Full citation: Connor, A.M., & MacDonell, S.G. (2006) Using historical data in stochastic estimation of software project duration, in Proceedings of the 19th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ). Wellington, New Zealand, NACCQ, pp.53-59.

Using historical data in stochastic estimation of software project duration

Dr Andy Connor, Professor Stephen G. MacDonell
SERL, School of Computing and Mathematical Sciences, Auckland University of Technology,
Private Bag 92006, Auckland 1142, New Zealand
aconnor@aut.ac.nz, stephen.macdonell@aut.ac.nz

Abstract

This paper presents a framework for the representation of uncertainty in the estimates used to predict the duration of software design projects. The modelling framework utilises Monte Carlo simulation to compute the propagation of uncertainty in estimates towards the total project uncertainty and therefore gives a project manager the means to make informed decisions throughout the project life. The framework also provides a mechanism for accumulating project knowledge through the use of a historical database, allowing effort estimates to be informed by, or indeed based upon, the outcome of previous projects.

Keywords: Software project management, cost and effort estimation.

1. INTRODUCTION

Estimation of cost and duration for software development activities is one of the most difficult aspects of software project management. The project manager often has the need to make estimations of effort and cost against which a project's success will be judged with incomplete data available. This is particularly true for projects in competitive bidding scenarios where estimates need to be made during the bid phase. A high bid could result in losing the contract or a low bid could result in a major loss. From an estimate, the management often decides whether to proceed with the bid for the project. Industry has a need for accurate estimates of effort and size at a very early stage in a project. Methods for improving the reliability of estimation without greatly increasing the overhead will prepare project managers to cope with the challenges in this area in the next decade.

This paper outlines the development of a methodology for introducing probabilistic modelling for the estimation of duration for 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 either late, over budget or not to requirements. Software cost estimates made early in the software development process are often based on wrong or incomplete requirements.

In this paper, uncertainties in effort estimates are linked to a project work breakdown structure. The degree of uncertainty is modelled by applying different probability distributions to the estimate. The tool detailed in this paper allows this uncertainty to be propagated through the work breakdown structure through the use of Monte-Carlo simulation. This provides an indication of the range of likely outcomes, not just a single estimate. The project risk management process can therefore be informed by pessimistic, optimistic and realistic estimates.

A key feature of the tool is its ability to capture and utilise project duration data for use in providing more accurate estimates for future projects. The use of such corporate knowledge is particularly appropriate for organisations that produce variants of a product or undertake very similar projects. However, the tool does not mandate the use of historical data therefore allowing it to be applied to both typical and atypical projects. For atypical projects, the underlying work breakdown structure can be modified to introduce new tasks for which historical data is not available and still produce a meaningful estimate. The use of historical data is a significant advancement on previous work (Connor & MacDonell, 2005). The tool is currently prototyped in Excel using a freely available add-in, Simular (Machain, 2005) to conduct the simulation.

2. ESTIMATING SOFTWARE PROJECT

Estimation of costs and effort requirements continues to be a weak link in software project management. In terms of new software development, it is not uncommon for effort or cost estimation to be done at the project concept (tendering) stage and for this single 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.

Estimates tend to be developed using a number of techniques, namely expert opinion, project analogy

(use of historical data) or parametric models (Briand et al, 1999; Heemstra, 1990). 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.

While 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 work breakdown structure 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 work breakdown structure items would tend to be represented as single point values, further reducing uncertainty in downstream tasks.

The aim of this research is to develop a simple approach for cost and effort estimation that does not require the overhead of more formal approaches that include COCOMO-II (Boehm et al, 2000). Our current work is focused on the investigation, design, development and testing of the proposed methodology. The research involves capturing new methods in a platform that will prepare software project managers for the future challenges of the future, where project timescales will shorten and the need for more refined understanding of software estimation will increase.

