Full citation: Connor, A.M., & MacDonell, S.G. (2005) Stochastic cost estimation and risk analysis in managing software projects, in Proceedings of the ISCA 14th International Conference on Intelligent and Adaptive Systems and Software Engineering (IASSE). Toronto, Canada, ISCA, pp.140-144.

STOCHASTIC COST ESTIMATION AND RISK ANALYSIS IN MANAGING SOFTWARE PROJECTS

Dr A.M. Connor, Professor S.G. MacDonell
SERL, School of Computing and Mathematical Sciences, Auckland University of Technology,
Private Bag 92006, Auckland 1142, New Zealand
andrew.connor@aut.ac.nz, stephen.macdonell@aut.ac.nz

Abstract

This paper presents an overview of the use of stochastic modelling as an approach to assessing the impact of uncertainty in effort and cost estimations in software projects. Uncertainty in input values is modelled using probability distributions and this uncertainty is propagated through the model to provide risk information using Monte Carlo simulation. Statistical analysis of the outputs of the simulation provides a means for identifying where the highest risk in the estimates lies. Understanding this risk, in terms of both its impact and its likelihood, allows activities to be undertaken to mitigate the risk prior to submitting a tender, therefore increasing the confidence with which the bid/no-bid decision is made.

1. INTRODUCTION

This paper outlines the development of a methodology for introducing stochastic modelling of cost and effort estimation into software development projects. Software development, more so than many other disciplines, is plagued by vague or shifting requirements and a lack of understanding regarding product complexity that often leads to projects being delivered late, over budget or not to requirements.

In this paper, uncertainty in cost and effort estimates is linked to a project work breakdown structure. The uncertain estimates can be utilised during the development of a tender submission for a software project to identify the main areas of risk in the submission. Cost estimates can be generated by a variety of approaches [1], many of which are deterministic. By analysing the uncertainty in estimates it is possible to identify the best utilisation of resource in the preparation of the tender documentation in order to minimise the risk in the bid submission.

2. SOFTWARE DESIGN LIFECYCLE

The software lifecycle is a term used to describe the various phases through which software travels. The software lifecycle runs from the point of conception to retirement. The phases include the traditional software development phases and the service management phases, combined into a single cycle. The phases of the software lifecycle are shown in Figure 1.

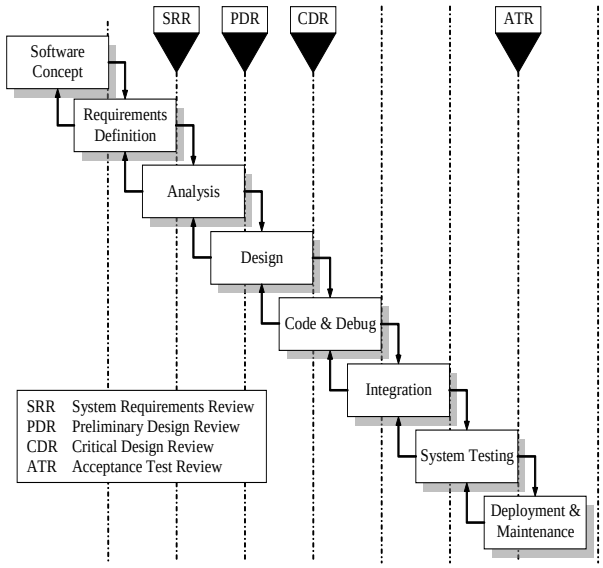

Figure 1: The software lifecycle

In this paper, the specifics of the software lifecycle and the tasks undertaken at each stage are not considered. Effort estimates are given at the phase level in the software lifecycle.

3. EFFORT AND COST ESTIMATION

In terms of new software development, it is not uncommon for cost estimation to be done at the project concept (tendering) stage and for this estimate to have a lifespan right through until the maintenance phase of the lifecycle, where the management model shifts towards bug fixes and enhancements which are treated as separate projects having their own cost/benefit analysis.

Cost estimates tend to be developed using a number of techniques, namely expert opinion, project analogy (use of historical data) or parametric models [1,2]. In some cases, organisations will use a Pert estimate to combine estimates from different sources into a three-point estimate, with minimum, maximum and "most likely" cost estimates.

