Table of Contents

Go Back

Challenges of System Dynamics to Deliver Requirements Engineering Projects: faster, better and cheaper

Ddembe Williams
Faculty of Engineering, Science and Technology
London South Bank University
103 Borough Road
London SE1 OAA, UK

Tel: +44 (020) 7815 7460 Fax: +44 (020) 7815 7499 e-mail: d.williams@sbu.ac.uk

Abstract

The success of Requirements Engineering (RE) projects for complex software systems critically depends upon the effectiveness of RE process improvement. This paper presents a model for improving the delivery of requirements engineering projects. The paper applies system thinking/ system dynamics (SD) to the complexity and dynamics of the RE process. The review of the state of the art and practice in the RE literature indicate six categories of problems that have motivated the work reported in this paper. Poorly defined requirements process cause projects to fall behind schedule, go over budget and result in poor quality of system specification. The paper seeks to understand these problems from a feedback control viewpoint since there is lack of agreement on the nature of deficiencies in current RE processes. This is a cornerstone of the paper: the system dynamics model developed should therefore not be viewed as an answer to the above problems but as a tool to help researchers and RE process stakeholders develop the understanding required to answer them. In developing such a model the paper fills an important gap in both RE ands SD modelling literature. The paper suggests that the model makes a useful contribution both in providing the foundation for theory building on RE projects and in improving the management of RE projects in learning and training situations.

Key Words: Feedbacks, Requirements Engineering Projects, System Dynamics

1.0 Introduction

Improving the requirements engineering (RE) process is a key issue for the delivery of effective, better quality software systems that meet user expectations, are delivered on time, and developed within budget. Existing process models have ignored important issues of enacting the RE process, as well as not specifically addressing the effects of RE process improvements on the cost of resources, quality and schedule (Finkelstein, 1994; Finkelstein and Kramer, 2000; Abdel-Hamid, 1996). The existing approaches to process modelling have not applied feedback concepts to large-scale and complex RE processes (McChesney, 1995; Paulk et al, 1993; Kellner et al, 1999). These feedback structures are responsible for requirements volatility or behavioural patterns that Curtis et al (1988) refer to as "ever fluctuating requirements". Despite their influence on system development phases, current RE processes neither address RE process issues, nor do they describe or explicitly model the underlying activities incorporating the requirements change dynamics (Williams, Hall and Kennedy, 1999; Williams and Kennedy, 1999a/b; Davis et al, 1993; Zahedi, 1995).

Although substantial progress today has been made in terms of methods, techniques and tools used within the RE phase of systems development, little attention has been paid to the understanding of the RE process itself. System designers and programmers often begin designing and programming the desired systems too early, before they actually understand the users' or customers' requirements. Since designing and programming systems is very expensive, ill-defined requirements (Bubenko et al, 1994) that cause projects to fall behind schedule (Abdel-Hamid and Madnick, 1991; Macaulay, 1996; Abdel-Hamid, 1996) and to be over budget (Standish Group, 1986) are a significant problem. For systems software development to be successful, there has to be a balance between the technical worldview of designers and programmers, and the social worldview of users and other stakeholders. This paper examines these problems from a feedback control point of view in order to facilitate understanding from a modelling and analysis perspective. The cornerstone of this paper is to apply SD modelling effort to provide a fulcrum for debate and enhanced understanding of the RE process problems.

1.1 Definition of Key terms

A variety of theoretical dimensions and terms have been used to define, quantify and operationalise the RE process research. Although there is still lack of consensus about the precise meaning and the exact boundaries of these theoretical terms used in the RE field, to put this paper into context, key terms are clarified and their relationships to the RE process modelling and analysis when put together in this paper namely: "*requirements engineering*", "*process*", "*model*" and "*RE process system dynamics model"*.

Requirements Engineering is used to describe a systematic process of developing requirements through an iterative co-operative process of analysing the problem, documenting the resulting observations in a variety of representation formats and checking the quality and accuracy of the understanding gained (Pohl, 1993; Macaulay, 1996). The success of RE projects for large-scale complex software systems depends critically on the effectiveness of the RE processes. RE process modelling is becoming an important field of research in requirements engineering, a subset of systems (software) engineering (Curtis et al, 1992; Finkelstein, 1994; Finkelstein and Kramer, 2000). The transformation of systems concerns into "what" the system needs in order to achieve its operational goals is stated in a *requirements document,* a product of the RE process (Melchisedech, 2000; Kamsties et al, 2000). Current RE process modelling and analysis tools have focused on improved documentation rather than on productivity improvements of engineers and the quality of the resulting specification. While some tools do not provide adequate support for developing an understanding of the problem, others even do not have any specific methodology to carry out the process of determining information requirements (Adelman, 1992). Little attention has been paid to understanding of the RE process itself in order to improve quality of the system delivered.

Process modelling refers to logically capturing and abstracting the system's components, relationships and behaviour, with respect to modelling objectives (McChesney, 1995). Requirements engineering process modelling, as part of software process modelling, involves a description and enactment of the model entities and their relationships. The entities of a process are: activities, products, resources and roles performed by agents. In terms of enacting the RE process, this is a challenging focus for process modelling because of the creative problem solving involved (Paulk et al, 1993; Kellner et al, 1999). In the RE process, co-ordination of stakeholders' interactions during this process is a complex intellectual artefact. The relationships between variables determine the process performance and level of requirements changes over time.

A model is an abstraction of a representation of a real or conceptual complex system. A model is designed to display significant features and characteristics of the system which one wishes to study, predict, modify or control (Law and Kelton, 1991). Thus a model includes some, but not all, aspects of the system being modelled. Process models can be descriptive, prescriptive, iconic or symbolic (Finkelstein et al, 1994). A model is valuable to the extent that it provides useful insights, predictions and answers to the questions like What if ? analysis it is used to address. One of the main motivations for developing a simulation model other than any other method is that it is an inexpensive way to gain important insights when the costs, risks or logistics of manipulating the real system of interest are prohibitive (Law and Kelton, 1991). Simulations are generally employed when the complexity of the system being modelled is beyond what static models or other techniques can usefully represent (Kellner et al, 1995). Complexity in the RE process is often encountered in real systems and can take any of the following three forms: uncertainty, dynamic behaviour and feedback mechanisms. System Dynamics simulation is being applied to an increasing variety of problems because of its ability to model and evaluate complex and dynamic systems that could not be otherwise modelled, particularly those inherent with feedback structures (Forrester, 1961; Senge and Forrester, 1980; Coyle, 1996; Sterman, 2000). Advances in hardware and software technology have greatly reduced the two major weaknesses of System Dynamics simulation, cost and the unavailability of the data necessary for building and validating the model.

A Requirements Engineering Process System Dynamics Model is an abstract representation of the RE process organisation, its characteristics, inheritance, feedback structure, delays, and interrelationships of the RE activities and key variables using system dynamics modelling approach. The RE Process Model is structured into a series of activities or phases that take inputs and transform these into meaningful output measures. Although a lot of research on complex process models in recent literature has focused on the whole software development life cycle, this paper focuses on RE process as an initial phase of the systems development life cycle. To enable process modelling and analysis, RE metrics proposed by Costello and Liu (1995) and Davis et al (1993) have been integrated in section Five to facilitate process measurement and performance prediction of such variables like cost, schedule, quality and customer satisfaction. Although there are already other studies in the area of software process simulation modelling (Abdel-Hamid and Madnick, 1991; Akhavi and Wilson, 1993; Burke, 1997; Gruhn, 1992; Gruhn, 1993; Kellner, 1991; Kusumoto, 1997; Madachy, 1994; Raffo, 1996; Raffo and Kellner, 1999; Tvedt, 1996), this is the first time System Dynamics simulation modelling has been applied to RE process investigation (Kellner, 1999; Williams, Hall and Kennedy, 2000).

The remaining part of this paper is divided in five sections. Section Two, reviews current literature on both the state of the art and practice in requirements engineering, and the challenges facing requirements engineering stakeholders. Section Three discusses the usefulness of system dynamics to requirements engineering, while Section Four discusses the challenges of SD to deliver requirements engineering projects. Section Five presents potential propositions drawn from the models and suggests how they may be tested. Section Six presents initial findings drawn from the research and suggested further work.

2.0 The requirements engineering process environment

The RE process is arguably one of the most important processes in software and systems engineering (Finkelstein, 1994; Curtis et al, 1988). The importance of requirements engineering has also been highlighted by Chatzoglou (1997) as one of the first stages of the systems development process, and the most important, because it is critical for the success of the whole development process stage. The RE process has only recently started being treated as important by the practitioners and the academic community. The aim of this paper is to examine and evaluate current RE processes from a feedback control point of view. The paper aims to apply the System Dynamics (SD) modelling approach in examining the various acknowledged weaknesses in current RE process practices (Curtis, et al; 1988; Finkelstein and Kramer, 2000).

The environment, in which requirements engineering process takes place, is arguably dynamic, uncertainty and is also characterised by feedback loops among its activities (Kellner, 1999). For example, key variables such as engineer productivity and error detection rates in the RE process change over time. Curtis et al (1988) for example found that “requirements were ever fluctuating”. Another example, in RE process the decision to hire or not to hire a new requirements engineer has multiple impacts and implications over the entire course of the project. The RE process is an iterative process in nature and some of the activities may have to be conducted several times due to the presence of feedback loops. The decision to adjust the mean time to review requirements or not also has many implications.

2.1 Characteristics of requirements engineering projects

The transformation of systems concerns into "what" the system needs in order to achieve its operational goals are stated in a *requirements document,* a product of the RE process (Pohl, 1993). Current RE tools have focused on improved documentation rather than on the quality of the resulting specification. While some tools do not provide adequate support for developing an understanding of the problem, others do not have any specific methodology to carry out the process of determining information requirements (Yadav et al, 1988). Analysis of information in the requirements documents by Melchisedech (2000) and Kamsties et al (2000) provide an insight into the current state of requirements engineering practices. Tables 1 and 2 provide a classification of the information found in the requirements documents. Melchisedech (2000) used case study research to analyse information contained in three requirements engineering documents for three projects. The goal of the investigation was to gain insights into the state of the practice. Table 1 summarises the main RE project characteristics in terms of resources, schedule and roles involved.

Table 1: Characteristics of Software Development Process [Adapted from Melchisedech (2000)]

Project Characteristics	**Project A (Development System)**	**Project B (GUI Front End)**	**Project C (On-line System)**
Duration	26 months	29 months	Not yet finished
Effort	48 man-months	155 man-months	Not yet finished
No of Project Team	2-7	8-13	2-3
Duration of RE phase	**5 months**	**4 months**	**12 months**
Effort of RE phase	**13 man-months**	**2.5 man-months**	**8 man-months**
No of Systems analysts (engineers)	**2-3**	**1**	**1**
Financing	**Fixed price project**	**In-house project**	**In house project**
Prototyping	GUI-Prototype	Pre-version	Prototype
Maintenance	Customer	Developer	Developer
Validation Role	Customer	User	User

Table 1 presents an analysis of RE schedule, cost and effort within the overall software development process, which is consistenct with the findings of Boehm (1981) and Zahedi (1995). Requirements engineering is an important phase of the systems development life cycle. The granularity of analysis was the requirements document and the documented development process undertaken is based on fine-grain models. Table 2 provides a classification of information found in the requirements documents based on field studies reported in Kamsties et al (2000).

