A Simple Estimate of the Cost of Software Project Failures and the Breakeven Effectiveness of Project Risk Management

Stephen P. Masticola
Siemens Corporate Research
755 College Road East, Princeton NJ 08540
Stephen.masticola@siemens.com

Abstract

Based on easily obtained data, a rough (but simple) estimate of the cost to an enterprise of software project failures can be formed. Using similar analysis, it is possible to estimate the effectiveness that a software quality improvement project will need to have upon the project failure rate in order to break even economically. This provides an informal calculation that can be used to cost-justify either a more complete investigation or risk management.

1. Introduction

The Standish Group's "CHAOS Report," [1] a widely respected survey of software projects in industry and government, estimated that, in the year 2004, only 29% of software projects in large enterprises succeeded (i.e., produced acceptable results that were delivered close to on-time and on-budget) . (See Figure 1.) 53% were "challenged" (significantly over budget and schedule), and 18% failed to deliver any usable result. The projects that are in trouble have an average budget overrun of 56%. This represents a serious and chronic risk-control problem. This paper gives a simple way for project managers and others to relate the Standish Report data to their own enterprise.

1.1. Running Example: XYZ Corporation

Suppose that the (fictitious) XYZ Corporation spends about $100 million per year on software development. Given these figures, it can be estimated that budget overruns on troubled projects would cost XYZ over $28.4 million per year. This doesn't include lost revenue or other indirect costs caused by troubled projects. Section 2 shows the basis for these estimates.

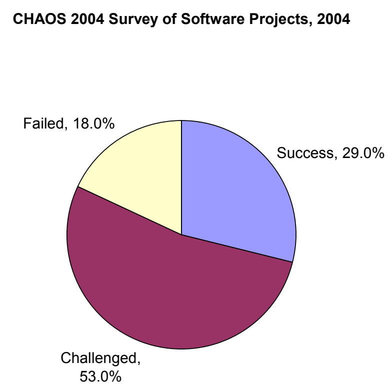

Figure 1: CHAOS database survey results of 50,000 completed commercial and government software projects for the year 2004.

For the purpose of this paper, I will define *risk management* to mean any activity that is intended to help managers to understand or reduce the risk of serious budget overruns in software development projects. Risk management thus includes a very wide range of activities. For example, model-driven approaches, automated defect detection tools, improved inspection technology, better process, and, indeed, economic analysis itself can all be considered as risk management measures. (This is not an all-encompassing definition because it excludes the management of other kinds of risk, such as the risk of losing customers due to late delivery or low quality.)

Most risk management activities have an associated cost, though, but it is still easy to justify this cost. For example, if a given risk management measure costs 5% of the budget of a software project, then the risk management only has to increase the probability of success of that project by 8.9 percentage points to justify its cost, regardless of the size of the project. This is the *breakeven effectiveness* for the

First International Workshop on the Economics of Software and Computation (ESC'07)
0-7695-2955-0/07 $20.00 © 2007 IEEE

improvement. This paper also shows how to calculate breakeven effectiveness under different assumptions. Section 3 shows these calculations.

Figure 2 illustrates this point for a hypothetical project with a \$10 million initial budget. Effectiveness to savings lines are drawn for risk management addenda of 0%, 1%, 2%, 5%, and 10% of the initial project budget. If, for example, the risk management effort historically increases the probability of a project's success by 10 percentage points and is estimated to cost 2% of the original budget (thin solid black line), it can save the project \$360,000 in overrun costs, on the average.

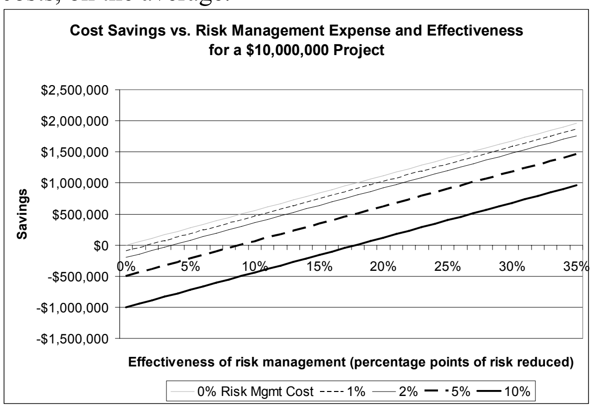

Figure 2: Cost savings from avoided overruns vs. cost and effectiveness of risk management for an example \$10 million project.