The aim of the proposed platform is not to replace existing methods, but augment them by providing additional tools to enhance future capability to make accurate estimations. Monte-Carlo simulation provides a suitable means of introducing a powerful yet simple to use stochastic element to the cost estimation of software projects and this method has been adopted in this work.

A Monte-Carlo method is a technique that involves using random numbers and probability to solve problems using simulation. The approach has been used in a variety of problem domains, including cost estimation (Vrijland et al, 1986; Crossland et al, 2003).

Computer simulation utilises 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, where the same results will be achieved every time 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 transformed into a stochastic model. The Monte Carlo method is just one method 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 modelled system.

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, statistics and confidence intervals as illustrated in Figure 1.

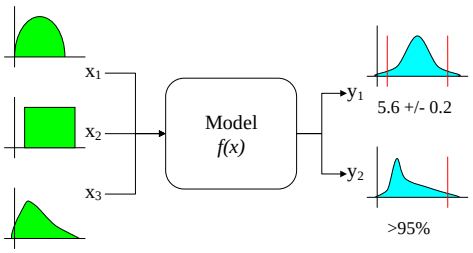

Figure 1: Schematic showing the principle of stochastic uncertainty propagation.

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

1. Create a parametric model, $y = f(x_1, x_2, ..., x_q)$
2. Generate a set of random inputs, $xi_1, xi_2, ..., xi_q$
3. Evaluate the model and store the results as y_i
4. Repeat steps 2 and 3 for i = 1 to n
5. Analyse 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 (Crossland et al, 2003) 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.

3. OVERVIEW OF THE PROTOTYPE TOOL

The research described in this paper is currently at the proof of concept stage, and as a result a simple prototype tool has been produced.This tool supports the initial estimation of effort required to undertake

tasks in a project work breakdown structure as well as on-going refinements. It also supports the recording of actual durations of a project on its completion to allow this data to be used in future projects.

3.1 Initial Estimation

The initial estimation of project duration is conducted by applying probability distributions to nominal tasks in the project work breakdown structure, as illustrated in Figure 2.

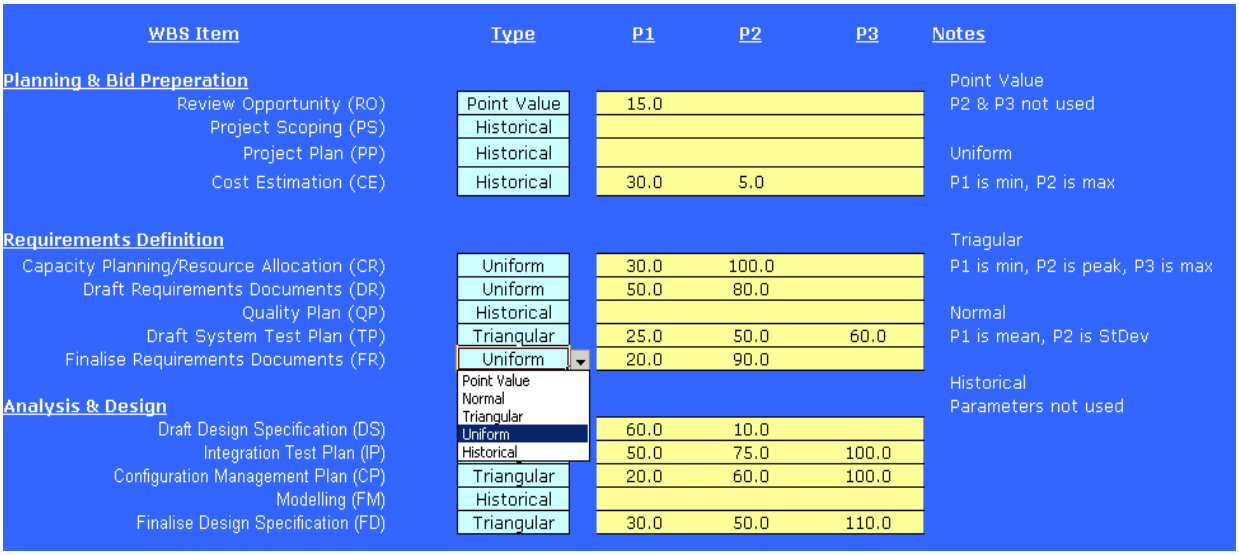

Figure 2: Data input screen.

