



Looking At Schedule Vs. Effort In Software Engineering, Systems Engineering and Systems Overall

John E. Gaffney, Jr.
301-721-5710
j.gaffney@lmco.com
Center For Process Improvement Excellence
Corporate Engineering and Technology
Lockheed Martin Corporation
July 16, 2008

7/16/08

1



Focus Issues and Questions

- An issue of considerable importance to proposal managers, program managers, technical planners and to software engineering and systems engineering managers is how schedule compression or stretch-out affects engineering costs or overall project costs
- Schedule compression (or stretch-out) can be defined as the amount or percentage of reduction (increase) of a project or software or systems engineering schedule with respect to some ideal or nominal value as related to cost or productivity
- Knowing what a relationship between cost (and productivity) and schedule (duration) can help us to answer such questions as:
 1. "What is the optimum duration to perform this task (e.g., development of a software system)?" Here, optimum might mean with respect to minimizing cost. More generally, it might mean with respect to some utility or value function of cost and duration, in which the relative importance or utilities of the cost and schedule (duration) values are stated. In the extreme, a project might be cost driven or schedule driven.
 2. "Can schedule (duration) and cost (effort) be traded off; if so, what is the tradeoff?"

7/16/08

2

Basic Approach, Mathematical Model Structures and Considerations

- The basic approach considered here is to use a parametric or top-down mathematical model to represent relationships among size, S, cost (effort), K, and schedule or duration, T
 - Note: S could be equivalent new source statements in the case of software, e.g. in the COCOMO tool or equivalent new requirements in the case of systems engineering estimation, e.g., in the COSYSMO tool*
- Use actual project data to establish the values of the models' parameters
 - Ideally, establish what projects in the data base were compressed or extended beyond a “normal” completion time
 - Practically, there may not be data that enables this to be done as none of the projects for which data is available had compressed schedules
 - Absent much data, use engineering judgment

7/16/08

3

Mathematical Cost Models/Tools Structure-1

- The (parametric/top-down) mathematical models considered here:
 - Relate three variables:
 - S: size
 - K: effort/cost
 - T: schedule/duration
 - Are of two principal forms:
 - Implicit variables
 - Dependent and Independent variables
- Implicit: Variables not characterized as independent or dependent:
 - R=Function (S,K,T);
 - R is some constant
 - This form of a cost model provides a <S,K,T> “tradeoff space.” This form recognizes that it may be difficult to differentiate independent and dependent variables
 - Not further discussed in this presentation

7/16/08

4

Mathematical Cost Models/Tools Structure-2

- Dependent and independent variables identified:
 - Examples: The COCOMO/R and COSYSMO/R software and systems engineering models are of the form: $K=A*S^E*D$; where:
 - D is the product of a set of cost drivers that modify a baseline productivity, A
 - E is a number related to whether productivity increases or decreases as S increases
 - K is the dependent variable
 - COCOMO/R: $K=Func(S,T^*)$;
 - T* means that schedule is represented as a cost driver with several alternative cost-multiplicative values to represent the affect of schedule compression on cost (increase). There is another relationship of the form: $T=a*K^b$ that is also used; it does not explicitly represent the effect of S
 - COSYSMO/R : $K=Func(S)$;
 - Schedule is not represented in the cost equation currently. However, Lockheed Martin is working on doing so
 - Example: SLIM software engineering estimation model/tool:
 - Form: $S=A*(K^p)*(T^q)$
 - SLIM is of the form of a **Cobb-Douglas Production function**
 - The variables K and T are “factors of production” and may enable tradeoffs to be made between K and T, effort and schedule

7/16/08

5

Cobb-Douglas Production Function

- The Cobb-Douglas Production Function relates an output of a process to factors of production
 - An example is a C-D P F that relates software source statements, S, produced by a software development process, to factors of production, such as labor, K, and time or project duration, T
 - General Form: $O=A*(F_1^{Q1})*(F_2^{Q2})*.....*(F_N^{QN})$; O=Output; the F_i are the factors of production
- Examples of Use of the Production Function Form
 - Inputs: labor and capital; output=automobiles. From this form can be derived a “productivity,” actually a “unit cost,” labor hours per automobile. Notice the indicated possible tradeoff between capital (appropriately applied in terms of training and technology) and labor; more capital →less labor to obtain the same output
 - Inputs: invested capital, labor; output=\$ profit From this form can be derived a “productivity,” return-on-invested capital”
 - Inputs: invested capital, labor; output=\$ sales From this form can be derived a “productivity,” “sales per employee”
 - Inputs: inspection time, meeting time, amount of material inspected; output=number of defects discovered. Can use as a basis for “tuning” the inspection process and for identifying opportunities for process improvement

