Instantiating the r2 Software Estimating Framework for COCOMO: Integrating Duration, Effort, Cost, Defects, and Uncertainty

Mike Ross
r2Estimating, LLC
7755 E. Evening Glow Dr.
Scottsdale, AZ 85262-1295
480-488-8366
mike.ross@r2estimating.com
http://www.r2estimating.com

[1,2,3]***Abstract***—This paper references the basis, assumptions, and derivations of the r2 Software Estimating Framework (r2SEF) and then shows this framework instantiated for the COnstructive COst MOdel (COCOMO). The r2SEF is a set of general software effort, duration, and defects estimating relationships that are based on the notion that 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 people, project, and product attributes, and that, prior to project completion, ***everything is uncertain***. It is the author's contention that a primary goal of any thorough project estimating process should 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 within some corresponding specified confidence limits (probabilities of success). Instantiating the r2SEF for COCOMO extends COCOMO's capabilities to include ***defect estimation***, to permit ***schedule versus effort versus cost versus defects tradeoffs*** within associated limits, and to provide ***confidence-driven estimating*** based on people, project, and product uncertainties.

TABLE OF CONTENTS

---

[2] Initial, October 18, 2007

1. INTRODUCTION

Purpose

This paper reviews the basis, assumptions, and derivations of the r2 Software Estimating Framework (r2SEF)™ [8] and then shows this framework instantiated for several forms of the Constructive Cost Model (COCOMO) [1] [2].

Scope

Instantiating the r2SEF for COCOMO extends COCOMO's capabilities to include ***defect estimation***, to permit ***schedule versus effort versus cost versus defects tradeoffs*** within associated limits, and to provide ***confidence-driven estimating*** based on people, project, and product uncertainties. While this paper focuses on an instantiation of the r2SEF for COCOMO, the derivation process described herein can be used to create instantiations for other models such as Jensen (Seer) [4] and Norden-Putnam-Rayleigh (SLiM®) [7] [6].

Background

It is the author's contention that a primary goal of any thorough project estimating process should be to not only yield estimated values for the key management metrics (duration, effort, cost, and defects); it should also indicate whether or not these estimated values satisfy their corresponding project goals within some corresponding specified confidence limits (probabilities of success) [9].

2. r2 SOFTWARE ESTIMATING FRAMEWORK

The r2SEF is a set of general software effort, duration, and defects estimating relationships that are based on the notion that 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 people, project, and product attributes, and that, prior to project completion, ***everything is uncertain***. [8]

Fundamental Software Productivity Equation[4]

$$E_C^{\alpha_E} t_C^{\alpha_t} = \frac{S_e}{k_\eta \eta} \tag{1}$$

where

E_C ::= Software Construction Effort[5]

t_C ::= Software Construction Duration[6]

---

[4] [8] p. 14
[5] [8] pp. 9-10
[6] [8] p. 9

α_E ::= Effort Exponent[7]

α_t ::= Duration Exponent[8]

S_e ::= Effective Software Size[9]

η ::= Specific Efficiency[10]

k_η ::= Fundamental Software Productivity Equation proportionality constant[11]

Fundamental Software Defect Propensity Equation[12]

$$E_C^{\varphi_E} t_C^{\varphi_t} = \frac{\Phi_{[a,b]}}{k_\delta \delta_{[a,b]}} \qquad [\varphi_t \le 0] \tag{2}$$

where

φ_E ::= Defect Effort Exponent[13]

φ_t ::= Defect Duration Exponent[14]

$\Phi_{[a,b]}$ ::= Defect Count[15]

$\delta_{[a,b]}$ ::= Specific Defect Vulnerability[16]

k_δ ::= Fundamental Software Defect Propensity Equation proportionality constant[17]

Fundamental Software Management Stress Equation[18]

$$M = \frac{E_C}{k_M t_C^{\gamma}} \tag{3}$$

where

γ ::= Gamma (Economy Exponent)[19]

---

7 [8] p. 11
8 [8] p. 11
9 [8] p. 10
10 [8] pp. 11-15
11 [8] p. 12
12 [8] p. 29
13 [8] p. 26
14 [8] p. 26
15 [8] p. 25
16 [8] pp. 26-29
17 [8] p. 27
18 [8] p. 18

M ::= Specific Management Stress[20]

k_M ::= Fundamental Software Management Stress Equation proportionality constant[21]

Typical (Nominal Stress) Equation[22]

$$M_{\text{nom}} = \frac{E_{C\text{nom}}}{k_M t_{C\text{nom}}{}^{\gamma}} \tag{4}$$

$E_{C\text{nom}}$ ::= Nominal (Typical) Software Construction Effort

$t_{C\text{nom}}$ ::= Nominal (Typical) Software Construction Duration

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

Minimum Duration Limit (Brooks' Law)[23]

$$M_{\max} \geq \frac{E_C}{k_M t_C{}^{\gamma}} \Rightarrow M_{\max} = \frac{E_{t_{C\min}}}{k_M t_{C\min}{}^{\gamma}} \tag{5}$$

$E_{t_{C\min}}$ ::= Software Construction Effort at Minimum Software Construction Duration[24]

$t_{C\min}$ ::= Minimum Software Construction Duration[25]

$M_{\max}$ ::= Maximum-Achievable Specific Management Stress[26]

Minimum Effort Limit (Parkinson's Law)[27]

$$M_{\min} \leq \frac{E_C}{k_M t_C{}^{\gamma}} \Rightarrow M_{\min} = \frac{E_{C\min}}{k_M t_{E_{C\min}}{}^{\gamma}} \tag{6}$$

$E_{C\min}$ ::= Minimum Software Construction Effort[28]

$t_{E_{C\min}}$ ::= Software Construction Duration at Minimum Software Construction Effort[29]

---

19 [8] pp. 16-17

20 [8] pp. 16-18

21 [8] p. 17

22 Instantiation of the Fundamental Software Management Stress Equation at the sample mean management stress value for a particular data set.

23 [8] pp. 20-21

24 [8] pp. 20-21

25 [8] pp. 20-21

26 [8] pp. 19-20

27 [8] pp. 22-23

28 [8] pp. 22-23

29 [8] pp. 22-23

$M_{\min}$ ::= Minimum Practical Specific Management Stress[30]

Note that k_η, k_δ, *and* k_M *in the above equations are proportionality constants that resolve the system of units being used; their values being unity when effort is measured in person-weeks and duration is measured in calendar weeks. For the purposes of this paper, we will consistently scale in weeks; therefore, these proportionality constants will always have a value of one and thus disappear from the subsequent equations in this paper.*

3. COCOMO 81 INSTANTIATION

Category[31] Data Elements

- Category Name: text string
- Category Id: GUID
- Defect Units: text string
- COCOMO-Form Effort Equation Exponent B : real
- COCOMO-Form Duration Equation Exponent A : real
- COCOMO-Form Effort Equation Scale Factor $C_{E\,\text{nom}}$: real
- COCOMO-Form Duration Equation Scale Factor $C_{t\,\text{nom}}$: real
- Defect Density Scale Vector **ù** $_{\textbf{Scale}}$: array [-3.0, -2.5…3.0] of real
- *future enhancement: move EM rating scale table to here*
- Profile List: list of **Profile**

Profile Data Elements[32]

- Profile Name: text string
- Profile Id: GUID
- Effort Multiplier Vector **EM** : vector of triangularly-distributed real random variables
- Minimum Duration Percentage $P_{\min}$: percentage
- Nominal Duration Percentage P_{nom} : percentage
- Maximum Duration Percentage $P_{\max}$: percentage
- Defect Density ***ù*** $_{\boldsymbol{B}}$: triangularly-distributed real random variable

---

[30] [8] pp. 22-23

[31] "Category", within the context of this paper, refers to a collection of data elements that together describe a particular r2SEF calibration (typically the result of analyzing a particular historical data set or instantiating a particular estimating relationship/model). Default categories for COCOMO 81 include "Organic", "Semi-Detached", and "Embedded".

[32] "Profile", within the context of this paper, refers to a collection of data elements that together describe a particular project or set of closely related projects within an associated r2SEF Category.

Metrics Definitions

All of the COCOMO 81 metrics definitions contained in this section are taken from Ref. [1].