1.2 Purpose and scope of this paper

The purpose of this paper is to document the detailed quantitative economic justification for the above conclusions. By doing this, it is hoped that software project managers and senior executives of software companies can gain a concrete insight into the economics of troubled software projects and their remediation measures.

In particular, we present an estimation method that, while not sophisticated, is very easy and inexpensive to measure and compute. It may hopefully also serve as a powerful motivator for more sophisticated metrics and for risk management activities.

2. Estimating cost overruns

Let E_E be the expected expenditure of the enterprise on software development. The Standish Group conducts an ongoing survey of software project results in commerce, industry, and government. From the Standish Group's 2004 CHAOS report [1], the probabilities of a commercial or government software project's resolution are as follows:

- Standish Resolution Type 1 (complete success): 29.0%.
- Standish Resolution Type 2 (significant delays or cost overruns): 53.0%.
- Standish Resolution Type 3 (cancelled before delivering results): 18.0%.

Let P_f be the probability that a project has a type 2 or 3 termination. P_f is thus estimated at 71%, based on the Standish Group's 2001 figures.

The Standish Report also estimates the budget overrun of projects with type 2 or 3 termination to the original budget. Let R_O be this ratio. In 2004, the Standish Group estimated that R_O was 56%.

Let R_f be the ratio of the cost of projects with Standish Resolution Type 2 or 3 to the original budget. Thus:

$$R_f = 1 + R_O \quad [1]$$

Thus, R_f was 156% in 2004.

Let B be the initial budget of all software projects in XYZ Corporation. Thus we have:

$$E_E = P_f B R_f + (1-P_f)B \quad [2]$$

Solving for B:

$$B = E_E/(P_f(R_f - 1) + 1)$$

$$= E_E/(P_f R_o + 1) \quad [3]$$

Thus, for the XYZ Corporation, whose E_E (actual amount spent on software development) is \$100 million for the year 2005, B is estimated at about \$71.5 million, based on the Standish Group's 2004 data.

If B is normalized to 1, then *[2]* gives E_E as 1.398. Reducing P_f by one percentage point, E_E drops to 1.392. Thus, for every percentage point we can reduce P_f, we can save 0.6% of the original project budget, assuming that there is no cost to reduce P_f. Of course, there usually will be a cost for quality improvements, so it is important to understand how effective the improvements have to be in reducing P_f to pay for themselves. This is called the *breakeven effectiveness* of the risk management, and is discussed in Section 3.

Based on this, the annual amount of budget overruns E_O for Standish Resolution Type 2 or 3 projects in XYZ Corporation is estimated at:

$$E_O = E_E - B \quad [4]$$

This cost of overruns is over \$28.4 million for the year 2005 over all projects in XYZ Corporation.

2.1. Discussion

By a straightforward calculation, every one percent that a risk management activity can reduce the probability of failure P_f saves the company $0.01E_O/P_f$. With the values above, the savings from a one-percentage-point risk management company-wide in XYZ Corporation would be slightly over \$400,000.

Several words of caution are in order about the estimated budget overruns. First, the Standish Report data reflect only an average across the software development field, so any specific enterprise is probably different and would need to be specifically measured in the same way that the Standish Group did in order to get a specific picture.

Second, it is entirely possible that some percentage of software project failures in any company are caused by inaccurate estimation. An inaccurate initial estimation can lead to inadequate allocations of budget, or time, or both to the project, which can then cause it to fail.

Suppose that we had the ability to estimate projects perfectly (i.e., that $E_E = B$.) This would have two possible effects that would perturb E_E and B. Without doubt, XYZ Corporation would simply not attempt to begin a significant number of its Standish Resolution Type 2 and 3 projects if they had been initially estimated accurately. With better estimates, XYZ might also have spent more on the projects that it did start to assure that they completed with Standish Resolution Type 1 (success.)

There is not sufficient data available to break the cause of project failures in any enterprise down between errors in estimation versus errors in execution. If such data existed, it would undoubtedly be of great interest, but it is beyond the scope of this paper.

Finally, the act of estimating the cost of overruns, E_O, implicitly assumes that the effect of overrunning the budget is limited to the overrun itself. In practice, however, the impact of serious overruns can go far beyond this in lost revenue and lost opportunity for future profit. If budget overruns are serious enough, they can also permanently damage the business by making entire business units unprofitable. Huang and Boehm attempt to measure these effects in [4].

