Investigating Software Project Resource Planning: An Outline of a Proposed Multiple Case Study Research Approach

G A Bell and J O Jenkins
City University
School of Informatics
Department of Business Computing
Northampton Square, London EC1V 0HB, UK
Tel: 071 477 8000

Abstract

The use of system dynamics (SD) is proposed to assist project managers in examining the consequences of their resource distribution plans. Justification of the selection of this technique is given, in particular why it is proposed as an alternative to statistical forecasting techniques. The focus of the research is to provide insights into how to reduce schedule slippages, which is thought to be a common dynamic behavioural problem. One objective in reaching this goal is to identify common SD structures.

An overview of the original SD research contribution to the software engineering discipline is given. A brief description of the problem tackled and the model findings are outlined. Criticisms of the research process are developed to justify part of our approach. To ensure that this practical research investigation is both relevant to managers and scientifically rigorous, a selection of both qualitative and quantitative methods are suggested, to assist with the first objective. These are briefly justified and discussed in this paper.

Investigating Software Project Resource Planning: An Outline of a Proposed Multiple Case Study Research Approach

1 Introduction

Schedule slippage is a common software management problem, which can be attributed to poor project resource planning. The goals of this research are: to provide relevant causal explanation to this dynamic problem behaviour pattern; to develop feasible prevention strategies. SD (Forrester 1961) is proposed as a high level project management modelling technique, which can represent the most important software development life-cycle process activities. It is argued that SD will provide explicit causal understanding and assist in examining the dynamic consequences of various resourcing decisions.

In reviewing the SD literature, Abdel-Hamid's (1984) inquiry into software project management issues is thought to have an important contribution to the schedule slippage problem. A brief overview of the model will be given. After reviewing his work four significant and linked issues have been identified. These are: research design, model ownership, model size, and the time required to build such a model. A description of the research process is given and this highlights the model ownership problem. It is contended that the selected approach created a large model which is seen as a concern, because the verification test may not be undertaken. Therefore, management confidence in the model is reduced. This issue will be discussed further in the paper.

To shorten model development time, the first research objective is to identify fundamental project management variables and common SD structures given established criteria. An 'in house' model is being constructed which incorporates key features of the third process maturity level. Humphrey (1989) provides a clear summary of this level: *" The organisation has defined the process. This helps ensure consistent implementation and provides a basis for a better understanding of the process."*

This model will be validated against data collected from a UK company, and it can be thought of as the researchers' causal theory. It will then be taken to this organisation and be used to focus the dialogue. To ensure that the causal differences and the manager's world-view are captured, qualitative methods have been selected. When the model has passed the verification and validation tests it then can be classed as a local theory. An outline of part of the research design is given, and reasons why the case study approach has been chosen.

2 Initial Part of the Research Project

The software engineering discipline has produced both dynamic and static models to understand the relationship between project size, schedule and effort. Boehm's (1981) constructive cost model (COCOMO) can be thought of as a typical static model, for it has a unique variable project size which is taken as the starting point for the systematic calculation of all other attributes of the project. This model has three distinctive hierarchical levels which are: basic, intermediate and detailed, each becoming more complex as the required accuracy of estimation increases.

Doubts have emerged (Kemerer 1987) about the estimations calculated by both COCOMO and the software life-cycle model (SLIM) (Putnam 1978), especially when they have not been calibrated with measured data from that organisation. Humphrey (1989) believes that good management intuition supported by documentary evidence is more accurate than any cost model, though he does not support this argument with empirical evidence. He further suggests that managers should compare their earlier estimates with subsequent experience to improve their intuition.

Both academics and practitioners have now taken a keen interest in the software process model (SPM) concept . Their aim is to support management with planning, controlling and improving the software process (Kellner 1991). Humphrey (1989) identifies three SPM levels: universal, worldly and atomic. The universal SPM provides a high level overview of the

software development life-cycle process, for example Royce's (1970) waterfall model or Boehm's (1988) spiral model. Worldly process models are generally a description of procedures, and atomic models provide precise details such as algorithm specification.