Metrics with Associated Rating Scales

These metrics are triangularly-distributed random variables that each represent an associated parameter (required software reliability, data base size, product complexity, execution time constraint, main storage constraint, virtual machine volatility, computer turnaround time, analyst capability, applications experience, programmer capability, virtual machine experience, programming language experience, use of modern programming practices, and use of software tools) where

$$\begin{array}{l} <\textbf{\textit{metric}}>=\left[Low \quad Most\ Likely \quad High \right] \\ Low \le Most\ Likely \le High \end{array} \tag{7}$$

with rating scales described below.

< *metric* >	VLO	LO	NOM	HI	VHI	XHI
RELY	0.75	0.88	1.00	1.15	1.40	
DATA		0.94	1.00	1.08	1.16	
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
TIME			1.00	1.11	1.30	1.66
STOR			1.00	1.06	1.21	1.56
VIRT		0.87	1.00	1.15	1.30	
TURN		0.87	1.00	1.07	1.15	
ACAP	1.46	1.19	1.00	0.86	0.71	
AEXP	1.29	1.13	1.00	0.91	0.82	
PCAP	1.42	1.17	1.00	0.86	0.70	
VEXP	1.21	1.10	1.00	0.90		
LEXP	1.14	1.07	1.00	0.95		
MODP	1.24	1.10	1.00	0.91	0.82	
TOOL	1.24	1.10	1.00	0.91	0.83	

(8)

Effort Multiplier Vector

Each effort multiplier vector element is a triangularly-distributed random variable, the scales of each having been described in (8) above.

$$\mathbf{EM} = \begin{bmatrix} \textbf{\textit{RELY}}, \textbf{\textit{DATA}}, \textbf{\textit{CPLX}}, \textbf{\textit{TIME}}, \textbf{\textit{STOR}}, \\ \textbf{\textit{VIRT}}, \textbf{\textit{TURN}}, \textbf{\textit{ACAP}}, \textbf{\textit{AEXP}}, \textbf{\textit{PCAP}}, \\ \textbf{\textit{VEXP}}, \textbf{\textit{LEXP}}, \textbf{\textit{MODP}}, \textbf{\textit{TOOL}} \end{bmatrix} \tag{9}$$

r2SEF Metric Assignments

Software Productivity Equation – r2SEF Random Variable Form

The random variable form of the r2SEF software productivity equation is

$$\boldsymbol{E_C}^{\alpha_E}\boldsymbol{t_C}^{\alpha_t} = \frac{\boldsymbol{S_e}}{\boldsymbol{ç}} \tag{10}$$

We multiplicatively combine the two equations from the COCOMO 81 model definition [1]

$$\boldsymbol{MM} = C_{E\,\mathrm{nom}}\boldsymbol{KEDSI}^{B}\prod_{i=1}^{14}\left(\mathbf{EM}_i\right) \tag{11}$$

and

$$\boldsymbol{TDEV} = C_{t\,\mathrm{nom}}\boldsymbol{MM}^{A} \tag{12}$$

to yield

$$\left(\boldsymbol{MM}\right)\left(\boldsymbol{TDEV}\right) = C_{E\,\mathrm{nom}}\boldsymbol{KEDSI}^{B}\prod_{i=1}^{14}\left(\mathbf{EM}_i\right)C_{t\,\mathrm{nom}}\boldsymbol{MM}^{A} \tag{13}$$

Converting COCOMO units (person-months, months, KSLOC) to r2SEF form units (person-weeks, weeks, SLOC), forcing the exponent on $\boldsymbol{S_e}$ to unity, and arranging the factors to be consistent with the r2SEF form yields

$$\begin{aligned}
&\left(\frac{12\times 7}{365.25}\right)\boldsymbol{E_C}\left(\frac{12\times 7}{365.25}\right)\boldsymbol{t_C} = C_{E\,\mathrm{nom}}\prod_{i=1}^{14}\left(\mathbf{EM}_i\right)\left(\frac{\boldsymbol{S_e}}{1000}\right)^{B} C_{t\,\mathrm{nom}}\left(\left(\frac{12\times 7}{365.25}\right)\boldsymbol{E_C}\right)^{A} \\
&\boldsymbol{E_C}^{\left(\frac{1-A}{B}\right)}\boldsymbol{t_C}^{\left(\frac{1}{B}\right)} = \frac{\boldsymbol{S_e}}{\left(\dfrac{1000\left(\dfrac{12\times 7}{365.25}\right)^{\left(\frac{2-A}{B}\right)}}{\left(C_{E\,\mathrm{nom}}C_{t\,\mathrm{nom}}\displaystyle\prod_{i=1}^{14}\left(\mathbf{EM}_i\right)\right)^{\left(\frac{1}{B}\right)}}\right)}
\end{aligned} \tag{14}$$

Software Management Stress Equation – r2SEF Form

The random variable form of the r2SEF management stress equation is

$$\boldsymbol{E_C} = M\boldsymbol{t_C}^{\gamma} \tag{15}$$

From the COCOMO 81 model definition

$$\boldsymbol{TDEV} = C_{t\,\mathrm{nom}}\boldsymbol{MM}^{A} \tag{16}$$

Converting COCOMO units (person-months and months) to r2SEF form units (person-weeks and weeks) forcing the exponent on $\boldsymbol{E_C}$ to unity, and arranging the factors to be consistent with the r2SEF form yields

$$\begin{gathered}
\left(\frac{12\times 7}{365.25}\right)\boldsymbol{t_C} = C_{t\,\mathrm{nom}}\left(\left(\frac{12\times 7}{365.25}\right)\boldsymbol{E_C}\right)^{A} \\
\boldsymbol{E_C} = \left(\frac{\left(\dfrac{84}{365.25}\right)^{\left(\frac{1-A}{A}\right)}}{C_{t\,\mathrm{nom}}{}^{\left(\frac{1}{A}\right)}}\right)\boldsymbol{t_C}{}^{\left(\frac{1}{A}\right)}
\end{gathered} \tag{17}$$

Software Defect Propensity Equation – r2SEF Random Variable Form

The random variable form of the r2SEF defect propensity equation is

$$\begin{gathered}
\boldsymbol{E_C}{}^{\varphi_E}\boldsymbol{t_C}{}^{\varphi_t} = \frac{\boldsymbol{\ddot{O}}_{[\boldsymbol{a},\boldsymbol{b}]}}{\boldsymbol{\ddot{a}}_{[\boldsymbol{a},\boldsymbol{b}]}} \\
\text{where } \varphi_E \geq 0 \text{ and } \varphi_t \leq 0
\end{gathered} \tag{18}$$

There is no defect estimating relationship within COCOMO 81; therefore, one has been derived from Ref. [5] as follows:

From Ref. [5], defect count is assumed to be linearly proportional to software size; therefore, defect density is constant. We refine this assumption such that defect density ω is constant for a given specific efficiency and specific management stress; however, defect density increases with increasing specific management stress and decreases with increasing specific efficiency. Defect density is therefore given as

$$\begin{gathered}
\boldsymbol{\grave{u}}_{\boldsymbol{b}} \equiv \frac{\boldsymbol{\ddot{O}}_{[\boldsymbol{a},\boldsymbol{b}]}}{\left(\dfrac{\boldsymbol{S_e}}{1000}\right)} \\
\therefore \left(\frac{\boldsymbol{S_e}}{1000}\right) = \frac{\boldsymbol{\ddot{O}}_{[\boldsymbol{a},\boldsymbol{b}]}}{\boldsymbol{\grave{u}}_{\boldsymbol{b}}}
\end{gathered} \tag{19}$$

Converting Equation (11) to r2SEF form variables and units yields

$$
\begin{aligned}
&\boldsymbol{MM} = C_{E\,\mathrm{nom}} \boldsymbol{KEDSI}^{B} \prod_{i=1}^{14} \left(\mathbf{EM}_i\right) \\
&\left(\frac{12\times 7}{365.25}\right) \boldsymbol{E_C} = C_{E\,\mathrm{nom}} \prod_{i=1}^{14} \left(\mathbf{EM}_i\right) \left(\frac{\boldsymbol{S_e}}{1000}\right)^{B}
\end{aligned}
\tag{20}
$$

Substituting Equation (19) into Equation (20) yields

$$
\left(\frac{84}{365.25}\right) \boldsymbol{E_C} = C_{E\,\mathrm{nom}} \prod_{i=1}^{14} \left(\mathbf{EM}_i\right) \left(\frac{\boldsymbol{\ddot{O}_{[a,b]}}}{\boldsymbol{\grave{u}_b}}\right)^{B}
\tag{21}
$$

