Identifying Good Independent Variables for Program Control

Bob McCann, Lockheed Martin Aeronautics

Abstract. Many of the behaviors and adverse outcomes that we see in software-intensive programs are the result of "misaligned incentives" between the goals of the individuals involved and those of the larger organization. These interact and play out in recurring dynamics that are familiar to both software developers and managers, but which are still poorly understood. By characterizing the forces within these dynamics explicitly in the form of the "acquisition archetypes" described in this paper we can come to understand the underlying mechanisms that cause these problems, and identify mitigations to help mitigate and prevent them.

Abstract

There is an important distinction between program control and program tracking. Control is predictive and proactive using a causal production model that clearly identifies input variables. Tracking is outcome based and decisions based on tracking are reactive and uses output measures. This article analyzes one such causal model and uses it to identify the limits of control (what you can and cannot accomplish with the identified control). The purpose here is to demonstrate how to pick a good control variable from the set of variables derivable from the model.

Introduction

To set expectations, the purpose of this article is to design a better brick not to build a subdivision. In what follows, it is important to distinguish control variables from tracking and oversight metrics. Tracking and oversight metrics are always after the fact, lagging indicators. Control variables are always predictive variables—specifically ones that are open to adjustment. It is also important to distinguish causal models from empirical ones. Causal models make a clear quantitative link between causes and effects while empirical models show general trends without regard to causality and detailed understanding of production processes. Causal models are useful for controlling individual process flows. Empirical models are generally useful for bounding the cost and schedule estimates for proposals before the detailed production processes have been chosen or designed.

Given what we know about delivering products and services using defined, repeatable processes[1], how do we pick the best controls for the project? Surprisingly, the answer comes from basic mathematical concepts. Answer: Pick the variables with the best condition numbers, but what does that mean? The idea is that the process output should be single valued and behave smoothly with respect to small changes in the control variables; in short the process output measure should be a smooth mathematical function of the inputs[2]. This article will demonstrate the application of that principle to a simple causal cost-benefit model.

For instance, if there is a relationship between quality, cost and schedule, then which one is the best independent variable to use in managing the contract? There are three choices:

- **cost as an independent variable**
- **schedule as an independent variable**
- **quality as an independent variable**

Which is the best to use? How do we answer that question?

In what follows a specific simple model is explored to demonstrate how to answer this question. The concepts generalize for more complex models. The chosen model is causal rather than empirical. That is an important distinction because causal models more easily identify how specific actions drive results in the overall project performance. That makes clear that program decisions are controlling the program by directly addressing causes to produce predictable results.

Mathematical Foundations

For linear functions, the choice of which variable to use is a matter of convenience. Not so for curves and more complex functions. It is important that the dependent variable be well behaved when represented as a function of the independent variables. First of all, one should pick an independent variable for which the dependent function is single valued, and secondly the dependent variable should be well conditioned with respect to the independent variable—it should change smoothly and proportionally to small changes of the independent variable. For one-dimensional functions of a single variable, the answer is described by the slope of the function; the function should not have spikes and other singularities near the point of optimum performance.

The example model analyzed here is a simple model for a single sequential process flow with a series of developmental tasks that affect the quality and cost of the end product in a traceable way. For more complex situations, it is necessary to consider the condition number.[3] The condition number is the ratio of the maximum to minimum eigenvalues of the matrix of the first derivatives of the dependent functions with respect to the independent variables—the Jacobian Matrix. (This is a direct application of the inverse function theorem and the definition of condition number for linear systems.)

Schedule is too complex to discuss at the necessary level of detail; it would easily take a whole book, so it will not be treated directly here. Heuristically, for small enough changes in the independent variables, schedule can behave simply. To see this, consider a staffing curve consisting of intervals with constant staffing. Fixed costs will be irrelevant to the marginal analysis that follows. Since marginal cost for applied labor is roughly schedule times labor rate, we can conclude that Cost and Schedule have a more or less piecewise linear relationship and are thus more or less interchangeable in a sufficiently small neighborhood of the operating point. Of course that does not mean that schedule is easy to manage over the life of the project because that simplification does not help when changes become large or discontinuous such as when the workflow on a PERT network shifts from one critical path to another. In general, it takes sophisticated analysis using a robust professional quality-scheduling tool such as Primavera or Artemis to manage the critical path and the most likely near-critical paths of the project's PERT Network. That complexity often obscures the simple relationship that follows from looking at the relationship between cost and quality for individual process flows.