Table 2: Classification of Requirements Document [Adapted from Kamsties et al (2000)]

Category	**Project A (Development System)**		**Project B (GUI Front End)**		**Project C (On-line System)**	
Definition of Terms, Abbreviation, etc	Lines	%	Lines	%	Lines	%
	178	2.4	29	0.7	20	1.1
Problems, Goals	134	1.8	116	2.6	116	6.1
Current state	-	-	41	0.9	194	10.2
Solution ideas, alternatives	-	-	-	-	65	3.4
System Environment	46	0.6	20	0.5	50	2.6
Interfaces	3540	48.6	677	15.3	334	17.6
Internal structure	1077	14.8	-	-	-	-
Internal State/ data	213	2.9	1126	25.4	455	24.0
Functional Requirements	**1306**	**17.9**	**1861**	**42.1**	**510**	**26.9**
Non-Functional requirements	**77**	**1.1**	**67**	**1.5**	**35**	**1.8**
Design Constraints	101	1.4	43	1.0	-	-
Requirements on Future Versions	18	0.2	47	1.1	-	-
Project Planning, cost Benefits	134	1.8	133	3.0	106	5.6
Meta Information	467	6.4	265	6.0	8	0.4

Table 2 provides a detailed field studies analysis based on workshops on requirements engineering held with practitioners from 10 Small and Medium Enterprises (Kamsties et al, 2000). As illustrated in Tables 1 and 2, requirements engineering process generate different types of requirements. For example "Functional" and "Non-functional" requirements presented in Table 2. Functional requirements define the systems functionality that a supplier and any other third party must build into the system to enable to accomplish their tasks, there by satisfying the business requirements. Non-functional requirements, on the other hand identify a measurable criteria or a quality attribute of a function or how well a functional requirement must be accomplished. This level of analysis provides a basis for dynamic analysis of RE process metrics, which can aid requirements engineering stakeholders understand their processes better.

2.2 Challenges facing requirements engineering stakeholders

Earlier studies have shown (Weiringa, 1996) various roles in a RE process including requirements engineers, requirements manager and developers from the software development organisation (SDO) together with customer and user roles in the sponsoring organisation. This process ensures completeness and correctness of the RE process dynamic model structure being proposed. At the lower level, the product engineering and project management subsystems are parallel and interactive activities. While the product engineering subsystem produces the requirement specification as a by-product, the project management subsystem ensures that the product is delivered with the right quality, within scheduled time and cost. These processes may exist concurrently or parallel with other project processes. An SDO may have more than one projects at the time of undertaking a RE process project. A RE process paradoxically demands greater understanding of the domain knowledge therefore the experience of requirements engineers and training in the use of tools aids greater understanding, leads to fewer errors and improved quality in delivered specification and communication.

Improvements in technological development have facilitated the automation of the RE tools; however, this automation has focused on documentation of requirements and specification (Yadav, 1988; 1983, Finckelstein, Kramer and Nuzeibeh, 1994; Finkelstein and Kramer, 2000) rather than the whole process (Williams, Hall and Kennedy, 1999; Williams and Kennedy, 1999a/c). This shortcoming has meant that requirements stakeholders do not have a whole picture of the process, its cost, schedule and quality and therefore understanding is not complete for effective management and process improvement. Gillies and Smith (1994) argue that, while increased tool automation should decrease the significance of the human role, the reverse appears to be true. This argument is true in the RE process. Increased automation in the whole RE process requires more careful management, commitment from senior management and increased motivation and training for requirements engineers. Table 3 summarises the different aspirations and interests to different stakeholders in a RE process.

Table 3: Stakeholders in a RE Process [Adapted from Gilles and Smith, 1994]

Stakeholder	Aspirations and Interests
Requirements Engineer	Wants a tool that makes their job easier, more satisfying and more productive.
Customer/User	Wants a system specification with minimum errors that will describe the system they want with lowest price and in the shortest time. Wants usable system, with fewer errors
Project/ Process Manager	Wants to deliver on time with the right specification quality and to satisfy the customer.
Quality Manager	Wants to ensure that the delivered system specification is error-free and meets the aspirations of the customer.
Senior Management	Wants to see a return on investment, increased productivity, increase in quality of products and services and fears the likely failure of the project!

The fears and aspirations illustrated in Table 3 shape the success or failure of the RE process. For this paper, the main aim of the field studies was to study requirements engineering projects in progress and in their natural settings. Samples for field studies research are not as large as for survey research methods (Robson, 1993 pp. 144), therefore RE projects fall under what researchers have termed non-probability samples (Robson, 1993). RE projects are usually very few and there was no intention to make statistical generalisation to any population beyond the sample surveyed. However, external validity of the results can be improved by simulation modelling proposed as further work in this paper (Law and Kelton, 1991). Field studies involved some observation, focused interviews and document analysis.

2.3 Requirements volatility

Curtis et al, (1988), who studied a number of software engineering practices, report that, in many application areas, accurate problem domain knowledge is essential to the success of a project and requirements volatility causes major difficulties during development. A review of the state of practice in requirements engineering by Lubars et al (1993) reveal that, although most informants were able to describe the nature of requirements specifications that they produced, they were unable to describe the process by which they arrived at these specifications. Loucopoulos and Karakostas (1995) postulate that the uncertainty inherent in trying to discover and document requirements is also problematic. The importance of tracking requirements changes (volatility) is recognised by many organisations (Gotel and Finkelstein, 1996). A requirement specification is likely to change many times before proceeding to the design phase, and therefore needs to be subjected to evaluation in order to inspire confidence as to its validity. Loucopoulos and Karakostas (1995) report "this cyclic approach of acquisition, representation and evaluation involves a succession of propositions which are increasingly closer to end users' perceptions about the target system" (pp. 15).

Managing the project's requirements helps ensure that the effort invested in gathering, documenting, and analysing the requirements is not wasted through neglect over time. Tracking the status of each requirement through the specification is an important aspect of communication, replicability and traceability of data items (Caputo, 1998). Williams and Kennedy (1999a) demonstrated how project monitoring is improved if you can periodically report the percentage of your requirements set that exist in each possible status category. The RE process system dynamics model can facilitate the tracking of the changes in each status category. Figure 1 illustrates requirements change categories that may be monitored. Measuring the variability of attributes performance and tracking the results over time determine stability as a control feedback problem of a process with respect to a given variable and its attributes. Statistical process can be used to identify the cases of variability and correct the process performance to achieve a stable and predictable state.
Very few frameworks provide a way of tracking the status of requirements throughout the duration of the schedule of the RE process. As illustrated in Figure 1, Wiegers (1999) suggest that a requirement in a given time of the project life cycle can be proposed, approved, implemented, traced completely or even be deleted. With good requirements engineering tools the number of requirements can be analysed as their respective statuses vary at the end of each month as illustrated in Figure 1.

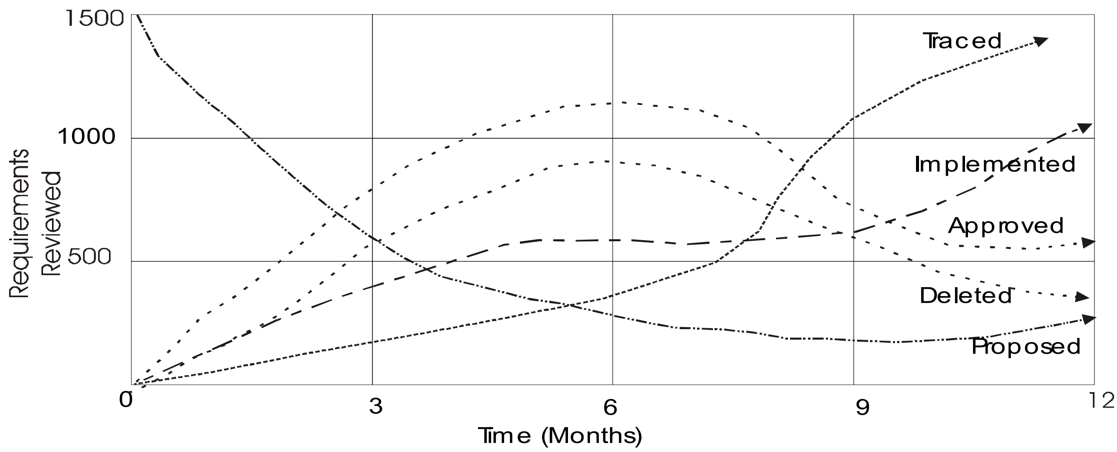

Figure 1: Distribution of the Requirements Status throughout RE Process Review.

Figure 1, which illustrates tracking the distribution by number of requirements, does not show whether the number of requirements in the baseline is changing over time, but it does illustrate how goals for completing verification of all approved requirements can be attained. Several other metrics can be tracked and analysed as illustrated in Figure 1. Software metrics may be used to measure attributes of software processes or a final product of the requirements engineering process – a specification presented in a requirements document as discussed in section 2.1.

2.4 Causes of requirements engineering project failures

An extensive literature review undertaken in this paper indicates that the success of the RE is an extremely important issue. This statement is collaborated by Finkelstein (1994); Curtis et al, (1992); and Chatzoglou and Macaulay, (1996). Requirements engineering literature, as analysed, confirms that current RE process models have inherent problems. Various authorities have indeed identified problems with the current RE process (Wieringa, 1995; Macaulay, 1996; Loucopoulos and Karakostas, 1995; Hofman and Lehner, 2001). While most observers will acknowledge that there are deficiencies in current RE process practice, there is no consensus on what they are. Understanding of the RE process and the facts that lead to its effectiveness is the prerequisite for improving the RE process practice (El Emam and Madhavji, 1995a; Newman and Robey, 1992; Chatzoglou, 1997). This lack of consensus is mainly due to poor understanding of domain knowledge and poor use of methods, techniques and tools. Macaulay (1996) reports inconsistency in positioning of requirements engineering process within the various software development life cycle models. These problems must be understood, in order to develop a systematic and sound model for effective RE process modelling and analysis. Six categories of problems in RE process modelling and analysis have motivated the work reported in this paper.

