Illustrating Probability in Software Cost and Schedule Estimating:

Know the Odds Before Placing Your Bet

Michael A. Ross
President & CEO
r2Estimating, LLC
7755 E. Evening Glow Dr.
Scottsdale, AZ 85266-1295
480-488-8366
mike.ross@r2estimating.com
http://www.r2estimating.com

[1,2,3]***Abstract***—This paper starts with the premise, "***The only thing certain about project performance is uncertainty.***" Over 25 years of software industry experience and project data tells us that cost and schedule estimating is essential to diligent affordability management, acquisition management, and project management. It also tells us that projects behave according to certain dynamic properties, that duration, effort, cost, and defects are all inexorably linked, that this linkage is influenced by people, project, and product characteristics; and that, prior to project completion, ***everything is uncertain***. A primary goal of any thorough project estimating process should therefore be to not only yield estimated values for these metrics; it should also indicate whether or not these values satisfy their corresponding goals with some corresponding desired probability of success. This paper describes a new chart (we call it a ***Ross Chart***™) that shows the goals satisfaction and confidence (probabilities of success) of two random dependent variables in a bivariate estimating relationship. Using an effort versus duration estimating relationship example, the paper describes a method for probabilistic treatment of bivariate relationships. It then presents an example project estimating scenario using a system of three synchronized Ross Charts to show the degree of risk in the duration, effort, cost, and defects that are associated with a particular estimating solution. In other words, this paper presents a process (methods and tools) for performing ***confidence-driven estimating***.

TABLE OF CONTENTS

---

[2] 2nd Edition, October 13, 2008

1. INTRODUCTION

Purpose

This paper describes a new chart (hereinafter referred to as a ***Ross Chart***™) that is a graphical depiction of the goals satisfaction and confidence (probabilities of success) of two dependent random variables a bivariate estimating relationship. Beginning with an effort[4] versus duration estimating relationship example, the paper describes how to treat these bivariate estimating relationships in a probabilistic manner and then brings in cost versus duration and defects versus duration to show an integrated estimate; the results displayed as duration-synchronized Ross Charts showing the degree of risk associated with the resulting estimates of duration, effort, cost, and defects. In other words, this paper presents a process (methods and tools) for performing ***confidence-driven estimating***.

Scope

While this paper focuses on an example from the software development world, the concepts presented herein can easily be applied to hardware development, systems engineering, operations and support; virtually any process where estimating relationships exist and where alternatives must be evaluated in the face of significant uncertainty.

Background

"The revolutionary idea that defines the boundary between modern times and the past is the mastery of risk: the notion that the future is more than a whim of the gods and that men and women are not passive before nature. Until human beings discovered a way across that boundary, the future was a mirror of the past or the murky domain of oracles and soothsayers who held a monopoly over knowledge of anticipated events."(Bernstein, 1996)

The only thing certain about project performance is uncertainty. Over 25 years of industry experience and project data tells us that cost and schedule estimating is essential to diligent affordability management, acquisition management, and project management. It also tells us that projects behave according to certain dynamic properties, that duration, effort, cost, and defects are all inexorably linked (correlated), that these correlations can be expressed as functions of product and project attributes, and that, prior to project completion, ***everything is uncertain***. A primary goal of any thorough project estimating process should therefore be to not only yield estimated values for these metrics; it should also indicate whether or not these estimated values satisfy their corresponding project goals (commitments) some corresponding desired confidence (probabilities of success). Recognition of this primary goal is exemplified in the 2004 Sambur-Teets memo, *"**High Confidence Estimates:** Estimate the software development and integration effort (staff hours), cost, and schedule at high (80-90%) confidence."* (Sambur, et al., 2004) and

[4] Effort is the cumulative result of people laboring to do work over elapsed time (Ross, 2007b). Effort is typically expressed in units such as person-months, person-weeks, or person-hours. Effort is directly proportional to cost where the constant of proportionality is the labor rate. Labor rate is typically expressed in currency units per effort units; e.g., dollars per person-hour.

in the NASA Cost Estimating Handbook, *"As a general rule, cost estimates at NASA should be presented at the 70% confidence level. As an entire portfolio of Projects, the budget should be presented at the 80% confidence level."* (National Aeronautics and Space Administration (NASA), 2004)

2. Charting Bivariate Relationships

r2 Software Estimating Framework

An example of two bivariate estimating relationships are those for software content productivity and software defect propensity found in the r2 Software Estimating Framework (r2SEF)™ (Ross, 2007b).

Fundamental Software Content Productivity Equation

$$E_p^{\alpha_E} t_p^{\alpha_t} = \frac{S_e}{\eta} \qquad \textbf{(1)}$$

where

E_p = Total Process Effort

t_p = Total Process Duration

α_E = Effort Exponent

α_t = Duration Exponent

S_e = Effective Content

η = Specific Efficiency

Fundamental Software Defect Propensity Equation

$$E_p^{\varphi_E} t_p^{\varphi_t} = \frac{\Phi_{[a,b]}}{\delta_{[a,b]}} \qquad [\varphi_t \le 0] \qquad \textbf{(2)}$$

where

φ_E = Defect Effort Exponent

φ_t = Defect Duration Exponent

$\Phi_{[a,b]}$ = Defect Count

$\delta_{[a,b]}$ = Specific Defect Vulnerability

Fundamental Software Management Stress Equation

$$M = \frac{E_p}{t_p^{\gamma}} \qquad \textbf{(3)}$$

where

γ = Gamma (Economy Exponent)

M = Specific Management Stress

Typical (Nominal Stress) Equation

$$M_{\text{nom}} = \frac{E_{p\,\text{nom}}}{t_{p\,\text{nom}}^{\ \gamma}} \tag{4}$$

$E_{p\,\text{nom}}$ = Nominal (Typical) Total Process Effort

$t_{p\,\text{nom}}$ = Nominal (Typical) Total Process Duration

M_{nom} = Nominal (Typical) Specific Management Stress

Minimum Duration Limit (Brooks' Law)

$$M_{\max} \geq \frac{E_p}{t_p^{\ \gamma}} \Rightarrow M_{\max} = \frac{E_{t_{p\min}}}{t_{p\min}^{\ \gamma}} \tag{5}$$

$E_{t_{p\min}}$ = Total Process Effort at Minimum Total Process Duration

$t_{p\min}$ = Minimum Total Process Duration

$M_{\max}$ = Maximum-Tolerable Specific Management Stress

Minimum Effort Limit (Parkinson's Law)

$$M_{\min} \leq \frac{E_p}{t_p^{\ \gamma}} \Rightarrow M_{\min} = \frac{E_{p\min}}{t_{E_{p\min}}^{\ \gamma}} \tag{6}$$

$E_{p\min}$ = Minimum Total Process Effort

$t_{E_{p\min}}$ = Total Process Duration at Minimum Total Process Effort

$M_{\min}$ = Minimum Practical Specific Management Stress

r2SEF Effort-Duration Estimating Relationship as an Example

We will use the software content productivity relationship shown in Equation **(1)** in conjunction with the software management stress relationship shown in Equations **(3)**, **(4)**, **(5)**, and **(6)** as an example in deriving the elements of a confidence-driven estimating method. Note that the same methodology can also be (and has been) applied to the defect propensity relationship in conjunction with the management stress relationship; the results of both derivations forming the basis of the duration-synchronized Ross Charts used in the Example Software Project Estimating Scenario found later in this paper.

To begin, a common thread in all but the simplest software estimation models is the desire to estimate total process effort (labor) E_p and total process duration (time) t_p as a function of the effective software content (size) S_e and some quantification of specific efficiency (reciprocal net environmental complexity) η. In the r2SEF this is accomplished by the bivariate effort-duration estimating relationship shown in Equation **(1)**.

In the typical estimating situation we try to determine reasonable expectations for both effort E_p and duration t_p. Using Equation **(1)** as our estimating relationship and having just declared effort and duration to be our dependent variables in a bivariate relationship, we need to instantiate effective content S_e and specific efficiency η in order to get any kind of meaningful result. A convenient way to illustrate the dynamics of this relationship is to chart effort as a function of duration for a given ***content/efficiency ratio*** ψ where

$$\psi \equiv \frac{S_e}{\eta} \qquad \textbf{(7)}$$

Solving Equation **(1)** for effort and substituting Equation **(7)** into the result yields

$$E_p = \left(\psi t_p^{-\alpha_t}\right)^{\frac{1}{\alpha_E}} \qquad \textbf{(8)}$$

Charting a specific software development process instance[5] of Equation (8) where $\alpha_E = 0.5462$ and $\alpha_t = 0.7463$ and for a given content/efficiency ratio that assumes effective content (size) of 35,000 Source Lines of Code (SLOC) with average efficiency ($\eta = \overline{\eta} = 22.79$), we get **Figure 1**. The instantiated form of Equation **(8)** is

$$E_p = \left(\frac{35{,}000\ \text{(SLOC)}}{22.79} t_p^{-0.7463}\right)^{\frac{1}{0.5462}} \qquad \textbf{(9)}$$

[5] This specific instance is based on the example Company X Avionics Projects data set described in (Ross, 2007b).

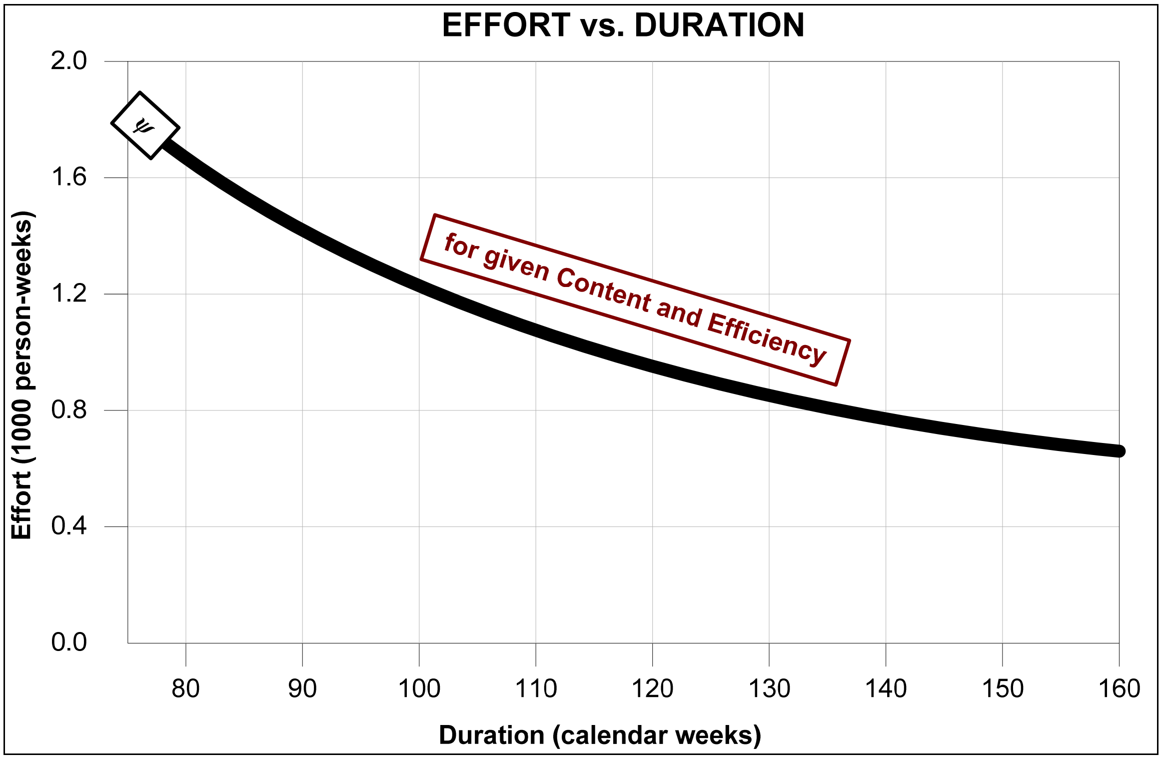

Figure 1: Example Estimating Relationship Curve (Content Productivity Law)

Note that, because α_E and α_t are positive, this particular example estimating relationship embodies a complementary (negative) relationship between effort and duration; i.e., more time means less effort (money) and vice versa. While this *tradeoff* behavior is consistent with some labor-dominant processes such as many of those found in software development, it may not be consistent with other types of development processes such as those where fixed average effort rate (level of effort) activities dominate. In these cases the appropriate estimating relationship will embody opposing (positive) behavior between duration and effort; i.e., more time means more labor and vice versa.