7/16/08

6



Some Software Engineering and Systems Engineering Examples-1

- General Form: $S=A*(K^p)*(T^q)$
- Some Actual Cases That Exemplify Different Situations That Can Be Experienced:
 - 1. $S=A_2*(K^{0.6288})*(T^{0.5555})$; $q/p=0.8831$
 - 2. $S=A_3*(K^{0.3333})*(T^{1.3333})$; $q/p=4.0000$
 - 3. $S=A_1*(K^{0.929})*(T^{0.079})$; $q/p=0.0807$
 - 4. $S=A_3*(K^{0.81})*(T^{-0.553})$; $q/p=-0.6827$
- Observations Concerning These Equations:
 - Case 1, Examples 1&2 :If $q&p$ both >0 ; K,T can be traded off for a given value of S : increasing T means decreasing K , for a given S
 - Case 2, Example 3: Low values of q/p , especially low q , mean little change in K , for a change in T , for a given value of S
 - Case 3, Example 4: If either q or p is <0 , this means that the equation for S is defined in a region for which schedule increases/decreases are matched with cost/effort increases/decreases, i.e., changes of each variable are of the same sign; that is, they are not traded off: increasing T means increasing K , for a given value of S

7/16/08

7

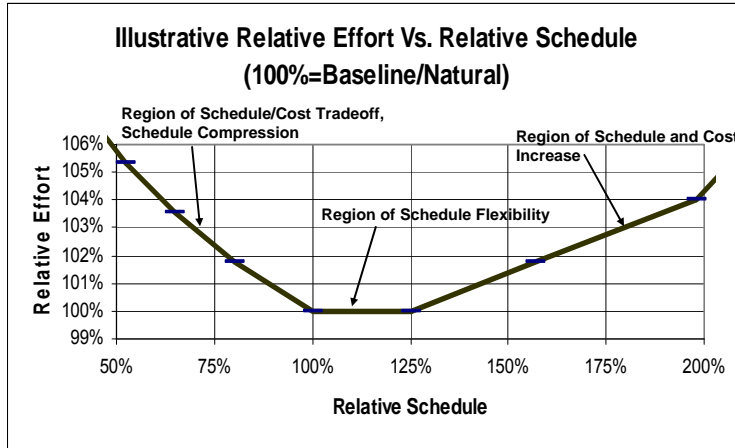


Some Software Engineering and Systems Engineering Examples-2

- Application of these equations:
 - An organization might experience any one or all three behaviors: for T increasing (illustrated in the plot on the next page): 1. K increasing, 2. K flat or 3. K decreasing The equations relating S to K and T are one-sided and might have to be fitted separately over various ranges
 - Always look at the data; don't just apply/fit equations !
- For small incremental changes in any one or all of the variables, e.g., $\Delta K/K$, % change in K ,:
 - $\Delta S/S = (\Delta A/A) + (p*(\Delta K/K)) + (q*(\Delta T/T))$; If $\Delta S/S = (\Delta A/A) = 0$;
 - Then, $\Delta K/K = -(q/p)*(T/\Delta T)$; For Example 3, a 4% reduction in schedule would result in only a 0.3% increase in effort. However, for Example 1, that same schedule reduction would result in a 3.5% increase in cost/effort. For Example 2, there would be 16% increase in effort; some would consider this to be not realistic

7/16/08

8



7/16/08

9



Some Recommendations and Questions

- Organizations should collect and carefully examine data for completed projects that are believed to have exhibited schedule compression as well as those that have not
- From this data, relationships such as described here should be developed that can be applied to the estimation of new projects or updating estimates for existent ones, e.g., to develop EAC (Estimate-At-Completion) values
- Absent sufficient data to develop a mathematical relationship between schedule and cost/productivity, a fall-back position can be to use expert engineering judgment/the delphi technique to establish criteria for identifying when the prospective schedule is compressed, too short, for a process being cost estimated, e.g., software engineering, systems engineering, and what the effect on the cost (effort) and productivity is expected to be under those circumstances
- Identify compatible/ harmonized approach to relating schedule and cost/productivity for COSYSMO/R and COCOMO/R
 - What do you think: Trade-off equations? Cost drivers?

7/16/08

10