Eliminating Software Task Overestimation
Using Economic Game Theory to Create a Competitive Incentive System

T.J. Glazier

Problem
It is common for software engineers to overestimate the time to successfully complete a task by as much as 250% of the actual time required. Systematic overestimation is an organizational behavior that is the product of tangible penalties for being late and no incentives for being early. This results in substantial overstaffing, dramatically higher costs, and grossly incorrect project schedules.

Goal
Eliminate overestimation by creating a system that incentivizes finishing early, creates deterrents for being late, and yields accurate task estimations.

The System

Step 1: Management determines that every engineer should spend at least 30 hours coding each week and a maximum of 40

Step 2: The tasks for the week are bid on by the engineers using hours, they must bid a minimum total of 30, max 40.

Step 3: System assigns each task to the lowest bidder

Step 4: Engineers finish their assigned tasks and are awarded credit equal to the bid amount.

Step 5: Management reviews credits awarded for performance

Why It Works/Analysis

Penalties for being over create upward pressure on estimate. + Competition for tasks creates downward pressure on estimates. = Accurate Estimates!

This system models and creates an open market economic system with the developers acting as companies, the bid is the product price, the time spent the product cost, and the over/under is the profit margin.

Benefits
1. System produces accurate estimates
2. Additional business intelligence (e.g. over/under for an engineer is a normalized performance score.)
3. Economic/technical analysis can be applied to produce additional metrics