3. CHARTING ESTIMATING RELATIONSHIP LIMITS

Often times estimating relationships have effectiveness limits; i.e., there is a range where they work and outside the range they tend to break down (historical data is sparse or non-existent outside the range). This is true for the content productivity relationship; therefore, there exists, for each instance of this relationship, unique minimum duration and minimum effort limiting functions.

Minimum Duration Limit—This limit is defined by Equation **(5)** where M_{max} quantifies the maximum management stress that the process being estimated can likely tolerate. **Figure 2** shows the region excluded by Equation **(5)** in red. Note that the curve described by the margin between the red region and the white region is actually the minimum duration limiting function (Equation **(5)** as an equality).

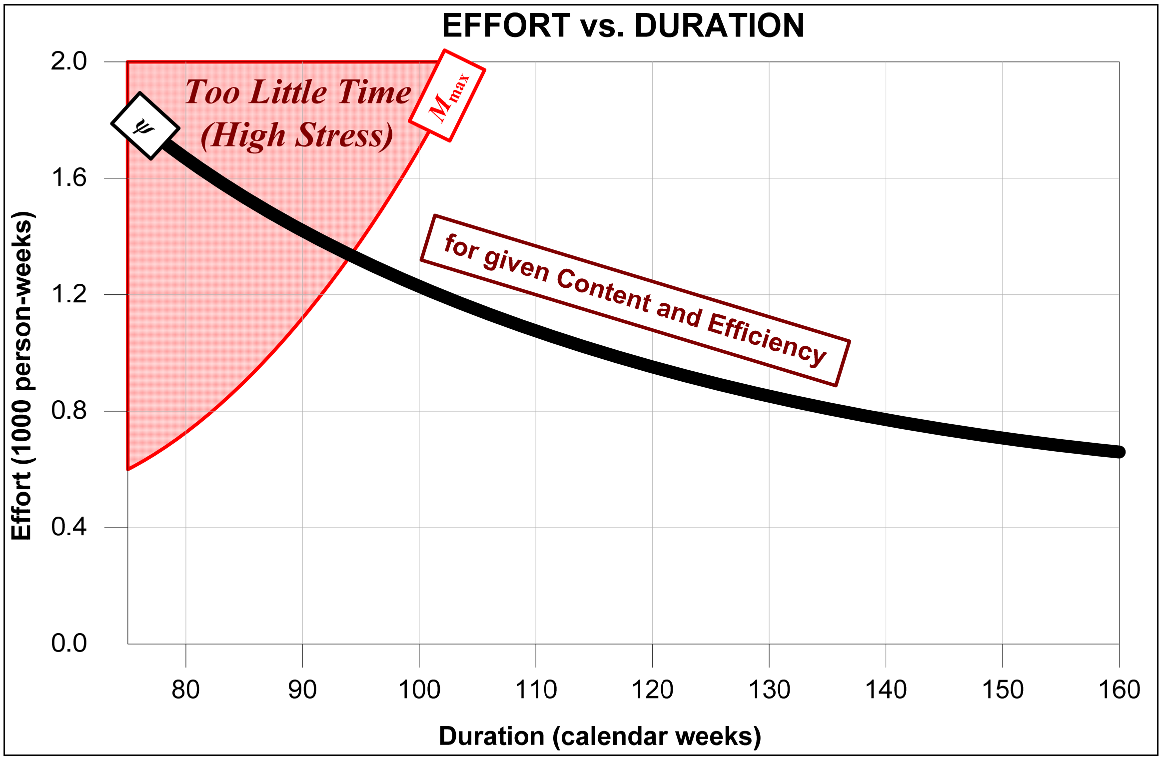

Figure 2. Minimum Duration Limit

Minimum Effort Limit—This limit is defined by Equation **(6)** where $M_{\min}$ quantifies the minimum management stress that the process being estimated needs to be reasonably productive. **Figure 3** shows the region excluded by Equation **(6)** in yellow. Note that the curve described by the margin between the yellow region and the white region is actually the minimum effort limiting function (Equation **(6)** as an equality).

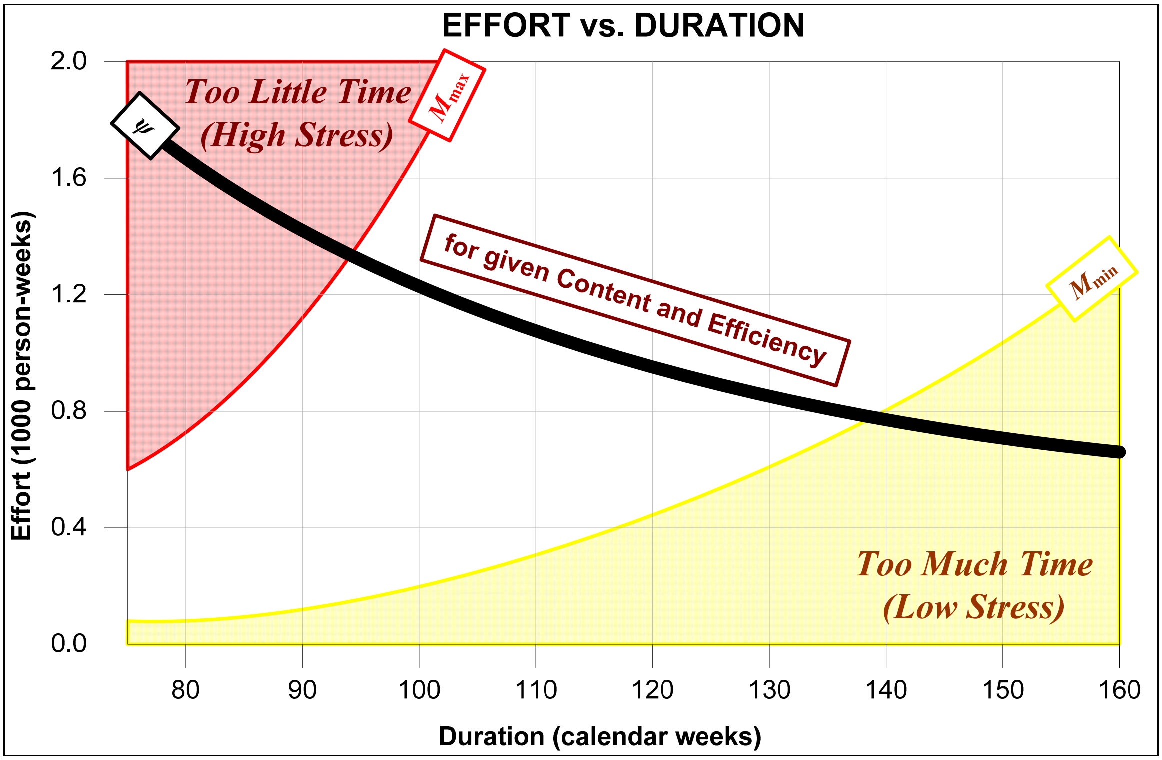

Figure 3. Minimum Effort Limit

4. Ross Chart Basics

Figure 3 is an example of using linear programming techniques to solve problems with multiple limits or constraints. We now extend the Cartesian format of Figure 3 with additional chart objects to yield what we will ultimately refer to as a Ross Chart.

Fundamental Ross Chart Layout

The Ross Chart uses, as its foundation, a two-dimensional Cartesian plane or grid. Each of two correlated dependent variables is represented by one of the two axes. In **Figure 3** we are using, as an example, software development effort (labor) and software development duration (time) as our two correlated dependent variables, represented on the y (vertical) axis and the x (horizontal) axis respectively. In our example, effort is measured in person-weeks and duration is measured in calendar weeks.

The estimating relationship between the two variables, in this case effort versus duration, is represented as a curve in the Cartesian plane (the bold black curve shown in **Figure 3**). The minimum duration and minimum effort limits are shown as red and yellow regions respectively.

Goals

Most processes are governed by management constraints. We define a ***management constraint***, within this context, to be a pair of numbers associated with some management measure or metric; these two numbers being a ***goal*** value and a ***desired confidence*** (probability of success) value. For our example, we will assume there exists a management constraint for each of effort and duration.

Figure 4 adds the goals associated with each constraint to our evolving Ross Chart example as interactive dynamic goal symbols (blue diamonds) that traverse each axis and represent the goal value associated with the corresponding variable (metric). In our example, the effort goal is displayed as 800 person-weeks and the duration goal is displayed as 110 calendar weeks.

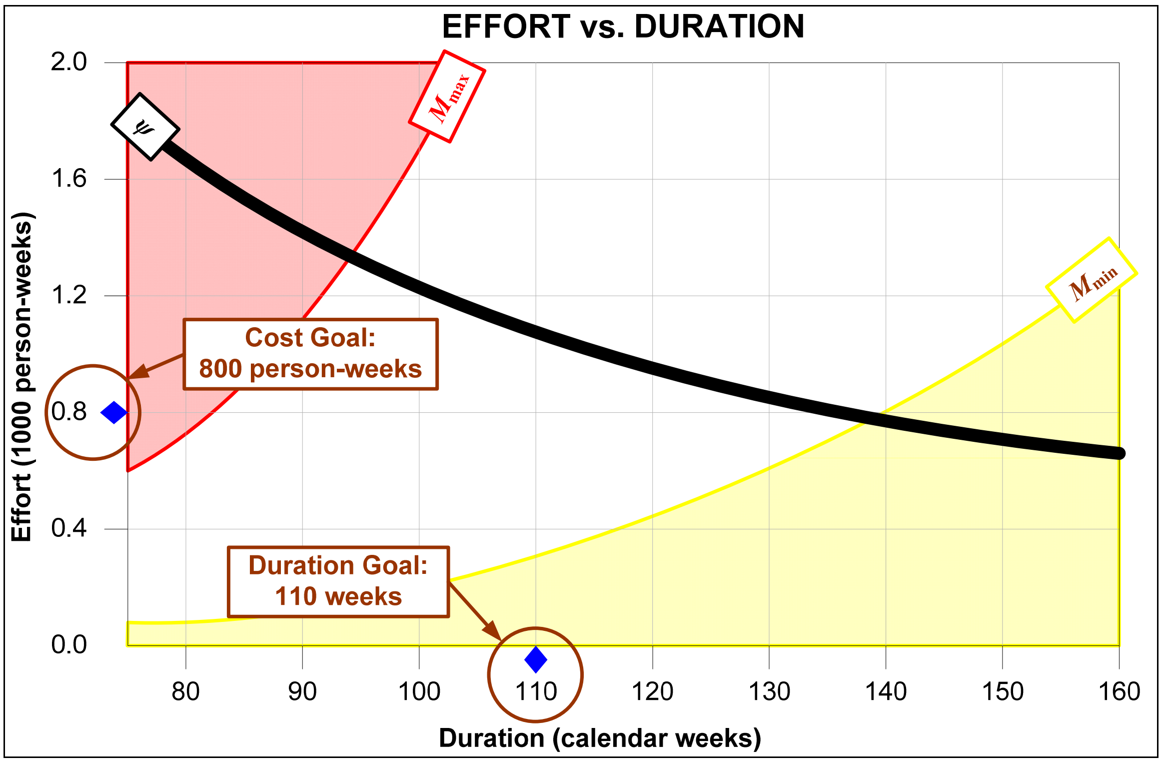

Figure 4. Goal Symbols

Reasonable Solutions

Figure 5, **Figure 6**, and **Figure 7** each show the addition of an interactive dynamic solution symbol (blue circle) to our evolving Ross Chart with projection lines to each axis, each Ross Chart a specific instance (effort-duration solution) on the estimating relationship curve.

