The Challenge of Accurate Software Project Status Reporting:
A Two Stage Model Incorporating Status Errors and Reporting Bias

Andrew P. Snow
Department of Computer Information Systems
Georgia State University, P.O. Box 4015
Atlanta, GA 30302-4015
(404) 651-0879  asnow@gsu.edu

Mark Keil
Department of Computer Information Systems
Georgia State University, P.O. Box 4015
Atlanta, GA 30302-4015
(404) 651-3830  mkeil@gsu.edu

Abstract
Project managers perceive and report project status. Recognizing that their status perceptions might be wrong and that they may not faithfully report what they believe, leads to a natural question – how different is true software project status from reported status? Here, we construct a two-stage model which accounts for project manager errors in perception and bias that might be applied before reporting status to executives. We call the combined effect of errors in perception and bias, project status distortion. The probabilistic model has roots in information theory and uses discrete project status from traffic light reporting. The true status of projects of varying risk were elicited from a panel of five experts, and formed the model input. Key findings suggest that executives should be skeptical of favorable status reports, and that for higher risk projects executives should concentrate on decreasing bias if they are to improve the accuracy of project reporting.

1. Introduction

In many cases, software project failures begin with inaccurate status reporting. The purpose of this study was to understand how reported status is impacted by: (1) the accuracy with which project managers can assess the true status of a software project, and (2) the faithfulness with which the perceived status is then communicated upward to executive management. Since both accuracy and faithfulness can affect the reported status of a project, the obvious research question is which of these variables has the greatest impact on the distortion of project status information received by senior executives.

Ascertaining the true status of a software project is often something easier said than done. At the project manager level, the perception of project status is influenced by whatever project management systems are in place to track project status as well as the perceptions and opinions expressed by those working under the project manager’s direction. In many instances, a project manager may misperceive the true status of a project. Although this is a general problem that can occur with any type of project, the literature e.g., [2, 7, 12, 21] suggests that information technology (IT) projects may be particularly challenging in this regard. The intangible nature of software makes it difficult to obtain accurate estimates of the proportion of work completed, which may promote misperceptions regarding project status.

If the project manager cannot always get an accurate picture of the true project status, then how can senior executives within a firm have any hope of obtaining accurate project status information? To make matters worse, project managers may sometimes censor themselves in reporting status, as they actually perceive it. If a project is perceived to be performing poorly, the project manager may withhold this information or misrepresent status to senior management. The net result is that senior managers may often be at the receiving end of status information that has been biased in some fashion. The case of the CONFIRM project (originally estimated at $55.7 million, and cancelled 3.5 years later with expenditures of $125 million) illustrates this point:

“On May 1, 1992, AMRIS’ vice-chairperson circulated a letter internally stating that “some people who have been part of CONFIRM RS management did not disclose the true status of the project in a timely manner. This has created more difficult problems—of both business ethics and finance—than would have existed if those people had come forward with accurate information”[17]. The case of CONFIRM is especially interesting because individuals who were close to the project apparently knew