From Figure 2, it can be seen that only four types of distribution (Point Value, Normal, Triangular and Uniform) may be selected manually, with the fifth option to be to determine the distribution from historical data. When this fifth option is selected, a much wider range of potential distributions will be tested against data values and a choice made as to which type of distribution best approximates the real data as discuss in section 3.3.

Once the input values have been set to their initial values, the Monte-Carlo simulation is initiated, typically for between 5000 and 10000 evaluations. In each evaluation, a sample is taken for each input distribution and the output determined. Following completion of the simulation, the results may be viewed with in the tool. Figure 3 shows the raw results and the statistics for the total project. The total project is the cumulative result of the six main project phases, Planning & Bid Preperation, Requirements Definition, Analysis & Design, Coding & Debugging, Integation and Testing and finally Deployment and Acceptance.

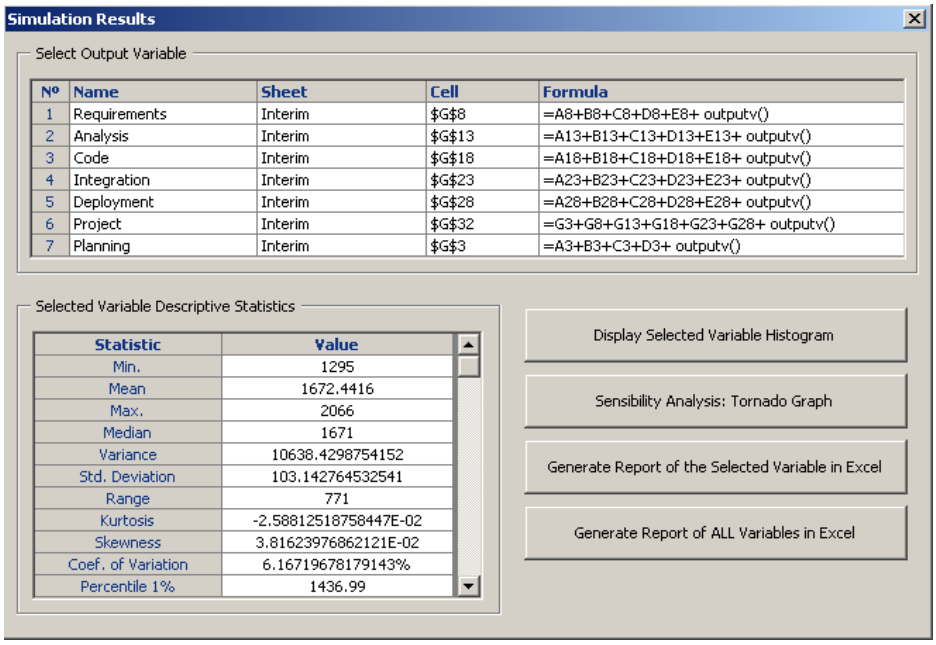

Figure 3: Raw results.

The key statistics for considering the total project are the mean, the standard deviation and the interquartile range. Kurtosis and skewness are also important to consider but will be discussed in interpreting results from individual phases. Analysis of these statistics indicates that the simulation has predicted a wide range of outcomes that may constitute a project risk. In addition to the statistics, the results for each output may be displayed graphically as a distribution of expected outcome. Figure 4 shows the expected outcome for the total project following completion of a simulation.