Minimum Duration Solution—**Figure 5** shows the blue solution circle positioned on what we refer to as the minimum duration solution. Note that the solution occurs at the intersection of the estimating relationship curve and the minimum duration limiting function. Mathematically, the x (duration) coordinate can be found by substituting Equation **(8)** instantiated with $t_{p\min}$ and $E_{t_{p\min}}$ into Equation **(5)** and solving for duration

$$M_{\max} = \frac{\left(\psi t_{p\min}{}^{-\alpha_t}\right)^{\frac{1}{\alpha_E}}}{t_{p\min}{}^{\gamma}} \qquad t_{p\min} = \left(\frac{\psi}{M_{\max}{}^{\alpha_E}}\right)^{\frac{1}{\gamma\alpha_E+\alpha_t}} \tag{10}$$

The y (effort) coordinate can then be found by solving Equation **(5)** for effort; i.e., projecting duration off of the minimum duration limit equality (the margin between the red region and the white region) onto the y (effort) axis.

$$M_{\max} = \frac{E_{t_{p\min}}}{t_{p\min}{}^{\gamma}} \qquad E_{t_{p\min}} = M_{\max} t_{p\min}{}^{\gamma} \tag{11}$$

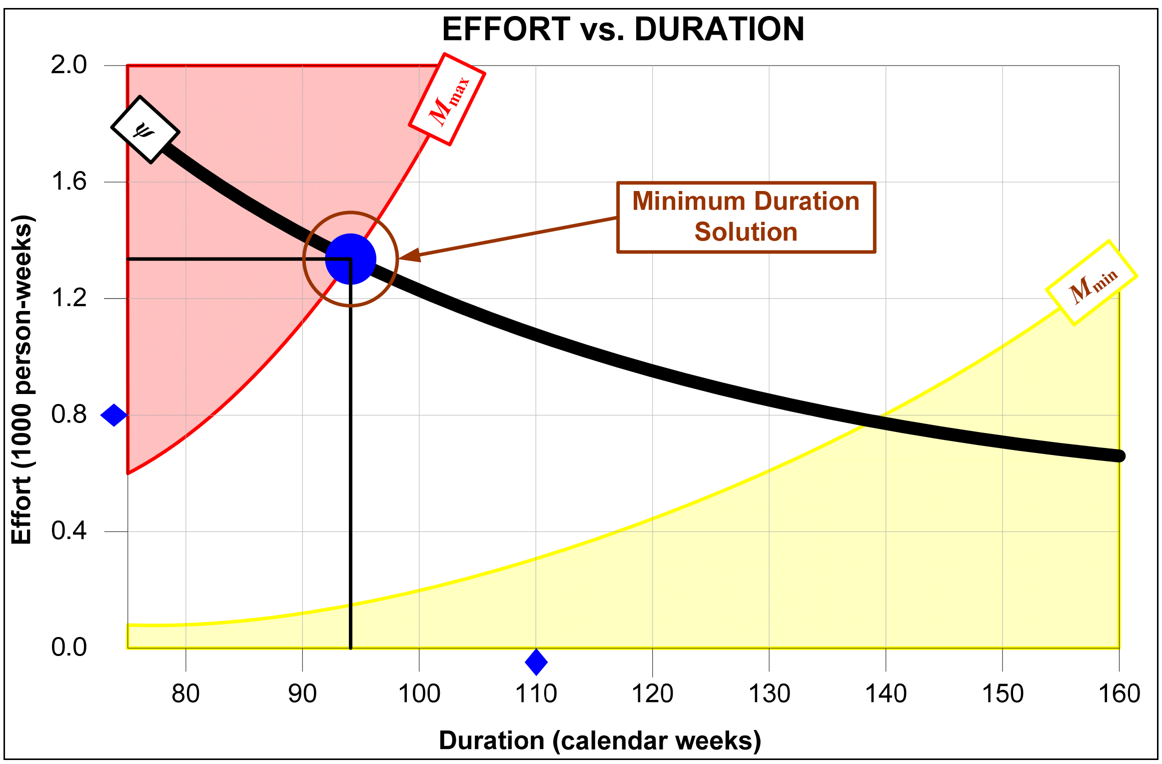

Figure 5. Minimum Duration Solution

Minimum Effort Solution—**Figure 6** shows the blue solution circle positioned on what is referred to as the minimum effort solution. Note that the solution occurs at the intersection of the estimating relationship curve and the minimum effort limiting function. Mathematically, the x (duration) coordinate can be found by substituting Equation **(8)** instantiated with $t_{E_{p\min}}$ and $E_{p\min}$ into Equation **(6)** and solving for duration.

$$M_{\min} = \frac{\left(\psi t_{E_{p\min}}{}^{-\alpha_t}\right)^{\frac{1}{\alpha_E}}}{t_{E_{p\min}}{}^{\gamma}}$$
$$t_{E_{p\min}} = \left(\frac{\psi}{M_{\min}{}^{\alpha_E}}\right)^{\frac{1}{\gamma\alpha_E+\alpha_t}} \quad \textbf{(12)}$$

The y (effort) coordinate can then be found by solving Equation **(6)** for effort; i.e., projecting duration off of the minimum effort limit equality (the margin between the yellow region and the white region) onto the y (effort) axis.

$$M_{\min} = \frac{E_{p\min}}{t_{E_{p\min}}{}^{\gamma}}$$
$$E_{p\min} = M_{\max} t_{E_{p\min}}{}^{\gamma} \quad \textbf{(13)}$$

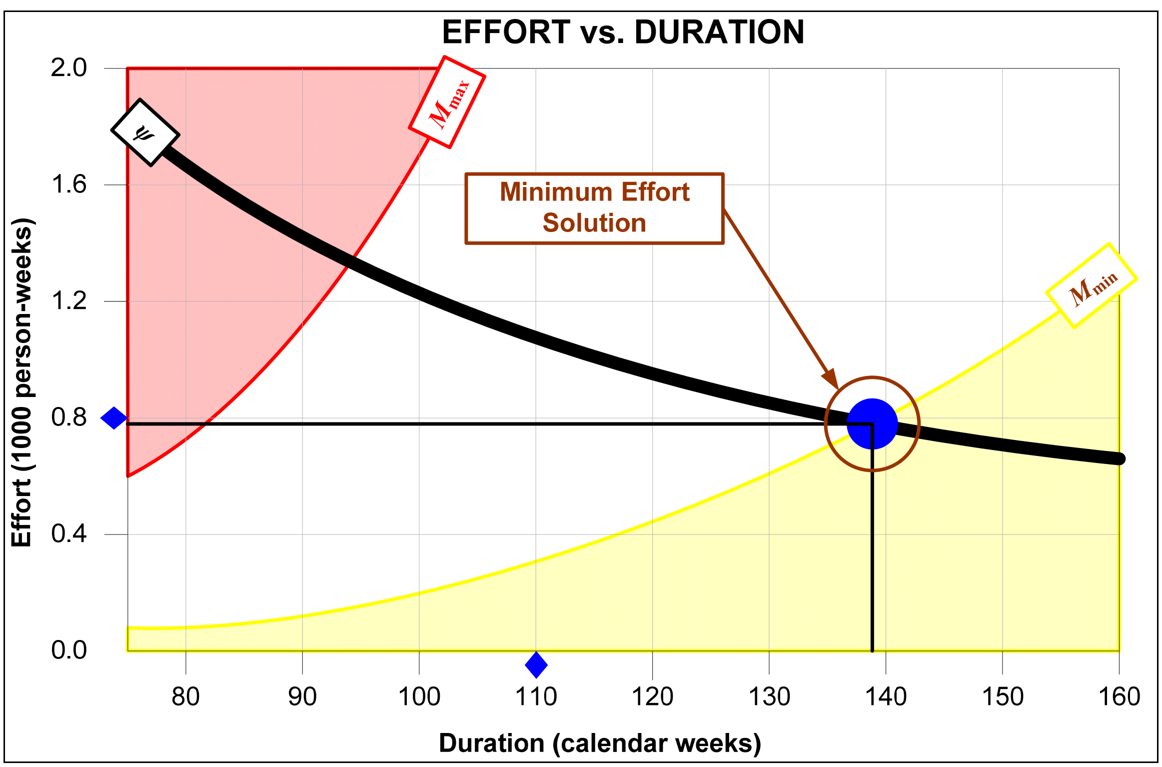

Figure 6. Minimum Effort Solution

Typical (Nominal Stress) Solution—**Figure 7** shows the blue solution circle positioned on what is referred to as the typical, nominal, or average stress solution. Note that the solution occurs at the intersection of the estimating relationship curve and the nominal stress curve. Mathematically, the x (duration) coordinate can be found by substituting Equation **(8)** instantiated with $t_{p\,\mathrm{nom}}$ and $E_{p\,\mathrm{nom}}$ into Equation **(4)** and solving for duration.

$$M_{\mathrm{nom}} = \frac{\left(\psi t_{p\,\mathrm{nom}}{}^{-\alpha_t}\right)^{\frac{1}{\alpha_E}}}{t_{p\,\mathrm{nom}}{}^{\gamma}}$$
$$t_{p\,\mathrm{nom}} = \left(\frac{\psi}{M_{\mathrm{nom}}{}^{\alpha_E}}\right)^{\frac{1}{\gamma\alpha_E+\alpha_t}} \quad \textbf{(14)}$$

The y (effort) coordinate can then be found by solving Equation **(4)** for effort; i.e., projecting duration off of the nominal stress curve onto the y (effort) axis.

$$M_{\mathrm{nom}} = \frac{E_{p\,\mathrm{nom}}}{t_{p\,\mathrm{nom}}{}^{\gamma}}$$
$$E_{p\,\mathrm{nom}} = M_{\mathrm{nom}} t_{p\,\mathrm{nom}}{}^{\gamma} \quad \textbf{(15)}$$

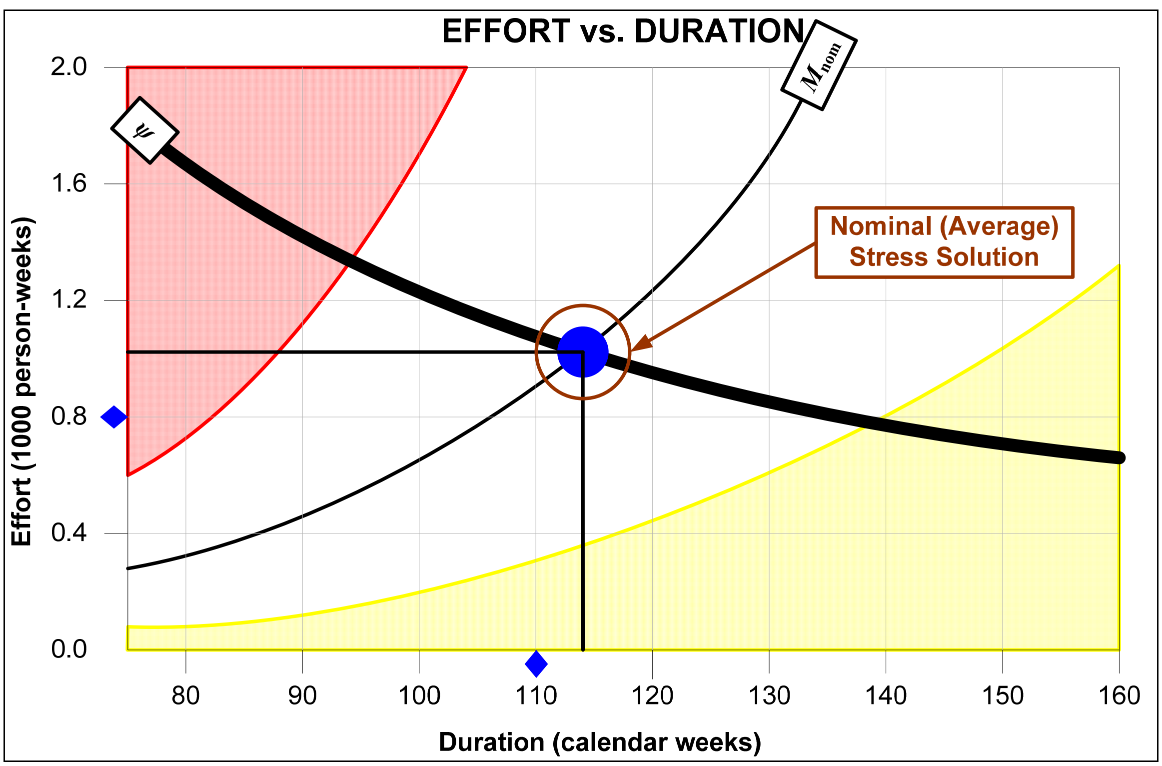

Figure 7. Nominal (Average) Stress Solution

Notice that the Ross Chart symbology makes it easy to determine whether or not a particular solution satisfies its goals. Notice also that from **Figure 7** we can deduce that there is no solution in this example scenario that can satisfy both goals; i.e., there is no point on the estimating relationship curve where the estimated values for both effort and duration are less than or equal to their corresponding goal values. We refer to this kind of circumstance as an ***over-constrained management situation***.