Whist this approach goes some way to mitigating risk in the cost estimation, there are two avenues that can be explored to further reduce risk. The first of these is the use of probabilistic modelling to gain a more realistic estimate of "most likely" cost. By assigning cost estimates against work breakdown structure items it is possible to use a Monte-Carlo simulation to provide a more realistic (and informative) estimate than that provided by a Pert estimate.

The second approach is to recognise that as a project matures so does the data that can be used in the cost estimation. During the concept phase, cost estimates against WBS items may simply be a wide range of values. As project tasks are undertaken, not only can these estimates be refined but the nature of the estimate can also be reconsidered. For example, it may be more appropriate to use a normal distribution, a three point (triangular) estimate or indeed even a point value. As the project further matures, completed WBS items would tend to be represented as single point values, further reducing uncertainty in downstream tasks.

In this paper, effort/cost refinement is being proposed as a tool in the bid preparation stage. However, it is beneficial to undertake a refinement of estimates at several significant gates in the software lifecycle, i.e. the transition from one lifecycle phase to the next. Once confidence in the approach has been gained it should be possible to apply this at a finer level of detail, essentially producing a self-refining cost model.

3.1 Monte-Carlo Simulation

A Monte Carlo method is a technique that involves using random numbers and probability to solve problems using simulation. Monte Carlo simulation has been used in a variety of problem domains, including cost estimation [3].

Computer simulation utilises computer models to imitate real life or make predictions. With a simple deterministic model a certain number of input parameters and a few equations that use those inputs produce a set of outputs, or response variables. A deterministic model implies that the same results will be achieved no matter how many times the model is re-evaluated.

Monte Carlo simulation is a method for iteratively evaluating a deterministic model using sets of random numbers as inputs. This method is often used when the model is complex, nonlinear, or involves more than just a few uncertain parameters. By using random inputs, the deterministic model is essentially transformed into a stochastic model.

The Monte Carlo method is just one of many methods for analysing uncertainty propagation, where the goal is to determine how random variation, lack of knowledge, or error affects the sensitivity, performance, or reliability of the system that is being modelled. Monte Carlo simulation is categorised as a sampling method because the inputs are randomly generated from probability distributions to simulate the process of sampling from an actual population. A distribution for the inputs that closely matches real data or best represents our current state of knowledge should be selected. The data generated from the simulation can be represented as probability distributions (or histograms) or converted to error bars, reliability predictions, tolerance zones, and confidence intervals.

The steps in Monte Carlo simulation corresponding to the uncertainty propagation are fairly simple, and can be easily implemented for simple models:

Step 1: Create a parametric model, $y = f(x_1, x_2, ..., x_q)$.
Step 2: Generate a set of random inputs, $xi_1, xi_2, ..., xi_q$.
Step 3: Evaluate the model and store the results as y_i.
Step 4: Repeat steps 2 and 3 for i = 1 to n.
Step 5: Analyze the results using histograms, summary statistics and confidence intervals

Monte Carlo simulation has been applied to modelling of uncertainty in cost estimations in a product breakdown structure [4] where historical project information is used to define the input probability distributions. This paper adopts a similar approach to the work breakdown structure representing the full life of a software project.

4. APPLICATION TO SOFTWARE LIFECYCLE

To illustrate the application of Monte-Carlo simulation to software project planning, a simple Excel tool has been developed that conducts a simulation for some basic effort

estimation data related to the software lifecycle. This can be applied in many ways throughout the software design process, but a specific example related to reducing risk in a tender submission is given.

A work breakdown structure has been generated for a generic software project, with tasks defined under the main lifecycle phases of;

- Planning and Bid Preparation
- Requirements Definition
- Analysis and Design
- Code and Debug
- Integrate and Test
- Deployment and Acceptance

The work packages in the work breakdown structure are in no way related to a specific design process, therefore actual day to day activities may be undertaken to satisfy more than one work package at any time. For example, in the bid preparation and planning phase, activities that support project scoping, the development of a project plan and a cost estimate will inevitably be conducted in parallel as there is co-dependence between tasks in each work package. However, in terms of the software lifecycle, the main reviews tend to be "gates" that limit a return to previous activities. For example, once the customer has approved the baseline design at the Critical Design Review then downstream activities will not include design unless it is at the customer request, which then is clearly a contractual change.