SD has inter-dependent variables, therefore enabling it to be associated with the dynamic modelling taxonomy. It is thought that this technique can examine management strategies to different dynamic problems particularly at a high or universal level. This generally agrees with Kellner's (1990) view, and he cites Abdel-Hamid's (1984) research as very promising.

It is thought that organisations at the first process maturity level are more likely to show symptoms, such as schedule slippages followed by significant additional effort allocated to the testing stage. The focus of the research is to provide a number of resource planning insights that can be used as evidence to support management intuition.

To achieve this goal, it is believed that common SD structures are needed. One research objective is to identify organisations that are at the third process maturity level called 'defined'. It is argued that organisations at this process maturity level are less likely to show problem behaviour characteristics, and should begin to have similar reference behaviour modes.

Currently, data is being used from a major company that is thought to be at the 'defined' level. In an attempt to identify common structure the following models have been reviewed (Abdel-Hamid 1984; Ledet 1992; Richardson and Pugh 1981; Roberts 1964, 1974). It is contended that premise description and partial model analysis are methods applicable to assist in identifying common SD structures (Morecroft 1983, 1985). These methods were designed as a direct result of Simon's (1957, 1976) principle of bounded rationality arguments. He believes that decision-making is affected by an individuals' cognitive limitations, for example they may not be able to anticipate correctly the consequences of various decisions.

The constructed 'in house' model will be the researchers' theoretical interpretation that replicates the data. The key features of the third process maturity level will be used to verify the model, which will then be statistically validated (Sterman 1984).

It is assumed that a greater participation with stakeholders in the modelling process will ensure that verification and validation are equally important. Each equation that is part of the theoretical model which simulates the system must be defended and justified. According to Barlas and Carpenter (1990): *"If a critic can show that one of the model equations does not make sense (does not agree with an obvious causality), then the model is refuted even if the aggregate model output matches the data."*

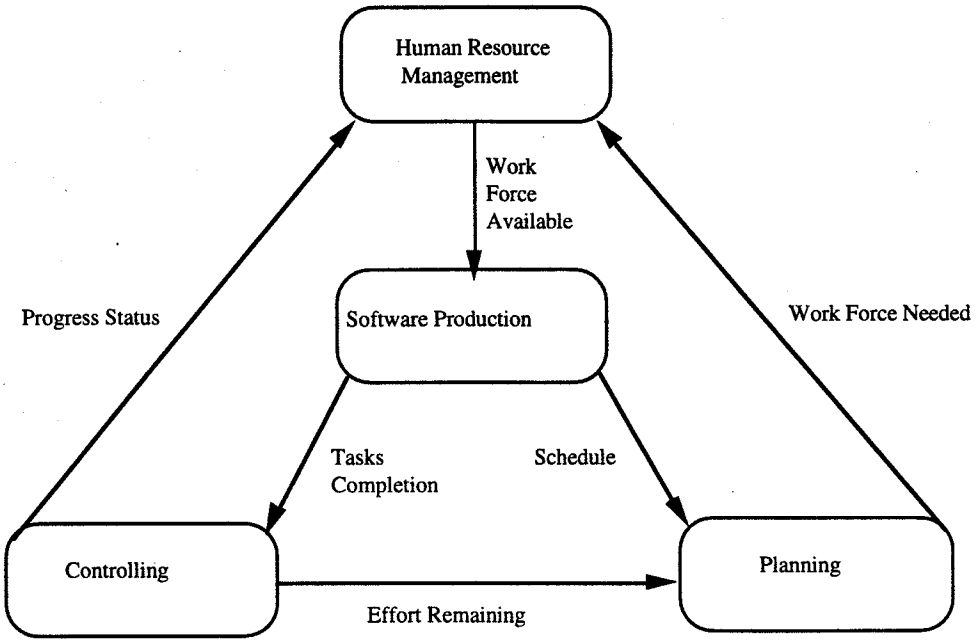
To ensure that the 'in house' model has captured the most important structures, qualitative methods will be used to capture the managers' understanding of the problem. When the model changes have been made, and passed the required confidence tests, it will be viewed as a local theory.

