practicalities of hosting a focus group, data was collected by interview.

11.1 Questionnaires

11.1.1 Overall Responses

	As number	As %
Number of respondents	6	
Number familiar with the existing models	4	66%
Number familiar with information systems development models	5	83%
Number who believe the existing models are adequate	0	0%
Number who believe the prototype model captures the strengths of the existing models	6	100%
Number who believe the prototype model addresses the weaknesses of the existing models	6	100%
Number who believe the prototype model introduces new strengths	6	100%
Number who believe the prototype model introduces new weaknesses	6	100%
Number who believe the prototype model is an improvement	6	100%

Figure 18. Summary of Questionnaire Responses

11.1.2 Strengths Identified

The strengths identified in the questionnaires, relating to the existing models, were:

- Accepted and/or approved by records management community,
- Thorough.

11.1.3 Weaknesses Identified

The weaknesses identified in the questionnaires, relating to the existing models, were:

- The model is too slow as a development model,
- There are no quality steps,
- Not accepted by the IT community,
- Time consuming,
- Too long spent on investigation and not enough action,
- Too inflexible for rapid implementation,
- Identifies requirements before considering the policies and standards,

11.2 Analysis of Interviews

Three of the reviewers were interviewed to gain further information, specifically in relation to the prototype model. The common themes discussed during these interviews are listed below.

11.2.1 Presentation

How the model is presented was raised by one interviewee. The issue of colours being close to each other, and not easily differentiated, was raised.

11.2.2 Description

Two interviewees were keen to discuss the lack of a description in the prototype model.

11.2.3 Lifecycle

Two themes became clear in discussions around the lifecycle of a system. The first was discussed in all three interviews and was concerned with the model's lack of inclusion of the maintenance phase of a system.

The second theme was mentioned twice and is concerned with the actual implementation of the system. It is normal practice to run some kind of pilot project or model office during the implementation of an information system or other IT project. This is not reflected in the prototype model.

11.2.4 Quality

Whilst the interviews were around the prototype model, the single most common issue discussed during the interviews, in every interview in fact, as well as in every questionnaire returned was that of a lack of quality in the existing models, either implicit or explicit. This ranged from a lack of testing to a lack of documentation review. All interviewees agreed that the prototype model addressed these concerns.

12 Conclusions

The research method used for this project was based on the systems development research method. It used the constant comparative method of data analysis to assess the usefulness of the prototype model developed, which then led to the identification of further work. Two pieces of work, a literature review and a critical review of existing models for the design and implementation of records keeping systems, led to the identification of gaps in those existing models. The prototype model seeks to address those gaps. Retrospectively, this methodology would still appear to be the most appropriate method for the work undertaken.

No research was uncovered that was able to describe failure rates in the design and implementation of records keeping systems in the same way that such statistics exist in the information systems world. This means that it would be difficult, at the least, to make any kind of useful assessment as to the ability of the prototype model to reduce the number of defects in the design and implementation of records keeping systems. However, it does seem likely that a model that explicitly defines the quality activities throughout the entirety of a systems lifecycle will contain fewer errors than one which does not. This is, to a degree, supported by statistics cited at the beginning of this paper regarding the major causes of failures and how to deal with their causes.

The existing models have more strengths than they do weaknesses; indeed the fact that there are models at all is a strong starting position. The existing models are based on an information systems development model that, whilst often criticised, is a model that is still frequently used because of its simplicity. This in itself is a benefit. This meant that the prototype model had to be one that easily captured the simplicity of the existing models in terms of its comprehension and usability.

The prototype model has essentially followed an evolutionary path found in information systems development and condoned by software testers especially to emerge as the W Model. It has retained the strengths of the original models and addressed the weaknesses to hopefully reduce the number of defects in the design and implementation of records management systems.

12.1 Recommendations

12.1.1 General

Recommendation 1

The prototype model needs to have much greater exposure in terms of assessing its relative strengths and weaknesses before it is clear as to whether this model will be of more use than the existing ones. It is therefore recommended that a larger group of reviewers be approached. An alternative would be to publish the prototype model as it is and to seek feedback from the publication's readership.

12.1.2 Presentation

Recommendation 2

The colour differentiation between the design and implementation steps and the quality steps should be greater. Blue and black can be confused. Black should be used for the development activities and another colour for the quality activities.