Content/Efficiency Ratio as a Random Variable

Up until this point, we have been treating the independent variables effective software content S_e, specific efficiency η, and the content/efficiency ratio ψ as certain; i.e., single-point values. Unfortunately, until project completion, we have exact values for neither effective content nor specific efficiency; these values are ***uncertain***; i.e., they have a range of possible outcomes. We, therefore, choose to represent content and efficiency as random variables $\boldsymbol{S_e}$ and $\boldsymbol{\eta}$, and instantiation becomes selecting an appropriate distribution function for each.[6] From this we write the ratio of these two random variables $\boldsymbol{\Psi}$, itself a random variable, as

$$\boldsymbol{\Psi} = \frac{\boldsymbol{S_e}}{\boldsymbol{\eta}} \qquad \textbf{(16)}$$

The choice of specific distributions for $\boldsymbol{S_e}$ and for $\boldsymbol{\eta}$ is a subject worthy of debate and a future paper. For convenience sake we have chosen to model both of these random variables as being triangularly distributed. Triangular distributions have the advantages of being mathematically simple, having a finite range, able to roughly approximate a Gaussian (normal) distribution, and able to model skew. Regardless of the distributions chosen, we need some way to determine the Cumulative Distribution Function (CDF) $\boldsymbol{D}(\psi_{\boldsymbol{\Psi}})$ of our content/efficiency ratio $\boldsymbol{\Psi}$. Finding a neat closed-form CDF that is the quotient of two random variables, triangularly distributed or otherwise, is problematic at best. We therefore recommend using Monte Carlo methods to determine the CDF of the content/efficiency ratio. This process is summarized as follows:

(1) Create ***randomly-ordered*** n-element vectors of distributed (triangularly or otherwise) possible outcomes $\mathbf{S_e}$ and $\boldsymbol{\eta}$ for each of effective content $\boldsymbol{S_e}$ and specific efficiency $\boldsymbol{\eta}$.

(2) Compute an n-element vector $\boldsymbol{\psi}$ for the content/efficiency ratio $\boldsymbol{\Psi}$ that is the vector quotient of $\mathbf{S_e}$ and $\boldsymbol{\eta}$ where the vector quotient is defined as

$$\boldsymbol{\psi} = \frac{\mathbf{S_e}}{\boldsymbol{\eta}} \equiv \begin{bmatrix} S_{e1}/\eta_1 \\ S_{e2}/\eta_2 \\ \vdots \\ S_{en}/\eta_n \end{bmatrix} \qquad \textbf{(17)}$$

(3) Use quantization (binning) and hit counting applied to the content/efficiency ratio vector $\boldsymbol{\psi}$ to produce a frequency distribution vector $\mathbf{F}_{\boldsymbol{\psi}}$ where each vector element consists of the bin lower and upper boundary values and a hit count value.

[6] We use the Arial bold italic typeface to indicate a random variable.

(4) Use accumulation (numerical integration) applied to the content/efficiency ratio frequency distribution vector $\mathbf{F}_{\Psi}$ to produce an ascending-sorted CDF vector $\mathbf{D}_{\Psi}$ where each vector element consists of a content/efficiency ratio value and its associated confidence probability (the probability that the associated content/efficiency ratio value will be greater than or equal to the actual final outcome value).

Analyzing estimating relationships such as the one previously described where the independent variables are uncertain and must be treated as random variables is a difficult, tedious, and time-consuming process when using a calculator or a spreadsheet. The number of variables combined with the non-intuitive nature of probabilistic mathematics makes it virtually impossible to analyze the solution space in a timely fashion. It is also difficult to present the results of such an analysis in a way that others can understand and accept. We therefore include, as part of our Ross Chart definition, interactive dynamic features for analyzing and presenting probabilistic bivariate estimating relationships as described in the following paragraphs; these features being well-suited for implementation as part of a dedicated software application.

5. Ross Charts and Uncertainty

Content/Efficiency Ratio Confidence Probability

Recall that the relationship between our two dependent variables effort and duration is based on the expected content/efficiency ratio which we now treat as a random variable. We begin by refining the definition of our estimating relationship curve shown in all of the figures thus far to represent the 50% probability (median) value of the content/efficiency ratio $\tilde{\psi}$. In other words, each point on the curve represents an effort-duration solution where there is a 50% probability that the final actual outcomes for both effort *and* duration will be less than or equal to their corresponding estimated values. We therefore rewrite Equation **(8)**, the equation of the estimating relationship curve, to reflect this 50% confidence probability assumption

$$E_{p50\%} = \left(\mathbf{D}_{\Psi}^{-1}\left[50\%\right] t_{p50\%}{}^{-\alpha_t}\right)^{\frac{1}{\alpha_E}} \qquad \textbf{(18)}$$

where

$\mathbf{D}_{\Psi}^{-1}\left[50\%\right]$ = Inverse-indexing the content/efficiency ratio CDF vector $\mathbf{D}_{\Psi}$ with a probability of 50% to yield the associated content/efficiency ratio value.

Focusing now on **Figure 8**, the current solution, as represented by the blue solution circle and its projection lines, is one of these 50% probability solutions. Because of the probabilistic nature of the estimating relationship, there is actually a family of estimating relationship curves, each curve corresponding to a different confidence probability of the content/efficiency ratio.

$$E_{p\phi} = \left(\mathbf{D}_{\Psi}^{-1}\left[\phi\right] t_{p\phi}{}^{-\alpha_t}\right)^{\frac{1}{\alpha_E}} \qquad \textbf{(19)}$$

where

ϕ = Confidence Probability

Figure 8 illustrates seven specific members of this family; the 50% probability curve (solid black line) and six additional member curves (dashed black lines) representing 1%, 10%, 30%, 70%, 90%, and 99% probabilities. While the number curves in this family is infinite, we have chosen this few to help illustrate the method we will use to transform uncertainty about content and efficiency into confidence associated with estimated values for effort and duration.

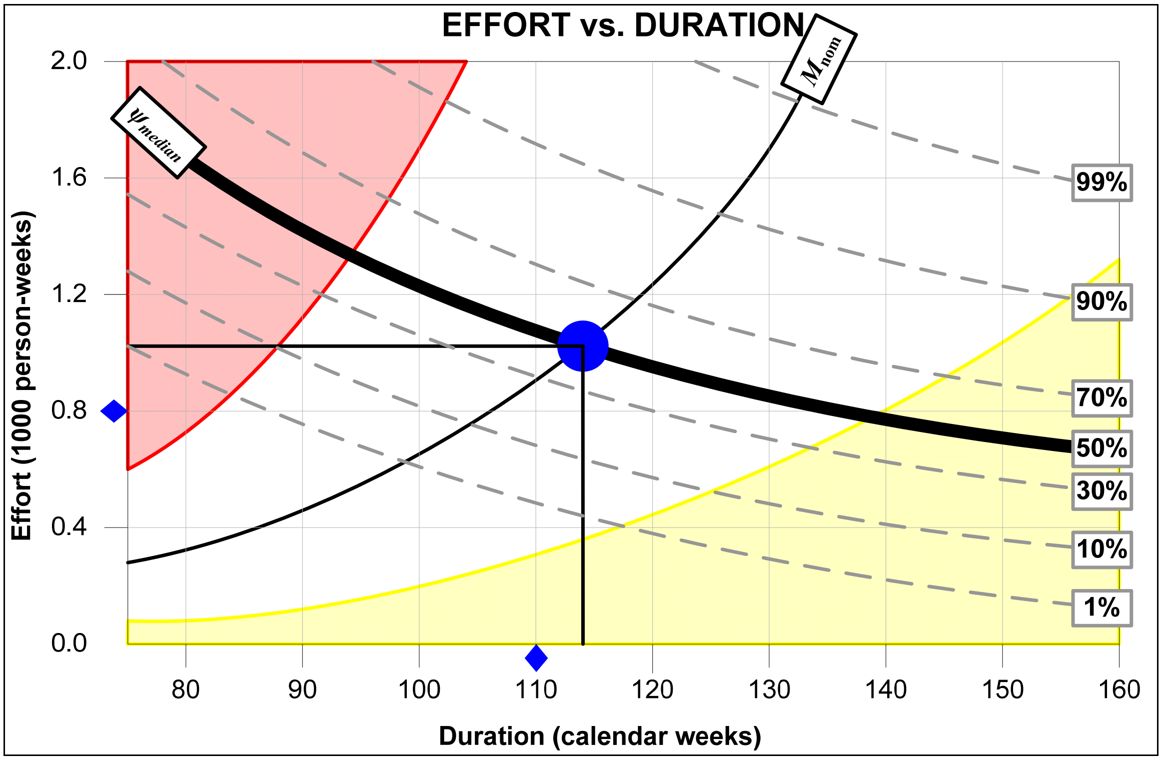

Figure 8: Family of Content/Efficiency Estimating Relationship Curves

Projecting Uncertainty

We now generalize the previously-described projection process to be applicable to the entire family of effort-duration estimating relationship curves. The x (duration) coordinate of a solution with management stress M for any confidence probability of the content/efficiency ratio can be found by substituting Equation **(19)** into Equation **(4)** and solving for duration.

$$M = \frac{\left(\mathbf{D}_{\Psi}^{-1}[\phi] t_{p\phi}{}^{-\alpha_t}\right)^{\frac{1}{\alpha_E}}}{t_{p\phi}{}^{\gamma}}$$

$$t_{p\phi} = \left(\frac{\mathbf{D}_{\Psi}^{-1}[\phi]}{M^{\alpha_E}}\right)^{\frac{1}{\gamma\alpha_E+\alpha_t}} \qquad \textbf{(20)}$$

The y (effort) coordinate can then be found by solving Equation **(4)** for effort; i.e., projecting duration off of the management stress curve (its position determined by M) onto the y (effort) axis.

$$M = \frac{E_{p\phi}}{t_{p\phi}^{\ \gamma}}$$

$$E_{p\phi} = M t_{p\phi}^{\ \gamma} \tag{21}$$

Focusing on the situation illustrated in **Figure 8**, we use the projection process defined by Equations **(20)** and **(21)** to map content/efficiency ratio (input) uncertainty to duration and effort (output) confidence probability. We use probability values between 1% and 99% to create a spectrum on each axis that reflects the cumulative distribution function (CDF) for the associated dependent variable. **Figure 9** shows these spectra displayed as what we choose to call dynamic cumulative distribution range symbols (decade-indexed black bar on each axis); dynamic since they move and change width as the blue solution circle is moved along the estimating relationship curve.[7]