How Does Quality Qualify?

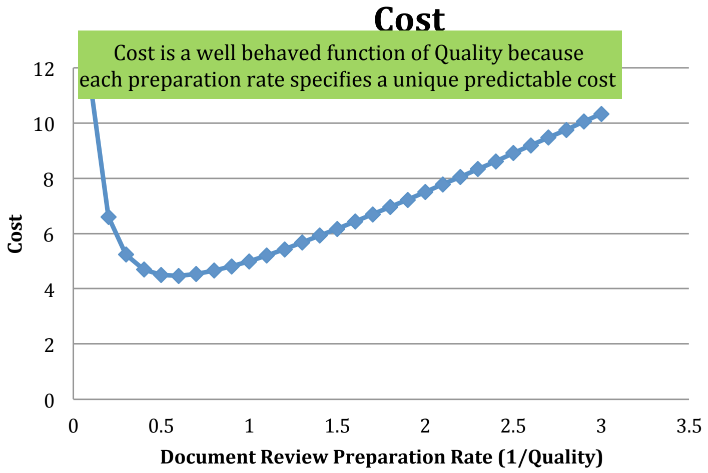
What about quality? With respect to quality, in the example model, marginal cost can be expressed as a linear function of defect injection rate[4] and a non-linear function of document review preparation rate.[5] When plotted as a function of preparation rate, it is clear that the function has a single minimum[6], see Figure 1. This choice meets the criteria identified above.

However, expressing quality as a function of cost is generally double valued and has a point of infinite slope at the cost minimum (the Functional inverse near points with a vertical slope is very poorly conditioned). Clearly, per Figure 2 and starting from the right, there is a region with two possible operating points—one with high quality and one (more likely) with lower quality. Then there is a point with a single optimal operating point and vertical slope, and last there is a region with no solutions. Clearly using cost as the independent variable creates a situation that is poor from the program control perspective—a multivalued relationship on the high cost side of the optimum cost point, a point of infinite condition number at the point of optimum performance, and a region with no stable solution on the lower side of the cost optimum. Programs managed by cost control while under cost pressure have a clear risk of driving off the cliff into chaos.

Please remember that program control is not the same as progress tracking. For example using Earned Value to track progress against budget and Earned Schedule to track progress against schedule both clearly have value but cannot be effective program controls at the task level on individual process flows because Earned Value is a non-causal tracking model. Managing the quality control variable (document review preparation rate) is a much more effective approach to program control for individual process flows at the task level.

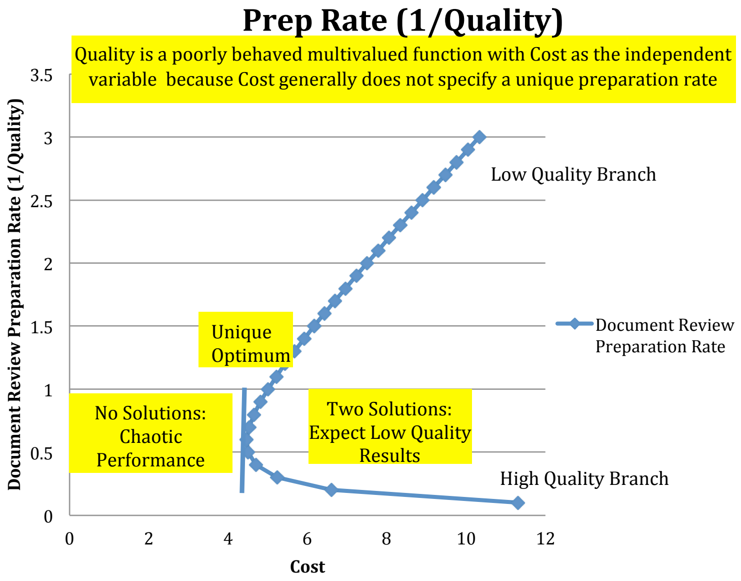
Conclusion

Thus, for the example cost-benefit model for a single process flow, we can conclude that of the three discussed, cost, schedule and quality, the best independent variable is quality as represented by the document review preparation rate. This is an example of what kind of analysis supports the decision of what control parameter to use proactively (leading indicator) in program control. The basic requirement is a quantitative, causal model with measurable, adjustable parameters for the known causes of variation rather than an empirical descriptive scaling model. The ones with the best functional behavior are the ones to use. Heuristically in the following figures, the right answer smiles and the other turns everything on its side and sticks its nose into chaos where it doesn't belong.