Recommendation 3

The labelling system in the existing models should be extended into the prototype model so that the design and implementation steps are all sequentially labelled and those labels correspond to a quality activity.

12.1.3 Description

Recommendation 4

The prototype model is only that at this stage, it does not have the descriptive elements of DIRKS which define the inputs and outputs of each step. This must be addressed.

Recommendation 5

It could be useful to include specific tools and techniques in the model, although there would also be a danger of making the model too rigid and inflexible. Tools and techniques should be investigated and their suitability assessed.

12.1.4 Lifecycle

Recommendation 6

Bearman's tactics are useful ones, but probably too late in the existing models. They should be incorporated into the prototype model as part of the requirements gathering step.

Recommendation 7

The prototype model does not currently explicitly define a staged approach to the implementation of a records management system. This is specific to the actual rollout and, as mentioned in the critical review, many systems implementers advocate this approach. The rates of failures in information systems projects also seem to support this view, as described in Section 6.1. A staged approach should be built into the model.

Recommendation 8

The inclusion of the maintenance phase of a systems lifecycle should be considered. This would at least encourage records managers and others implementing a records management system to consider the total cost of ownership of such a system.

12.2 Reflections

Not only is the topic of the dissertation one in which I was, and still am, interested, it is also one in which I have a level of background knowledge having studied software engineering and later completing an IT degree. I am also an experienced and qualified software tester and project manager. The software engineering especially gave me a solid foundation in development methods. Indeed it was this very background that first led me to consider this topic for my dissertation, as it seemed a useful thing to bring experience from one discipline into another.

12.2.1 Aims And Objectives

- I have never been completely comfortable with the title of my dissertation. I am not convinced that it accurately reflects what I wanted to do. Perhaps a better title would have been

‘The applicability of information systems development models in reducing the number of defects in the design and implementation of records management systems.’

- I have not really identified techniques, as stated in the original title, but developed a new model; furthermore ‘to improve quality’ is too subjective a term to be particularly useful, whereas ‘reduce the number of defects’ is measurable and therefore infinitely more useful.
- The aims and objectives of the project were sensible and useful, if perhaps a touch ambitious. I am not convinced that I have adequately dealt with my second objective, that of comparing and contrasting the DIRKS and ISO 15 489 model with existing information systems development models. Some of this has been done implicitly as part of other sections in the paper, but I don’t think that I have adequately explicitly defined this comparison in the paper.
- It soon became evident that I would struggle to reach any kind of conclusion to the final objective, that of reducing the number of defects in an implemented records

management system. However, this has occurred only because I have not been able to test the model yet, and I remain convinced that once it has been used it will reduce the number of defects and will prove to be a useful tool for records managers.

- The prototype model I have developed is not a new model as such. I have taken what was already available, that is the DIRKS and ISO 15 489 model, and extended it to a W Model in the same way that software testers have extended the Waterfall Model. This should mean that it retains the strengths of DIRKS and ISO 15 489 whilst simultaneously addressing its weaknesses.
- The prototype model does not include many specific tools or techniques, which is an area that I need to take further to ensure that the model is as useful as possible.
- I have come to the conclusion that I could have chosen a topic that would have made use of questionnaires and made life much easier for myself. That said, this topic has, and I believe this is more important, been one that interests me.
- I have some concerns as to whether I will be able to give the model I have developed any kind of exposure in the records management arena.

12.2.2 Planning And Delivery

- Planning the project was crucial. It removed any initial trepidation or fear caused by the size of the project by breaking it down into manageable pieces of work. Each piece of work was then given a target completion date which then enabled me to assess whether or not I was on course to complete the project on time.
- I initially set myself an aggressive timetable for the project. I wanted the project to be completed by the end of November so that I would be free over the Christmas period; the timetable allowed for this to happen and also included a degree of slack. In the event, I made use of the unallotted time and finished the project mid December.

- When I initially developed the plan I was able to factor in known events such as holidays and some work commitments, but inevitably wasn't able to account for everything. I spent the whole of October and much of November presenting end user training at a client site in central London, which impacted on my ability to work on my dissertation purely due to the length of my working day. It is possible to accomplish some work on the train such as proof reading and assessing feedback.
- Keeping the momentum going is imperative. I found it hard to get going again after a break for a holiday. However as I had made a solid start before this I was still on schedule overall.