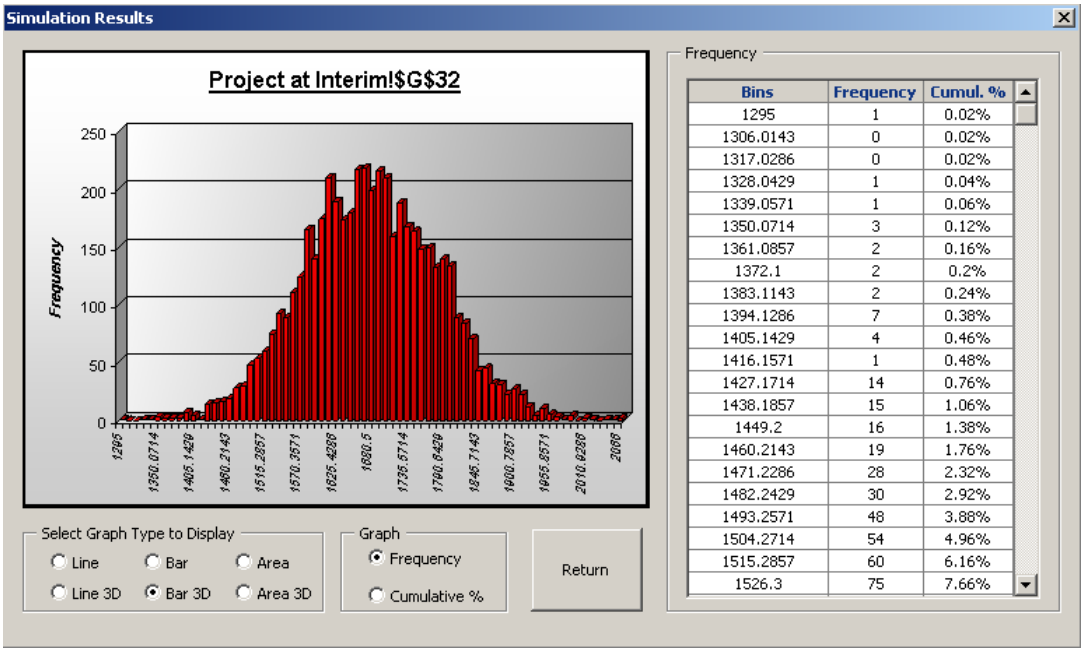

Figure 4: Project duration distribution.

While an indication of likely duration for the entire project is useful, a more granular analysis could be even more informative. As demonstrated in previous work (Connor & MacDonell, 2005), the contribution of risk of each phase of the project to the total duration may be gauged by analysing each phase. An indication of the risks in the total project 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.

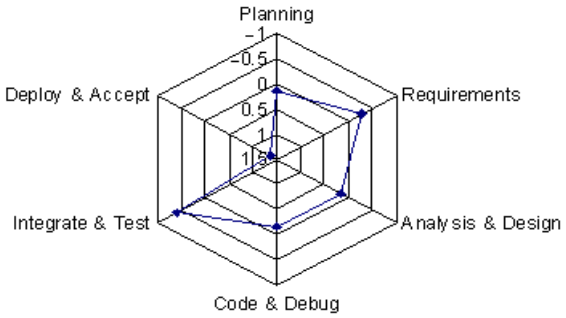

Figure 5: Plot of Kurtosis for each phase.

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. The nature of the distribution can be confirmed by plotting the results for this phase as in Figure 6.

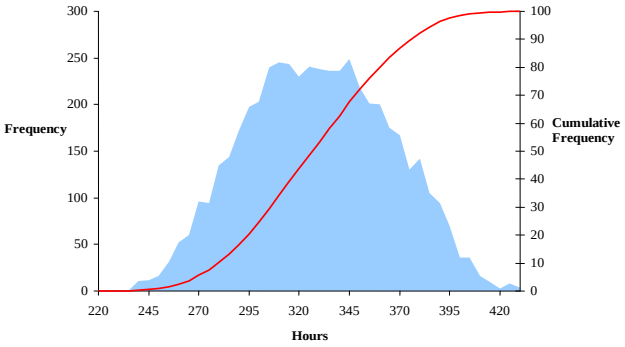

Figure 6: Distribution of results for Integration and Test phase.