Converting Equation (16) to r2SEF form variables and units yields

$$
\begin{aligned}
&\boldsymbol{TDEV} = C_{t\,\mathrm{nom}} \boldsymbol{MM}^{A} \\
&C_{t\,\mathrm{nom}} \left(\left(\frac{12\times 7}{365.25}\right) \boldsymbol{E_C}\right)^{A} = \left(\frac{12\times 7}{365.25}\right) \boldsymbol{t_C}
\end{aligned}
\tag{22}
$$

Ratio combining Equation (21) with Equation (22) and arranging the factors to be consistent with the r2SEF form yields

$$
\begin{aligned}
&\frac{\left(\frac{84}{365.25}\right) \boldsymbol{E_C}}{\left(\frac{84}{365.25}\right) \boldsymbol{t_C}} = \frac{C_{E\,\mathrm{nom}} \prod_{i=1}^{14} \left(\mathbf{EM}_i\right) \left(\frac{\boldsymbol{\ddot{O}_{[a,b]}}}{\boldsymbol{\grave{u}_b}}\right)^{B}}{C_{t\,\mathrm{nom}} \left(\left(\frac{84}{365.25}\right) \boldsymbol{E_C}\right)^{A}} \\
&\boldsymbol{E_C}^{\left(\frac{A+1}{B}\right)} \boldsymbol{t_C}^{-\left(\frac{1}{B}\right)} = \frac{\boldsymbol{\ddot{O}_{[a,b]}}}{\left(\boldsymbol{\grave{u}_b} \left(\frac{C_{t\,\mathrm{nom}}}{C_{E\,\mathrm{nom}} \prod_{i=1}^{14} \left(\mathbf{EM}_i\right)}\right)^{\left(\frac{1}{B}\right)} \left(\frac{84}{365.25}\right)^{\left(\frac{A}{B}\right)}\right)}
\end{aligned}
\tag{23}
$$

Duration Exponent α_t (alpha t)

We define the duration exponent α_t (Greek lower case alpha subscript t) to be the exponent on $\boldsymbol{t_C}$ in Equation (14); therefore,

$$
\begin{aligned}
&\alpha_t = \frac{1}{B} \\
&\text{inversely } B = \frac{1}{\alpha_t}
\end{aligned}
\tag{24}
$$

Effort Exponent α_E (alpha E)

We define the effort exponent α_E (Greek lower case alpha subscript E) to be the exponent on $\boldsymbol{E_C}$ in Equation (14); therefore,

$$\begin{aligned} &\alpha_E = \frac{1-A}{B} \\ &\text{inversely } A = 1 - B\alpha_E = \frac{\alpha_t - \alpha_E}{\alpha_t} \end{aligned} \tag{25}$$

Tradeoff Economy γ (gamma)

We define tradeoff economy γ (Greek lower case gamma) to be the exponent on $\boldsymbol{t_C}$ in Equation (17); therefore,

$$\gamma = \frac{1}{A} \tag{26}$$

Specific Efficiency ç (eta)

We define specific efficiency ç (Greek lower case eta) as the denominator of the right-side term in Equation (17). Specific efficiency is a random variable resulting from the random variable product of the effort multiplier random variables.

$$\begin{aligned} &\text{ç} = \frac{1000\left(\dfrac{12\times 7}{365.25}\right)^{\left(\frac{2-A}{B}\right)}}{\left(C_{E\,\text{nom}} C_{t\,\text{nom}} \prod_{i=1}^{14}\left(\mathbf{EM}_i\right)\right)^{\left(\frac{1}{B}\right)}} \\ &\therefore \text{ç} = \frac{1000\left(\dfrac{12\times 7}{365.25}\right)^{(\alpha_E+\alpha_t)}}{\left(C_{E\,\text{nom}} C_{t\,\text{nom}} \prod_{i=1}^{14}\left(\mathbf{EM}_i\right)\right)^{\alpha_t}} \end{aligned} \tag{27}$$

Since the "nominal" rating of each effort multiplier evaluates to unity, nominal efficiency $\bar{\eta}$ (Greek lower case eta bar) is therefore

$$\bar{\eta} = \frac{1000\left(\dfrac{12\times 7}{365.25}\right)^{(\alpha_t-\alpha_E)}}{\left(C_{E\,\text{nom}} C_{t\,\text{nom}} (1)\right)^{\alpha_t}} \tag{28}$$

Specific Management Stress $M_{\min}, M_{\text{nom}}, M_{\max}$

We define management stress M to be the coefficient on $\boldsymbol{t_C}$ in Equation (17); therefore,

$$M = \left(\frac{\left(\frac{84}{365.25}\right)^{\left(\frac{1-A}{A}\right)}}{C_{t\,\text{nom}}^{\left(\frac{1}{A}\right)}} \right)$$
$$\therefore M = \left(\frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t - \alpha_E}\right)}}{C_{t\,\text{nom}}^{\gamma}} \right) \tag{29}$$

Let P represent the percentage of $t_{C\bar{\eta}}$ in which we are interested and let t_{CP} represent the construction duration at that percentage of $t_{C\bar{\eta}}$

$$t_{CP} \equiv P t_{C\bar{\eta}} \tag{30}$$

Let

$$E_{t_{CP}} \equiv x E_{t_{C\bar{\eta}}} \tag{31}$$

when

$$t_{CP} = P t_{C\bar{\eta}} \tag{32}$$

Instantiating Equation (15) for P yields

$$E_{t_{CP}} = M_P t_{CP}^{\gamma}$$
$$M_P = \frac{E_{t_{CP}}}{t_{CP}^{\gamma}} \tag{33}$$

Instantiating Equation (10) for $\bar{\eta}$, $E_{C\bar{\eta}}$, and $t_{C\bar{\eta}}$ yields

$$E_{t_{C\bar{\eta}}}^{\alpha_E} t_{C\bar{\eta}}^{\alpha_t} = \frac{S_e}{\bar{\eta}} \tag{34}$$

Instantiating Equation (10) for $\bar{\eta}$, $E_{t_{CP}}$, and t_{CP} yields

$$t_{CP}{}^{\alpha_E} E_{t_{CP}}{}^{\alpha_t} = \frac{S_e}{\bar{\eta}} \tag{35}$$

Substituting Equation (35) into Equation (34) yields

$$E_{t_{C\bar{\eta}}}{}^{\alpha_E} t_{C\bar{\eta}}{}^{\alpha_t} = E_{t_{CP}}{}^{\alpha_E} t_{CP}{}^{\alpha_t} \tag{36}$$

Substituting Equation (31) and Equation (32) into Equation (36) and solving for x yields

$$\begin{aligned} E_{t_{C\bar{\eta}}}{}^{\alpha_E} t_{C\bar{\eta}}{}^{\alpha_t} &= \left(xE_{t_{C\bar{\eta}}}\right)^{\alpha_E} \left(Pt_{C\bar{\eta}}\right)^{\alpha_t} \\ x &= \frac{1}{P^{\left(\frac{\alpha_t}{\alpha_E}\right)}} \end{aligned} \tag{37}$$

Substituting Equation (37) into Equation (31) yields

$$E_{t_{CP}} = \left(\frac{1}{P^{\left(\frac{\alpha_t}{\alpha_E}\right)}}\right) E_{t_{C\bar{\eta}}} \tag{38}$$

Substituting Equation (38) and Equation (32) into Equation (33) and isolating the ratio $E_{t_{C\bar{\eta}}}$ to $t_{C\bar{\eta}}{}^{\gamma}$ yields

$$\begin{aligned} M_P &= \frac{\left(\frac{1}{P^{\left(\frac{\alpha_t}{\alpha_E}\right)}}\right) E_{t_{C\bar{\eta}}}}{\left(Pt_{C\bar{\eta}}\right)^{\gamma}} \\ P^{\left(\frac{\gamma\alpha_E + \alpha_t}{\alpha_E}\right)} M_P &= \frac{E_{t_{C\bar{\eta}}}}{t_{C\bar{\eta}}{}^{\gamma}} \end{aligned} \tag{39}$$

Instantiating Equation (15) for $E_{t_{C\bar{\eta}}}$ and $t_{C\bar{\eta}}$, substituting Equation (29) into the result, substituting that result into Equation (39), and solving for M_P yields

$$P^{\left(\frac{\gamma\alpha_E+\alpha_t}{\alpha_E}\right)} M_P = \left(\frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t-\alpha_E}\right)}}{C_{t\,\mathrm{nom}}{}^{\gamma}} \right)$$

$$M_P = \frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t-\alpha_E}\right)}}{P^{\left(\frac{\gamma\alpha_E+\alpha_t}{\alpha_E}\right)} C_{t\,\mathrm{nom}}{}^{\gamma}} \tag{40}$$