12.2.3 Existing Literature

- Quite early in the project it became evident that I would struggle to find much literature, or specifically research, around the DIRKS and ISO 15 489 models. Conversely, it just as early became evident that there was an abundance of both research and literature on information systems development models and methodologies and sorting the wheat from the chaff became a laborious and time consuming task. However, once this initial sift had taken place I was left with some extremely useful material.

12.2.4 Data Collection

- It appears to me, rightly or wrongly, that any dissertation for this course will score higher, or at least be easier to complete, if the topic lends itself to questionnaires as the data collection method, and indeed the data analysis as elements of the project are simplified.
- I could have confined the data collection to the pre-interview questionnaires but I do not believe that this would have been the best approach. By interviewing I was able to gain a better understanding of the topic area and to investigate specific areas which are subjective in nature, but this comment must be read in

conjunction with others in this section regarding my lack of experience as an interviewer.

- I had initially assumed that it would be easy to gather a large number of reviewers. In practise it proved somewhat more difficult than I had assumed. However, as this was a task completed relatively early in the project, I had time to approach alternative reviewers, which I had hoped had resolved the issue. Unfortunately, when it came to carrying out the review, several people who had indicated a willingness to participate were unable to do so due to a variety of work and personal reasons.
- I would ideally have liked the prototype model to have been reviewed by more people than it was to gain more feedback, but also from a slightly more diverse group of people to give a more rounded view. Even though the project is formally complete, I might still attempt to do so.
- My major concerns in regard to the collection of data are around my expertise, or rather lack of, in designing questionnaires and interviewing techniques. The fact that a large proportion of the questions set were answered with 100% results could infer that the questionnaire itself was poorly designed as opposed to the prototype model, which seems rather unlikely, adequately addressing the weaknesses identified in its first iteration.
- I interview people as part of my job on a fairly regular basis when writing requirements etc. or when recruiting, and I had read a couple of texts on interviewing techniques prior to undertaking the interviews, but I remain to be convinced that I am sufficiently competent as to be able to adequately elucidate the feedback from the reviewers.

12.2.5 Data Analysis

- The data analysis didn't work quite as I had planned. The difference was in the analysis of the pre-interview questionnaires. I had not intended to carry out any analysis of these questionnaires; their purpose was to try and get the interviewees primed for the interview, that is thinking along the right lines, a source of guidance. However, in the end I have carried out a simplistic analysis of the replies. This simplicity coupled with the size of the sample means that any conclusions drawn from the analysis of the questionnaires should be treated with a high degree of caution.
- The second part of my data analysis, the analysis of the interviews using the constant comparative method, worked reasonably well. But again, the sample was probably too small and I am not certain if I have presented the analysis either well or correctly.

12.3 Future Application of Approach and Methods

- Whilst the research method used for this project would not be directly applicable, research into the evolution of information systems development techniques from records management techniques would be an interesting piece of work to carry out. It would appear that techniques used in information systems development, such as data flow diagrams, might have their roots in work originally carried out by records managers. It might also be the case that that whilst information systems developers have taken these techniques, used them and often improved them, records managers have made little progress. To discover one way or another whether any of this is true would be an interesting and worthwhile piece of work.
- Systems software development would appear to have a lower risk of failure than other types of software development. This would suggest that there are lessons to be learnt from the industry. It would be a useful research project to discover why this is the case; however this project would require the participation of software quality assurance experts and is probably not within the remit of records management.

13 References

Avison and Wood-Harper. 1990. MultiView - an exploration in information systems development. McGraw Hill, Maidenhead. (In Avison & Fitzgerald 1995)

Bearman, David. 1994. Electronic evidence: strategies for managing records in contemporary organizations. Pittsburgh: Archives and Museum Informatics.

Beynon-Davies, Paul. 1998. Information systems development: an introduction to information systems. 3rd ed. McMillan.

Boehm, B. 1988. A spiral model of software development and enhancement. Computer. May 1988 p61-72.

British Standard BS EN ISO 15489-1. 2001. Information and documentation – Records Management – Part 1: General. British Standards Institute.

British Standard BS EN ISO 15489-2. 2001. Information and documentation – Records Management – Part 2: Guidelines. British Standards Institute.