Using this metric, a refinement in the estimate for the Integrate & Test phase could result in an increased confidence in the overall project by producing an overall distribution with a more pronounced “spike”, essentially implying a reduced level of risk.

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.

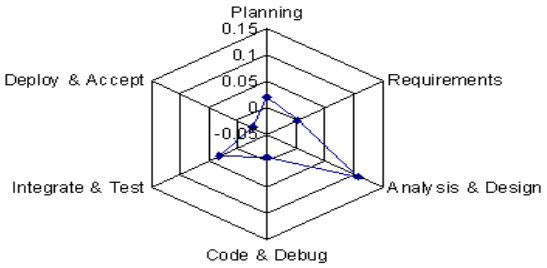

Figure 7: Plot of Skewness for each phase.

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 under run. In managing projects, it is as important to identify under run as to identify potential overruns. Under runs 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.

3.2 Estimate Refinement during Project Life

In addition to the use of the tool to provide an initial estimate for a project, it has significant benefit in being used throughout the project life. To demonstrate this, the input settings of the example used above have been modified so as to represent a project in mid-life. Activities that have occurred in the past and are completed have been assigned point values. Activities that are towards the tail end of the project lifecycle can have their estimates refined as more knowledge is available on which to base the estimation. In this example, the project is assumed to be at the end of the requirements definition phase, so all activities in the planning and requirements phases have been set to point values. The activities in the Analysis & Design phase have been revised to be less conservative and all other activity estimates have been untouched. Even these few changes have a significant effect on the overall project estimate as can be seen in Figure 8.

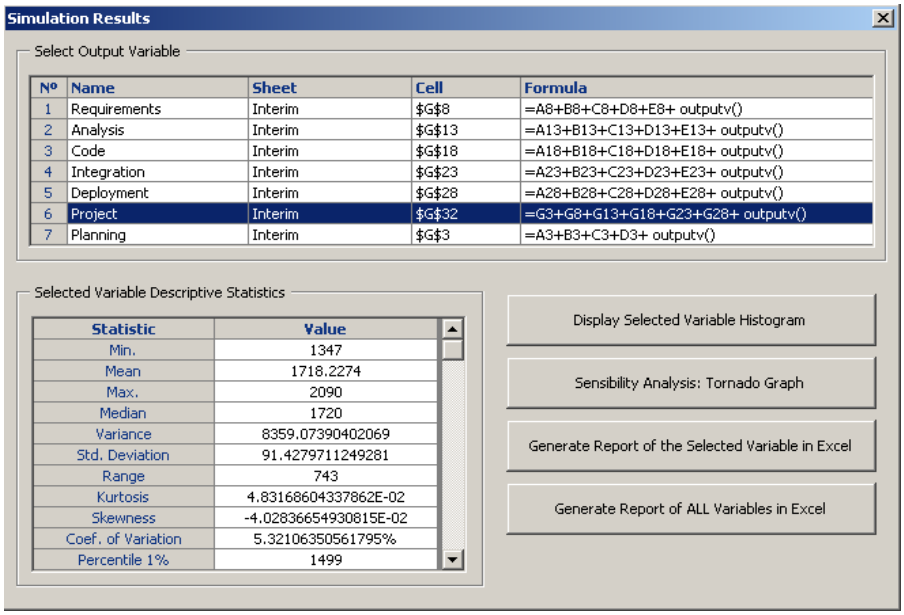

Figure 8: Revised simulation results.

Whilst the mean estimate has increased, the standard deviation has reduced and, more significantly, both the kurtosis and the interquartile range have more favourable values. This shows that even a small change in confidence in the input parameters can result in a more realistic set of output distributions.