3 An Overview of Abdel-Hamid's Model

As stated, Abdel-Hamid's work is seen as crucial to the research problem, therefore a brief description of his model is now given. Figure 1 shows that the SD model can be broken down into four interrelated subsystems and these are:

a) The human resource management subsystem
b) The software production subsystem
c) The controlling subsystem
d) The planning subsystem

Figure 1. An overview of Abdel-Hamid's SD Model.



a) During software production, the human resource management subsystem attempts to capture human activities such as; hiring, training, and the transfer of staff. Each subsystem affects other subsystems, for example the auxiliary equation 'Work Force Needed' is calculated in the planning subsystem; and this equation influences the human resource management subsystem.

b) The simulation of software production is fairly complex, therefore this subsystem is broken down into four sectors:

i) manpower allocation.
ii) software development productivity.
iii) quality assurance and rework.
iv) system testing.

i) The manpower allocation sector calculates the effort required for training, quality assurance, software production, rework and testing.

ii) The software development productivity sector simulates the design and coding processes.
iii) The quality assurance and rework sector considers the generation, detection, and correction of errors within the software development phase. This incorporates the design, coding, reviewing and testing phases but excludes the requirements phase.

iv) The system testing sector simulates the errors that the quality assurance and rework sector fails to detect when the software is being designed and coded. It also considers the bad fixes resulting from faulty rework that remain undetected until the System Testing phase. As the project progresses comparisons between estimates are essential.

c) The controlling subsystem, in this case, attempts to simulate three important elements within any control function:

i) measurement - what is happening in the process being controlled;

ii) evaluation - the assessment of the significance of measured data;

iii) communication - the reporting of the analysis to other parts of the system.

d) The Planning subsystem makes initial estimates at the start of the project, and these estimates are then changed as the project progresses.

3.1 A Research Finding From The Model

Abdel-Hamid (1989) describes his model as a comprehensive representation of software project management. He further claims that the model captures essential management functions such as planning, controlling and staffing, as well as software project functions such as design, coding, reviewing and testing.

With this model he then investigated Brooks's Law (1975) which states *"Adding manpower to a late software project makes it later."* Abdel-Hamid (1989) argued that Brooks's theories have not been fully investigated, especially, the effects on productivity when adding new staff to a late project. Historical data was collected from the NASA Goddard software engineering laboratories which showed similar dynamic problem behaviour characteristics as described by Brooks. Abdel-Hamid's model replicated this data to an acceptable tolerance level, before commencing a number of controlled experiments.

The results from the model do not fully support Brooks's Law (Abdel-Hamid 1989), and he concludes that adding more people to a late project causes it to become more expensive. The rise in project costs were found to be caused by an increase in training requirements for new staff which leads to greater communication overheads. For project schedules to be affected requires the average production rate of the new team to be significantly reduced, thus indicating very little productivity from the new employees. Abdel-Hamid suggests that this may not be the norm. For example, new staff that are sufficiently experienced need very little time to readjust to a new situation. Therefore, productivity increases rapidly over a short period of time.

The results of this investigation are interesting and thought provoking. What is of concern is whether the manager of the project verified the model and agreed with the findings. Abdel-Hamid does not explicitly discuss this issue.

4 Critical review of the research design approach

There is some confusion about what was the goal of Abdel-Hamid's research: was he trying to produce a comprehensive generic model and then tackle common behavioural problems; or did he identify Brooks's Law as a problem and then construct a comprehensive model ? Paich (1985) seems to support the latter argument, but when reviewing Abdel-Hamid's research this is not made clear. This dilemma is highlighted by Powell (1987) who seems to support the former view, and wrote: *"A System Dynamics model of the software development process has been studied to determine its usefulness in modelling software project management."*

A description of the research process is now given, which will lead into the model ownership problem. Initially Abdel-Hamid undertook an extensive software engineering literature review. He then interviewed 10 experienced managers from different software producing organisations. It is claimed that the insight gained from this dialogue complemented his own software experience. This information became the basis for a prototype.