Brooks, F. P. 1995. The mythical man-month: Essays on software engineering, Anniversary Edition. Addison-Wesley.

Calloway, L. J. and Knapp, C. A. 1995. Using grounded theory to interpret interviews. <http://csis.pace.edu/~knapp/AIS95.htm>

Calloway, L. J. 1988 An approach for assessing tools for designing dialog structures: a study of the dialog charts. Doctoral dissertation, New York University.

CAMBRIDGE DICTIONARY OF PHILOSOPHY. Cambridge University Press.

Cash, J.I. and Lawrence, P. (eds) 1989. The information systems research challenge: Qualitative research methods. Harvard Business School Research Colloquium.

Checkland. 1981. Systems thinking systems practice. Wiley. (In Avison and Fitzgerald 1995).

Checkland and Scholes. 1990. Soft systems methodology in action. Wiley. (In Avison and Fitzgerald 1995).

Cook, Terry. 1997. The impact of David Bearman on modern archival thinking: An essay of personal reflection and critique. Archives and Museum Informatics. May 1997, p15-37.

Cooper, H. M. 1988. The structure of knowledge synthesis. Knowledge in Society, Vol. 1, p104-126.

Degelman , Douglas. 2003. Literature search essentials.
http://www.vanguard.edu/faculty/ddegelman/index.cfm?doc_id=3380. 25 June 2003.

Dictionary.com <http://dictionary.reference.com/> 23 June 2004.

Dubin, R. 1978. Theory building. New York: The Free Press.

Dye, J. F., Schatz, I. M., Rosenberg, B. A., & Coleman, S. T. 2000. Constant comparison method: The Qualitative Report [On-line serial], 4(1/2).

DIRKS

Fanning, B. 2002. Records management standards:- A starting point. E-doc. 16(5) p60-61.

Fidel, R. 1993. Qualitative methods in information retrieval research. Library and Information Science Research. 15 p219-247.

Floyd, C., Mehl, W.-M., Reisin, F.-M., Schmidt, G., & Wolf, G. 1989 . Out of Scandinavia: Alternative approaches to software design and system development. Human-Computer Interaction. 4(4), p253-350

Galliers, R.D. 1992. Choosing information systems research approaches. Information Systems Research: Issues, methods and practical guidelines. (ed. R. Galliers), Blackwell Scientific Publications, p144-162.

Gane, Chris, and Sarson, Tish. 1979. Structured systems analysis: tools and techniques. Prentice Hall.

Glaser, B.G. and Strauss, A.L. 1967. The discovery of grounded theory: Strategies for qualitative research. Aldine Publishing Company, New York.

Gorman, G.E. and Clayton, P. 1997. Qualitative research for the information professional: A practical handbook. Library Association.

Gane, Chris, and Sarson, Tish. 1979. Structured systems analysis: tools and techniques. Prentice Hall.

Haig, Brian, D. 1995. Grounded theory as scientific method. Philosophy of Education Society. http://www.ed.uiuc.edu/EPS/PES-yearbook/95_docs/haig.html

Hooks, Ivy F., and Kristin A. Farry. 2001. Customer-centered products: creating successful products through smart requirements management. AMACOM.

International Council on Archives. 1996. Electronic records management: A literature review. ICA.

Jackson, Michael. 1975. Principles of program design. London. Academic Press.

Jones, Casper. 1998. Project management tools and software failures and successes. Crosstalk, The Journal of Defense Software Engineering. July 1998. p13-17.

Josephson, John. 1994. Abductive inference in reasoning and perception. <http://www.cse.ohio-state.edu/lair/Projects/Abduction/abduction.html>

Juran, JM. 1989. Juran on leadership for quality. New York : Free Press.

Kaner, Cem., Bach, James., Pettichord, Bret. 2001. Lessons learned in software testing. John Wiley & Sons.

Kerlinger, F.N. 1986. Foundations of behavioral research. Holt, Rinehart & Winston, New York.

Knapp, C. A. 1995 An investigation into the organizational and technological factors that contribute to the successful implementation of CASE technology. Doctoral dissertation, City University of New York Graduate Center.

Kaplan, B. and Duchon, D. 1998. Combining qualitative and quantitative methods in information systems research: A case study. MIS Quarterly. 12(4) p571-587.