Each work package needs to be assigned an effort (and cost) estimate, which can be developed using any traditional method. Each estimate can be defined using different probability distributions, namely a single value, a normal distribution, a triangular distribution or a uniform (rectangular) distribution. Future work will link the input distributions to a historical database of past projects giving an added level of refinement. At present, the choice of distribution and corresponding parameters should represent the confidence in the estimate itself.

For a given set of inputs, the tool runs a Monte Carlo simulation to predict a range of scenarios for the project. The data generated by the simulation can be used to analyse each phase of the project, or the project as a whole. The output includes a histogram of the likely duration of this particular project phase and a summary of statistics that relate to the distribution. These statistics provide data that can be used to inform the development project. The calculated statistics include the mean, the median, the standard deviation, the interquartile range, the skewness and the kurtosis of the distribution.

The skewness is a measure of symmetry, or more precisely, the lack of symmetry. Positive values for the skewness indicate data that are skewed right, indicating a likelihood that the project phase has the potential to overrun.

Kurtosis is a measure of whether the data are peaked or flat relative to a normal distribution. A negative kurtosis indicates a flatter distribution indicating a greater potential range in actual phase duration.

The mean, median, standard deviation and interquartile range also provide valuable information to determine where the most uncertain costs estimates are located.

4.1 Minimising Risk in the Bid/No-Bid Decision

The prototype tool can be used in the initial response to an invitation to tender in order to gauge the risk in the proposed project and as such inform the bid/no-bid decision. Figure 2 shows a generic process for tendering activities [3]. In this application of the tool, it is assumed that minimisation of risk is conducted in the development activities.

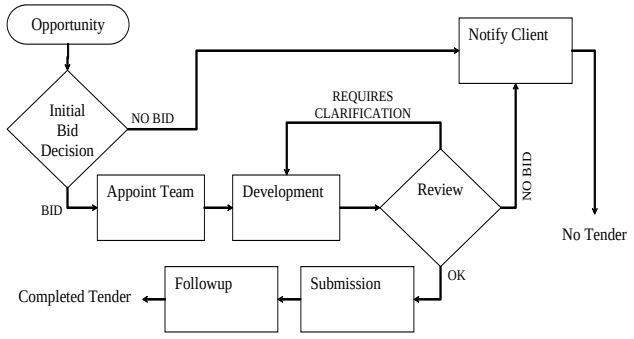

Figure 2: Generic tender process model

The development activities can be decomposed into a specific lower level model, in this case defining a process based on the work packages in the work breakdown structure. This lower level model is shown in Figure 3.

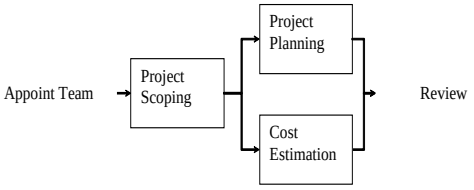

Figure 3: Development sub-model

In this model, a scope for the project is determined and a project plan and cost estimate are produced using additional lower level activities. These are not defined, but in the cost estimation area could include tasks such as "Obtain Expert Opinion", "Use Parametric Model" and "Analyse Historical Data". Project scoping activities

would relate to initial interpretation of customer requirements and gauging whether the technical capability and resources are available to enable the requirements to be met. Iteration around the development activities occurs after the review of the data generated as illustrated in the generic top level model of tendering activities.
A crucial input into this review is the risk in the project and this can be gauged from the cost/effort estimates developed by applying the prototype tool. Running the Monte Carlo simulation for this data, assuming that the total effort for the project is the sum of the effort required for each work package, provides a great deal of data with regards not just the project but each phase of the project separately.

Figure 4 shows results of the simulation for 5000 trials for the project. The histogram shows the predicted range of project duration between a minimum of 168 days and a maximum of 237 days.