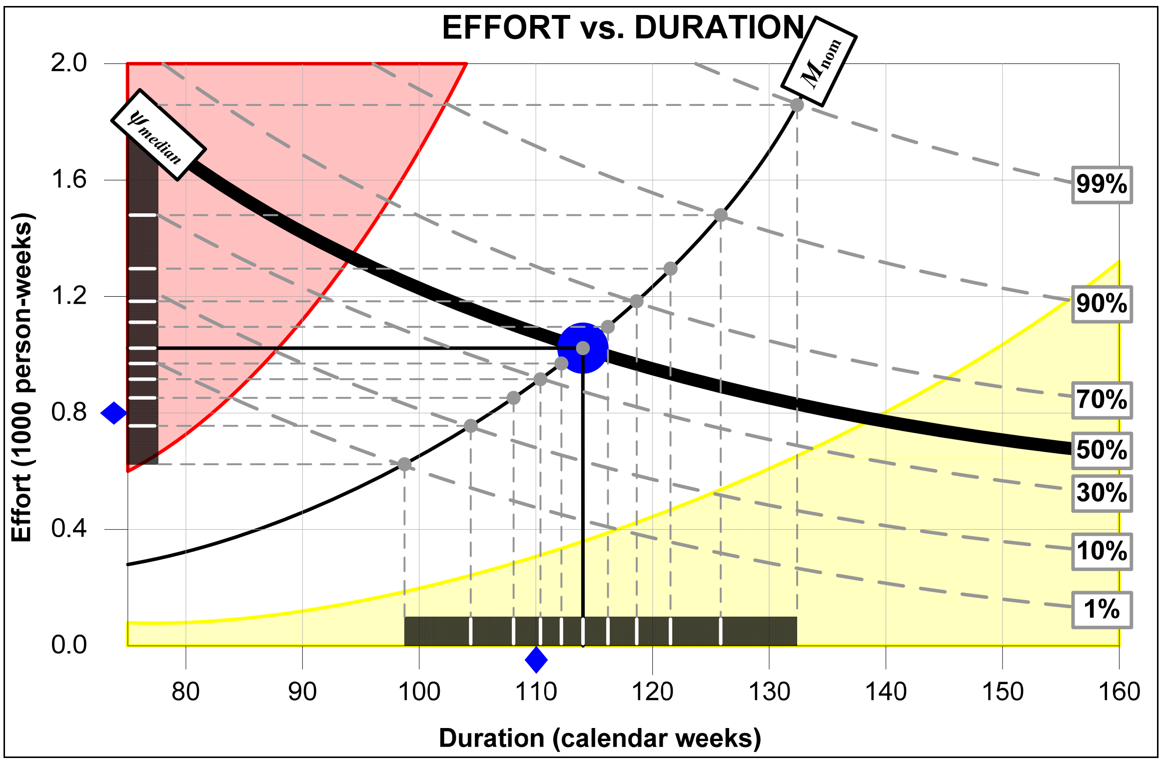

Figure 9: Cumulative Distribution Range Bars

Desired Confidence

We have already defined the notion of a constraint and described one of its two constituents, the goal. The other constituent is the desired confidence. It specifies the desired probability that the actual final outcome of the associated dependent variable will be less than or equal to the goal value.

[7] The *projection of uncertainty* process used in this paper is inspired by (Browne, 2001 pp. 6-10).

Figure 10 shows the desired confidence values associated with each constraint as interactive dynamic desired confidence symbols (blue confidence triangles) on each associated cumulative distribution range bar. Each represents the desired probability of success associated with its corresponding dependent variable. If the dependent variable value associated with the location of its blue confidence triangle is greater than the goal value, then the associated cumulative distribution range bar turns red; otherwise it is green. In other words, if the current solution does not meet the conditions of a particular constraint (the goal cannot be met with the desired probability of success) then the associated range bar will be red.

In our example, the desired confidence for both effort and duration are set to 70%; however, they could have been set to any desired probability value and need not have been set to the same value.

Interpreting the Ross Chart in **Figure 10**, we conclude that the typical (nominal stress) solution satisfies neither the effort goal of 800 person-weeks with a desired confidence of 70% nor the duration goal of 110 weeks with a desired confidence of 70%.

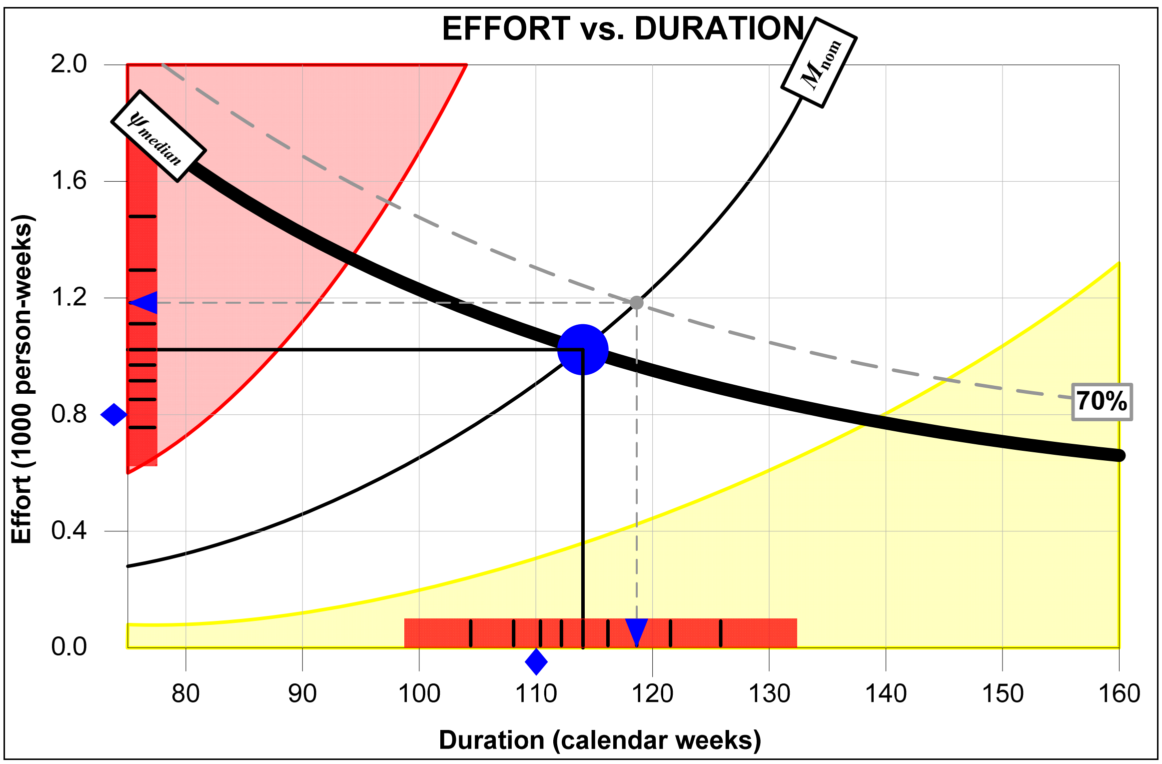

Figure 10. Desired Confidence Symbols

It is important to note here that the distributions of probabilities on the range bars in this example are neither symmetrical nor linear due to the nonlinear and skewed nature of the content and efficiency distributions involved.[8] This explains why the projection lines from the current solution

[8] In this example effective content is estimated as [Low = 25,000, Most Likely = 31,000, High = 51,500] SLOC and specific efficiency is estimated as [Low = 17.89, Most Likely = 22.79, High = 29.03] unitless, both triangularly distributed.

circle, which represent median (50% probability) values, do not intersect their associated cumulative distribution range bars at the center of their respective ranges.

Exploring the Solution Space

We can see that the solution shown in Figure 10 does not satisfy the duration constraint (the duration range bar is red). What would it take to satisfy this constraint? One possibility is to accept more management stress by dragging the blue solution circle to the left along the estimating relationship curve until the duration goal of 110 calendar weeks is satisfied with a confidence probability of 70% (duration range bar turns green) as shown in **Figure 11**. Notice that this action has made the situation worse with respect to satisfying the effort constraint.

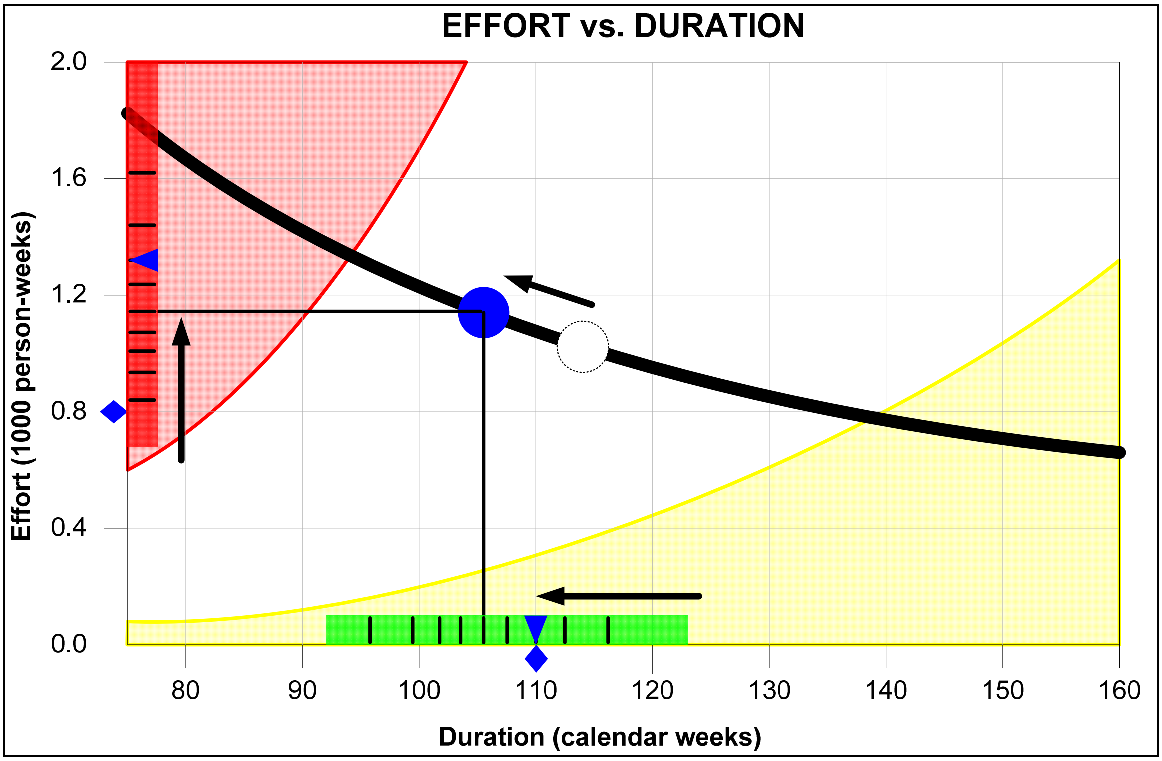

Figure 11: Compressing the Time to Meet the Duration Constraint

Suppose, in our example, we are able to negotiate a staffing budget increase from 800 person-weeks to 1,200 person-weeks and our stakeholders are willing to accept a riskier 50% desired confidence for effort instead of the original 70%. We can illustrate these changes by moving the effort goal (dragging the blue effort goal diamond up) to 1,200 person-weeks and moving the effort desired confidence (dragging the blue effort confidence triangle down) to 50% as shown in **Figure 12**. Note that these changes cause the effort range bar to turn green since there is now greater than a 50% probability that we can satisfy the new effort goal of 1,200 person-weeks. We now have a satisfactory solution; i.e., we have a solution where both management constraints have been satisfied (both range bars are now green).