A further 17 managers from different companies were interviewed separately to enhance the systemic concept of managing a software project. However, this research process created a fundamental problem that there were 27 different world-views supposedly incorporated within this model. However, no dialogue took place between the owners of these views. There was no group consensus on software project causality to enable it to be thought of as a 'generic' software project model. Moreover, did these managers see Brooks's Law as a problem ? This model is unlikely to gain acceptance as the basis of a project management tool, because each causal feedback link must be accepted by the manager.

In illustration of this consider one local problem situation. A manager is required to transfer staff from one project to another (late running) project. One manager approached by us stated he would transfer his experienced staff rather than his inexperienced staff. In Abdel-Hamid's model, if the work force level sought is less than the current total work force level, then the new recruits would be the first to be transferred out of the project. Abdel-Hamid's model was therefore refuted by this manager and this suggests that the model may not be universally accepted.

This example further highlights the model ownership problem: whose view should the model reflect, that of the analysts or the project managers ? It believed that to investigate the problem situation more closely qualitative methods must be used. The purpose is to capture management understanding which is reflected in the model, thus reducing the verification or model ownership problem.

The research process applied in the development of the model ensured that layer after layer of equations were added to the model, thus reducing transparency and causing the *artichoke effect* (Miser and Quade 1988). Essentially the *artichoke effect* occurs when more equations are incorporated into the model to increase comprehensiveness. This could reduce confidence in the model, because the verification is not feasible. Managers who are very busy may not be able to spent much time trying to understand such a model. It may then be perceived as a black box and therefore more likely to be discarded.

This theory is confirmed by Powell (1987) who wrote:" *The model is large and complicated, and considerable effort is therefore required to be able to explore successfully any policy option. This difficulty will limit the wide use of the model until the analysts have become thoroughly familiar with it. A further consequence of the size and complexity of the model is that it may never be possible to achieve a status of full validation.*" Presently, the software engineering group which Powell belonged to have dropped Abdel-Hamid's model.

In summary, the research process was extremely rigorous, which led to the development of a large model. It is contended that verification is very important to ensure the model is owned by the manager, therefore dynamic insights are meaningful. It is not clear whether the manager of the problem behaviour agreed with the causality of the model or its findings. Moreover, it is argued that a dynamic problem should be identified first before an SD model is constructed. This could reduce the number of equations. It is proposed before a problem is examined common SD structures of the software development life cycle should be identified to decrease model building time.

The theoretical 'in house' resource model which is being developed will be verified with features from the third process maturity level, and validated against past data. To assist in developing the local theory, certain methods have been selected within the case study approach.

5 Why The Case Study Approach ?

It is our belief that, through dialogue with this manager (the expert) a local theory will be derived. The purpose of the case study approach is close to our objective, in that, it is primarily used for an exploratory study. Benbasat *et al* (1987) have identified eleven case study characteristics, ten of which are thought to be applicable to this investigation:

- Phenomenon is examined in a natural setting.
- Data are collected by multiple means.
- One or few entities (person, group, or organisation) are examined.
- The complexity of the unit is studied intensively.
- Case studies are more suitable for the exploration, classification and hypothesis development stage of the knowledge building process: the investigator should have a receptive attitude towards exploration.
- No experimental controls or manipulation are involved.
- The results derived depend heavily on the integrative powers of the investigator.
- Changes in site selection and data collection methods could take place as the

investigator develops new hypotheses.

- Case research is useful in the study of "why" and "how" questions.
- The focus is on contemporary events.