These problems are partly due to lack of well-defined process models, innovative process modelling approaches and historical RE project data which are consequences of intangibility, uncertainty and lack of precision in managing the RE projects. These problems can be summarised in order of their importance as:

1. Requirements engineering process stakeholders and researchers in the RE field have not used the feedback control viewpoint in exploring these problems;
2. Requirements fluctuations due to frequent change of requirements in process improvement over time cause instability and uncertainty in the overall process performance effectiveness;
3. Software project managers, including requirements managers, lack quantitative data regarding project status, progress and quality. There is a need to articulate data collection methods, in order to collect objective data that can be used as a basis for estimation and planning. The uncertainty that exists in requirements engineering process management makes project planning and control process difficult;
4. There is a lack of precision in estimating RE project size. There is need to scope the RE process effectively. Over or under estimation of the scope of the RE process can lead to serious quality and productivity problems;
5. Some factors of major consequence to RE process are intangible (soft variables); these are, more difficult to measure, and therefore usually ignored in modelling the RE process;
6. There is a lack of communication between requirements engineering process stakeholders.

These problems and the underlying causes deserve to be investigated, as they present fundamental difficulties currently experienced by requirements stakeholders and researchers (Finkelstein, 1994; Curtis et al, 1992). There is an inherent lack of understanding and recognising what the real problems are in carrying out software development. These problems have been exacerbated by the lack of process measurement to support stakeholders to respond to changes in RE projects, central to the management of the RE process. Indications are that this lack of know—how and the need for engineer productivity improvements are among the many yet unanswered questions about the RE process effectiveness. These problems have arisen because of misunderstandings of the requirements engineering process and also because requirements engineering is the first phase of the systems development process. Understanding the requirements engineering process has become an important issue for practitioners and researchers (Chatzoglou, 1997; Williams, 2000; Williams, Hall and Kennedy, 2000).

1.3.2 Sources of Misunderstanding of the RE Process

Five sources of misunderstandings of the requirements engineering process have been identified:

1. The technical focus of the requirements engineers and non-technical focus of requirements customers and users discourage shared understanding of the potential benefits of the requirements engineering solutions being developed (Williams, 2000; Williams, Hall and Kennedy, 2000);

2. The managerial tools and techniques for RE process assessment and process improvement focus on wrong viewpoints (Chatzoglou and Macaulay, 1996; Christie, 1994);
3. RE process factors affecting project performance effectiveness have been ignored and there is rarely empirical valid evidence of objective data and valid relationships between and among data (Henry and Kafura, 1995);
4. Field studies reported on the RE process have failed to establish a clear link to the whole RE process performance and tend to focus on a narrow set of variables (Hofman and Lehner, 2001);
5. There is lack of a high level (coarse grain) process modelling methodology instead of the current low level (fine grain), that focuses on requirements engineering with committed resources for model research, development and continuous improvement (Williford and Chang, 1999; Williams, Hall and Kennedy, 2000).

Although many process modelling technologies are currently mature enough to provide the basis for effective process definition and to facilitate process improvement efforts, no standard approach exists for RE process, simulation or enactment (compared for example, to the Software Engineering Institute process assessment procedure, which has come the closest to being a fully developed technology and an accepted industry standard for software processes). RE process enactment, the use of the process definition to manage and/or control RE processes, is largely at the research stage and support for a true process-driven environment, except that proposed in this paper, is not yet a reality.

3.0 Challenges of system dynamics to deliver RE projects

In order to improve our understanding of the RE process and to facilitate debate in the RE field, there is a need to use alternative process modelling approaches that provide insights into the complex process behaviour commonly found in requirements engineering. Christie (1994) suggests that, like any other process, the software processes can contain multiple feedback loops, associated with correction of errors. There is also a time delay for key activities, which may range from minutes to years. The complexity resulting from such effects and their interactions make it almost impossible for human (mental) analysis to predict consequences. Traditional process modelling approaches do not shed much light on these behavioural issues and the most common method of resolving them is to run the actual process enactment and observe the consequences. This can be a risky and costly way (Law and Kelton, 1991) to perform RE process improvement. To avoid this risk of playing with real systems and the associated costs, simulation modelling can be applied to support RE project planning and control, costing, tracking and prediction (Abdel-Hamid and Madnick, 1991; Raffo, 1995). System Dynamics-based simulation can thus provide a powerful tool to aid activity-based modelling and can provide insights into problems of the RE process, while also providing significant advantages over more traditional costing techniques (Boehm, 1981).

Some Software Development Organisations (SDO) may be more interested in the economic benefits of increased engineer process productivity or reduced error rework. There is evidence where SDOs have been able to show improvements in terms of ROI of reduced error rework rates and increased productivity. The benefits of an System Dynamics model proposed in this paper can also be calculated in terms of cost avoidance such as error rework, increased engineer capability, reduced expected delivery time or improved specification quality. These benefits confirm the usefulness of SD in RE process modelling and analysis, as follows:

- The requirements engineering process is dynamic i.e. it develops over a period of time, this period varies between 3 and 24 months for small and large complex projects respectively (Williams, 2001/b);
- There is existence of feedbacks in the RE process i.e. For example the more time customers spend reviewing documents; the more errors are likely to be found. Equally, the

fewer reviewers read the requirements documents per months, the fewer errors they will find (Williams, 2001/a);

- The RE process is continuous i.e. the requirements flow through different activities within the RE process as demonstrated in Figure 1 (Williams, Hall and Kennedy, 2000);
- The requirements engineering process involves a number of stakeholders with different viewpoints as illustrated in Table 3 (Williams and Kennedy, 1997).

The challenges of SD for developing and maintaining RE process modelling and analysis lies in the following:

- System dynamics RE process models permit the modelling of complex processes at various levels of granularity; this enables the concentration on the required level of interest without being overwhelmed by lower level details. Field observation shows activity level stakeholders at a monthly reporting frequency easily explain details.
- A RE model's notation "pseuedocode" underlying the SD elements provides an effective way of precisely describes a process with greater fidelity than one could command using English language descriptions (e.g., in a procedures manual).
- The RE process model developed can be used by stakeholders in training situations as, one can simulate the process to gain a better understanding of it;
- The use of systems thinking and application of SD modelling notation in the requirements engineering process provides the requirements engineer with a balanced perspective between "hard" (Davis and Vick, 1977; Boehm, 1981) and "Soft" (Checkland and Scholes, 1990, Roberts, 1964; Williams and Kennedy, 1997) systems problem solving paradigms (Galliers, 1984).
- RE process simulation model notations can be used to support process enactment. The resulting prototype from the SD can be converted into other programming environments for software development (Finkelstein et al, 1994).

There is an overwhelming need for an interactive and dynamic RE process which can support modelling in what if…? dynamic analysis mode. The requirements engineering process, as a more highly social process, needs approaches or problem-solving paradigms that can capture both the quantitative and qualitative issues commonly found in complex systems (Williams and Kennedy, 1997). These tools and techniques can be used in the RE process modelling to improve its effectiveness. Methodological pluralism or paradigms that support both hard and soft should be welcomed, regardless of inherent differences in epistemological and ontological issues (Galliers, 1984), particularly when researching systems requirements engineering process performance.

4.0 Dynamic hypothesis of the RE process success

The RE problems identified in 2.4, suggest several variables are important in a RE process. The process modelling and analysis approach used in this paper involves the identification of activities, their entities, variables, their measures and interactions that are present in the RE process. Entities, variables and measures must be represented as constituent elements of the RE process model to facilitate model analysis. SD models are rational structures that generate a formal representation of behaviour, which should fit the empirical behaviour of the system being studied. Finkelstein (1994) suggests that an analysis of software areas of interest can be viewed from a process or product perspective. Many researchers have proposed tools to support the production of computer-based products. However, little attention has been paid to RE process modelling. The focus of these tools in software systems research has been the technical specification advancement, in the case of computers, as individual tools (Finkelstein Kramer, 2000). Capturing the RE process data has often been one of the most difficult, due to informality involved in the process. Process simulation provides insights and understanding on how likely the process will perform in terms of quality, cost and schedule as patterns of interest in this paper. The utility of a SD model in providing explanatory and prescriptive insights about the likely RE process performance can be validated using case studies as empirical evidence to provide a basis for improving the RE practices.

The development of requirements engineering process models that integrate organisation wide processes is seen as an important area of research (Finkelstein, Kramer and Nuseibeh, 1994). This is because, for a RE process to be cost effective to the customer, the software development organisation must use its assets (human resources, organisational knowledge, financial and material) to add value to its RE process inputs. This process should produce outputs that satisfy customer expectations and also increase the value of the organisation by making profit. The effectiveness of the RE process must be related to the quality of the product, the cost of resources and the duration the product developed. Requirements Engineering Process Effectiveness is used as the measure of the accuracy and completeness of goals achieved in RE process. Customer is the owner of the RE project usually with limited resources, facing a number of investment choices and attempting to achieve a satisfactory return on investment.

Software Development Organisation (SDO) is the contracted organisation for the RE projects with limited financial and technological resources and know-how, which seeks to profit from its work outputs - the product specification. Various approaches have been proposed for evaluating the success of RE process (El Emam and Madhavji, 1995a/b; Newman and Robey, 1992). Newman and Robey (1992) found the process modelling approach very appealing, particularly its applicability with respect to complex dynamic RE process. User/customer satisfaction has also been widely used as a measure of RE process success (El Emam and Madhavji, 1995a/b). The perceived utility seeks the opinion of customer/user during a requirements review meeting about both the requirements and the whole process. Raffo and Kellner (1999) contend, "requirements pose a major problem to software development as a consequence of measurement". Jones (1992) contends that 50% or more of all software defects are requirement errors. Hence, a software development organization that does not address the quality of the requirements is more prone to produce poor-quality products. Jones (1994) identifies several other legitimate areas for measurement, like cost of the project, whether the schedule commitments are being met, customer satisfaction, etc. It is clear from this discussion that the key concerns is the desire of RE process stakeholders to reduce cost of resource, minimise schedule and improve the RE process effectiveness. The key question is, how can requirements engineer productivity be increased to improve the quality of product specification delivered to the customer?

4.1 Problem Statement

The concerns, considered as part of the dynamic hypothesis, lead this paper to suggest an important problem statement about the RE process dynamics:

The RE process effectiveness changes over time as a result of variations among product specification quality, customer satisfaction and cost of resources. Based on the dynamic hypothesis presented in Figure 2, one can deduce that requirements engineering customers seem to have mixed satisfaction levels about the SDO's RE product specification quality, an end product of the RE process. During some periods, customers' reviews discover many errors in the requirements document, while at times customers discover a few errors. As customer's document review duration increases, more errors are discovered. These errors are documented as requirements changes. After some delay, the accumulation of requirements changes triggers productivity process improvements necessary for reworking errors. Sometimes, additional requirements engineers may be acquired; this increase may need engineers to be trained on new tools and methods. However, the training initially reduces the productivity of engineers. When the productivity per requirements engineer decreases, the observed quality decreases. This increased training effort further reduces specification quality. However, after the training delay, the productivity per requirements engineer increases and observed quality increases. This increase in productivity process improvements initially increases the cost of resources and inevitably reduces customer satisfaction, but later improves the specification quality, thus improving customer satisfaction. This problem statement may be used as a basis for defining relevant variables.

