Chapter 49
The Mythical Lines of Code Metric: A Form of Moral Hazard

Charley Tichenor
Marymount University, USA

ABSTRACT

Using the lines of code (LOC) metric in software project management can be a financial moral hazard to an organization. This is especially true for upper management who handles an organizational budget and strategic plan. Software project managers have their own budgets. However, if they fail to meet the budget, the organization's cash flow, rather than the project manager's personal cash flow, will suffer. This chapter will discuss the practice of software project management, the field of software metrics, game theory, and the game theory issue of moral hazard. It will demonstrate why using LOC as a metric can present a moral hazard to senior management and an organization.

INTRODUCTION

Cost overruns are significant problems encountered by the software development industry. One example is the United States Federal Bureau of Investigation's (FBI) Sentinel software development project. As reported by the Cato Institute in Washington, DC (DeHaven, 2010):

Sentinel is currently 32% over-budget and two years behind schedule. Worse, an independent assessment of the project concluded that the project will need another \$351 to \$453 million and won't be finished for another 6 to 8 years. As the inspector general notes, "… the longer the full implementation of Sentinel takes, the more likely it is that already implemented hardware and software features will become obsolete.

In 2011, a report from the Office of the Under Secretary of Defense, Acquisition, Technology, and Logistics indicated that after studying nine ERP systems, "ERP schedule delays and cost overruns were common and sometimes large" (G. Bliss, personal memorandum, February 23, 2011).

DOI: 10.4018/978-1-6684-3702-5.ch049

Those working in the software development industry can identify others within their organization or similar organizations.

A major cause of cost overruns in software development is the use of unsound cost forecasting methods. Often, software cost estimations are poorly taught or overlooked in IT academic programs. Operations research/analytics programs rarely address the issue or will teach unsound methods. Probably less than 10 successful software cost estimation management consulting firms operate in the U.S.

An effective approach to forecasting software development costs is to look for a strong statistical regression correlation between a predictor variable (representing the amount of software developed) and the resulting cost to develop the software. While many predictor variables have appeared in practice, the LOC variable has a strong following in both government and industry. Written in a programming language, programmers generate LOC. According to the LOC metric approach, as software requirements increase, more LOC will be required. Work to write the LOC (as well as perform testing, deployment, and project management tasks) will increase as the number of LOC increase.

This chapter will detail the LOC metric and show how fundamentally in error its use can be.

BACKGROUND

The Force Multiplier

In the military, a force multiplier is “a capability that, when added to and employed by a combat force, significantly increases the combat potential of that force and thus enhances the probability of successful mission accomplishment” (Force multiplier, 2016). Many military examples discuss the employment of a force multiplier. For example, suppose that a modern army unit of 10 soldiers was to transport back in time to face 100 of Napoleon’s soldiers. Who would have the competitive advantage? Napoleon’s soldiers were equipped with single shot rifles; the modern army unit is equipped with automatic rifles with magazines. This advanced technology gives the modern army a competitive advantage over Napoleon’s army. This is a force multiplier. The modern army unit has helmets and body armor. The modern army unit would enter the battle dug into foxholes. Napoleon’s army would enter the battle without armor and firing from a standing position. The modern army unit would have better field rations, camouflaged uniforms, vaccinations, automatic deposits of their paychecks, and a military family support group program. These force multipliers give the modern army unit a competitive advantage over Napoleon’s. The modern army unit 10 soldiers could probably defeat Napoleon’s unit of 100. Although it is more expensive “per unit” to recruit, train, and equip modern army soldiers, a force of 10 would achieve a victory at a significant overall cost reduction in terms of resources and manpower.

The multiplier effect can also work in reverse. Suppose that the modern army unit of 10 had weapons that always misfired. In that scenario, Napoleon’s army would easily defeat them with a complete loss of expensive resources and soldiers.

The multiplier effect is not limited to military combat. Business, government, and military organizations need administrative force multipliers to be more efficient and effective. Consider the organization of 2016 compared to its counterpart organization of 1966: replace letters and the desk in-box with e-mail; replace electric typewriters with word processors; and use computers to significantly reduce the manpower to perform payroll and other accounting calculations, supply chain accountability, inventory