3.3 Updating Historical Cost Database

The use of a historical database provides a powerful tool for learning from previous experience and using this knowledge to inform future project estimates. The current implementation of the tool uses a simple means to capture and utilise historical data.

Historical data is captured within the Excel tool, simply as a list of actual effort required for each project broken down by project phase. The historical database is limited to typical projects, where typical is defined by the nature and scope such that they are within the expertise of the developers. The inclusion of atypical projects in the database does actually introduce an element of risk in the project estimates.

When new data is added to the database, it is necessary to refit a distribution to the data using the inbuilt functions of SimulAr. Figure 9 shows the original data set used for the Draft Requirements activity along with the best fit distribution. In this instance, the best distribution fit is achieved by using a logistical distribution and the quality of the fit is poor, as shown by the difference between the lines indicating the real data and theoretical distribution.

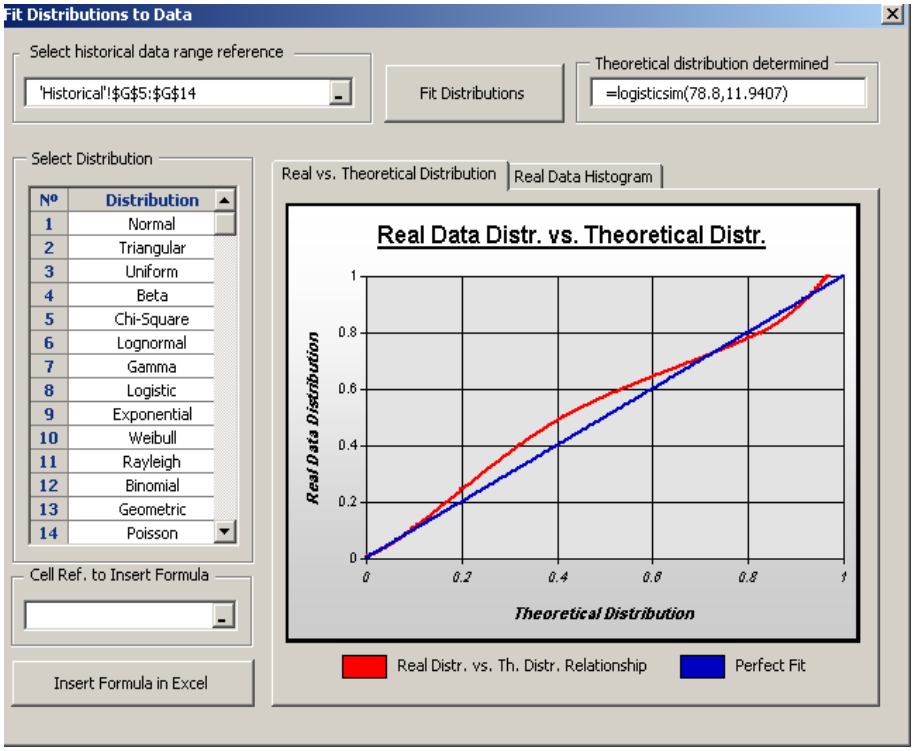

Figure 9: Original fit of distribution to data.

Both the type and the value for the approximate distribution must be revised when new data is added. Even adding just one more entry into the database allows a higher quality of fit to be obtained, as illustrated in Figure 10.