4.2 Defining Key Variables

The above problem statement implies that a system dynamics RE process model should show a tendency towards fluctuations in the specification quality of the product, the cost of resources, and customer satisfaction as a result of process improvements. The problem statement identifies these important variables that represent an example of quantitative process management, as suggested in simulations for support of the RE process improvement. The simulation model can be used for control process performance and making adjustments within acceptable limits. The initial key variables identified in the literature and confirmed by field observation are important in explaining the variability in the RE process effectiveness. These key variables are as follows {units of measure or dimensions are given in curly brackets}:

Process effectiveness is related to the quality of the product, the cost of resources and the duration of the product specification. This is desired as a mean effectiveness of key indicators like cost effectiveness, quality, customer satisfaction and schedule pressure {0-1 Scale, Fraction};

Requirements Changes: these are number of errors in pages discovered by customers per month in the review process and requested for change {pages/month};

Documents Reviewed: this represents the requirements document pages reviewed by customers per month {pages/month};

Process Improvements: are the productivity improvements in terms of pages per person required per month as a result of requirements changes generated {pages/person/month};

Specification Quality: is a fractional level of correctness and completeness of the product specification quality delivered in relation to the quality objective set by the SDO and Customer {0-1 Scale, Fraction};

Customer Satisfaction: is a customer's fractional level of content with the quality of services provided as incorporated in the product specification by the Software Development Organisation based on perceived specification product value and the cost of resources passed on to the Customer {0-1 Scale, Fraction};

Schedule Pressure: relates to a fractional variation in the average RE process Scheduled completion time and the Expected Delivery Time of the product specification {0-1 Scale, Fraction};

Cost of Resources: relates to the total cost of resources in terms of wages, documentation, training and initial set-up costs in pounds sterling {GBP};

Errors Discovered: is the accumulated number of errors or defects identified by customers as a result of the review process having been generated by requirements engineers {errors};

Expected Delivery Time: is the minimum time required to process all initial and the additional requirements requests to deliver product specification to the Customer. {Months}.

These variables provide a reasonable starting point for conceptualising the feedback structure governing the dynamics in the requirements engineering process. A description of the quantitative behaviour of these key variables form also a reference mode necessary to reinforce the understanding gained from the problem statement identified.

4.3 Reference Modes

Inherent problems of the RE process are due to the productivity of the tools and requirements engineers, and the resulting product specification quality. The measurement theory used in this paper combines the product, process and resource variables with Wolstenholme's (1990) approach, to define graphical behaviour of variables, attributes, their relationships and interaction.
This historical behaviour of key variables is used in the studies of software measurement (Fenton, 1991 and Boehm, 1981; De Marco, 1982) and those used in requirements engineering practice studies (Costello and Liu, 1995; Melchisedech; 2000; Kamsties et al, 1997).

These data can be used to postulate or depict the probable effect of process improvements on effort, cost, quality, schedule and customer satisfaction during the RE process project. In Figure 2 (Williams, Hall and Kennedy, 2000), the influence of engineering productivity improvements on the RE process effectiveness is partly due to organisational improvements in capability to undertake RE projects.

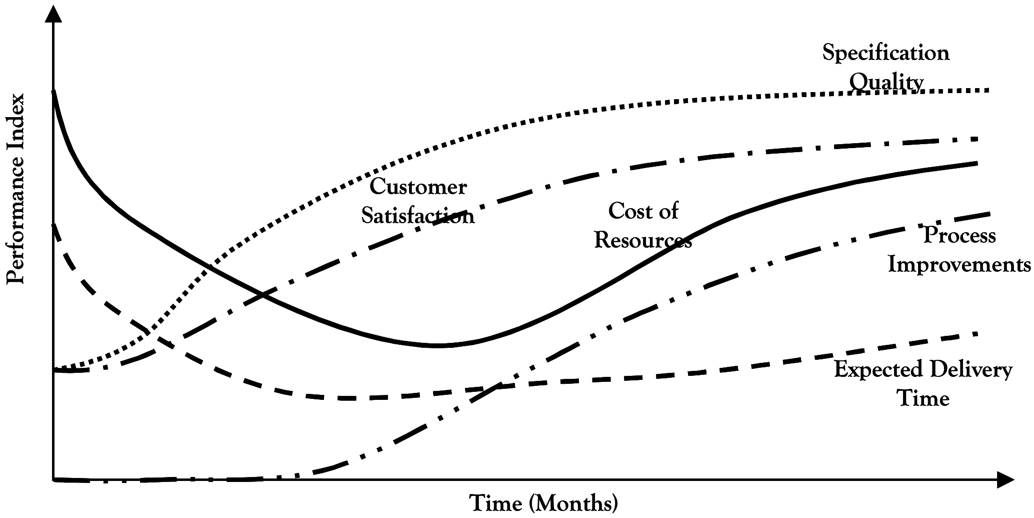

Figure 2: Reference Modes of Key Variables

Figure 2 indicates a mixture of growth and declines in specification quality, cost of resources, customer satisfaction, expected delivery time and process improvements.
Figure 3 (Williams, Hall and Kennedy, 2000), indicate a mixture of variations and trends in errors discovered, document reviewed, requirements changes, and process effectiveness over the 15 months period.

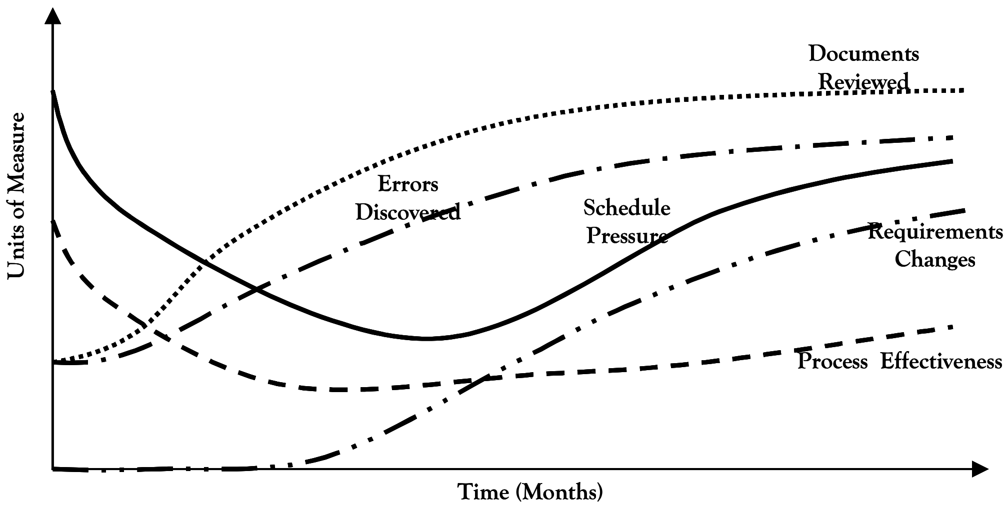

Figure 3: Reference Modes of Key Variables

For example patterns of variables presented in Figures 2 and 3 indicate a mixture of variations and trends. The fluctuations in variables presented in Figure 2 and 3 over time form the basis for interventions of RE process stakeholders through managerial activities concerned with providing resources, monitoring and control, and changing the resource flow path during the RE process (Williams, Hall and Kennedy, 2000). These reference modes provide significant insight into the underlying dynamics present in a system.

4.4 Dynamic hypothesis and key propositions

Models, tools and techniques supported by SD facilitate richer shared understanding (Senge, 1990) of domain knowledge and their structures (Meadows, 1982). Figure 4 demonstrates the relationships among RE process cost, quality and schedule and how they impact on the overall RE process performance effectiveness (Williams, 2001/b). Kliendhorfer et al (1993) emphasise that a test of a good theory lies in its ability to predict, shape or change the surrounding world. Figure 4 offers a useful basis for research on requirements engineering process success and a number of propositions that can be drawn from it. These propositions can be tested using data from field studies and validated by problem owners. The system dynamics theory of requirements engineering process encompasses interactions between the RE activities and the whole process relating to the work itself. Figure 4, provides a theoretical statement of the underlying hypothesis proposed in this thesis. It is these interactions as depicted in Figure 4 that determine the process behaviour patterns over time as a result of effects from process improvements. The initial RE process feedback structure (Figure 4) contains eight dominant feedback loops, of which four are Reinforcing loops (R), and other four are Balancing loops (B).

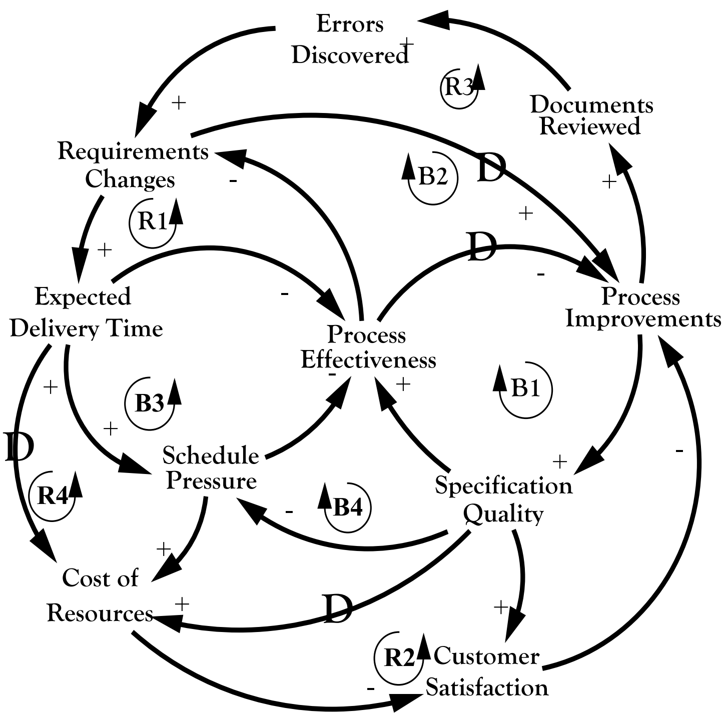

Figure 4: Dynamic Hypothesis of the RE Process

Reinforcing loop **(R)** or positive feedback is a representation of growing or declining actions. In this case, a positive loop response reinforces the original perturbation. Balancing loop **(B)**, also referred to as counteracting, is a goal seeking negative feedback process that seeks stability or return to control. When a feedback loop responds to a variable change opposing the original perturbation, the loop is negative or goal seeking (Goodman, 1974). In the SD modelling approach, feedback loops in causal loop diagrams or influence diagrams indicate the direction of

influence and polarity of the loop (Richardson and Pugh, 1981; Richardson, 1996; Goodman, 1974; Roberts, 1964; Dangerfield, 1992). Polarity (negative or positive) of the entire loop is determined by tracing consequences of an arbitrary change in one loop variable. In this thesis material or information delay is denoted by the symbol (D) in influence diagrams.
Dangerfield and Roberts (1995) contend that SD causal loop diagrams reflect content whilst stock-and flow diagrams represent structure of the model.