Cost

Cost is a well behaved function of Quality because each preparation rate specifies a unique predictable cost

Figure 1 Cost as a Function of Document Review Preparation Rate (Dimensionless Scales)


Prep Rate (1/Quality)

Quality is a poorly behaved multivalued function with Cost as the independent variable because Cost generally does not specify a unique preparation rate

Low Quality Branch

Unique Optimum

No Solutions: Chaotic Performance

Two Solutions: Expect Low Quality Results

High Quality Branch

Figure 2 Document Review Preparation Rate as Related to Cost (Dimensionless Scales)

Disclaimers:

Copyright 2013 Lockheed Martin. Non-Export Controlled. Releasable to Foreign Persons. Non-Proprietary Information.

ABOUT THE AUTHORS

Bob McCann is a staff systems engineer at Lockheed Martin Aeronautics in Fort Worth, Texas. He is currently an Institute of Electrical and Electronics Engineers Certified Software Development Professional and has nearly 20 years of experience in computational physics and high performance computing including nine years at Princeton Plasma Physics Laboratory working in the U.S. Department of Energy-controlled fusion program, as well as about 10 years experience in design and development of relational databases of various kinds. Mr. McCann has served as a member of the Lockheed Martin IS&S Metrics Process Steering Committee and currently works on improving systems engineering processes, methods, and metrics. He has also studied Aikido since 1982 and has taught Aikido for 20 years.

He has a Bachelor of Arts in physics with a concentration in mathematics from Shippensburg University, a Master of Science in physics from University of Maryland, a Master of Science in computer science from Southwest Texas State University, and a Master of Science in computer systems management/software development management at the University of Maryland University College.

Lockheed Martin Aeronautics
P.O. Box 748, Mail Zone 2893
Fort Worth, Texas 76101
Phone: 817-935-4037
Fax: 817-935-5272
E-mail: bob.mccann@lmco.com

NOTES

1. Process based organizations can be very predictable and deliver high quality products and services at low cost. Such organizations can use an assessment tool such as the CMMI Institute's Capability Maturity Model Integrated to understand how well they are organized and how effective that structure can be (can do everything it needs to do). However, measuring performance is the organization's responsibility and is not an explicit part of the assessment model. <http://cmmiinstitute.com>
2. See <http://en.wikipedia.org/wiki/Function_(mathematics)>: In mathematics <http://en.wikipedia.org/wiki/Mathematics>, a function <http://en.wikipedia.org/wiki/Function_(mathematics)#cite_note-1> is a relation <http://en.wikipedia.org/wiki/Binary_relation> between a set <http://en.wikipedia.org/wiki/Set_(mathematics)> of inputs and a set of permissible outputs with the property that each input is related to exactly one output.
3. Please follow the hyperlinks to see a full development of the mathematical concepts where that understanding is a bit rusty.
4. Ron Radice, "High Quality, Low Cost Software Inspections," Paradoxicon Publishing, 2001. This 14 chapter, 478 page book provides a firm foundation for using semi-formal review of intellectual work products, documents in general and code in particular, as an effective program control.
5. See p. 32 for a derivation of the cost function: Robert T. McCann, Cost-Benefit Analysis of Quality Practices <http://www.amazon.com/Cost-Benefit-Analysis-Quality-Practices-Robert/dp/0769546595/ref=sr_1_2?s=books&ie=UTF8&qid=1332876341&sr=1-2> Note that this IEEE Ready Note is a compilation and extension of three Crosstalk articles:

- Robert McCann, "How Much Code Inspection is Enough?" CrossTalk, July 2001
- Robert McCann, "When is it Cost Effective to use Formal Software Inspections?" CrossTalk, March 2004
- Robert McCann, "The Relative Cost of Interchanging, Adding, or Dropping Quality Practices," CrossTalk, June 2010

6. Ibid, Figure 3-2, p. 34