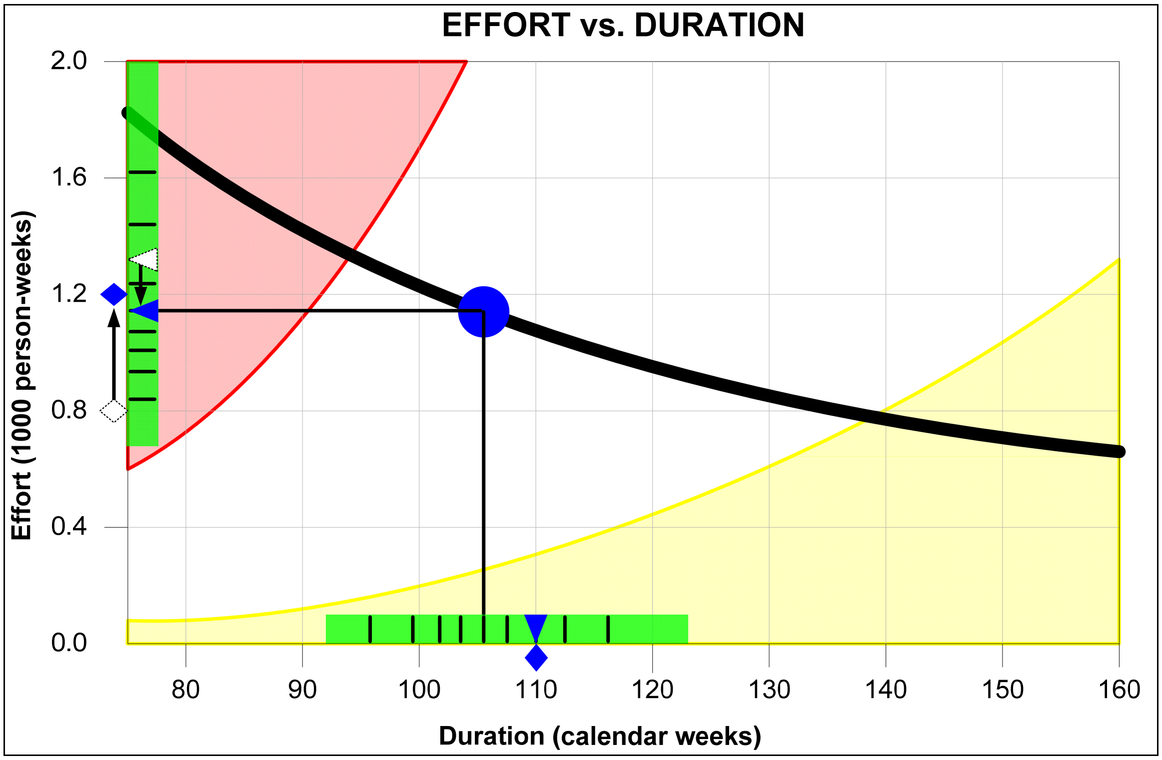

Figure 12: Relaxing the Effort Goal and Desired Confidence

6. EXAMPLE SOFTWARE PROJECT ESTIMATING SCENARIO

Assumptions and Constraints

To demonstrate the application and capabilities of Ross Charts, we present a proposed project with the following assumptions and constraints:

- Typical aerospace contractor developing real-time embedded software.
- Typical efficiency and associated uncertainty for this business sector.
- Typical defect vulnerability and associated uncertainty for this business sector.
- Effective software size estimate:
  [45,000; 50,000; 60,000] SLOC (triangular).
- Cost of labor assumptions:
  40 person-hours per person-week; $100 per person-hour.
- Duration constraint:
  Goal ≤ 104 weeks (24 months);
  Confidence ≥ 80%.
- Effort constraint:
  Goal ≤ 2,000 person-weeks (40,000 person-hours);
  Confidence ≥ 50%.

- Cost constraint:
  Goal ≤ $10,000,000;
  Confidence ≥ 80%.
- Defects constraint:
  Goal ≤ 50 defects remaining at delivery;
  Confidence ≥ 90%.

We use the *r2ESTIMATOR*™ estimating tool[9] and its Ross Chart capabilities to show various estimation alternatives for the above-described project. Note that in the automated implementation, the blue solution circle, blue confidence triangles, and blue goal diamonds are all draggable.

[9] *r2ESTIMATOR*™ is developed and distributed by *r2ESTIMATING*®, *LLC*; http://www.r2estimating.com.

Initial Examination of the Solution Space

Average Management Stress Solution*— Figure 13** (with **Table 1**) is a synchronized set of three Ross Charts that are displaying the average management stress solution for the above-described scenario. Note that the duration and delivered defects constraints are ***not satisfied. On the other hand, the effort and cost constraints *are* satisfied. Accepting this solution implies accepting a delivered defects confidence of 78.2% and a duration confidence of less than 1% as shown in **Table 1**.

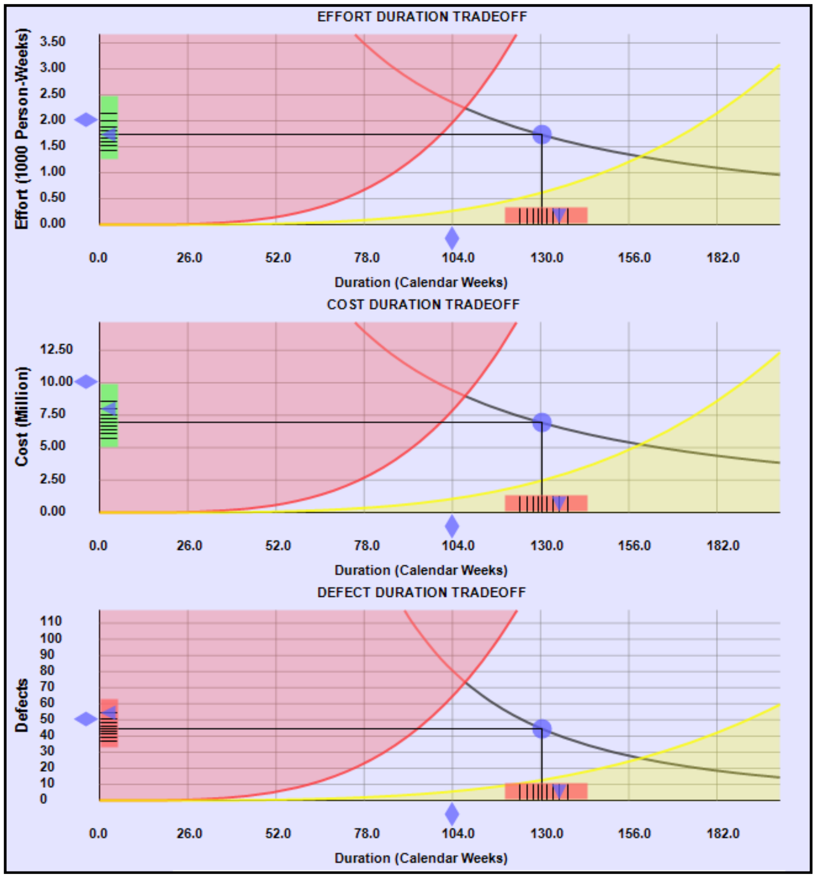

Figure 13: Synchronized Ross Charts: Typical (Nominal Management Stress)

Table 1: Project Metrics – Typical

	Expected Value	Goal Value	% Goal Probability	% Risk Tolerance	Value at Risk Tolerance
Duration (Weeks)	130.8	104.0	0.0	80.0	135.7
Effort (PW)	1.7217K	2.0000K	81.4	50.0	1.7217K
Effort (PH)	68.866K	80.000K	81.4	50.0	68.866K
Cost	6.8867M	10.000M	99.3	80.0	7.9176M
Defects	44	50	78.2	90.0	54

Minimum Duration Solution*— Figure 14** (with **Table 2**) is a synchronized set of three Ross Charts that are displaying the minimum duration solution for the above-described scenario. Note that all of the constraints are ***not satisfied. Accepting this solution implies accepting an effort confidence of 22.7%, a cost confidence of 74.3%, a delivered defects confidence of less than 1%, and a duration confidence of 19.7% as shown in **Table 2**. We can conclude from this solution that, regardless of what happens to effort, cost, and delivered defects, extreme schedule compression will still not satisfy the aggressive duration constraint.

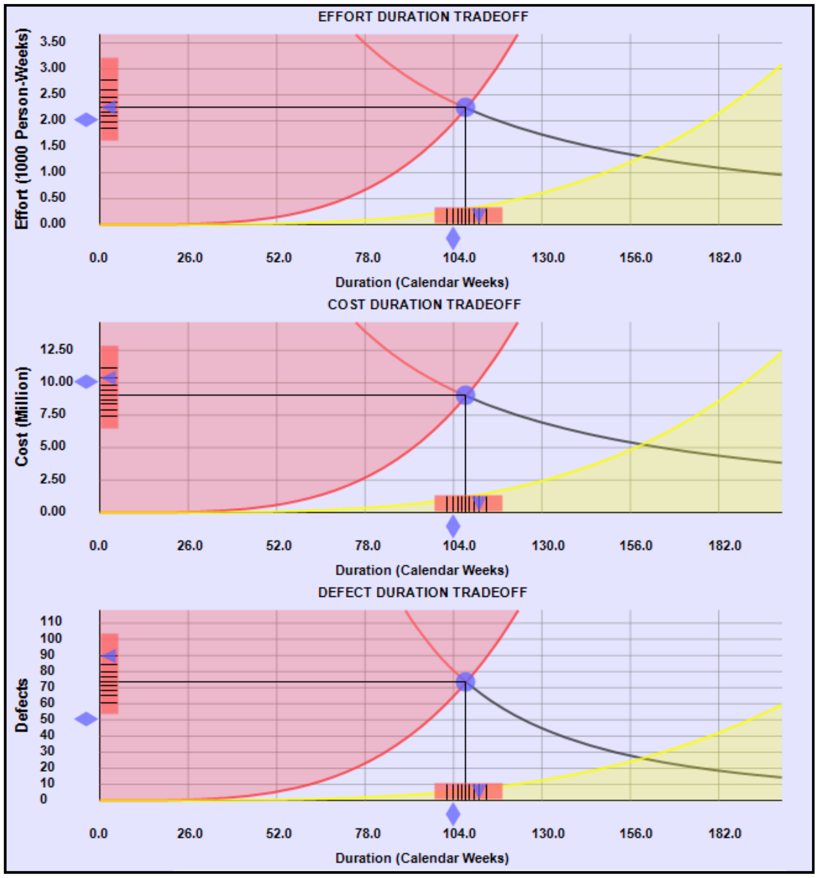

Figure 14. Synchronized Ross Charts: Minimum Duration

Table 2: Project Metrics – Minimum Duration

	Expected Value	Goal Value	% Goal Probability	% Risk Tolerance	Value at Risk Tolerance
Duration (Weeks)	107.7	104.0	19.7	80.0	111.9
Effort (PW)	2.2430K	2.0000K	22.7	50.0	2.2430K
Effort (PH)	89.719K	80.000K	22.7	50.0	89.719K
Cost	8.9719M	10.000M	74.3	80.0	10.315M
Defects	73	50	0.1	90.0	89

***Minimum Effort Solution*— Figure 15** (with **Table 3**) is a synchronized set of three Ross Charts that are displaying the minimum effort solution for the above-described scenario. Note that the effort, cost, and delivered defects constraints are all comfortably satisfied; however, our duration situation is much worse; i.e., we have less than a 1% chance of satisfying the duration constraint.