The interaction among these key variables throughout the project life cycle determines the completion time, resources required and the quality of the specification delivered. In return, the Customer/User may perceive the process as satisfactory or may not be satisfied with the quality of the specification delivered. A further review of the requirements document may call for process improvements in the form of error rework. This error rework, in turn may demand additional resources in terms of effort {person-months} and consequently cause delay in the progress of the RE project. These dynamics are explicitly illustrated in Figure 4 as changes – schedule loops (R1 and B3), quality – productivity loops (B1, B2 and B4) and cost – product value loops (R2, R3 and R4). The dynamics captured in Figures 2, 3 and 4 provide a basis for understanding such effects of RE process improvements on quality, cost, schedule and effort in the RE process.

After a number of review meetings in fieldwork and also with two independent consultants, the influence diagram presented in Figure 4 was accepted as the dynamic hypothesis of the RE process problem identified in this thesis. This development was iterative both referencing what is known in the literature and in practice (Bass et al, 1998; Lin et al, 1997). Based on the key variables identified in 4.2 brief description of the main feedback loops and the dynamics among key variables in the model were accepted as those that play important roles to problem owners. It was also possible to derive propositions from the model. The dynamics portrayed in Figures 2, 3 and 4 provide a basis for understanding such effects of quality, cost, schedule and effort on the RE process. These variables are modelled in three broad groupings as change requests – schedule loops, quality – productivity loops and cost – product value loops. Crossed influences in Figure 4 with a bold "**D**" denote a time delay in the same direction of the arrow.

4.4.1 Change Requests - Schedule Loops (R1 and B3)

R1 and B3 are minor loops that monitor and control the RE process progress, based on the requirements changes requested and actual process effectiveness. For example, in Figure 4, showing the Reinforcing loop R1, an increase in Requirements Changes will cause an increase in expected Delivery Time; this in turn decreases Effectiveness. Meanwhile an increase in process effectiveness decreases the desire to request requirements change any further.

Proposition 1: *The extent of decrease in expected delivery time will depend on the magnitude of requirements changes*

The effect of the Balancing loop B3 is to increase requirements changes as a result of reduced process effectiveness leading to longer expected delivery time and higher project schedule pressure. Qualitative analysis of these two loops supports the historical pattern of the behaviour for expected delivery time, requirements changes and process effectiveness as identified in Figures 2 and 3.

4.4.2 Quality –Productivity Loops (B1, B2 and B4)

The dynamics portrayed in the interactions resulting from B1, B2 and B4 (Figure 4) present major balancing loops in the system. In the B1 balancing loop, an increase in process effectiveness overtime decreases the need for process improvements. In turn, an increase in Process Improvements increases specifications quality, which again increases the process effectiveness, hence closing the balancing loop. On the other hand, balancing loop B2 is triggered by increases in Requirements changes and leads to an increase in process improvements, which in turn improves the specification quality. An increase in the specification quality increases the process effectiveness; in turn this increase reduces the need for further requirements changes as illustrated in Figure 4.

Proposition 2: *The effect of RE process effectiveness depends significantly on the resulting product specification quality.*

In loop B4, an increase in the specification quality reduces the schedule pressure; in turn an increase in schedule pressure decrease process effectiveness. On the other hand, an increase in Process effectiveness decreases Productivity Process Improvements, hence closes the loop. The balancing loops (B1, B2, B3) indicate that continuous measurement of the product specification quality in the RE process results in improvements, consequently leading to fewer requirements changes and a shorter expected delivery time. This qualitative analysis, as illustrated in Figure 4, highlights the importance of the three balancing loops in bringing about improved process effectiveness and lesser cost of resources.

4.4.3 Cost – Product Value Loops (R2, R3 and R4)

The Cost-Product Value loops in Figure 4 represent dominant loops in the system. Reinforcing loop R2 is one of the most dominant loops in the model. An increase in Process improvements increases the Specification Quality, which in turn increases the cost of resources. Continued increase in the cost of resources decreases customer satisfaction, while an increase in customer satisfaction decreases the need for Process Improvements, which in turn increases the Specification Quality.

Proposition 3: *The more customers spend reviewing requirements documents the more errors are likely to discovered.*

In conducting qualitative analysis, it was assumed that there has been some positive increases in error discovered leading to an increase the Requirements changes. This Reinforcing loop R3 follows B2, except that an increase in Process Improvements after some delay leads to an increased Document reviewed. The more time customers spend reviewing requirements documents the more errors are likely to be discovered. Increases in errors discovered in turn increases Requirements changes, thus closing the loop. The fewer customer reviewers read requirements documents per month, the more errors they will find.

Proposition 4: *The extent of decrease in customer satisfaction will depend on the extent to which cost of resources and expected delivery time increase.*

A key reinforcing loop R4 is one of the longest and the most dominant loops, in addition to loop R2. An increase in Process Improvements increases the possibility for more documents to be reviewed; consequently more errors will be discovered. An increase in errors discovered increases the ability for the SDO to make more requirements changes. The more changes made the longer the expected Delivery time becomes. After some delays the increase in expected delivery time increases the cost of resources, which in turn reduces customer satisfaction. A decrease in customer satisfaction increases the need for Process Improvements to cope with the outstanding change backlog. Elimination of possible requirements changes and costs as a result of Documents Review results in a better estimation of the Expected Delivery Time and a decrease in cost of resources and increased Customer Satisfaction. An increase in understanding of requirements indicates high effectiveness of the RE process modelling. Considering the complexity of the problems that SD attempts to solve, there is a definite need for a structured approach in its implementation (Williams and Kennedy, 1997). The next section describes the RE process model based on sectors of the various subsystems identified (Morecroft, 1979). The understanding of the RE process problems can be gained through using resulting system dynamics model presented in proposed Section Five.

5.0 Requirements engineering system dynamics model structure

Lack of RE process historical data has made it difficult for many organisations to simulate their maturity level before the actual process assessment. However, it may be possible that some organizations have information that allows them to use simulation more appropriate to a higher-level organization. Raffo and Kellner (1999) contend that organisations situated at higher maturity levels can only achieve this state "*if they have either a deeper understanding of, or have more detailed empirical data on their processes than is typical for their level*". Building models is relatively easy - empirically validating them in the real world is hard (Christie, 1980). An important phase in this research was the verification, validation and an establishment of a credible model. In this regard, having a higher level of process maturity provides a framework that facilitates this validation exercise (Coyle, 1996; Senge and Forrester, 1980).

The interacting three subsystems (Process organisation, Product engineering, Project Management) and their 14 sectors identified in Table 4 endogenously present the RE process modelling problem presented in the problem statement. The sector names are selected based on the key business process of each process categories presented in Table 4, summarising major functions of each subsystem and their sectors (Morecroft, 1979). Process Organisation subsystem main goal is to align organisational resources required to support a new "request for proposal" submitted by the Customer. There are four main sectors in process organisation, including: Project Scope sector, Process Integration sector, Customer Services sector and Project Finance sector.

Table 4: Model Sectors and their Main Functions in the RE Process Model

Sub-System	Sector	Major Functions
1. Process Organisation		
	1.1 Project Scope Sector	Main function is to ensure that the organisation producing requirements for specific action adheres to standard process and there is a business case in the project in terms of project scope and finances.
	1.2 Process Integration Sector	
	1.3 Customer Services Sector	
	1.4 Project Finance	
2. Product Engineering		The main function of this sector is the acquisition of resources, process technology and provision of support environment. This is to ensure that customer acceptance and satisfaction with the product quality.
	2.1 Process Technology Sector TE	
	2.2 Development\Process Sector	
	2.3 RE Support Environment	
	2.4 Process Schedule Sector	
3. Project Management		The main function is to plan and control achievable commitments regarding cost, schedule and resources.
	3.1 Quality Assurance	
	3.2 Planning and Control Sector	
	3.3 RE Procurement and Contract Management	
	3.4 Document Management Sector	
		The main function is based on perception of the need for new system customer monitoring the performance of the project in terms of evaluating the request for funds from the SDO.
External	3.5 Customer/User Sector	

The Product engineering subsystem integrates all the requirements development activities that produce a product specification (requirements document) that is correct and consistent with the customer's requirements. The product engineering subsystem involve eliciting (capturing) requirements and specifying them correctly and completely and comprises of four main sectors, including: Technology sector, Development process sector, Support environment sector and Process Schedule sector as presented in Table 4..

The Project management subsystem mainly focuses on the planning and controlling requirements engineering projects that are delivered on time and within budget. The project management descriptive subsystem comprises four main prescriptive as follows: quality assurance sector, procurement management sector, planning and control sector and document management sector.

Customer /User Organisation Sector shows the structure of customer behaviour in the requirements engineering process. The customer initially is swayed by the perceived value of a product, estimating project cost and evaluating the worth of a project for investment purposes. Although the customer does not usually engage in the requirements development process, the customer is usually influenced by the same kinds of determinants as the SDO in the requirements engineering process. Another important characteristic of the Customer/User organisation is the Customer confidence. The Customer confidence in the SDO influences the degree to which it respects the SDO's estimates of the value and cost of the project. The level of confidence also affects the Customer's faith in the technical capabilities of the SDO (Roberts, 1964). Simulation results indicate that for any given set of requirements specification project, the cost of requirements engineering services determines the level of customer satisfaction in return the financial resources of the customer provide the funds required for RE project accomplishment. The next section discusses key model equations.

5.1 Key Model Equations

The model sectors presented in Table 4 consists of about 343 equations with 47 level and 48 rate variables. In the model, the "request for proposal" and inflation are assumed to be exogenous. There are fourteen endogenous interacting sectors within three subsystems identified (Morecroft, 1979). Using STELLA software Research Version 6.0 (HPS, 1996-2000), the variables identified to characterise the RE process and their relationships are explicitly defined and modelled for intensive simulation experiments. The model simulates for 15 months in five equally spaced graphic display representing three months each. To prevent instability in the model from delays and to ensure that output from the model delays approximates reasonably (Vapenikova and Dangerfield, 1987-1991) closely to that of the requirements engineering process, the model is simulated using Runge Kunta 4 order of integration. A shorter time step (DT) of 0.0625 is set as a sixteenth of the smallest fourth order delay in the model representing approximately one day of the simulation time unit. Such control statements are aimed at achieving a good compromise between the accuracy of results and the speed of simulation (HPS, 1996-2000).

Customer_satisfaction(t) = Customer_satisfaction(t - dt) + (Chg_in_Satisfaction) * dt
INIT Customer_satisfaction = 1 {fraction}
Chg_in_Satisfaction = (Total_revenue_from_Customer/Cost_of_Resources-Customer_satisfaction)*Perceived_Quality/Technical_eff_Adjt_Time {fraction/month}

Customer satisfaction is an important soft variable in many requirements engineering projects and software engineering projects. Customer satisfaction is the level of content experienced by the customer based on the Specification quality and funds requested for RE services. In the field study observations, the customer was most satisfied when a product specification quality was highly valued, and when it became available for the customer's use, particularly when the customer's costs to acquire the product was low. The customer is disappointed when he perceives that the product specification "cost more than is worth". Customer satisfaction will be best attained if the SDO opts for middle range of project durations. Customer satisfaction is initialised at 1 and is dependent on changes in total revenue, cost of resources and perceived quality by the customer. Too-short project duration results in huge costs and engineer productivity improvements inefficiencies. However too-long RE process duration delays force the specification product quality low when the value of the finished product value is almost negligible. These extreme cases both force the RE process project to be abandoned.