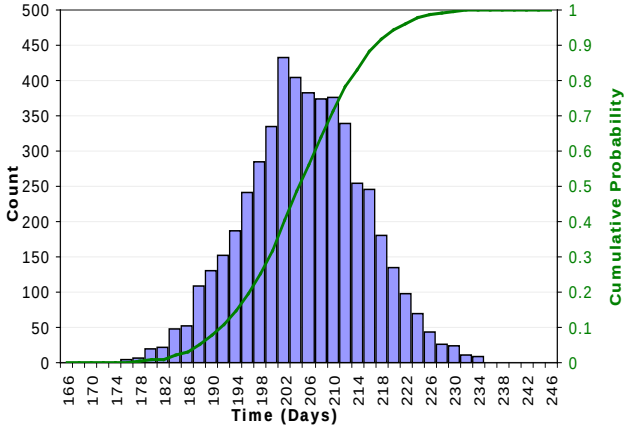

Figure 4: Histogram of simulation results

An indication of where the risks in the total project lie can be obtained by looking at the statistics associated with each individual phase of the project, particularly the Kurtosis, Skewness, Standard Deviation and the Interquartile Range. These statistics describe the shape and the spread of the distribution. This data can be plotted for each phase of the project to allow comparison to be made. For example, Figure 5 plots the Kurtosis of each phase such that the phase that is furthest away from the centre has the greatest risk.

Project phases which exhibit a negative Kurtosis value have a more broad shape than a normal distribution, therefore the most negative value indicates a distribution that is tending towards being wide and flat. Using this metric, a refinement in the estimate for the Deploy & Accept phase could result in an increased confidence in the overall project by producing an overall distribution with a more pronounced "spike".

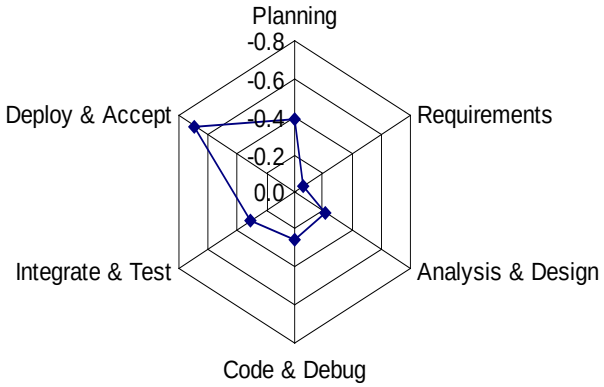

Figure 5: Plot of Kurtosis for each phase

Figure 6 shows the histogram of results for just this project phase. Examining this distribution shows that it is tending to be "wider and flatter" than a normal distribution. The statistics for this distribution are given in Table 1. These statistics can be analysed to confirm that the Deploy & Accept phase is likely to give rise to some risk in the project effort estimates.

Distribution Statistic	Value
Min	12
Max	31
Mean	22
Median	22
Variance	11.8
Standard Deviation	3.4
0.25 Quartile	19
0.75 Quartile	25
Interquartile Range	6
Skewness	-0.019
Kurtosis	-0.69

Table 1: Statistics for Deploy & Accept simulation results

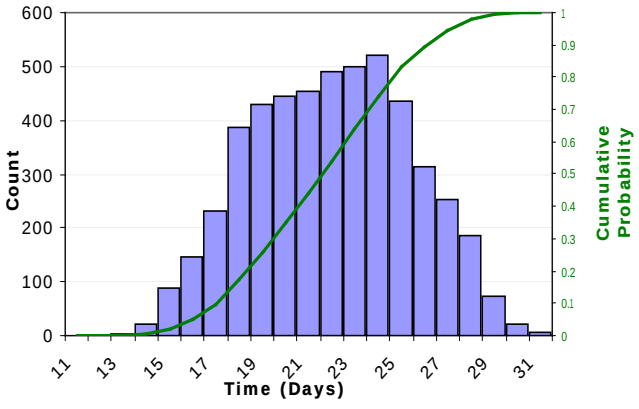

Figure 6: Histogram of simulation results for Deploy & Accept phase

Figure 7 plots the Skewness of each phase such that the phase that is furthest away from the centre has the greatest risk of overrun.

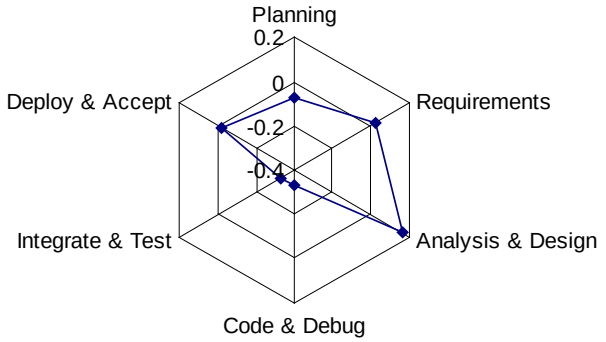

Figure 7: Plot of Skewness for each phase

Project phases which exhibit a positive Skewness value have a larger right tail than left tail, indicating that the phase is more likely to overrun than be completed early. Using this metric, a refinement in the estimate for the Analysis & Design phase could result in an increased confidence in the overall project by producing an overall distribution that is more centrally distributed or has a larger left tail, indicating likelihood to underrun. In managing projects, it is as important to identify underrun as to identify potential overruns. Underruns provide a degree of slack to compensate for overrun in either the project or the wider portfolio and can also be used to shift resource between tasks or projects.