3. Breakeven effectiveness of risk management activities

The remainder of this paper assumes that process changes are available to improve the probability of success of the projects, and that the estimation of the project budget is exactly as accurate as it is now (i.e., that with B as the total estimated cost of all projects, the probability of failure is P_f.) These process changes may come with some added expense. We express this expense of risk management, R_J as a fraction of B. Spending R_JB on risk management activities reduces the average probability of failure P_f by a fraction, ΔP_f. ΔP_f is in units of the probability of failure (or success) of the project, and hence is expressed as percentage points in this paper. (ΔP_f may be derived from historical data or by other means. (Estimating ΔP_f through historical data, modeling, or other techniques is a topic for future work.)

We now want to know the *breakeven effectiveness* of the risk management activity. This is the value of ΔP_f at which the savings from additional project successes pays for the cost of the risk management activity.

Keeping E_E constant, we have:

$$E_E = R_JB + (P_f - \Delta P_f)BR_f + (1-(P_f - \Delta P_f))B \quad [5]$$

Substituting *[2]* for E_E in *[5]* and solving for ΔP_f, we have:

$$\Delta P_f = R_J/R_O \quad [6]$$

3.1. Discussion

An example helps explain this. Assume that, under current conditions, an additional 5% of the estimated budget is spent on risk management activities. If the risk management activities increase the chance of success by 8.9 percentage points, they have paid for themselves, regardless of the project size.

Thus, effective risk management activities can pay for themselves quite readily. Key in developing effective risk management activities is measuring their effectiveness in economic terms. The act of measuring the process is, in and of itself, a risk management activity, and (as long as it can be done in harmony with the project) is likely to be helpful to project success.

4. Related Work

Wagner [2] presents a sophisticated theoretical model for calculating the cost of quality using defect detection techniques. Slaughter, Harter, and Krishnan [6] shows a complete and detailed analysis of a large software project.

While undoubtedly much more sophisticated than the model presented here, these models also requires much more information and therefore a much higher startup cost to get the needed measures. In contrast, this method is a simple back-of-the-envelope calculation that may be sufficient to motivate senior management to put measurement systems in place that could gather data for a similar cost model.

Demirörs et. al. [5] show an empirical approach and a case study for calculating various costs of software quality (such as the cost to respond to failures in the field) based on easily obtainable metrics (such as the level of effort to respond to those failures.) This approach is useful in risk management, is a historical method, or at least requires some historical information about the project to be predictive.

5. Conclusions and Future Work

Most companies that invest heavily in software development are losing a large fraction of their annual revenue to troubled software projects. The good news is that investments in risk management will more than pay for themselves, even if they are only modestly effective.

In the future, any company that discovers how to lower the cost of its troubled software projects will have a significant competitive advantage.

Future work in this area should include:

- Developing techniques to measure or estimate the effectiveness of specific risk management activities within a single specific software project;
- Including the effects of lost revenue and lost opportunity for profit in the estimate for E_O;
- Sensitivity analysis.

Thanks to André Bondi and Bob Schwanke for their help in preparing this paper.

10. References

[1] J. Johnson. *My Life Is Failure: 100 Things You Should Know to be a Successful Project Leader.* Standish Group International, West Yarmouth, MA, 2006.

[2] S. Wagner. "A model and sensitivity analysis of the quality economics of defect-detection techniques." Proceedings of the 2006 International Symposium on Software Testing and Analysis (ISSTA '06), Portland, ME., USA, July 17-20, 2006, pp. 73-83.

[3] D. Galin, "Towards an inclusive model for the costs of software quality." *Software Quality Professional*, 6(4):25-31, 2001.

[4] L. Huang and B. Boehm. "Determining How Much Software Assurance is Enough? A Value-based Approach. In *Proceedings of the 7th Workshop on Economics-Driven Software Research (EDSER-7), 2005.*

[5] O. Demirörs, Ö. Yildiz, A.S. Güceğlioğlu, I. Bulvari. "Using cost of software quality for a process improvement initiative." *Proceedings of the 26th Euromicro Conference*, Maastricht, 2000.

[6] S.A. Slaughter, D.E. Harter, and M.S. Krishnan. "Evaluating the cost of software quality." *Communications of the ACM*, 41(8):67-73, 1998.