Cost_of_Resources(t) = Cost_of_Resources(t - dt) + (Operating_Costs) * dt
INIT Cost_of_Resources = 2500 {GBP}

Cost of Resources is an accumulation of various foregone values of the resources needed for the RE process completion. Business costs may vary in many forms with different projects. In the field observations these costs were a result of monthly expenditure on documentation, training and Engineer Salaries, and facilities used. Chatzoglou (1997) found that the use of "resources" usually means a number of people involved, time and money allocated is not enough for successful completion of the requirements engineering process time. Cost of resources is initialised at £ 2,500 representing the initial set-up costs for the project.

Operating_Costs = SUM(Other_Overheads, Documentation_Cost,Training_Cost,Engineer_Salaries, Taxes) {GBP/month}

Monthly Operating Costs are costs incurred as a result of undertaking the requirements engineering process. The costs cover document production cost, training cost for requirements engineers, and wages and taxes paid to the Government.

Requirements_Changes = (Requirements_management-Total_Error_Reworked) {pages}

Requirements Changes are the number of changes requested by the customer as a result of review of the requirements baselined in the requirements engineering management Database.

Requirements_Volatility = (Requirements_management-Requirements_Changes)/Initial_Process_definition {fraction}

Requirements volatility is an important metric indicating how initial process definition changes over the requirements process life cycle.

Quality_Effectiveness = Specification_Quality/SDO's_Quality_Objective {fraction}

The effectiveness of the resulting product specification is regarded as the indicator for the RE process success relative to SDO wide set quality objective. The closer to 1, the higher the effectiveness. Specification Quality is the relative product specification quality as perceived by reviewers of the Requirements document. Hence the relative specification quality is regarded as the indicator of specification quality. The specification quality is a fractional level of correctness and completeness in relation to the quality objective incorporated in the product quality.

Specification_Quality = (Fraction_Quality*Perceived_Quality) {fraction}

Commercial requirements management tools are used to manage this intense work as they are effective in keeping track of documents from product version to specifications (Weigers, 1999). Key variables like Planned Review and Average Requirements Documents are important drivers for the model.

Documents_Reviewed(t) = Documents_Reviewed(t - dt) + (Review_Rate - Chg_in_req_doc - Rework_Rate) * dt
INIT Documents_Reviewed = 1 {pages}

Documents Reviewed are the accumulated number of pages reviewed as a result of review rate. The level of this state variable significantly depends on the review rate and the required changes in the requirements document. Extending automation support for stakeholder environments enables further support for paperless exchange of information and more effective review of requirements engineering artefacts.

Errors_Discovered(t) = Errors_Discovered(t - dt) + (Error_Discovery_rate) * dt
INIT Errors_Discovered = 3.5 {errors}

Errors Discovered are errors or defects discovered in the requirements document by reviewers. Initially there are 3.5 errors discovered until after time 0. An error is a problem or technical defect detected in the requirements document from an activity, such as a misspelling in the requirements document or a flaw in requirements Use case model.

Error_Discovery_rate =
(Potential_Errors*Frac_Error_Discovorable*Effect_of_Review_Rate)/Time_to_detect_Errors
{errors/month}

Error Discovery Rate is a monthly rate at which errors are discovered in a requirements document as a result of the review process. This rate depends on fraction of errors discoverable and the potential errors discoverable. The more time customer spends reviewing the requirements document the more errors are likely to be found.

Expected_Delivery_Time = MEAN(Indicated_Completion_Time,Scheduled_Completion_Time) {months}

Expected Delivery Time represents the time required to process and deliver the product specification. The requirements process will continue until all activities have been completed.

Process_Effectiveness = (Quality_Effectiveness*Cost_effectiveness*Schedule_Pressure) {fraction/month}

Process Effectiveness is the mean effectiveness of key indicators in the RE process including quality, cost and schedule. The effectiveness is the degree to which RE process purpose is achieved in terms of the specification quality. cost and schedule. For example, the RE process effectiveness depends on the probability of successful completion of the project on a planned delivery time and the resulting product specification. An effectiveness higher than 0.65 {fraction} is considered to be high.

Process_Improvements =
(Applied_Rigour*Actual_Technical_Effectiveness*Customer_satisfaction*Technology_transfer_to_Customer*Requirements_Management*Process_Effectiveness) {pages/month}

Process Improvements is the level of capability developed by the SDO in implementing requirements engineering best practices within their organisation. This aggregated variable significantly depends on applied rigour, technical effectiveness and customer satisfaction. During fieldwork observations, one project manager indicated that "*the ability to transfer technology and efficiently would also be seen as an important area of process improvements*".

Schedule_Pressure =
Effect_of_schedule_press_on_frac/Time_Perceived_Still_Remaining*Schedule_Compression_Factor*Schedule_Switch {fraction/month}

Schedule Pressure relates to a monthly fractional variation in the average RE process completion time and the expected delivery time of the product specification.

Due to the large size of the model (35 pages), other equations can be supplied on request .The next section discusses the simulation results.

5.2 Requirements Engineering Systems Dynamics Model Interface

An intuitive interface for the RE process system dynamics model (DYNASIS) was developed. DYNASIS is a prototype research model resulting from the application of Dynamic Synthesis Methodology to the modelling and analysis of requirements engineering process (Williams, 2000; Williams, 2001a/b, 2002). This RE process system dynamics model, whose interface is illustrated in Figure 5 predicts the performance of the RE process in terms of cost, quality of specifications and schedule of the RE projects.

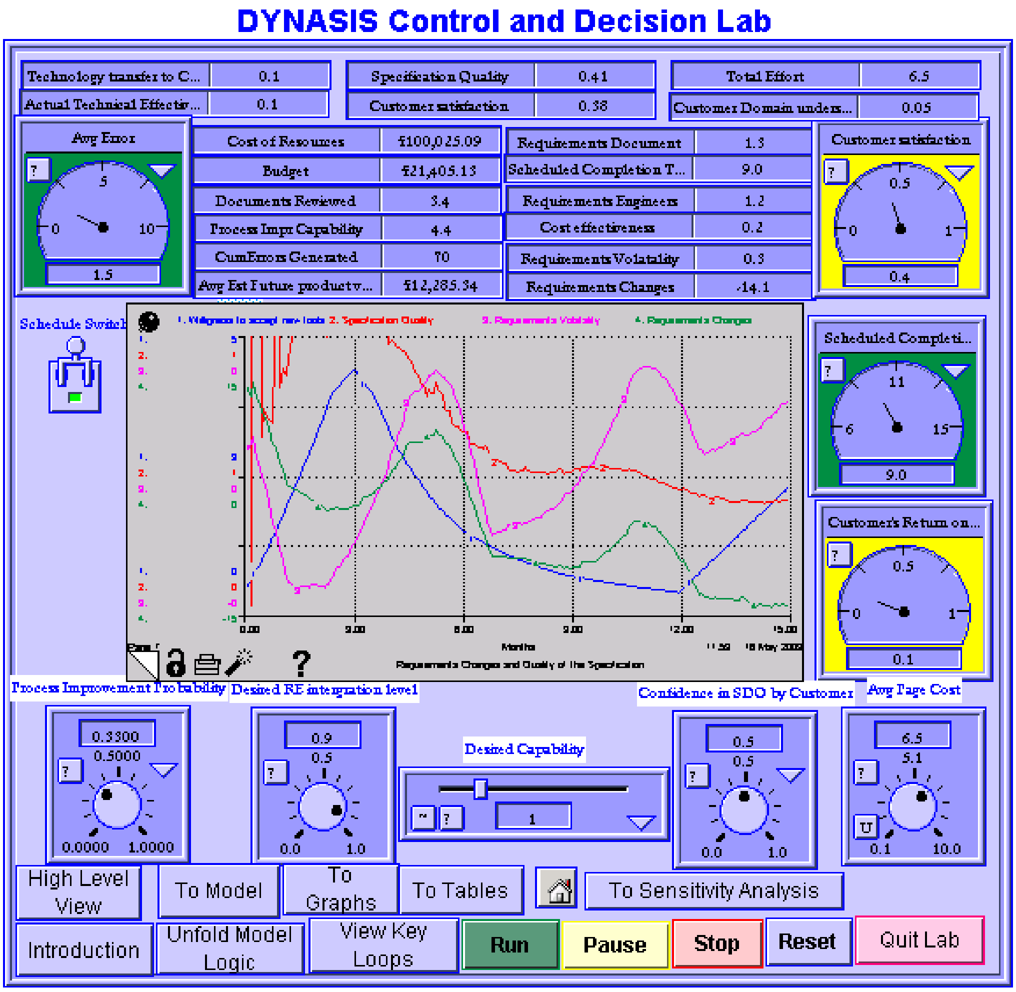

Figure 5: Main Interface for DYNASIS Tool

As illustrated in Figure 5, a requirements engineer or manager who uses DYNASIS can evaluate the completeness, consistency and accuracy of the model in predicting RE process performance against known performance measures as well as gain requirements engineering planning and control expertise through training with tool (Williams, Hall and Kennedy, 2000; Williams, 2001a/b).

The simulation results presented in Figures 6 and 7 show some of the complexity that can be derived and simulated with DYNASIS Model. The description of the RE process system dynamics model discussed in this paper appropriately fits a medium-sized, single complex requirements engineering project. Characterisation of the RE process validates the use of the Dynamic Synthesis Methodology in solving process–based problems as proposed by Williams (2001a/b; 2002).

5.3 Simulation Results

The simulation results can be directly compared with the reference mode behaviour, identified in Figure 3 and 4. As expected, completion time for the RE project slightly reduces initially and then continues to track from month seven till the final simulation time. The requirements changes continue to increase gradually, and the schedule pressure decreases initially and after some instability slowly rises during the rest of the seven months remaining. These results suggest patterns of behaviour similar to the reference mode presented initially and the supporting dynamic hypothesis.

Figure 6 shows interesting dynamics into the effects of estimated schedule Adjustment Time on requirements changes (4), willingness to accept changes (1) and requirements volatility (3) and specification quality (2). The second set of simulation runs included relaxing the assumption of Integrity Coefficient of the SDO {0.5 fraction}, and replacing it with RE process stakeholder's 'best' estimate of 0.95 {fraction}. The increase in Integrity Coefficient of the SDO significantly shows an initial study increase to month three in Willingness to accept changes (4), and then eases off in a Z-shaped behaviour until month 12. These changes have a profound influence on requirements volatility (3) and requirements changes (4) as indicated in Figure 6.