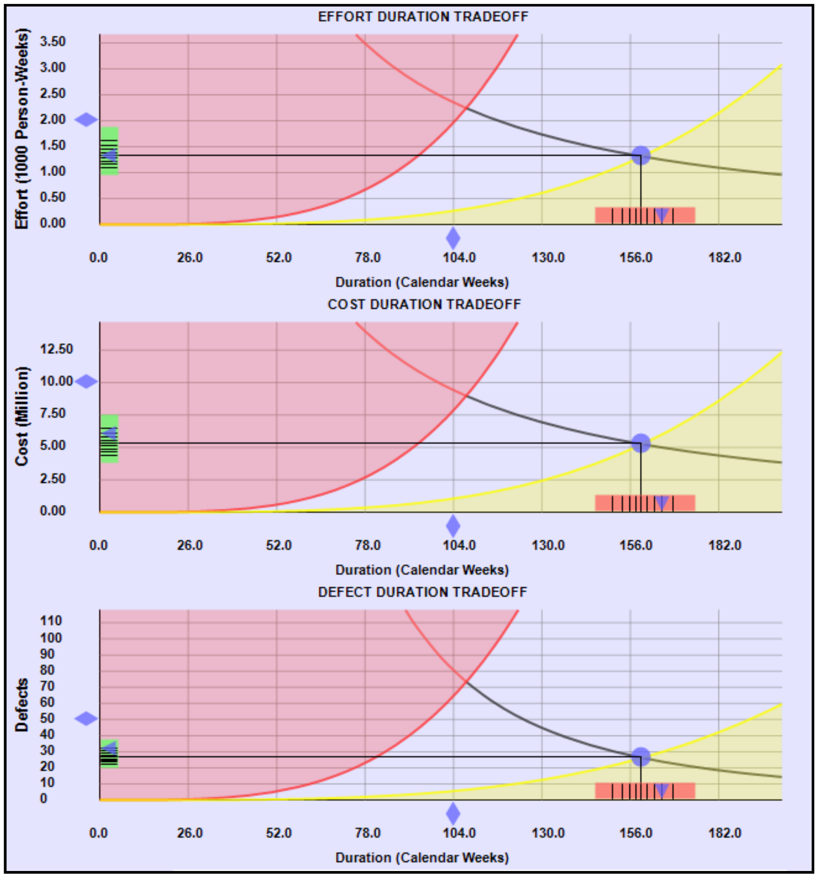

Figure 15: Synchronized Ross Charts: Minimum Effort

Table 3: Project Metrics – Minimum Effort

	Expected Value	Goal Value	% Goal Probability	% Risk Tolerance	Value at Risk Tolerance
Duration (Weeks)	159.5	104.0	0.0	80.0	165.6
Effort (PW)	1.3127K	2.0000K	99.9	50.0	1.3127K
Effort (PH)	52.506K	80.000K	99.9	50.0	52.506K
Cost	5.2506M	10.000M	100.0	80.0	6.0366M
Defects	26	50	100.0	90.0	32

Identifying and Analyzing Solution Alternatives

Identifying Solution Alternatives— The preceding three solution alternatives illuminate an all-too-common situation where, for the given size and efficiency, the goals and the desired probabilities are mutually impossible to achieve. We described this set of circumstances earlier as an over-constrained situation. In this case, an overly-aggressive duration goal with an 80% desired probability of success cannot be achieved without compressing the schedule beyond the minimum duration limit. This extreme schedule compression also increases effort, cost, and delivered defects to the point that their respective constraints cannot be satisfied. Over-constrained situations present an interesting challenge to the cost analyst and program manager. It is generally not a good political strategy to present findings that conclude a particular project *can't be done*. It is much more prudent to offer alternative strategies that allow the decision-makers to compromise and make fact-based choices.

A reasonable way to list potential solution alternatives is to consider changing each of the project's assumptions (estimating relationship independent variables) and constraints (goals with desired confidence probabilities), individually, and in combination. For our example project, a partial list of alternatives might look like the following:

- Change Assumptions
    - Reduce the effective software size (i.e., postpone or eliminate functionality),
    - Reduce the uncertainty range around effective software size (i.e., refine the size estimate and secure functionality freezes to reduce variability and potential for growth),
    - Increase efficiency (i.e., better people, better processes/tools, less complex product, etc.),
    - Reduce the uncertainty range around efficiency (i.e., lock down decisions about the product technology and the development environment).
- Change Constraints
    - Relax one or more of the goal values,
    - Relax one or more of the desired confidence probabilities.

The number of possible alternatives is virtually endless; we analyze five of these in the following pages.

Reduced Functionality Solution— *"How much functionality would we have to postpone or eliminate in order to satisfy the given effort, cost, delivered defects, and duration constraints?"* **Figure 16** (with **Table 4**) is a synchronized set of three Ross Charts that are displaying a reduced effective software size solution for the above-described scenario. By postponing or eliminating 30% of the original functionality (i.e., reducing the effective software size to [31,500; 35,000; 42,000] SLOC), we can satisfy all of the original constraints.

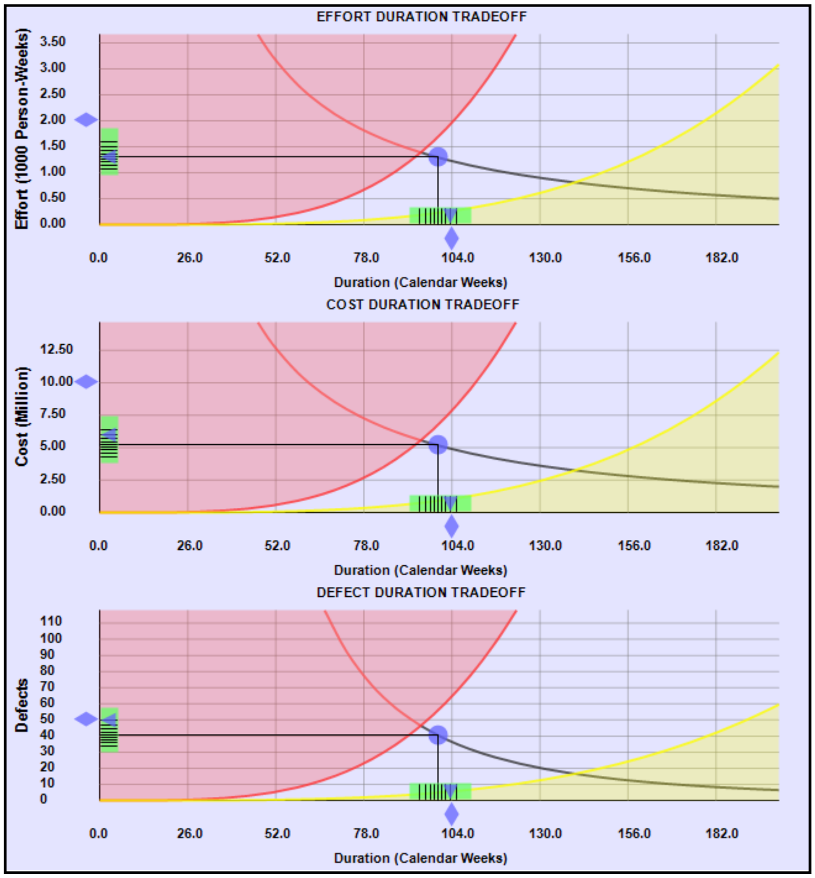

Figure 16: Synchronized Ross Charts:
Reduced Effective software size

Table 4: Project Inputs & Project Metrics – Reduced Effective software size

	Low	Likely	High
Size (ESLOC)	31500	35000	42000
Efficiency	17.88633	22.78683	29.02996
Defect Vulnerability	0.46937	0.54605	0.63525

	Expected Value	Goal Value	% Goal Probability	% Risk Tolerance	Value at Risk Tolerance
Duration (Weeks)	100.0	104.0	80.9	80.0	103.8
Effort (PW)	1.2927K	2.0000K	99.9	50.0	1.2927K
Effort (PH)	51.708K	80.000K	99.9	50.0	51.708K
Cost	5.1708M	10.000M	100.0	80.0	5.9448M
Defects	40	50	92.0	90.0	49

Relaxed Duration Goal Solution— *"By how much would we have to slip the duration goal in order to preserve its 80% desired confidence probability while satisfying the given effort, cost, and delivered defects constraints?"* **Figure 17** (with **Table 5**) is a synchronized set of three Ross Charts that are displaying a relaxed duration goal solution for the above-described scenario. Slipping the duration goal by 30.8 weeks (a 30% schedule slip from 104 weeks to 134.8 weeks) while preserving its 80% confidence probability, we can satisfy all of the other original constraints.

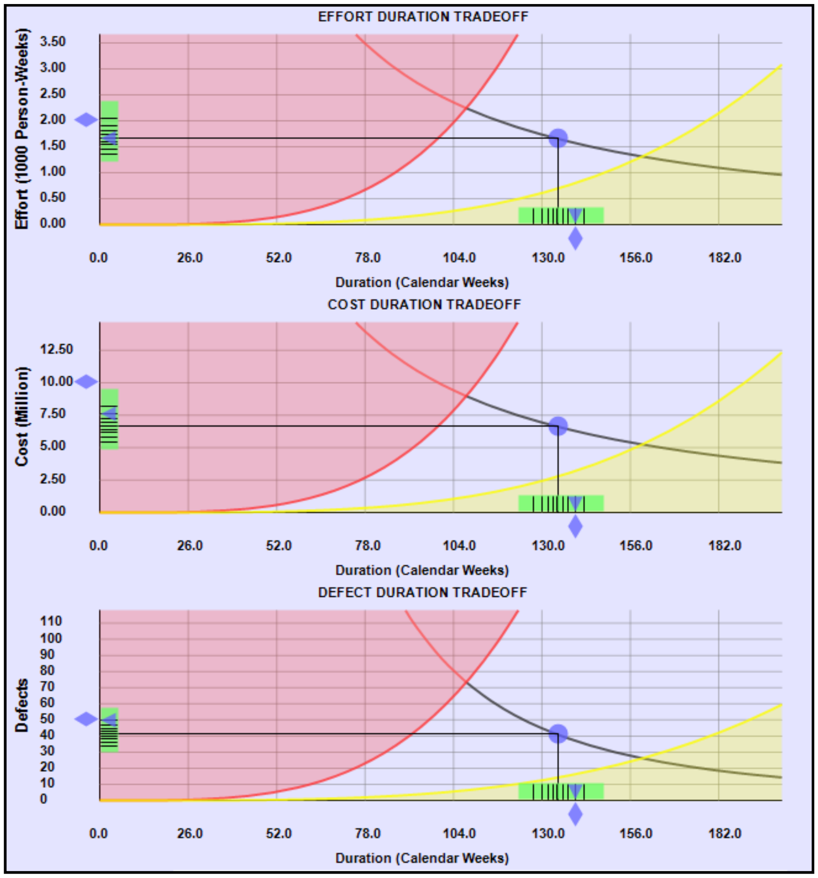

Figure 17. Synchronized Ross Charts: Relaxed Duration Goal

Table 5: Project Metrics – Relaxed Duration Goal

	Expected Value	Goal Value	% Goal Probability	% Risk Tolerance	Value at Risk Tolerance
Duration (Weeks)	134.8	140.0	80.3	80.0	139.9
Effort (PW)	1.6519K	2.0000K	87.8	50.0	1.6519K
Effort (PH)	66.076K	80.000K	87.8	50.0	66.076K
Cost	6.6076M	10.000M	99.8	80.0	7.5967M
Defects	41	50	90.6	90.0	50

High Duration Risk Solution— *"How much schedule risk must we accept in order to preserve the original duration goal while satisfying the given effort, cost, and delivered defects constraints?"* **Figure 18** (with **Table 6**) is a synchronized set of three Ross Charts that are displaying an extremely high duration risk solution for the above-described scenario. Satisfying the effort, cost, and delivered defects constraints within the 104-week duration goal implies a duration confidence of nearly zero. This is definitely not a good alternative.