Laplante, Phillip, A. and Neill, Colin, J. 2004. The demise of the Waterfall Model is imminent and other urban myths of software engineering. ACM Queue. Vol. 1, No. 10, February 2004, p10-15.

Lincoln, Y. S., & Guba, E. G. 1985. Naturalistic inquiry. Newbury Park, CA: Sage.

Lundeberg *et al.* 1982. Information systems development - A systematic approach. Prentice Hall, Englewood Cliffs, New Jersey. (In Avison & Fitzgerald 1995).

Martin, P.Y. and Turner, B.A. 1986. Grounded theory and organisational research. The Journal of Applied Behavioural Science. 22(2) p141-157.

McGovern, Timothy, J. and Samuels, Helen, W. 1997. Our institutional memory at risk: collaborators to the rescue. CAUSE/EFFECT. Volume 20, Number 3, Fall 1997, p. 19-21, 49-50.

Merriam, S. B. 1998. Qualitative research and case study applications in education. San Francisco: Jossey – Bass Inc.

Morgan, D. L. 1988. Focus groups as qualitative research. Newbury Park, CA: Sage Publications.

Mumford, N. 1995. Effective requirements analysis and systems design: the ETHICS method. Macmillan. (In Avison & Fitzgerald 1995).

Mumford, E., Hirschheim, R.A., Fitzgerald, G. and Wood-Harper, A.T. (eds). 1985. Research methods in information systems. North-Holland Publishers.

Myers, M.D. and Avison, D.E. (eds). 2002. Qualitative research in information systems: A reader. Sage Publications.

Orlikowski, W.J. and Baroudi, J.J. 1991. Studying Information Technology in Organisations: Research Approaches and Assumptions. Information systems research. 2(1) p143-169.

Ould, Martyn, A. and Unwin, Charles. 1986. Testing in software development. CUP.

Phillip Crosby Associates II. www.phillipcrosby.com. 23 June 2004.

Pirsig, Robert M. 1984. Reissue edition. Zen and the art of motorcycle maintenance. Bantam Books.

Reynolds, P.D. 1971. A primer in theory construction. NY: MacMillan Publishing Company.

Royce, W. W. 1970. Managing the development of large software systems: concepts and techniques. Proceedings of IEEE WESCON, p. 1-9.

Shewart, W.A. 1931. Economic control of quality manufactured products. Van Nostrand.

Slife, B.D. & Williams, R.N. 1995. What's behind the research? Discovering hidden assumptions in the behavioral sciences. Thousand Oaks, CA: Sage Publications, Inc.

Standish Group, The. 1995. Chaos. <http://www.standishgroup.com/chaos.html>.

Strauss, A and Corbin, J. 1990. Basics of qualitative research: Grounded theory, procedures and techniques. Sage Publications.

SWITZERLAND. 1992. International Organisation for Standardisation. ISO 9000.

THE NEW INTERNATIONAL WEBSTERS DICTIONARY OF THE ENGLISH LANGUAGE. 1995. Trident Press International.

Weaver, Philip L. 1993. Practical SSADM4. Pitman Publishing.

Yeates, Don. 1991. Project management for information systems. Pitman Publishing.

Yeates, Don., Shields, Maura. and Helmy, David. 1994. Systems analysis and design. Pitman Publishing.

Yourdon, E. and Constantine, L.L. 1978. Structured design. Yourdon Press, New York.

14 Bibliography

Allen, D.P. 1997. Evaluating Hybrid Systems. ARMA Records Management Quarterly. 31(1) p10-14.

Avison. D.E., Fitzgerald G. 1995. Information systems development: methodologies, techniques and tools. 2nd Ed. McGraw Hill.

Avison, David and Fitzgerald, Guy. 2002. Information systems development: methodologies, techniques and tools (Information Systems Series) Higher Education.

Avison, David, Fitzgerald, Guy and Powell, Philip. 2003. Editorial. Information Systems Journal. 13 (4), p299-300.

Babbie, E. 1990. Survey research methods. 2nd Edition CA: Wadsworth Publishing Co.

Bemberger, J. 1997. Essence of the capability maturity model. IEEE Computer. June 1997. p. 112-114.