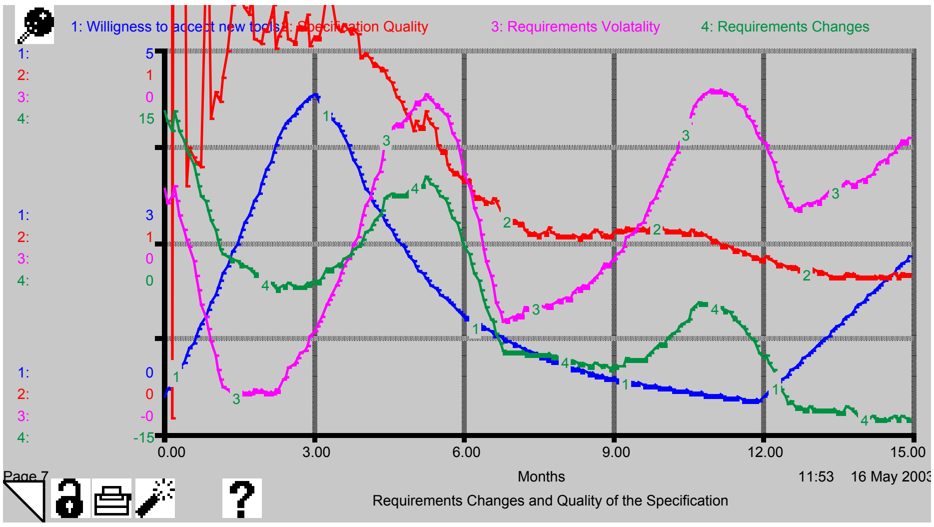

Figure 6: Simulation Output: Requirements changes and Volatility

In Figure 5, requirements volatility (3) and requirements changes (4) initially fall to 0 in month three, then rise steadily to ease off in month 5. As illustrated, a change in requirements leads to changes in requirements volatility (3) metric. The change also in volatility influences the level of the specification quality (2) being delivered to the customer.

In addition, the assumption of constant values of the Customer return On Investment (0.15 fraction) and Delay in Process improvements (1.5 months), outlined in the equations in the process technology and procurement management sectors, were replaced by 1 and 3 respectively. The results of this second validation base case run are presented in Figure 7

Figure 7 exhibits a more stable behaviour showing slow increase in requirements management database size (3) and cost of resources (5). It is interesting to note that the behaviour of Expected Delivery Time (4) has a direct impact on the requirements engineers (1) on the project. As the expected delivery time collapses in month five, immediately the SDO recruits requirements engineers to try to bring the project on schedule. The same pattern is repeated in month seven. The total effort (2) in person-hours on the project reflects the level of requirements changes (Figure 6) on the project, while cost of resources (5) on the project continue to rise. The effects of increased requirements changes (3) from month 1 to month 9 have a significant impact in product delivery times. One of the key reasons why RE project fail.

Figure 7: Simulation Output: Requirements Engineers, Cost of resources and Expected Delivery Time

Results from RE process system dynamics model analysis clearly show that the model provides a basis for debate on many issues in RE process. The author would like to concur with Wolstenholme et al (1990), who conclude that it is *"the process of interpreting the results obtained by super-imposing the model which gives rise to an understanding of how the results are actually generated"*. The RE process model relationships are based on the views of requirements engineering customers, thus their insights captured into the model shed light on fundamental aspects of the real requirements engineering process. However, it should be noted that the model has not studied the performance of a RE process under a number of different scenarios. Undertaking scenario analysis will undoubtedly unfold some weaknesses in the model. This process points the model into areas of further work. In order to test the propositions suggested in section 4.4, it is necessary to verify, validate and establish credibility of the model. There model has been validated by problem owners i.e. requirements engineers and managers. The author intends to carry out a programme of research to further test propositions identified in section 4.4. The aim will be to undertake field studies that will confirm characteristics of RE projects relevant to requirements engineering research. The understanding of the RE process gained in this paper and characterised in the DYNASIS model maybe generalised to other types of requirements engineering projects.

6.0 Conclusion and Future Work

The paper indicates that significant research has been carried out in relation to requirements engineering tools and techniques. Thus the model has the potential to make a significant contribution to the study and evaluation of requirements engineering process, whether its domain is defence or business, such as that described in case study. The system dynamics model was developed to represent a common class of problems in the requirements engineering process, but it has some limitations. Different RE domains will require model calibrations to realistically reflect specific projects. The limitation of the requirements engineering process model developed in this paper suggests the following important issues for the broader application:

A further research programme may be directed towards testing and empirical validation of the model. Further work may be directed towards field validation and improvement of the model as a tool for requirements engineering process feedback structure and simulation model optimisation.

Further research may also be directed at extending the model both in scope and purpose to include decision engineering to support the scenario analysis.
Requirements Engineering Metrics Definition Tool

Lack of data pertaining to RE is a big problem hampering research in systems software engineering field as a whole (Christie, 1994). Further development of the tool can be a basis for research directed at establishing a data collection method for potential requirements engineering metrics programme.

Another study may be directed toward understanding how SDOs estimate RE processes project costs. In this type of investigation, an understanding of what makes IT managers decide to invest funds into the RE process effort may be explained.

A further promising extension of the investigation reported in this paper is a study of two or more requirements engineering projects in parallel rather than in sequence, either within a SDO or with a single customer. In such a project aspects of investment decisions and resources allocations policies should present an opportunity and a foundation for the necessary extensions of the RE process model reported in this paper.

While the RE process problems have not been solved, the contributions of this paper will undoubtedly enhance our understanding of the RE process modelling. The RE process system dynamics model developed provides a fulcrum for debate and enhanced understanding of the RE process. The model should not be viewed as an answer to the problems, but rather as a vehicle for exploring many of the problems reported in the literature. Currently RE still has no general standards except those supporting the defence sector. The use of DYNASIS tool to collect more comprehensive empirical data for process enactment and improvement will allow the model to support prescriptive framework that may make a significant contribution to requirements engineering debate in terms of standardisation, process enactment and technology transfer.

Acknowledgments

The author is grateful for the suggestions and comments made by the anonymous reviewer.

References

Abdel-Hamid, T. (1996). The Slippery Path to Productivity Improvement, IEEE Software, pp. 43-53, July

Abdel-Hamid, T., and Madnick, S., (1991). Software Project Dynamics. Prentice-Hall, Englewood Cliffs, NJ.

Adelman, L. (1992). *Evaluating Decision Support and Expert Systems.* Wiley. New York:

Bass-Lenclemets, P and R. Kazman (1998) Software architecture in Practice. Reading MA.: Addison_Wesley

Boehm, B., (1981). Software Engineering Economics. Prentice-Hall, Englewood Clliffs, NJ.

Brookes, F. P., (1978). *The Mythical Man-month*. Addison-Wesley, Reading, MA.

Bubenko, J. Rolland, C., Loucopoulos, P. and deAutonellis V (1994). *Facilitating "Fuzzy to formal" Requirements Engineering*. Proc. IEEE International Conference RE pp: 154-7

Caputo, Kim (1998) CMM Implementation Guide: Choreographing Software Process Improvement, Reading, MA.: Addison-Wesley

Chatzoglou. P. D. and Macaulay, L.A. (1996). A review of existing models for project planning and estimation and the need for a new approach, *International Journal of Project Management*, Vol. 14, No. 3, pp. 173-183.

Chatzoglou. P. D., (1997). Factors affecting completion of the requirementscapture stage of projects with different characteristics, *Information and Software Technology Vol. 39* pp 627-640

Checkland, P. B. and Scholes, J. (1990). "*Soft Systems Methodology in Action*", Wiley, Chichester

Christie, A. M. (1980). Comparison of experimental and analytic simulations of reactor structural response to a simulated core disruptive accident. Nuclear Engineering and Design 60.

Christie, A. M. (1994) Enactable process speciÆcation in support of process-centered environments. Information and Software Technology 36 (11).

Christie, A. M. (1999) Simulation in Support of CMM-based process improvement, Journal of Systems and Software Vol: 46 pp 107-112

Costello, R. J. and Liu, D. B. (1995). Metrics for requirements engineering. *The Journal of Systems and Software*, 29(1):39-63.

Coyle, R. G., (1996). *System Dynamics Modelling*--A Practical Approach, London: Chapman & Hall. Pp. 413

Curtis, B. Krasner, H. and Iscoe, N. (1988). A Field Study of the Software Design Process for Large Systems. *Communications of the ACM*, 31(11): 1268-1287.

Curtis, B., Kellner, M. I. and Over, J. (1992). Process Modelling, *Communications of the ACM,* 35(9), Sept. 1992, pp. 75-90.

Dangerfield, B. (1992) The System Dynamics Modelling Process and DYSMAP2, *European Journal of Operational Reseaerch Society*, Vol. 59, No 4 pp 203-209

Dangerfield, B. and Roberts, C., (1995) Projecting Dynamic Behaviour in the Absence of a model: An experiment. *System Dynamics Review* 11,: 2 pp 157-172

Davis, A. (1993). *Software Requirements: Objects, Functions and States*. Prentice-Hall International, Englewood Cliffs, New Jersey.

Davis, A., Overmeyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh, A., Kincaid, G., Ledeboer, G., Reynolds, P., Sitaram, P., Ta, A. and Theofanos, M. (1993). *Identifying and Measuring Quality in a Software Requirements Specification*. I 1st International Symposium on Software Metrics pp. 141-152

Davis, C. and Vick, C. R. (1977). The Software Development System, IEEE Trans. Software Engineering SE-3(1) pp. 69-84.

DeMarco, T. (1982). Controlling Software Projects. Yourdon, New York.

El Emam, K. and Madhavji, N.H. (1995a). A field study of requirements engineering practices in information systems development. In Second IEEE International Symposium on Requirements Engineering, pages 68-80. IEEE Computer Society Press.

El Emam, K. and Madhavji, N.H. (1995b). Measuring the success of requirements engineering processes. In Second IEEE International Symposium on Requirements Engineering, pages 204-211. IEEE Computer Society Press.

Fenton, N., (1991) Software Measurement: Why a Formal Approach? In Formal Aspects of Measurement: Tim Denvir, Ros Herman and R.W. Whitty (Eds)., Computing Workshop, Springer-Verlag

Finkelstein, A., J (1994). "Requirements Engineering: a review and research agenda" in Proc 1st Asian & Pacific Software Engineering Conference, (IEEE CS Press) pp10-19.

Finkelstein, A., J. and Emmerich, W (2000). "The Future of Requirements Management Tools" in Information Systems in Public Administration and Law, Quirchmayr, G, Wagner R. and Wimmer, M (Eds.) (Oesterreichische Computer Gesellschaft).

Finkelstein, A., J. and Kramer, J (2000). "Software Engineering: A road map" in The Future of Software Engineering, Finkelstein, A., J. Eds. (ACM Press.) pp3-24

Finkelstein, A., J. Kramer and B., Nuseibeh, (1994). Software Process Modelling and Technology, Research Studies Press Ltd.