Figure 8 shows the distribution of results for just the Analysis and Design phase of the project. Examining this distribution shows that it is non-symmetric, with the right tail longer than left around the mean value of 51 days.

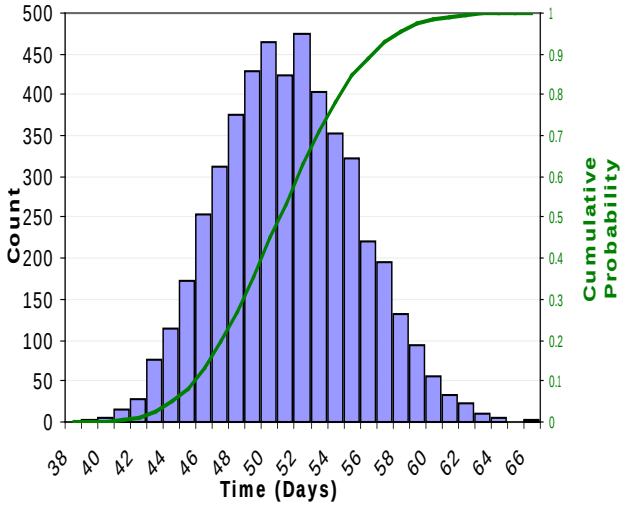

Figure 8: Histogram of simulation results for Analysis & Design phase

If the cumulative estimates for the total project length are likely to exceed either the time allowed for the project by the client, or that the costs associated with the effort will become too high, then the question arises as to how best to allocate resource in the bid development activities so as to reduce this risk, providing greater confidence in the tender submission. By using the statistical information for each phase estimation, it is possible to determine which project tasks can be undertaken in the bid and planning phase so as to best reduce the risk in the project.

For the example given, targeting activities to reduce uncertainty in both the Analysis & Design and the Deploy & Accept phases are likely to produce more certain cost and effort estimates to support the bid/no-bid decision. For instance, in the Analysis & Design phase such activities might include more extensive use of prototype solutions, or the secondment of a client representative onto the wider development team.

5. CONCLUSIONS

This paper has presented a methodology for tracking the uncertainty in project estimates and has shown how modelling this uncertainty using probability distributions can inform both the submission of bids for projects and the subsequent project management itself.

Throughout this paper, reference has been made to the ability to use statistical information with regards the uncertainty propagation to inform the ordering and priority of project tasks. It is a challenge for future work to explore this concept further by developing more detailed process models and defining dependencies between tasks and how tasks relate to the underlying data that can be used to drive the dynamic ordering of the process.

Even without the enhancement of process management, the approach detailed in this paper has considerable benefit. This is particularly true if project estimates are kept in a suitable database that can be used to inform future cost estimates for other projects. Organisational learning over time should see the distributions shift and reshape as more certainty and confidence is gained. Future work will link the simulation tool to such a database to allow historical data to be used to generate input probability distributions.

6. REFERENCES

[1] Briand, L.C. *et. al.*, "Assessment and comparison of common software cost estimation modeling techniques", Proceedings of the International Conference on Software Engineering, pp 313-323, 1999

[2] Heemstra , F.J. “Software cost estimation models”, Proceedings of the Jerusalem Conference on Information Technology, pp 286-297, 1990

[3] Vrijland, M. S. A. *et. al*., “Monte Carlo Method in Cost Estimations”, Norwegian Assoc of Cost & Planning Engineering,, pp A. 2. 1-A. 2. 7, 1986

[4] Crossland, R., Sims Williams, J.H. & McMahon, C.A., “An object-oriented modeling framework for representing uncertainty in early variant design”, Research in Engineering Design, Vol 14, pp 173-183, 2003

[5] Barr, G., Burgess, S.G., Connor, A.M. and Clarkson, P.J. “Tendering for engineering contracts” *Proceedings of Design for Excellence: Engineering Design Conference (EDC 2000)*, pp 499-506, 2000