Bolland, R.J. Jr. 1979. Control, causality and information systems requirements. Accounting, Organizations and Society. 4(4) p259-272).

Cain, P. 2002. Model requirements for the management of electronic records (MoReq): a critical evaluation. Records Management Journal. 12(1) p14-18.

Card, D.N., McGarry, F.E., and Page, G.T. 1987. Evaluating software engineering technologies. IEEE Transactions on Software Engineering. 13 (4) p517-554.

Capability Maturity Model for Software. 2004. <http://www.sei.cmu.edu/cmm/>. 12 September 2004.

Cribbs, J., Roe, C. and Moon, S. 1992. An evaluation of object-oriented analysis and design methodologies. SIGS Books, New York.

- Denzin, N.K. and Lincoln Y.S. (eds). 1994. Handbook of qualitative research. Sage.
- European Commission. 2001. Model requirements for the management of electronic records – MoReq specification.
- Fanning, B. 2002. Records management standards:- A starting point. E-doc. 16(5) p60-61.
- Fichman, R.G. and Kemerer, C.F. 1991. Object-oriented and conventional analysis and development methodologies: comparison and critique. Center for Information Systems Research, Sloan School of Management, M.I.T., CISR WP. No. 230, 1991.
- Fresko, M. 2001. ERM...getting on with it. Conspectus. September 2001 p20-21.
- Fowler, Jr., F.J. 1993. Survey research methods. CA: Sage Publications.
- Gable, G. 1994. Integrating case study and survey research methods: An example in information systems. European Journal of Information Systems. 3(2) p112-126.
- Galal, G.H. and McDonnel, J.T. 1997. Knowledge based systems in context: Methodological approach to the qualitative issues. AI and Society. (11) p104-121.
- Glazier, J.D. 1992. Qualitative research for the information professional. Library Association.
- Gregory, K. 2002. Software essentials. Advisor today. 97(3) p88-89.
- Hughes, J. and D,H. 2000. Grounded theory: Never knowingly understood. Information Systems Review. (1) p181-197.
- Jayaratna, N. 1994. Understanding and evaluating methodologies - NIMSAD. McGraw Hill.

Kolarik, William, J. 1995. Creating quality: concepts, systems, strategies, and tools. McGraw-Hill.

Mason, J. 2002. Qualitative researching. Sage Publications.

Martin, J. 1990. Information engineering: Book I introduction. Prentice Hall, New Jersey.

Martin, J. 1990. Information engineering: Book II planning and analysis. Prentice Hall, New Jersey.

Martin, J. 1990. Information engineering: Book III design and construction. Prentice Hall, New Jersey.

National Archives, The. 2002. Functional requirements for electronic records management systems – Volume 1 statement of requirements.

National Archives, The. 2002. Functional requirements for electronic records management systems – Volume 2 references.

Patton, M.Q. 1990. Qualitative evaluation and research methods. 2nd. Ed. Sage Publications.

Redmill, Felix. 2004. It's only a model. Professional Tester. November 2004. pp16-17

Ricks, B., Swaford, A. and Gow, K. 1992. Information and image management: a record systems approach. 3rd Edition South western Publishing.

Salant, P., & Dillman, D.A. 1994. How to conduct your own survey. NY: John Wiley and Sons.

Slater, M. (ed). 1990. Research methods in library and information studies. Library Association.

Silverman, D. 1993. Interpreting qualitative data. Sage Publications.

Smit, J. 1999. Grounded theory methodology in IS research: Glaser versus Strauss. South African Computer Journal. November p309-340.

Strauss, Anselm, L. 1987. Qualitative analysis for social scientists. Cambridge University Press, New York.

Swanson, K., McComb, D. Smith, J. and McCubbrey, D. 1991. The application software factory: Applying total quality techniques to systems development. MIS Quarterly. 15(4) p567-579.

Tudor, D.J. & Tudor, I. 1996. Systems analysis and design - A comparison of structured methods. NCC Blackwell.

United States of America Department of Defense. 2002. DOD5015.2STD – Design criteria standard for electronic records management software applications.

Urquart, C. 1996. Strategies for conversation and systems analysis in requirements gathering: A qualitative view of analyst-client communication. The Qualitative Report. (4:1-2).

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

Young, Ralph R. 2001. Effective requirements practices. Addison Wesley.