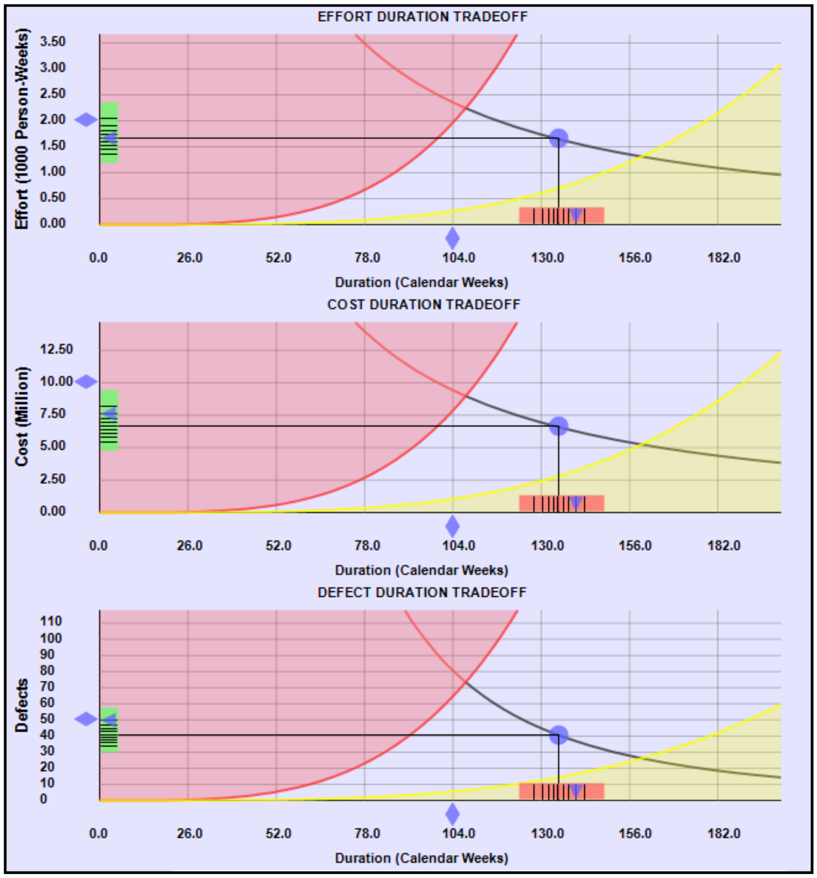

Figure 18. Synchronized Ross Charts: High Duration Risk

Table 6: Project Metrics – High Duration Risk

	Expected Value	Goal Value	% Goal Probability	% Risk Tolerance	Value at Risk Tolerance
Duration (Weeks)	135.1	104.0	0.0	80.0	140.3
Effort (PW)	1.6458K	2.0000K	88.4	50.0	1.6458K
Effort (PH)	65.833K	80.000K	88.4	50.0	65.833K
Cost	6.5833M	10.000M	99.8	80.0	7.5689M
Defects	41	50	91.4	90.0	49

Composite Solution #1— *"Show me a reasonable solution given that I really need to deliver something useful (at least 80% of the total functionality) and given that I can negotiate an additional 9 weeks into the schedule."* **Figure 19** (with **Table 7**) shows a composite solution with:

- Reduced functionality by postponing 20% to later release; reduced uncertainty by half by scrubbing and freezing requirements; [39,000; 40,000; 42,000] SLOC
- 113-week duration goal (9 week negotiated increase) with 80% confidence

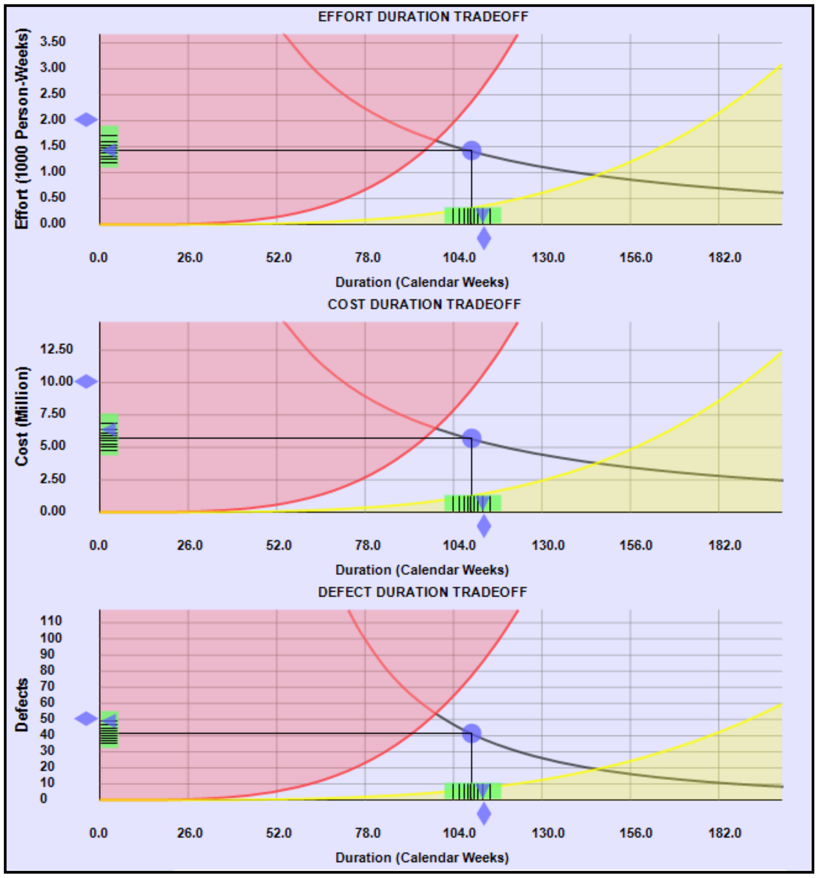

Figure 19. Synchronized Ross Charts: Composite Solution #1

Table 7: Project Inputs & Project Metrics – Composite Solution #1

	Low	Likely	High
Size (ESLOC)	39000	40000	42000
Efficiency	17.88633	22.78683	29.02996
Defect Vulnerability	0.46937	0.54605	0.63525

	Expected Value	Goal Value	% Goal Probability	% Risk Tolerance	Value at Risk Tolerance
Duration (Weeks)	109.3	113.0	80.5	80.0	112.9
Effort (PW)	1.4077K	2.0000K	100.0	50.0	1.4077K
Effort (PH)	56.308K	80.000K	100.0	50.0	56.308K
Cost	5.6308M	10.000M	100.0	80.0	6.3448M
Defects	41	50	92.4	90.0	49

Composite Solution #2— *"Given that I must deliver all of the functionality as fast as possible, to what values for effort, cost, and duration can I commit and maintain the associated desired confidence probabilities?"* **Figure 20** (with **Table 8**) shows a composite solution with the goals:

- 111.9-week duration; 80% confidence
- 2,243 person-week effort; 50% confidence
- $10,315,000 cost; 80% confidence
- 90 delivered defects; 90% confidence

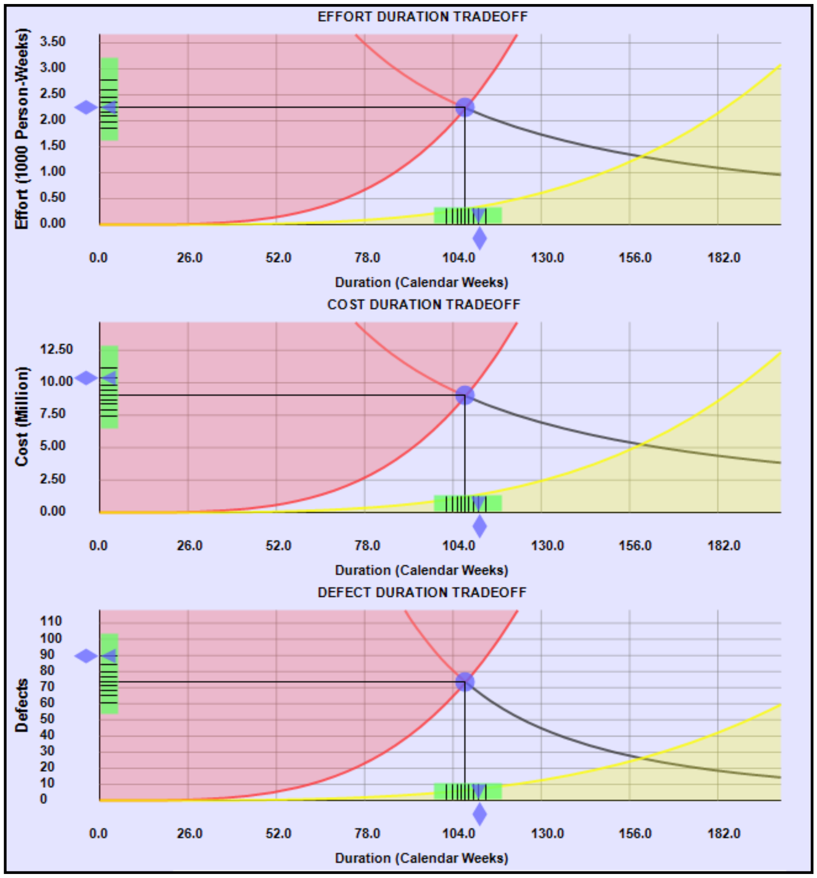

Figure 20. Synchronized Ross Charts: Composite Solution #2

Table 8: Project Metrics – Composite Solution #2

	Expected Value	Goal Value	% Goal Probability	% Risk Tolerance	Value at Risk Tolerance
Duration (Weeks)	107.7	111.9	80.2	80.0	111.9
Effort (PW)	2.2430K	2.2430K	50.0	50.0	2.2430K
Effort (PH)	89.719K	89.720K	50.0	50.0	89.719K
Cost	8.9719M	10.315M	80.0	80.0	10.315M
Defects	73	90	90.1	90.0	89

7. SUMMARY AND CONCLUSION

Purpose Revisited

This paper describes a new chart (called a Ross Chart) that is a graphical display of the goals satisfaction and confidence (probabilities of success) of two dependent random variables a bivariate estimating relationship. It has a unique set of display elements coupled with interactive dynamic properties for analyzing and presenting this type of relationship.

Areas for Further Study

The following are suggestions for furthering Ross Chart capability:

- Consider mechanisms for allowing interactive dynamic changes to the estimating relationship independent variables; i.e., allow the curve itself to move in real time as a result of making changes to its constituent independent variables.
- Consider a linkage to an historical project database with an ability to select relevant past projects and display them as points in scatter plot fashion on the Ross Chart.
- Examine the notion of expanding Ross Chart display elements and dynamic behavior to multi-dimensional chart forms (e.g., radar charts) in order to display probabilistic *multivariate* estimating relationships.

REFERENCES

Bernstein, Peter L. 1996. *Against the Gods: The Remarkable Story of Risk.* New York City : John Wiley & Sons, Inc., 1996. ISBN 0-471-29563-9 (paper).

Browne, John. 2001. Probabilistic Design: Course Notes. Melbourne, Australia : Swinburne University of Technology, 2001.

National Aeronautics and Space Administration (NASA). 2004. *2004 NASA Cost Estimating Handbook (CEH).* Washington : NASA, 2004. Online Handbook: http://ceh.nasa.gov.

Ross, Michael A. 2007b. Next Generation Software Estimating Framework: 25 Years and Thousands of Projects Later. *Proceedings, Joint ISPA / SCEA 2007 Conference.* New Orleans, LA, USA : The International Society of Parametric Analysts and The Society of Cost Estimating and Analysis, June 2007b.

Sambur, Marvin R. and Teets, Peter B. 2004. Revitalizing the Software Aspects of Systems Engineering. *Memorandum for See Distribution.* Washington, District of Columbia, USA : Under Secretary of the Air Force, September 20, 2004. 04A-003.

BIOGRAPHY

Michael A. Ross *has over 30 years of experience in software engineering as a developer, manager, process champion, consultant, instructor, and award-winning international speaker. Mr. Ross is currently the President and CEO of r2Estimating, LLC. Mr. Ross's previous experience includes three years as Chief Engineer of Galorath Inc. (makers of the SEER suite of estimation tools), seven years with Quantitative Software Management, Inc. (makers of the SLIM suite of software estimating tools) where he was Vice President of Education Services, and 17 years with Honeywell Air Transport Systems (formerly Sperry Flight Systems) and 2 years with Tracor Aerospace where he developed or managed the development of embedded software for avionics systems installed various commercial airplanes and for expendable countermeasures systems installed in various military aircraft. He also co-founded Honeywell Air Transport Systems' SEPG, served as its focal for software project management process improvement, and served as a Honeywell corporate SEI CMM assessor. Mr. Ross did his undergraduate work at the United States Air Force Academy and Arizona State University, receiving a Bachelor of Science in Computer Engineering.*