Therefore

$$M_{\min} = \frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t-\alpha_E}\right)}}{P_{\max}{}^{\left(\frac{\gamma\alpha_E+\alpha_t}{\alpha_E}\right)} C_{t\,\mathrm{nom}}{}^{\gamma}}$$

$$M_{\mathrm{nom}} = \frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t-\alpha_E}\right)}}{P_{\mathrm{nom}}{}^{\left(\frac{\gamma\alpha_E+\alpha_t}{\alpha_E}\right)} C_{t\,\mathrm{nom}}{}^{\gamma}} \tag{41}$$

$$M_{\max} = \frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t-\alpha_E}\right)}}{P_{\min}{}^{\left(\frac{\gamma\alpha_E+\alpha_t}{\alpha_E}\right)} C_{t\,\mathrm{nom}}{}^{\gamma}}$$

Defect Duration Exponent φ_t (phi t)

We define the defect duration exponent φ_t (Greek lower case phi subscript t) to be the exponent on $\boldsymbol{t_C}$ in Equation (23); therefore,

$$\varphi_t = -\left(\frac{1}{B}\right) = -\alpha_t$$

$$\text{inversely } B = -\left(\frac{1}{\varphi_t}\right) \tag{42}$$

Defect Effort Exponent φ_E (phi E)

We define the defect effort exponent φ_E (Greek lower case phi subscript E) to be the exponent on $\boldsymbol{E_C}$ in Equation (23); therefore,

$$
\begin{aligned}
\varphi_E &= \frac{A+1}{B} = 2\alpha_t - \alpha_E \\
\text{inversely } A &= B\varphi_E - 1 = \frac{-\varphi_E - \varphi_t}{\varphi_t}
\end{aligned}
\tag{43}
$$

Specific Defect Vulnerability $\ddot{\boldsymbol{a}}_{[\boldsymbol{a},\boldsymbol{b}]}$ (delta)

We define specific defect vulnerability $\ddot{\boldsymbol{a}}_{[\boldsymbol{a},\boldsymbol{b}]}$ (Greek lower case delta) as the denominator of the right-side term in Equation (23); therefore,

$$
\begin{aligned}
\ddot{\boldsymbol{a}}_{[\boldsymbol{a},\boldsymbol{b}]} &= \grave{\boldsymbol{u}}_{\boldsymbol{b}} \left(\frac{C_{t\,\mathrm{nom}}}{C_{E\,\mathrm{nom}} \prod_{i=1}^{14} \left(\mathbf{EM}_i\right)} \right)^{\left(\frac{1}{B}\right)} \left(\frac{84}{365.25}\right)^{\left(\frac{A}{B}\right)} \\
\therefore \ddot{\boldsymbol{a}}_{[\boldsymbol{a},\boldsymbol{b}]} &= \grave{\boldsymbol{u}}_{\boldsymbol{b}} \left(\frac{C_{E\,\mathrm{nom}} \prod_{i=1}^{14} \left(\mathbf{EM}_i\right)}{C_{t\,\mathrm{nom}}} \right)^{\varphi_t} \left(\frac{84}{365.25}\right)^{\varphi_E + \varphi_t}
\end{aligned}
\tag{44}
$$

To maintain duration confidence consistency between effort and defects, we choose to treat defect vulnerability as a single value rather than as a random variable. The value we choose is the median of the random variable. Since the "nominal" rating of each effort multiplier evaluates to unity, we define median defect vulnerability $\overset{\frown}{\delta}$ (Greek lower case delta frown) to be

$$
\begin{aligned}
\overset{\frown}{\delta}_{[a,b]} &= \mathrm{median}\left(\grave{\boldsymbol{u}}_{\boldsymbol{b}} \left(\frac{C_{E\,\mathrm{nom}} \prod_{i=1}^{14} \left(\mathbf{EM}_i\right)}{C_{t\,\mathrm{nom}}} \right)^{\varphi_t} \left(\frac{84}{365.25}\right)^{\varphi_E + \varphi_t} \right) \text{ or} \\
\overset{\frown}{\delta}_{[a,b]} &= \overset{\frown}{\omega}_b \left(\frac{C_{E\,\mathrm{nom}}\ \mathrm{median}\left(\prod_{i=1}^{14} \left(\mathbf{EM}_i\right)\right)}{C_{t\,\mathrm{nom}}} \right)^{\varphi_t} \left(\frac{84}{365.25}\right)^{\varphi_E + \varphi_t}
\end{aligned}
\tag{45}
$$

4. COCOMO II INSTANTIATION

Category Data Elements[33]

- Category Name: text string
- Category Id: GUID
- Defect Units: text string
- Minimum entropy $B_{\min}$: real
- Schedule equation exponent offset λ : real
- COCOMO-Form Effort Equation Scale Factor $C_{E\,\mathrm{nom}}$: real
- COCOMO-Form Duration Equation Scale Factor $C_{t\,\mathrm{nom}}$: real
- Defect Density Scale Vector $\boldsymbol{\grave{u}}_{\mathbf{Scale}}$: array [-3.0, -2.5…3.0] of real
- *future enhancement: move SD and EM rating scale tables to here*
- Profile List: list of Profile

Profile Data Elements[34]

- Profile Name: text string
- Profile Id: GUID
- Scale Driver Vector $\mathbf{SD}$: vector of triangularly-distributed real random variables
- Effort Multiplier Vector $\mathbf{EM}$: vector of triangularly-distributed real random variables
- Minimum Duration Percentage $P_{\min}$: percentage
- Nominal Duration Percentage P_{nom} : percentage
- Maximum Duration Percentage $P_{\max}$: percentage
- Defect Density $\boldsymbol{\grave{u}}_{\boldsymbol{B}}$: triangularly-distributed real random variable

Metrics Definitions

All of the COCOMO II metrics definitions contained in this section are taken from Ref. [2].

Metrics with Associated Rating Scales

These metrics are triangularly-distributed random variables that each represent an associated parameter (product reliability and complexity, required reuse, platform difficulty, personnel experi-

[33] "Category", within the context of this paper, refers to a collection of data elements that together describe a particular r2SEF calibration (typically the result of analyzing a particular historical data set or instantiating a particular estimating relationship/model).

[34] "Profile", within the context of this paper, refers to a collection of data elements that together describe a particular project or set of closely related projects within an associated r2SEF Category.

ence, facilities, required software reliability, data base size, documentation match to life-cycle needs, product complexity, required reusability, execution time constraint, main storage constraint, platform volatility, analyst capability, applications experience, programmer capability, platform experience, language and tool experience, personnel continuity, use of software tools, multisite development, precedentedness, development flexibility, architecture / risk resolution, team cohesion, and process maturity) where

$$\begin{aligned} &< \boldsymbol{metric} >= \begin{bmatrix} L & M & H \end{bmatrix} \\ &L \le M \le H \end{aligned} \tag{46}$$

with rating scales described below.

< *metric* >	XLO	VLO	LO	NOM	HI	VHI	XHI
RCPX	0.73	0.81	0.98	1.00	1.30	1.74	2.38
RUSE			0.95	1.00	1.07	1.15	1.24
PDIF				1.00	1.29	1.81	2.61
PERS	2.12	1.62	1.26	1.00	0.83	0.63	0.50
PREX	1.59	1.33	1.12	1.00	0.87	0.71	0.62
FCIL	1.43	1.30	1.10	1.00	0.87	0.73	0.62
RELY			0.92	1.00	1.10	1.26	
DATA			0.90	1.00	1.14	1.28	
DOCU		0.81	0.91	1.00	1.11	1.23	
CPLX		0.73	0.87	1.00	1.17	1.34	1.74
RUSE			0.95	1.00	1.07	1.15	1.24
TIME				1.00	1.11	1.29	1.63
STOR				1.00	1.05	1.17	1.46
PVOL			0.87	1.00	1.15	1.30	
ACAP		1.42	1.19	1.00	0.85	0.71	
AEXP		1.22	1.10	1.00	0.88	0.81	
PCAP		1.34	1.15	1.00	0.88	0.76	
PEXP		1.19	1.09	1.00	0.91	0.85	
LTEX		1.20	1.09	1.00	0.91	0.84	
PCON		1.29	1.12	1.00	0.90	0.81	
TOOL		1.17	1.09	1.00	0.90	0.78	
SITE		1.22	1.09	1.00	0.93	0.86	0.80
PREC		6.20	4.96	3.72	2.48	1.24	0.00
FLEX		5.07	4.05	3.04	2.03	1.01	0.00
RESL		7.07	5.65	4.24	2.83	1.41	0.00
TEAM		5.48	4.38	3.29	2.19	1.10	0.00
PMAT		7.80	6.24	4.68	3.12	1.56	0.00

(47)

Effort Multiplier Vector

Each effort multiplier vector element is a triangularly-distributed random variable, the scales of each having been described in (47) above.