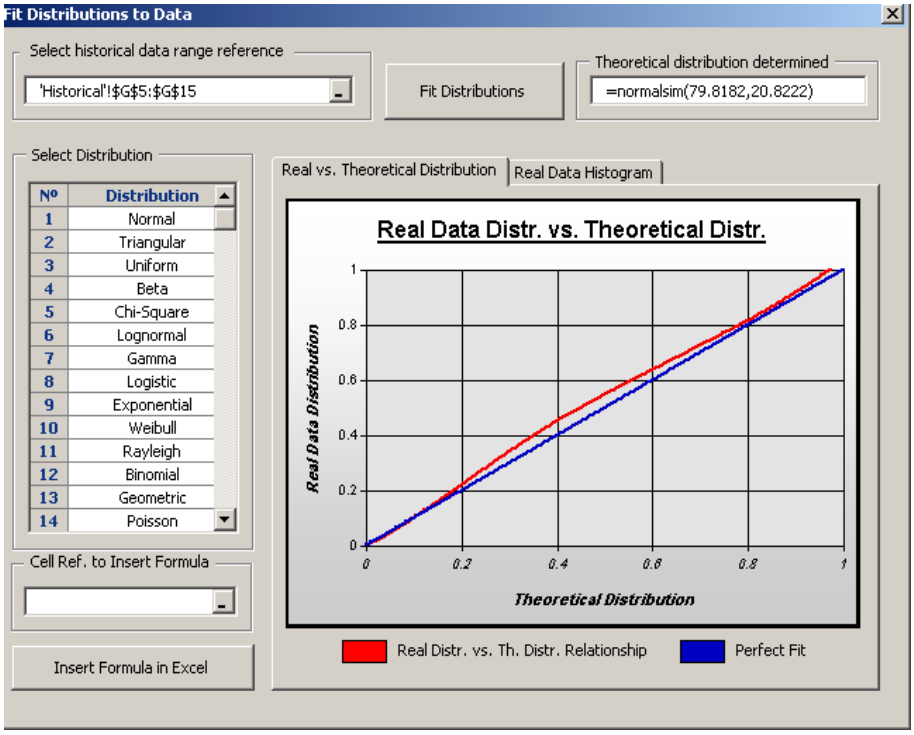

Figure 10: Revised fit of distribution to data.

4. CONCLUSIONS

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

steps required for establishing initial software duration estimates and then tracking and refining those estimates throughout the life of the project. Establishment of this process early in the life cycle will result in greater accuracy and credibility of estimates and a clearer understanding of the factors that influence software development costs.

By linking estimates to a historical database of real project data, the approach has the capability to make accurate estimates early in the lifecycle with relatively low risk, despite the fact that the project requirements may be incomplete or inaccurate. The data in the historical data base is the actual duration of previous projects, for which estimates would have been made in similar circumstances when requirements were incomplete. For each and every project, corporate knowledge can be enhanced by comparing estimates at intervals through out the lifecycle with the final cost or duration data at the end of the project.

The overall approach is simplistic in its nature and can therefore be utilised by a wide range of businesses to further understand their development processes. Adopting the tool will improve risk management approaches for software projects. It is thought that the approach is particularly applicable to projects conducted using an agile development methodology and future work will clarify the benefits of adoption with this focus.

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 understanding whether "rich" data can be captured to provide insight into relationships and issues not immediately obvious.

5. REFERENCES

1. Boehm, B.W. et al (2000) Software cost estimation with COCOMO II, Prentice Hall

2. Briand, L.C. et al. (1999) Assessment and comparison of common software cost estimation modeling techniques, Proceedings of the International Conference on Software Engineering, pp. 313-323

3. Connor, A.M. & MacDonell, S.G., (2005) Stochastic cost estimation and risk analysis in managing software projects", Proceedings of the ISCA 14th International Conference on Adaptive Systems and Software Engineering (IASSE-2005), pp 400-404 [CD-ROM]

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

5. Heemstra, F.J. (1990) Software cost estimation models, Proceedings of the Jerusalem Conference on Information Technology, pp. 286-297

6. Machain, L., Simular add-in for Excel, Retrieved July 8th 2005, from http://www.simularsoft.com.ar/

7. Vrijland, M. S. A. et al. (1986) Monte Carlo Method in Cost Estimations, Norwegian Assoc. of Cost & Planning Engineering, pp. A. 2. 1-A. 2. 7