Forrester, J.W., (1961), Industrial Dynamics. Cambridge, MA.: MIT Press, 1961.

Galliers R.D. (ed) (1984) Information Systems Research: Issues, Methods and Practical Guidelines. Information Systems Series., Alfred Waller Ltd Henley-on-Thames

Gillies, A.C and P.Smith. (1984). *Managing Software Engineering: Case Studies and Solutions* Chapman and Hall.

Goodman, Michael R. (1974). *Study Notes in System Dynamics.* Cambridge, Mass.: Wright-Allen Press, Inc.

Gotel, O and Finkelstein, A., J. (1996). "An Analysis of the Requirements Traceability Problem" in Software Change Impact Analysis, Arnold, R. & Bohner, S Eds. (IEEE Computer Society Press).

Gruhn, V., (1991) . Validation and verification of software process models, Ph.D. Thesis University of Dartnund.

Gruhn, V., (1992) Software process simulation on arbitrary levels of abstraction. Computational Systems Analysis, pp. 439-444.

Gruhn, V., (1993). Software process simulation in MELMAC. SAMS 13, 37-57.

Henry and Kafura, (1995). Quantitative Evaluation of Software Process Improvements. *Journal of Systems Software,* Vol. 28, pp 169-177.

Hofmann, H. F. and Lehner. F. (2001) Requirements Engineering as a Success Factor in Software Projects, *IEEE Software* July/Agust pp58-66

HPS - High Performance Systems (1999-2000). *STELLA Users Guide.* Lyme Road, Hanover, NH: High Performance Systems.

HPS - High Performance Systems, Inc. (1996-2000). *STELLA*: An Introduction to System Thinking.

Jones, Capers (1994) "Rivatalising Software project Management" American Programmer, Vol 6 (7) pp 3-12 , June

Jones, P. (1992). The Mystery of the IT Managers. *Infomatics.* September

Kamsties, E. and Rombach, H. D. (1997). A Framework for Evaluating System and Software Requirements Specification Approaches. In Manfred Broy and Bernaard Rumpe, Editors. Proceedings of the RTSE'97-Workshop on Requirements Targeting Software and Systems Engineering pp 381-296, Munich, Germany, April, 1998.

Kamsties, E., Hormann, K. and Schlich, M. (2000). Requirements Engineering in Small and Medium Enterprises: State-of-the-practice, problems, solutions and technology transfer. (ICRE 2000).

Keller, R., Lajorie, R., Madhavji, N. H., Bruckhaus, (1995). A Comprehensive Process Model for Studying Software Process Paper, Technical Paper CRIM.

Kellner, M. (1989). Software process modelling: value and experience. In: SEI Technical Review. Software Engineering Institute., Carnegie Mellon University, Pittsburgh, PA, pp. 23-54.

Kellner, M., (1991). Software process modeling support for management planning and control. In: Proceedings of the First International Conference on the Software Process, Redondo Beach, California, October 21-22. IEEE Computer Society Press, Los Alamitos, CA, pp. 8-28.

Kellner, M.I., Madachy, R. and Raffo, D.M. (1999). Software process simulation modelling: Why? What? How?, Journal of Systems and Software, Vol. 46, pp. 91-105.

Kleindorfer, P. R., Kunreuther H.C. and Schoemaker P.J.H (1993). Decision Sciences: An integrative Perspective Cambridge University Press.

Kusumoto, S., et al., (1997). A new software project simulator based on generalized stochastic petrinet. In: Proceedings of the 19th International Conference on Software Engineering (ICSE-19), Boston, Massachusetts, May 17-23. IEEE Computer Society Press, Los Alamitos, CA, pp. 293-302.

Law, A. M. and Kelton, W. D. (1991). *Simulation Modelling and Analysis*, 2nd ed. New York: McGraw-Hill.

Lin. C. Y., Abdel-Hamid, T and Sherif, J (1997) Software-Engineering Process Simulation Model (SEPS)0, Journal of Systems Software, Vol 38 pp 263—277.

Loucopoulos, P. and Karakostas, V. (1995). *System Requirements Engineering*., McGraw-Hill.

Lubars, M., C. Potts and C. Richter, (1993). *A Review of the State of the practice in Requirements modelling,* IEEE International Symposium on Requirements Engineering, San Diego, CA, USA IEEE Computer Society Press

Macaulay, L. A., (1996). *Requirements Engineering: Applied Computing*., Springer

Madachy, R., (1994). A software project dynamics model for process cost, schedule and risk assessment, Ph.D. Dissertation. Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, CA.

McChesney, I. R. (1995). Toward a classification scheme for software process modelling approaches, *Information and Software Technology*, Vol: 37 (7) pp363-374.

Meadows, D. M,. (1982). *Lessons from Global Modelling and Modellers Features* Vol.: 14 pp: 111-121

Melchisedech, R. (2000). Investigation of Requirements Documents Written in natural Language. ICRE pp 89-99. IEEE.

Morecroft, J. D. W. (1979). An integrated approach to Industrial Dynamics. Working Paper, Sloan School of Management, MIT Cambridge, MA 02139.

Morecroft, J. D. W. (1980). A critical review of diagramming tools for conceptualisation feedback systems models. IEEE Conference on Cybernetics and Society.

Morecroft, J. D. W. (1982). A Critical Review of Diagramming tools for conceptualising feedback models, Dynamica Vol 8 (1), pp 20-29

Newman, M. and D. Robey (1992). A Social Process Model of User-Analyst Relationships. MIS Quarterly/ June pp: 249-266.

Paulk, M. C., B. Curtis., M.B Chrissis and C. V. Weber., (1993). Capability Maturity Model for Software., Version 1.1 Software Engineering Institute, CMU/SEI/-87-TR-24

Pohl, K. (1993). The three dimensions of requirements engineering. In Rolland, C., Bodart, F., and Cauvet, C., editors, 5th International Conference on Advanced Information Systems Engineering, pages 275-292. Springer-Verlag.

Raffo, D, (1996). Modelling software processes quantitatively and assessing the impact of potential process changes on process performance, Ph.D. Dissertation. Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA

Raffo, D. and Kellner, M., (1999). Modeling software processes quantitatively and evaluating the performance of process alternatives. In: K. El Emam and N. Madhavji (Eds.), Elements of Software Process Assessment and Improvement. IEEE Computer Society Press, Los Alamitos, CA

Raffo, D.M., (1995). Modeling software processes quantitatively andassessing the impact of potential process changes on process performance. Dissertation, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA.
Richardson, G. P. (1996). *Modelling for management I and II*: Simulation in Support of Systems Thinking. Dartmouth, Aldershot, UK
Richardson, G. P. and A. L. Pugh III., (1981). *Introduction to System Dynamics Modelling with DYNAMO*, MIT Cambridge, MA.: Productivity Press
Roberts, E. B. (1964). The Dynamics of Research and Development. Harper and Row, Publishers, New York.
Robson, C., (1993). Real World Research: A resource for Social Scientists and Practitioner- Researcher, BlackWell, Oxford.
Senge, P. M. (1990). The Fifth Discipline: The Art & Practice of The Learning Organization. Doubleday, New York.
Senge. P and Forrester, J. W (1980) Tests for Building confidence in system dynamics models, in Legasto, A J.W. Forrester, and Lyneis (eds) (1980) System Dynamics TIMMS studies in Management Sciences 14 New-Yorl : North-Holland, 209-228
Standish Group Report (1986). A Survey of Software engineers Standish Group Denver USA
Sterman, J. D(2000) Business Dynamics: Systems Thinking and Modelling for a Complex World, Irwin, McGraw-Hill.
Tvedt, J., (1996). A modular model for predicting the impact of process improvements on software development time, Ph.D. Dissertation. Computer Science and Engineering Department, Arizona State University, Tempe, AZ.
Wang, Y., Court, I., Ross, M., Staples, G and King, G. (1997). Mutual Transformation of the Capability levels between the current SPA models, Technical Report SI-RCSE-WANG97-SP123, pp 1-12.
Weiringa, R. J., (1996). Requirements Engineering: Frameworks for Understanding., John Wiley
Wiegers, K. E. (1999). *Software Requirements*, Microsoft Press.
Williams, D. and Kennedy, M. (1999a). A Framework for Improving the Requirements Engineering Process Effectiveness, International Council on Systems Engineering 1999 Conference, Brighton, UK, 5- 11 June
Williams, D. and Kennedy, M. (1999c). Component Based Modelling and simulation: A Case for Simulation Software Architecture, International System Dynamics 1999 Conference, Wellington, New Zealand, 20-23 July
Williams, D. Hall, T. and Kennedy, M. (1999). A Framework for Improving the Requirements Engineering Process modelling, In Eds: C. Hawkins, G. King, M. Ross and G. Staples Managing Quality : Proc. of Software Quality Management Conference, Southampton, UK, 29 - 31 March. pp 203-217
Williams, D., (2001/a) Systems Requirements Engineering: A process-based skill building approach, McMillan Global Publishing, Chichester (forthcoming)
Williams, D., (2001/b) A System Dynamics Theory of Requirements Engineering Process. In Proceedings of the International System Dynamics 2001 Conference, Atlanta, USA, 23-30 July
Williams, D., Hall, T. and Kennedy, M. (2000). A Framework for Improving the Requirements Engineering Process Management, *Software Quality Journal*, Vol. 8 (2) pp 133-147 February. ISSN: 0963-9314
Williams, D.W. (1994). *The Evaluation of SD and Software Project Management Using* ***ithink*** *Tool*, Unpublished MSc Thesis, SCISM, South Bank University.
Williams, W., and Kennedy M. S (1997). A viewpoints conceptual framework for improving the organisational information requirements: A System Dynamics perspective. In Yaman Barlas, Vedat G. Dicker and Seckin Polat Eds. 15th International System Dynamics Conference, Istanbul, Turkey, Vol 2, pp: 475-479. System Dynamics Society.
Williford, J and Chang, A. (1999) Modelling the FEDEx IT division: a system dynamics approach to strategic IT Planning, *Journal of Systems and Software* Vol. 46 pp 203-211
Wolstenholme, E .F. (1993). "*The Evaluation of Management Information Systems: -A System Dynamics Approach",* Wiley, Chichester
Wolstenholme, E. F., A. Gavin, K. M. Watts, and S. Henderson, (1990). The design of Dynamic Methodology for the Assessment of Computerised Information Systems. Procc. Of the 1990 International System Dynamics Conference, Eds, D. F. Andersen, G. P. Richardson, and J. D. Sterman. III: pp 1346-1354
Yadav, S.B., Bravoco, R.R., Chatfield, A.T., and Rajkumar, T.M. (1988). Comparison of analysis techniques for information requirement
Yadav. S. (1983). Determining an Organisation's Information Requirements: A State of the Art Survey, DATA BASE, Spring
Zahedi, F. (1995). *Quality Information Systems*, Boyd & Frazer Publishing Company Danvers, Mass., USA