For Model Name = Early Design :

$$\mathbf{EM} = \left[\boldsymbol{RCPX}, \boldsymbol{RUSE}, \boldsymbol{PDIF}, \boldsymbol{PERS}, \boldsymbol{PREX}, \boldsymbol{FCIL}\right] \qquad (48)$$

For Model Name = Post Architecture :

$$\mathbf{EM} = \begin{bmatrix} \boldsymbol{RELY}, \boldsymbol{DATA}, \boldsymbol{DOCU}, \boldsymbol{CPLX}, \boldsymbol{RUSE}, \boldsymbol{TIME}, \boldsymbol{STOR}, \boldsymbol{PVOL}, \\ \boldsymbol{ACAP}, \boldsymbol{AEXP}, \boldsymbol{PCAP}, \boldsymbol{PEXP}, \boldsymbol{LTEX}, \boldsymbol{PCON}, \boldsymbol{TOOL}, \boldsymbol{SITE} \end{bmatrix} \tag{49}$$

Scale Driver Vector

Each scale driver vector element is a triangularly-distributed random variable, the scales of each having been described in (47) above.

$$\mathbf{SD} = \left[\boldsymbol{PREC}, \boldsymbol{FLEX}, \boldsymbol{RESL}, \boldsymbol{TEAM}, \boldsymbol{PMAT} \right] \tag{50}$$

r2SEF Metric Assignments

Software Productivity Equation – r2SEF Form

The random variable form of the r2SEF software productivity equation is

$$\boldsymbol{E_C}^{\alpha_E} \boldsymbol{t_C}^{\alpha_t} = \frac{\boldsymbol{S_e}}{\boldsymbol{ç}} \tag{51}$$

We multiplicatively combine the two equations from the COCOMO II.1999 model definition

$$\boldsymbol{PM} = C_{E\,\mathrm{nom}} \prod_{i=1}^{n} \left(\mathbf{EM}_i \right) \boldsymbol{KESLOC}^{B} \tag{52}$$

and

$$\boldsymbol{TDEV} = C_{t\,\mathrm{nom}} \boldsymbol{PM}^{A} \tag{53}$$

to yield

$$\left(\boldsymbol{PM} \right) \left(\boldsymbol{TDEV} \right) = C_{E\,\mathrm{nom}} \prod_{i=1}^{n} \left(\mathbf{EM}_i \right) \boldsymbol{KESLOC}^{B} C_{t\,\mathrm{nom}} \boldsymbol{PM}^{A} \tag{54}$$

where B is defined as

$$B \equiv \mathrm{median}\left(\boldsymbol{B} \right) = \mathrm{median}\left(B_{\min} + 0.01 \sum_{i=1}^{5} \mathbf{SD}_i \right) \tag{55}$$

And where A is defined as

$$A \equiv \lambda + 0.2\left(B - B_{\min} \right) \tag{56}$$

Converting COCOMO units (person-months, months, KSLOC) to r2SEF form units (person-weeks, weeks, SLOC), forcing the exponent on $\boldsymbol{S_e}$ to unity, and arranging the factors to be consistent with the r2SEF form yields

$$\left(\frac{12\times 7}{365.25}\right)\boldsymbol{E_C}\left(\frac{12\times 7}{365.25}\right)\boldsymbol{t_C} = C_{E\text{nom}}\prod_{i=1}^{16}\left(\mathbf{EM}_i\right)\left(\frac{\boldsymbol{S_e}}{1000}\right)^B C_{t\text{nom}}\left(\left(\frac{12\times 7}{365.25}\right)\boldsymbol{E_C}\right)^A$$

$$\boldsymbol{E_C}^{\left(\frac{1-A}{B}\right)}\boldsymbol{t_C}^{\left(\frac{1}{B}\right)} = \frac{\boldsymbol{S_e}}{\left(\dfrac{1000\left(\dfrac{84}{365.25}\right)^{\left(\frac{2-A}{B}\right)}}{\left((C_{E\text{nom}})(C_{t\text{nom}})\prod_{i=1}^{n}\left(\mathbf{EM}_i\right)\right)^{\left(\frac{1}{B}\right)}}\right)} \tag{57}$$

Software Management Stress Equation – r2SEF Form

The random variable form of the r2SEF management stress equation is

$$\boldsymbol{E_C} = M\boldsymbol{t_C}^{\gamma} \tag{58}$$

From the COCOMO II.1999 model definition

$$\boldsymbol{TDEV} = C_{t\text{nom}}\boldsymbol{PM}^A \tag{59}$$

Converting COCOMO units (person-months and months) to r2SEF form units (person-weeks and weeks) forcing the exponent on $\boldsymbol{E_C}$ to unity, and arranging the factors to be consistent with the r2SEF form yields

$$\left(\frac{12\times 7}{365.25}\right)\boldsymbol{t_C} = C_{t\text{nom}}\left(\left(\frac{12\times 7}{365.25}\right)\boldsymbol{E_C}\right)^A$$

$$\boldsymbol{E_C} = \left(\frac{\left(\dfrac{84}{365.25}\right)^{\left(\frac{1-A}{A}\right)}}{C_{t\text{nom}}^{\left(\frac{1}{A}\right)}}\right)\boldsymbol{t_C}^{\left(\frac{1}{A}\right)} \tag{60}$$

Software Defect Propensity Equation – r2SEF Random Variable Form

The random variable form of the r2SEF defect propensity equation is

$$\boldsymbol{E_C}^{\varphi_E}\boldsymbol{t_C}^{\varphi_t} = \frac{\boldsymbol{\ddot{O}}_{[\boldsymbol{a},\boldsymbol{b}]}}{\boldsymbol{\ddot{a}}_{[\boldsymbol{a},\boldsymbol{b}]}} \tag{61}$$

where $\varphi_E \geq 0$ and $\varphi_t \leq 0$

There is no defect estimating relationship within COCOMO II; therefore, one has been derived from Ref. [5].

From Ref. [5], defect count is assumed to be linearly proportional to software size; therefore, defect density is constant. We refine this assumption such that defect density ω is constant for a given specific efficiency and specific management stress; however, defect density increases with increasing specific management stress and decreases with increasing specific efficiency. Defect density at delivery is therefore given as

$$
\begin{aligned}
\boldsymbol{\omega}_{\boldsymbol{\beta}} &\equiv \frac{\boldsymbol{\Phi}_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}{\left(\dfrac{\boldsymbol{S_e}}{1000}\right)} \\
\therefore \left(\frac{\boldsymbol{S_e}}{1000}\right) &= \frac{\boldsymbol{\Phi}_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}{\boldsymbol{\omega}_{\boldsymbol{\beta}}}
\end{aligned}
\tag{62}
$$

Converting Equation (52) to r2SEF form variables and units yields

$$
\begin{aligned}
\boldsymbol{MM} &= C_{E\,\mathrm{nom}}\boldsymbol{KEDSI}^{B}\prod_{i=1}^{n}\left(\mathbf{EM}_i\right) \\
\left(\frac{12\times 7}{365.25}\right)\boldsymbol{E_C} &= C_{E\,\mathrm{nom}}\prod_{i=1}^{n}\left(\mathbf{EM}_i\right)\left(\frac{\boldsymbol{S_e}}{1000}\right)^{B}
\end{aligned}
\tag{63}
$$

Substituting Equation (62) into Equation (63) yields

$$
\left(\frac{84}{365.25}\right)\boldsymbol{E_C} = C_{E\,\mathrm{nom}}\prod_{i=1}^{n}\left(\mathbf{EM}_i\right)\left(\frac{\boldsymbol{\Phi}_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}{\boldsymbol{\omega}_{\boldsymbol{\beta}}}\right)^{B}
\tag{64}
$$