As stated, the theoretical model will focus the dialogue. Qualitative methods will be employed to capture the data, before a formal SD model is constructed. One strength of this approach is that first hand understanding of the phenomena under investigation can be accumulated. However, one weakness is the potential lack of objectivity, because of the researchers' stake in the outcome. This can be resolved by undertaking another investigation where the circumstances are similar.

As researchers, there is an obligation to meet certain positivistic requirements. For example: to produce generalisations; to search for objectivity. It is believed that a single exploratory case study may not match these characteristics. Therefore, another organisation must be found, and a second case study undertaken. It is hypothesised that companies at the third process maturity level, which have similar organisation, project management, process management and technology features will produce basic explanatory structures.

A common concern addressed at this approach is " How can you generalise from case studies ?" Yin (1989) argues "*...case studies, like experiments, are generalizable to theoretical propositions and not to populations* .." , and he further contends, *"..the investigator's goal is to expand and generalise theories (analytic generalisation) and not to enumerate frequencies (statistical generalisation)."*

6 Selected Methods

A brief overview and reasons why the following methods have been selected for this investigation are now given.

6.1 Literature Review

To enable closer involvement with a project manager software engineering literature must be reviewed, especially from the following topic areas:

Organisation This is concerned with the leadership of the software organisation. It covers a number of management aspects including resource allocation, policy making and training.

Project Management This deals with activities such as planning, tracking and project control.

Process Management As the company increases in maturity a process infrastructure should emerge. This essentially covers the definition of the current process, data gathering and analysis.

Technology This area is concerned with identifying and installing technology when needed.

6.2 Open-Ended Interview

The initial part of this research is to identify common variables and structures at the third process maturity level, that are believed to affect resource planning strategies. The first case study investigation is viewed as being exploratory, and examines the resourcing problem in its natural setting. The developed 'in house' theoretical model will be used to focus the conversation. The open-ended interview attempts to capture the richness of the problem as perceived by the manager.

6.3 Cognitive Mapping, Premise Description and Partial Model Testing

The cognitive mapping method (Eden 1988), is thought to be appropriate in capturing data from the open-ended interview. It is a technique that can structure and analyse the stakeholder's perspective. The identification of causal loops is central to the investigation. Eden *et al* (1992) contends that " *loops imply the possible existence of dynamic consideration within cognition - that is, the cognition has acknowledged either implicitly or explicitly growth, decline or feedback control.*"

Two other methods are being used to assist in identifying common structures. These are premise description and partial model testing (Morecroft 1983, 1985). Premise description is related to the formulation of equations that represent a decision. Partial model testing examines the premise of the decision to the simulated behaviour.

7 Conclusion

Abdel-Hamid's model is regarded as an important 'landmark' within the software engineering discipline. Criticisms of the rigorous research process have been developed. There is no evidence to suggest that the manager of the project, which exhibited similar characteristics as described by Brooks's Law, verified the model. Therefore, it is argued that model ownership is an issue, which implies dynamic insights gained are questionable.

One objective of the research, is to identify basic explanatory SD structures which capture fundamental software development life-cycle features that can affect project resourcing. This could reduce model building time, and be used to explain different local dynamic behavioural problems.

It is thought that, there is a three stage process to achieve this objective. First, an 'in house' model is being constructed, which can be regarded as the researchers' theory that replicates historical data collected from an organisation. This company is identified with the 'defined' level.

Second, to ensure that the model has captured the most important variables and structures it will be taken to a major UK company. This initial case study is regarded as an exploratory investigation into developing a local theory. Because it is believed that model verification is crucially important before any controlled experiments are undertaken. The chosen qualitative methods will assist in confirming, refuting and improving the model, to ensure relevance.

Finally, another case study will be undertaken to produce basic explanatory SD structures assuming certain organisational, project management, process management and technology criteria are met.

References

Abdel-Hamid, T.K. (1984) *The Dynamics of Software Development Project Management: An Integrative System Dynamics Perspective*, Ph.D dissertation, Sloan School of Management, MIT.

