Incorporating Cost-Benefit Analyses into Software Assurance Planning

Martin S. Feather, Burton Sigal, Steven L. Cornford
Jet Propulsion Laboratory,
4800 Oak Grove Drive, Pasadena, CA 91109, USA
Martin.S.Feather@Jpl.Nasa.Gov
Burton.Sigal@Jpl.Nasa.Gov
Steven.L.Cornford@Jpl.Nasa.Gov

Patrick Hutchinson
Wofford College,
429 N. Church St.
Spartanburg, SC 29303
hutchinson@wofford.edu

SYNOPSIS

The objective is to use cost-benefit analyses to identify, for a given project, optimal sets of software assurance activities. Towards this end we have incorporated cost-benefit calculations into a risk management framework. The net result is the capability to rapidly explore the costs and benefits of sets of assurance decisions.

Keywords: Software assurance; Validation and Verification (V&V); risk management; Return On Investment (ROI), software quality; software process improvement; NASA.

PROBLEM

Software development efforts must select assurance activities judiciously, so as to make most effective use of limited resources. Rarely, if ever, can a project afford to apply all such possible activities to all portions of the software. The goal might be to minimize risk with the available resources, or to reduce risk to an acceptable level, minimizing the resources needed to do so. In either case, it is an optimization problem to trade costs (of performing activities) and benefits (the value of risk reduction).

Cost-benefit analyses of *individual* activities have been reported (e.g., reinspections - [Biffl et al, 2000]; regression testing – [Graves et al., 1998]). Studies of overall process improvement also exist (e.g., [McGarry et al, 1998] that relate software defects, productivity, development cycle time and effort estimation to process ratings akin to the Software Engineering Institute (SEI)'s Capability Maturity Model (CMM)- [McGarry et al, 1998]). However, the middle ground, of quantitatively planning the suite of activities to apply to a given project, is relatively under explored. This middle ground is the focus of our ongoing efforts.

We have incorporated the key elements of cost-benefit calculations into a risk management tool. The net result is the ability to study the implications of multiple interrelated decisions on selection of software assurance activities.

Our starting point is a NASA-developed risk management framework, the effect Detection and Prevention (DDP) tool for risk assessment, planning and management [Cornford et al, 2001]. DDP deals with requirements, risks and risk mitigations. Risks are quantitatively related to requirements, to indicate how much each risk, should it occur, impacts each requirement. Mitigations are quantitatively related to risks, to indicate how effectively each mitigation, should it be applied, reduces each risk. A set of mitigations achieves benefits (requirements are met because the risks that impact them are reduced by the selected mitigations), but incurs costs (the sum total cost of performing those mitigations). The main purpose of DDP is to facilitate the judicious selection of a set of mitigations, attaining requirements in a cost-effective manner. DDP has the capability to represent and reason simultaneously with a multitude of mitigations, and their effectiveness at reducing multiple risks. In actual usage, DDP application sessions have dealt with ranges of 50 – 150 each of requirements, risks and mitigations.

APPROACH

Motivated by the desire to aid software assurance planning, we have extended DDP's cost-benefit aspects in the following manner:

Assurance activities are subdivided into:

- *Preventions* – assurance activities that reduce the likelihood of problems occurring, e.g., training of programmers reduces the number of mistakes they make. Preventions incur a cost of performing the prevention activity.
- *Detections* – assurance activities that detect problems, with the assumption that detected problems will be corrected, e.g., unit testing detects coding errors internal to the unit. Detections incur two costs: that of performing the detection activity, e.g., performing the unit test, and that of repairing the problems they detect, e.g., correcting a problem found during unit test. Note that the cost of performing a detection is a function of the detection itself, while the cost of repairing a problem it uncovers is a function the problem. DDP computes the total cost by summing the detection's performance cost and the problem repair costs for the expected number of problems that it detects.
- *Alleviations* – assurance activities that decrease the impact (severity) of problems should they occur, e.g., programming a module to be tolerant of out-of-bound values input to it from another module. Alleviations incur a cost of performing the

alleviation activity.

Assurance activities are performed at certain times, where times are represented as values taken from an ordered, enumerated set, definable by the user, e.g., "requirements phase", "design phase", "code phase", "test phase". These times affect the cost and benefit calculations as follows:

- Earlier-phase preventions and detections reduce the likelihood of problems that get through to later stages. In particular, preventions that precede detections reduce the number of problems there to be detected, and therefore decrease the number of detected problems that incur repair costs.
- Repair costs escalate over time, e.g., the oft-repeated observation that a requirements bug detected and repaired at requirements time costs (say) 10 times as much to repair if only detected and repaired at coding time, 100 times as much at test time, etc.

RESULTS

The above extensions to cost and benefit calculations have been integrated into DDP. As users make selections of assurance activities, DDP calculates and displays the levels to which requirements are attained and the sum total costs of the assurance activities (both performance and repair costs).

One consequence of this is that DDP can identify those additional assurance activities whose selection will decrease both risk and net development costs. This occurs when an earlier-phase activity (e.g, requirements inspection) is selected, and leads to the early discovery, and therefore inexpensive correction, of problems that would otherwise be uncovered only in later phases (e.g, testing) when repair costs are higher. Cost considerations such as these are discussed in [Kaner, 1996].

Return-On-Investment (ROI) calculations can be derived if requirements are valued on the same scale as costs. DDP computes the costs (as described above); DDP also computes benefits – the attainment of requirements – derived from its quantitative computations of risk.

Population of DDP with cost-benefit data for software assurance activities is our current focus. DDP can and has been used successfully in a mode where experts make estimates of effectivity data. We have pre-populated DDP with these estimates, and in DDP applications, users can customize them to the task in hand. For the area of software assurance planning, we are seeking measurement-based data that will have greater fidelity than estimates. In areas of inspection and testing, data does seem to be available. However in general there seems a lack of data on, say, each of the key process areas of CMM level 3. We are now looking to the ongoing efforts of the consortium http://www.cebase.org to gather such data, which we would then add to the pre-populated DDP tool.

ACKNOWLEDGEMENTS

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology. Discussions with John Kelly (JPL), Tim Kurtz (NASA Glenn), Tim Menzies (Univ. British Columbia) and Hoh In (Texas A&M) have been most useful in helping us formulate our ideas.

REFERENCES

[Biffl et al, 2000] S. Biffl, B. Freimut and O.Laitenberger. "Investigating the cost-effectiveness of reinspections in software development", *23rd Int. Conference on Software Engineering*, 2001, pp. 155-164.

[Cornford et al, 2001] S.L. Cornford, M.S. Feather & K.A. Hicks. "DDP – A tool for life-cycle risk management", *IEEE Aerospace Conference*, Big Sky, Montana, Mar 2001, pp. 441-451.

[Graves et al, 1998]. T. Graves, M. Harrold, J. Kim, A. Porter and G. Rothermel. "An Empirical Study of Regression Test Selection Techniques". *20th Int. Conference on Software Engineering*, 1998, pp. 267-273.

[Kaner, 1996]. C. Kaner. "Quality Cost Analysis: Benefits and Risks", Software QA Vol 3, #1, p. 23, 1996.

[McGarry et al, 1998] F. McGarry, S. Burke & B. Decker. Measuring the impacts individual process maturity attributes have on software products. Proceedings, *5th* International *Software Metrics Symposium*, 1998, pp. 52-60