

## **System Dynamics in Project Management: Assessing the Impacts of Client Behaviour on Project Performance**

**ALEXANDRE G. RODRIGUES and TERRY M. WILLIAMS**

Department of Management Science, Strathclyde Business School  
University of Strathclyde, Glasgow G1 1QE, United Kingdom.  
Tel: +141 552 4400, Fax: +141 552 6686  
email: [mgtsoci@mansci.strath.ac.uk](mailto:mgtsoci@mansci.strath.ac.uk)

Major projects generally involve a "client" and a "contractor". The client often essential for approving intermediate milestones or documentation, can however, cause detrimental effects to the project, requiring changes to workscope or the product definition, delaying documentation approval or essential information, requiring too much reporting, or tightening milestone schedules. If the relationship starts to deteriorate, positive dynamics can be set up, and many such are discussed in this paper. These effects must be analysed and quantified: to keep control of the project costs and time, to estimate correctly the true cost of contract amendments, and where necessary to make auditable claims against the client. Traditional project management tools are based around decomposing the project into constituent cost or time elements. These techniques have proven inadequate on their own for analysing and managing modern complex projects, which are increasingly becoming more complex and intra-related. A holistic approach must be used, particularly to deal with systemic effects such as multiple client actions. A natural technique for modelling such effects qualitatively is cognitive maps / influence-diagram; a natural extension is to develop these into System Dynamics (SD) models, able to capture both hard auditable systemic effects, and the softer "human" effects that play an important part but are harder to quantify. Academic and practical work has continued to demonstrate the usefulness of SD in project management, particularly supporting dispute resolution.

The Client/contractor relationship during the project can be characterised by two different communication processes. The first is the continual reporting of milestone progress. As the contractor fails to meet milestones the Client wants compensation while his trust in the contractor reduces. On the other hand, the need to sacrifice early milestones for later success may not be accepted by the Client, and his reluctance may be reinforced if the contractor cannot demonstrate the benefit. Consequently, the contractor often avoids reporting delays hoping that staff will catch up. As communication deteriorates, Client/contractor co-operation reduces and conflicts become counter-productive. The second process is the continuous review of the system definition and functionality required. During development, intermediate sub-products are assessed so that misinterpretations of the requirements can be identified early. While the aim is to evolve towards a common and clear understanding, the result is often the demand of more functionality than that which the contractor believes to have been agreed.

An influence diagram in the paper shows key "vicious circles" in the underlying feedback structure. The introduction of changes is usually balanced by negotiated schedule adjustments based on an estimate of extra direct effort. However, the immediate

secondary effects is work being performed out of its sequence. This can also be caused by Client delays in documentation approval. Doing work out of sequence increases the number of errors, and when these are detected in areas thought to be stable staff start to lose trust in the current system's requirements, leading to poorer productivity and work quality. As delays occur, the Client's trust reduces, tolerating fewer schedule adjustments and demanding more progress reports, diverting staff from development work. Further changes conceded to persuade the Client to agree with delays set up reinforcing loops. A second Influence Diagram in the paper shows how Client behaviour interferes with project dynamics: aggravating delays and restricting schedule adjustments puts pressure on management to increase the work rate by hiring more staff, putting pressure on staff to work quicker, using over-time, reducing QA, and increasing activity concurrency. When over-applied, the disruptive long-term effects of such decisions become dominant and reinforcing. Again, these effects are difficult to estimate.

Frequently, initial contractual agreements become violated and changes need to be agreed. To avoid costly legal disputes, effective negotiation during the project is needed, but this is only possible with improved understanding and quantification of the impacts of changes. SD provides an appropriate approach, but needs to be embedded within the existing tactical decision-making structure. The authors have developed a project management integrated model (PMIM), providing a generic framework using SD models integrated with traditional procedures in supporting on-going project planning and monitoring. The use of SD models is focused on three major roles: estimating, risk analysis (e.g. assessing the impact of Client behaviour) and progress monitoring and diagnosis. The PMIM considers the use of SD models at both the strategic and operational management levels.

The authors are currently following a large-scale software project to improve and validate the PMIM. This field work comprises the development of new SD models. The project ("KDCOM") is to develop a Command and Fire Control System for the Republic of Korea Navy. The prime contractor is BAeSEMA, and they are directly involved in the development of the software component which is being implemented in two main builds (SWB1 and SWB2). Traditional planning and control procedures are being used based on a detailed WBS. Gantt Charts, logical networks and new approaches like Boehm's spiral model are being used. Client behaviour is of crucial importance; of particular relevance is the subjectivity involved in interpreting contractual agreements, which might be exacerbated by cultural differences, so the introduction of changes is a major risk. Traditional tools are recognised as inadequate to estimate and quantify the impact of these changes. For the case-study, SWB1 is to be used as the basis to develop, validate, and to calibrate the SD models. The models will then be used amongst other tools to support SWB2.