Abdel-Hamid, T.K. (1989) *The Dynamics of Software Project Staffing: A System Dynamics Based Simulation Approach*, IEEE Transactions on Software Engineering, Vol.15 (2), pp 109-119.

Barlas, Y., and Carpenter, S. (1990) *Philosophical Roots of Model Validation: Two Paradigms*, System Dynamics Review, Vol. 6 (2), pp 148 - 166.

Benbasat, I., Goldstein D.K., and Mead, M. (1987) *The Case Research Strategy In Studies of Information*. MIS Quarterly, Vol. 11 (3).

Boehm, B.W. (1981) *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, NJ.

Boehm, B.W. (1988) *A Spiral Model of Software Development and Enhancement*, IEEE Computer.

Brooks, F.P. Jr. (1975) *The Mythical Man-Month*, Addison-Wesley, Reading, Mass.

Eden, C. (1988) *Cognitive mapping*, European Journal of Operational Research, Vol. 36, pp 1-13.

Eden, C., Ackermann, F., and Cropper, S. (1992) *The Analysis of Cause Maps*, Journal of Management Studies, Vol. 29, No. 3.

Forrester, J.W. (1961) *Industrial Dynamics*, MIT Press, Cambridge, Mass.

Humphrey, W.S. (1989) *Managing the Software Process*, Addison-Wesley, Reading, Mass.

Kellner, M.I. (1990) *Supporting Software Processes through Software Process Modelling*, Proceedings of the 6th International Software Process Workshop: Support for Software Process, Hakodate, Japan.

Kellner, M.I. (1991) *Software Process Modelling Support for Management Planning and Control*. Proceedings of the 1st International Conference on the Software Process, Redondo Beach, California, USA.

Kemerer, C.F. (1987) *An Empirical Validation of Software Cost Estimation Models*, Communications of the ACM, Vol. 30 (5), pp 416-429.

Ledet, W.J. (1992) *Balancing Time-To-Market, Customer Understanding and Creative Innovation*, M.S. Thesis, Sloan School of Management, MIT.

Miser, H.J., and Quade, E.S. (Eds) (1988) *Handbook of Systems Analysis: Craft Issues and Procedural Choices*, John Wiley , New York.

Morecroft, J.D.W (1983) *System Dynamics: Portraying Bounded Rationality*, Omega,Vol.11 (2), pp 131-142.

References

Morecroft, J.D.W (1985) *Rationality in the Analysis of Behavioural Simulation Models*. Management Science, Vol.31 (7), pp 900-916.

Paich, M. (1985) *Generic Structures*. System Dynamics Review, No.1 (1), pp 126 - 132.

Powell, F. (1987) *Study of a Software Development Process Dynamic Model*, Mitre Document 10314, Bedford, Mass.

Putnam, L.H. (1978) *A General Empirical Solution to the Macro Software Sizing and Estimating Problem*, IEEE Trans. Software Engr. pp 345-361.

Richardson, G.P., and Pugh A.L. (1981) *Introduction to System Dynamics Modelling with Dynamo*, MIT Press, Cambridge, Mass.

Roberts, E.B. (1964) *Dynamics of Research and Development*, Harper and Row, New York.

Roberts, E.B. (1974) *A simple Model of R&D Project Dynamics*. In Managerial Applications of System Dynamics, Roberts, E.B. (1978) (Eds) MIT Press, Cambridge, Mass.

Royce, W.W.(1970) *Managing the Development of Large Software Systems*, Proceedings of IEEE WESCON.

Simon, H.A. (1957) *Rationality and Decision Making*. In Models of Man, John Wiley , New York.

Simon, H.A. (1976) *Administrative Behaviour*, Free Press, New York.

Sterman, J.D. (1984) *Appropriate Summary Statistics for Evaluating the Historical Fit Of System Dynamics*, Dynamica Volume 10, Part 2.

Yin, R. (1989) *Case Study Research: Design and Methods*, Sage Publications, Inc., Newbury Park, CA.