15 Appendix A. Pre-Interview Briefing Notes

Dear all,

For my dissertation, the final part of my MSc in Records Management, I have chosen to develop a new model for the design and implementation of records management systems. To do this I have critically analysed the existing models by comparing and contrasting them to models in the field of information systems development. This allowed me to identify gaps in the existing models and to develop a new model. I now need people to review my 'prototype model' for me.

My question is, therefore, would you act as a reviewer for me and complete the attached questionnaire and would you be prepared to extend this to an interview to further develop any suggestions you might have?

Please feel free to pass this to colleagues, but could you please keep me informed of who it has gone to as I am obviously keen to release this only when I believe it is complete.

I will need your responses by 16 November, which is 3 working weeks, and interviews in the last 2 weeks of November for those willing to take part.

Thanks in advance for your help.

16 Appendix B. Pre Focus Group Questionnaire

Are you familiar with the development model in ISO 15 489 or DIRKS?

Are you familiar with any information systems development models?

Which ones?

Do you believe the DIRKS and ISO 15 489 model covers the necessary activities for the development of a system?

What are the strengths of the DIRKS and ISO 15 489 Model?

What are the weaknesses of the DIRKS and ISO 15 489 Model?

In terms of the prototype model :

Does it capture the strengths of the DIRKS and ISO 15 489 model?

Does it address the weaknesses of the DIRKS and ISO 15 489 model?

Does it introduce new strengths?

Does it introduce new weaknesses

Overall, is it an improvement ?

What other improvements could be made?

Other comments

17 Appendix C. Reviewers

- David Bowen, Audata Ltd, who has 30 years experience with computers and 15 with records management.
- David Ryan, formerly of The National Archives, who was Head of Digital Preservation there and is now Director of Records for the Royal Household.
- Clive Whitton, Pfizer Ltd, responsible for electronic document management.
- David Cox, Audata Ltd, who has 35 years experience with computers and systems development.
- Liz Smith, Diagonal Solutions, who has 20 years experience of implementing both bespoke and commercial off the shelf records management systems.
- Jon Moss, Audata Ltd. Jon initially trained as a bio-chemist before completing an MSc in Information Systems. Jon is employed as an ERMS Consultant.
- Adrian Brown, the National Archives. Adrian has many years experience with both information systems and records management.

18 Appendix D. Schedule

Activity	Start	Finish	Duration
Complete Dissertation Proposal	24 May 2004	15 June 2004	3 weeks
Identify suitable people to review prototype model	24 May 2004	15 June 2004	3 weeks
Identify suitable people to review dissertation	24 May 2004	15 June 2004	3 weeks
Undertake literature review	14 June 2004	25 June 2004	2 weeks
Complete background, including literature review write up	28 June 2004	09 July 2004	2 weeks
Define methodology	12 July 2004	16 July 2004	1 week
Justify methodology	19 July 2004	30 July 2004	1 week
Holiday (Yippee)	02 Aug 2004	13 Aug 2004	2 weeks
Conduct critical evaluation of RM models	16 Aug 2004	27 Aug 2004	2 weeks
Conduct gap analysis	31 Aug 2004	10 Sept 2004	2 weeks

Construct prototype	13 Sept 2004	17 Sept 2004	1 week
Design guidance notes for reviewers	20 Sept 2004	21 Sept 2004	2 days
Send prototype and guidance notes for review	22 Sept 2004	22 Sept 2004	1 day
Start write up of recommendations	23 Sept 2004	01 Oct 2004	1 week
Start write up of reflections	23 Sept 2004	01 Oct 2004	1 week
Conduct focus group	04 Oct 2004	04 Oct 2004	1 day
Write up transcripts of focus group	05 Oct 2004	08 Oct 2004	4 days
Analyse data produced by focus group	11 Oct 2004	05 Nov 2004	4 weeks
Write up conclusions from data analysis	08 Nov 2004	19 Nov 2004	2 weeks
Complete write up recommendations	22 Nov 2004	26 Nov 2004	1 week
Complete write up reflections	22 Nov 2004	26 Nov 2004	1 week
Issue draft dissertation for review	29 Nov 2004	10 Dec 2004	2 weeks
Amend dissertation to reflect comments of reviewers	13 Dec 2004	17 Dec 2004	1 week