The model development approach was based on PMIM. An SD model aggregates several WBS packages representing more complex tasks. The paper shows the generic structure of the SD model that captures a development task (the SD-DTModel). The model has two parallel processes: engineering and management. Within the engineering process defects are generated, detected in the reviews, and removed by rework; however, some

“escape” (i.e. are not detected or not properly reworked). This provides a measure for the product quality and thus it is possible to assess the cost/time/quality trade-off quantitatively. The management process comprises the internal monitoring of progress and re-planning. While the SD-DTModel was being developed its structure had to be validated. The model was checked against the specified software development process, and the inclusion of all relevant factors was checked in informal interviews. The structuring was based on four steps described in the paper. In the final step, the model was calibrated to reproduce observed past behaviour. The critical issue here is that the model considers many variables and parameters which are either intangible or not measured. We distinguish between input metrics which are model parameters and output metrics which are variables reflecting the results of the simulation; some of the metrics collected can be directly translated into input metrics, while others will have to be matched by the model's output metrics.

Our example analyses the impact of changes to the design phase of a SWB1 sub-component; an SD-DTModel is used to aggregate the intermediate stages of this phase. The manpower budget assumed the project-wide policy of allocating 19% to each of QA (design reviews) and rework activities. The software sub-component, estimated at 5.5 KDSI (lines of code), represents about 12% of SWB1. Given the short time period management were not considering changing the staff level during the work. We focus on managing changes in the design phase, when a detailed plan is not available for later stages. Our first step in the example was to calibrate the model to reproduce the steady behaviour implicit in the plan. This included calibrating policy parameters under direct management control (e.g. length of the review period), and intangible process parameters, reflecting intrinsic characteristics of the development process (e.g. the effect of learning on productivity). The relationships are defined in the model by input functions; as these cannot be measured directly we have used approximations based on past data, managers/staff experience (i.e. mental models), and literature. The immediate benefit from this calibration was the explicit definition and estimation of some important metrics which otherwise would remain "hidden" in the plans. At this stage, such an exercise may help identify unrealistic assumptions and suggest readjustments in the plan.

Our next step was to investigate the impact of design changes. This required definition and quantification of: a measure for the magnitude of the changes (defined as the % of work needing to be re-done), a description of how these changes are introduced over time (defined by an S-shaped curve), and the direct effects of changes (e.g. on error generation). Assumptions were tested separately with simple scenarios. When changes are introduced the final schedule immediately needs to be re-assessed. This usually involves two levels of decision-making: agreement between the contractor and Client regarding the schedule milestone, and how management passes this on to the development staff. This policy was represented in the model by the schedule adjustment given to staff in proportion to the changes introduced - referred to as the "adjustment fraction". The paper shows the results of a 30% magnitude of total demanded changes being introduced rather smoothly. The model allowed us to assess the impacts of these changes and to test several schedule adjustment policies. The main purpose of this experimental investigation was to analyse

how cost and schedule should be traded against quality (measured in number of defects escaped).

The parameter "adjustment fraction" was changed to investigate how the schedule agreed with the development staff should be readjusted in the face of introduced changes. The output graphs in the paper show the behaviour produced when "schedule pressure" is put at its maximum as an attempt to cope with the changes and still achieve the original schedule. The impacts on the general behaviour are an immediate steep decrease in the amount of work completed while later schedule slippage cannot be avoided and cost exceeds budget. There is also a cut in man-power undertaking reviews; as a result of this poorer QA activity more defects escape. Although this restrictive scheduling policy does not avoid a delay, the results look attractive: a 14% over-run and a 13% over-expenditure (less than half of the proportion of changes) and the increase in the amount of defects detected was in the same proportion as the changes (about 30%), reflecting how staff implicitly assess QA under schedule pressure ("if I have to re-design 30% more then I should find 30% more defects"). However, the number of defects escaped suffered a steep increase of 72% above the "normal" level (i.e. the level in the base case). This result is a consequence of increased error generation and decreased QA effectiveness; the consequences of this will emerge later in the project when corrective actions are more difficult and expensive.

We used the model to test more tolerant schedule extension policies to reduce the negative effects of schedule pressure. Alleviating schedule pressure takes effect when the "adjustment fraction" exceeds 20%, where increasing the schedule adjustment to balance the changes results in later completion and more over-expenditure, but considerable decreases in defects escaping. Beyond 60%, the quality of the designs does not improve significantly while cost and schedule keep increasing, suggesting this is an appropriate level of schedule adjustment. Practically, this means that given the total duration agreed with the Client, if 30% of changes are introduced then a total of  $30\% \times 60\%$  days of schedule delay should be passed onto staff; in our example model this would give an actual 4% time over-run. In the case where the schedule milestone could not be extended the model would suggest an adjustment fraction around 40%. For the details the reader can refer to the numerical results shown in the paper.

In summary, analysing client behaviour is of growing importance in project management. Traditional techniques have proven inadequate. The holistic and integrative perspective of SD provides an appropriate approach, but needs a well defined framework integrating it within traditional tools. The authors have proposed such a framework (the PMIM) and described its use in practice in a large software project. In the example above we have used an SD model to investigate how schedule adjustments should be managed and negotiated.

*Acknowledgement* -- This work has been funded by JNICT - Comissão Permanente INVOTAN/NATO and Programa PRAXIS XXI, Portugal; and supported by BAeSEMA Ltd., United Kingdom.