Converting Equation (53) to r2SEF form variables and units yields

$$
\begin{aligned}
\boldsymbol{TDEV} &= C_{t\,\mathrm{nom}}\boldsymbol{MM}^{A} \\
C_{t\,\mathrm{nom}}\left(\left(\frac{12\times 7}{365.25}\right)\boldsymbol{E_C}\right)^{A} &= \left(\frac{12\times 7}{365.25}\right)\boldsymbol{t_C}
\end{aligned}
\tag{65}
$$

Multiplicatively combining Equation (64) with Equation (65) and arranging the factors to be consistent with the r2SEF form yields

$$\left(\frac{84}{365.25}\right)\boldsymbol{E_C} C_{t\,\mathrm{nom}}\left(\left(\frac{84}{365.25}\right)\boldsymbol{E_C}\right)^{A} = C_{E\,\mathrm{nom}}\prod_{i=1}^{n}\left(\mathbf{EM}_i\right)\left(\frac{\text{Ö}_{[a,b]}}{\text{ù}_{b}}\right)^{B}\left(\frac{84}{365.25}\right)\boldsymbol{t_C}$$

$$\boldsymbol{E_C}^{\left(\frac{A+1}{B}\right)}\boldsymbol{t_C}^{-\left(\frac{1}{B}\right)} = \frac{\text{Ö}_{[a,b]}}{\left(\text{ù}_{b}\left(\frac{C_{t\,\mathrm{nom}}}{C_{E\,\mathrm{nom}}\prod_{i=1}^{n}\left(\mathbf{EM}_i\right)}\right)^{\left(\frac{1}{B}\right)}\left(\frac{84}{365.25}\right)^{\left(\frac{A}{B}\right)}\right)} \tag{66}$$

Duration Exponent α_t (alpha t)

We define the duration exponent α_t (Greek lower case alpha subscript t) to be the exponent on $\boldsymbol{t_C}$ in Equation (57); therefore,

$$\begin{aligned}\alpha_t &= \frac{1}{B}\\ \text{inversely } B &= \frac{1}{\alpha_t}\end{aligned} \tag{67}$$

Effort Exponent α_E (alpha E)

We define the effort exponent α_E (Greek lower case alpha subscript E) to be the exponent on $\boldsymbol{E_C}$ in Equation (57); therefore,

$$\begin{aligned}\alpha_E &= \frac{1-A}{B}\\ \text{inversely } A &= 1 - B\alpha_E = \frac{\alpha_t - \alpha_E}{\alpha_t}\end{aligned} \tag{68}$$

Tradeoff Economy γ (gamma)

We define tradeoff economy γ (Greek lower case gamma) to be the exponent on $\boldsymbol{t_C}$ in Equation (60); therefore,

$$\gamma = \frac{1}{A} \tag{69}$$

Specific Efficiency ç (eta)

We define specific efficiency ç (Greek lower case eta) as the denominator of the right-side term in Equation (57). Specific efficiency is a random variable resulting from the random variable product of the effort multiplier random variables.

$$\boldsymbol{\varsigma} = \frac{1000\left(\frac{12\times 7}{365.25}\right)^{\left(\frac{2-A}{B}\right)}}{\left(C_{E\,\mathrm{nom}}C_{t\,\mathrm{nom}}\prod_{i=1}^{n}\left(\mathbf{EM}_i\right)\right)^{\left(\frac{1}{B}\right)}}$$

$$\therefore \boldsymbol{\varsigma} = \frac{1000\left(\frac{12\times 7}{365.25}\right)^{(\alpha_E+\alpha_t)}}{\left(C_{E\,\mathrm{nom}}C_{t\,\mathrm{nom}}\prod_{i=1}^{n}\left(\mathbf{EM}_i\right)\right)^{\alpha_t}} \tag{70}$$

Since the "nominal" rating of each effort multiplier evaluates to unity, nominal efficiency $\bar{\eta}$ (Greek lower case eta bar) is therefore

$$\bar{\eta} = \frac{1000\left(\frac{12\times 7}{365.25}\right)^{(\alpha_t-\alpha_E)}}{\left(C_{E\,\mathrm{nom}}C_{t\,\mathrm{nom}}(1)\right)^{\alpha_t}} \tag{71}$$

Specific Management Stress $M_{\min}, M_{\mathrm{nom}}, M_{\max}$

We define management stress M to be the coefficient on $\boldsymbol{t_C}$ in Equation (60); therefore,

$$M = \left(\frac{\left(\frac{84}{365.25}\right)^{\left(\frac{1-A}{A}\right)}}{C_{t\,\mathrm{nom}}^{\left(\frac{1}{A}\right)}}\right)$$

$$\therefore M = \left(\frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t-\alpha_E}\right)}}{C_{t\,\mathrm{nom}}^{\gamma}}\right) \tag{72}$$

Let P represent the percentage of $t_{C\bar{\eta}}$ in which we are interested and let t_{CP} represent the construction duration at that percentage of $t_{C\bar{\eta}}$

$$t_{CP} \equiv P t_{C\bar{\eta}} \tag{73}$$

Let

$$E_{t_{CP}} \equiv x E_{C\bar{\eta}} \tag{74}$$

when

$$t_{CP} = Pt_{C\bar{\eta}} \tag{75}$$

Instantiating Equation (58) for P yields

$$\begin{aligned} E_{t_{CP}} &= M_P t_{CP}{}^{\gamma} \\ M_P &= \frac{E_{t_{CP}}}{t_{CP}{}^{\gamma}} \end{aligned} \tag{76}$$

Instantiating Equation (51) for $E_{C\bar{\eta}}$ and $t_{C\bar{\eta}}$ yields

$$E_{C\bar{\eta}}{}^{\alpha_E} t_{C\bar{\eta}}{}^{\alpha_t} = \frac{S_e}{\bar{\eta}} \tag{77}$$

Instantiating Equation (51) for $E_{t_{CP}}$ and t_{CP} yields

$$t_{CP}{}^{\alpha_E} E_{t_{CP}}{}^{\alpha_t} = \frac{S_e}{\bar{\eta}} \tag{78}$$

Substituting Equation (78) into Equation (77) yields

$$E_{C\bar{\eta}}{}^{\alpha_E} t_{C\bar{\eta}}{}^{\alpha_t} = E_{t_{CP}}{}^{\alpha_E} t_{CP}{}^{\alpha_t} \tag{79}$$

Substituting Equation (74) and Equation (75) into Equation (79) and solving for x yields

$$\begin{aligned} E_{C\bar{\eta}}{}^{\alpha_E} t_{C\bar{\eta}}{}^{\alpha_t} &= \left(xE_{C\bar{\eta}}\right)^{\alpha_E} \left(Pt_{C\bar{\eta}}\right)^{\alpha_t} \\ x &= \frac{1}{P^{\left(\frac{\alpha_t}{\alpha_E}\right)}} \end{aligned} \tag{80}$$

Substituting Equation (80) into Equation (74) yields

$$E_{t_{CP}} = \left(\frac{1}{P^{\left(\frac{\alpha_t}{\alpha_E}\right)}}\right) E_{C\bar{\eta}} \tag{81}$$

Substituting Equation (81) and Equation (75) into Equation (76) and isolating the ratio $E_{C\bar{\eta}}$ to $t_{C\bar{\eta}}{}^{\gamma}$ yields

$$M_P = \frac{\left(\frac{1}{P^{\left(\frac{\alpha_t}{\alpha_E}\right)}}\right) E_{C\bar{\eta}}}{\left(P t_{C\bar{\eta}}\right)^{\gamma}} \tag{82}$$

$$P^{\left(\frac{\gamma\alpha_E + \alpha_t}{\alpha_E}\right)} M_P = \frac{E_{C\bar{\eta}}}{t_{C\bar{\eta}}^{\ \gamma}}$$

Instantiating Equation (58) for $E_{t_{C\bar{\eta}}}$ and $t_{C\bar{\eta}}$, substituting Equation (72) into the result, substituting that result into Equation (82), and solving for M_P yields

$$P^{\left(\frac{\gamma\alpha_E + \alpha_t}{\alpha_E}\right)} M_P = \left(\frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t - \alpha_E}\right)}}{C_{t\,\mathrm{nom}}^{\ \gamma}}\right) \tag{83}$$

$$M_P = \frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t - \alpha_E}\right)}}{P^{\left(\frac{\gamma\alpha_E + \alpha_t}{\alpha_E}\right)} C_{t\,\mathrm{nom}}^{\ \gamma}}$$

Therefore

$$M_{\min} = \frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t - \alpha_E}\right)}}{P_{\max}^{\left(\frac{\gamma\alpha_E + \alpha_t}{\alpha_E}\right)} C_{t\,\mathrm{nom}}^{\ \gamma}}$$

$$M_{\mathrm{nom}} = \frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t - \alpha_E}\right)}}{P_{\mathrm{nom}}^{\left(\frac{\gamma\alpha_E + \alpha_t}{\alpha_E}\right)} C_{t\,\mathrm{nom}}^{\ \gamma}} \tag{84}$$

$$M_{\max} = \frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t - \alpha_E}\right)}}{P_{\min}^{\left(\frac{\gamma\alpha_E + \alpha_t}{\alpha_E}\right)} C_{t\,\mathrm{nom}}^{\ \gamma}}$$

Defect Duration Exponent φ_t (phi t)

We define the defect duration exponent φ_t (Greek lower case phi subscript t) to be the exponent on $\boldsymbol{t_C}$ in Equation (66); therefore,

$$\begin{aligned} &\varphi_t = -\left(\frac{1}{B}\right) = -\alpha_t \\ &\text{inversely } B = -\left(\frac{1}{\varphi_t}\right) \end{aligned} \tag{85}$$

Defect Effort Exponent φ_E (phi E)

We define the defect effort exponent φ_E (Greek lower case phi subscript E) to be the exponent on $\boldsymbol{E_C}$ in Equation (66); therefore,

$$\begin{aligned} &\varphi_E = \frac{A+1}{B} = 2\alpha_t - \alpha_E \\ &\text{inversely } A = B\varphi_E - 1 = \frac{-\varphi_E - \varphi_t}{\varphi_t} \end{aligned} \tag{86}$$

Specific Defect Vulnerability **ä** *(delta)*

We define specific defect vulnerability **ä** (Greek lower case delta) as the denominator of the right-side term in Equation (23); therefore,

$$\begin{aligned} &\boldsymbol{\ddot{a}}_{[\boldsymbol{a},\boldsymbol{b}]} = \boldsymbol{\grave{u}}_{\boldsymbol{b}} \left(\frac{C_{t\,\mathrm{nom}}}{C_{E\,\mathrm{nom}} \prod_{i=1}^{n} \left(\mathbf{EM}_i\right)} \right)^{\left(\frac{1}{B}\right)} \left(\frac{84}{365.25}\right)^{\left(\frac{A}{B}\right)} \\ &\boldsymbol{\ddot{a}}_{[\boldsymbol{a},\boldsymbol{b}]} = \boldsymbol{\grave{u}}_{\boldsymbol{b}} \left(\frac{C_{t\,\mathrm{nom}}}{C_{E\,\mathrm{nom}} \prod_{i=1}^{n} \left(\mathbf{EM}_i\right)} \right)^{-\varphi_t} \left(\frac{84}{365.25}\right)^{\left(\frac{\frac{-\varphi_E-\varphi_t}{\varphi_t}}{-\left(\frac{1}{\varphi_t}\right)}\right)} \\ &\therefore \boldsymbol{\ddot{a}}_{[\boldsymbol{a},\boldsymbol{b}]} = \boldsymbol{\grave{u}}_{\boldsymbol{b}} \left(\frac{C_{E\,\mathrm{nom}} \prod_{i=1}^{n} \left(\mathbf{EM}_i\right)}{C_{t\,\mathrm{nom}}} \right)^{\varphi_t} \left(\frac{84}{365.25}\right)^{\varphi_E+\varphi_t} \end{aligned} \tag{87}$$

To maintain duration confidence consistency between effort and defects, we choose to treat defect vulnerability as a single value rather than as a random variable. The value we choose is the median of the random variable. Since the "nominal" rating of each effort multiplier evaluates to unity, we define median defect vulnerability $\overset{\frown}{\delta}$ (Greek lower case delta frown) to be

$$\overset{\frown}{\delta}_{[a,b]} = \text{median}\left(\grave{\boldsymbol{u}}_{\boldsymbol{b}} \left(\frac{C_{E\,\text{nom}} \prod_{i=1}^{n} \left(\mathbf{EM}_i \right)}{C_{t\,\text{nom}}} \right)^{\varphi_t} \left(\frac{84}{365.25} \right)^{\varphi_E + \varphi_t} \right) \text{ or}$$

$$\overset{\frown}{\delta}_{[a,b]} = \overset{\frown}{\omega}_b \left(\frac{C_{E\,\text{nom}} \,\text{median}\left(\prod_{i=1}^{n} \left(\mathbf{EM}_i \right) \right)}{C_{t\,\text{nom}}} \right)^{\varphi_t} \left(\frac{84}{365.25} \right)^{\varphi_E + \varphi_t} \tag{88}$$

5. TRADEOFF RELATIONSHIPS AND CONFIDENCE-DRIVEN ESTIMATING

At this point in the paper, we choose to select the COCOMO II instantiation effort-duration tradeoff relationship as an example to graphically illustrate tradeoff relationships and confidence-driven estimating. Note that the method can and has been applied to the defects-duration and cost-duration tradeoff relationships as well. Examples of all three tradeoff relationships in a duration-synchronized format appear later in the paper.

Charting Tradeoff Relationships

To begin, a common thread in all but the simplest software estimation models is the desire to estimate construction effort (labor) E_C and construction duration (time) t_C as a function of the effective software size S_e and some quantification of specific efficiency (reciprocal net environmental complexity) η. For COCOMO II this has been accomplished by Equation (57).

In the typical estimating situation we try to determine reasonable expectations for both effort E_C and duration t_C. Ignoring, for the moment, the notion of uncertainty and confidence, we use a single-point estimating form of Equation (57) 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 software size 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 ***size/efficiency ratio*** ψ.

$$\psi \equiv \frac{S_e}{\eta} = \frac{S_e}{\left(\dfrac{1000\left(\dfrac{84}{365.25}\right)^{\left(\frac{2-A}{B}\right)}}{\left((C_{E\,\mathrm{nom}})(C_{t\,\mathrm{nom}})\prod_{i=1}^{n}\left(\mathbf{EM}'_i\right)\right)^{\left(\frac{1}{B}\right)}} \right)} \tag{89}$$

where $\mathbf{EM}'$ is a vector of single-point values for each of the COCOMO II effort multipliers.

Solving the single-point estimating form of Equation (57) for effort, assuming a unit system with person-weeks and calendar weeks, and substituting Equation (89) into the result yields

$$E_C^{\left(\frac{1-A}{B}\right)} t_C^{\left(\frac{1}{B}\right)} = \frac{S_e}{\left(\dfrac{1000\left(\dfrac{84}{365.25}\right)^{\left(\frac{2-A}{B}\right)}}{\left((C_{E\,\mathrm{nom}})(C_{t\,\mathrm{nom}})\prod_{i=1}^{n}\left(\mathbf{EM}'_i\right)\right)^{\left(\frac{1}{B}\right)}} \right)}$$

$$E_C = \left(\left(\frac{S_e}{\left(\dfrac{1000\left(\dfrac{84}{365.25}\right)^{\left(\frac{2-A}{B}\right)}}{\left((C_{E\,\mathrm{nom}})(C_{t\,\mathrm{nom}})\prod_{i=1}^{n}\left(\mathbf{EM}'_i\right)\right)^{\left(\frac{1}{B}\right)}} \right)} \right)^{\left(\frac{B}{1-A}\right)} t_C^{-\left(\frac{1}{1-A}\right)} \right) \tag{90}$$

$$E_C = \psi^{\left(\frac{B}{1-A}\right)} t_C^{-\left(\frac{1}{1-A}\right)}$$

Charting Equation (90) for a given effective software size and efficiency (set of single-point effort multiplier values) (and therefore a given size/efficiency ratio ψ) is shown in Figure 1 below.

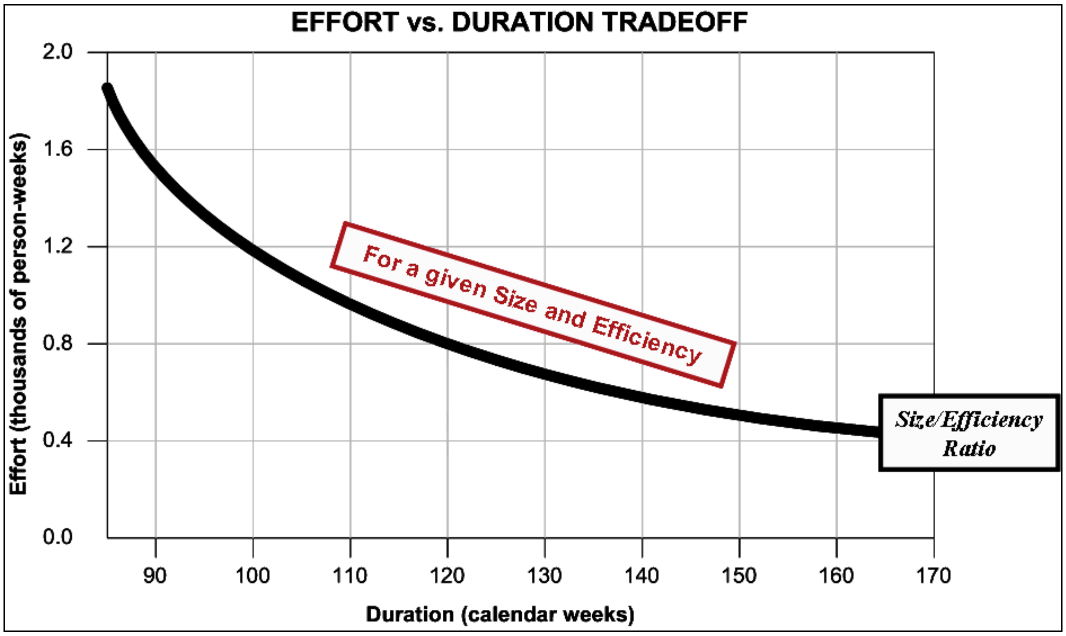

Figure 1: Example Tradeoff Curve (Software Productivity Law)

Charting Tradeoff Limits

Often times correlations have effectiveness limits; i.e., there is a range where they work and outside the range the correlation tends to break down (historical data is sparse or non-existent outside the range). This is true for the software productivity relationship; there exists, for a given software development project, unique minimum duration and minimum effort limiting functions.

Minimum Duration Limit—Each and every software development project, by its nature (divisibility or potential for concurrency), can effectively handle only so many additional people at a given time; therefore, there exists, for each and every software development project, some minimum achievable development duration. [8] In the r2SEF, this limit is defined by Equation (5) where $M_{\max}$ quantifies the maximum management stress that the project can withstand. Equation (84) solved for $M_{\max}$ instantiates this limit for COCOMO II. Therefore, substituting Equation (84) solved for $M_{\max}$ into Equation (5) and solving for effort E_C yields

$$M_{\max} \geq \frac{E_C}{k_M t_C^{\gamma}}$$

$$M_{\max} = \frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t - \alpha_E}\right)}}{P_{\min}^{\left(\frac{\gamma\alpha_E + \alpha_t}{\alpha_E}\right)} C_{t\,\mathrm{nom}}^{\gamma}}$$

$$E_C \leq \left(\frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t - \alpha_E}\right)}}{P_{\min}^{\left(\frac{\gamma\alpha_E + \alpha_t}{\alpha_E}\right)} C_{t\,\mathrm{nom}}^{\gamma}}\right) t_C^{\gamma}$$

$$E_C \leq \left(\frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\left(\frac{1-A}{B}\right)}{\left(\frac{1}{B}\right)-\left(\frac{1-A}{B}\right)}\right)}}{P_{\min}^{\left(\frac{\left(\frac{1}{A}\right)\left(\frac{1-A}{B}\right)+\left(\frac{1}{B}\right)}{\left(\frac{1-A}{B}\right)}\right)} C_{t\,\mathrm{nom}}^{\left(\frac{1}{A}\right)}}\right) t_C^{\left(\frac{1}{A}\right)} \tag{91}$$

$$E_C \leq \left(\frac{\left(\frac{84}{365.25}\right)^{(1-A)}}{P_{\min}^{\left(\frac{1}{1-A}\right)} C_{t\,\mathrm{nom}}}\right)^{\left(\frac{1}{A}\right)} t_C^{\left(\frac{1}{A}\right)}$$

Figure 2 shows the region excluded by this limit 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 (91) as an equality).

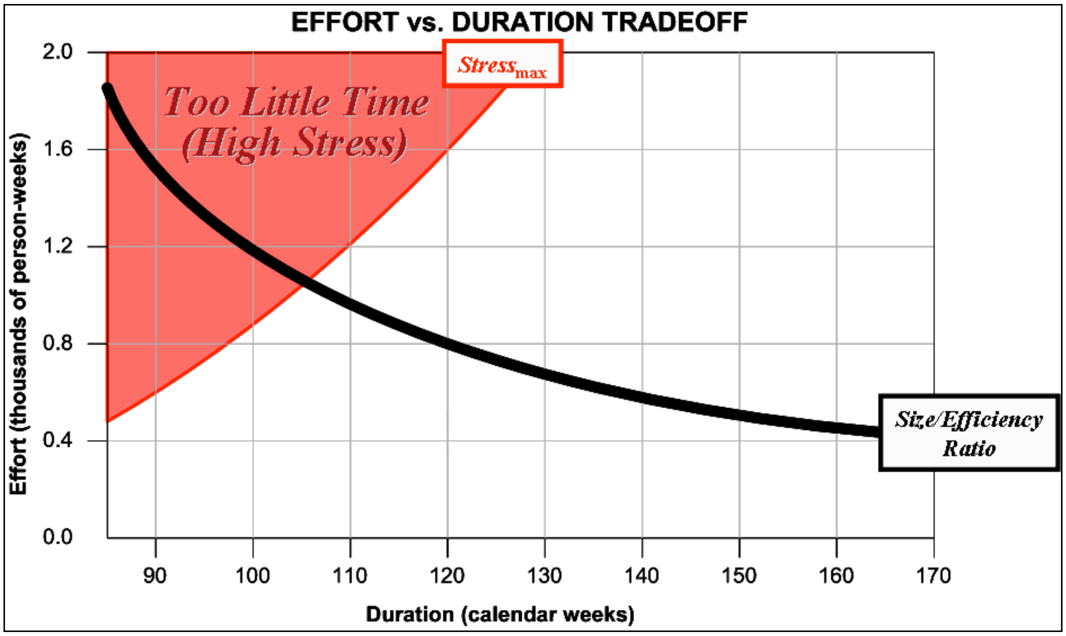

Figure 2. Minimum Duration Limit

Minimum Effort Limit—Theoretically, a software development project is not limited by some maximum development duration. Rare is the software engineer who complains about having too much time to develop software. However, we submit that there exists, for each and every project, some point of maximum productivity; i.e., some point that represents the most efficient use of labor on the project. [8] In the r2SEF, this limit is defined by Equation (6) where $M_{\min}$ quantifies the minimum management stress that the project needs to be productive. Equation (84) solved for $M_{\min}$ instantiates this limit for COCOMO II. Therefore, substituting Equation (84) solved for $M_{\min}$ into Equation (6) and solving for effort E_C yields

$$M_{\min} \geq \frac{E_C}{k_M t_C{}^{\gamma}}$$

$$M_{\min} = \frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t - \alpha_E}\right)}}{P_{\max}{}^{\left(\frac{\gamma\alpha_E + \alpha_t}{\alpha_E}\right)} C_{t\,\mathrm{nom}}{}^{\gamma}}$$

$$E_C \geq \left(\frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t - \alpha_E}\right)}}{P_{\max}{}^{\left(\frac{\gamma\alpha_E + \alpha_t}{\alpha_E}\right)} C_{t\,\mathrm{nom}}{}^{\gamma}}\right) t_C{}^{\gamma}$$

$$E_C \geq \left(\frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\left(\frac{1-A}{B}\right)}{\left(\frac{1}{B}\right) - \left(\frac{1-A}{B}\right)}\right)}}{P_{\max}{}^{\left(\frac{\left(\frac{1}{A}\right)\left(\frac{1-A}{B}\right) + \left(\frac{1}{B}\right)}{\left(\frac{1-A}{B}\right)}\right)} C_{t\,\mathrm{nom}}{}^{\left(\frac{1}{A}\right)}}\right) t_C{}^{\left(\frac{1}{A}\right)} \tag{92}$$

$$E_C \geq \left(\frac{\left(\frac{84}{365.25}\right)^{(1-A)}}{P_{\max}{}^{\left(\frac{1}{1-A}\right)} C_{t\,\mathrm{nom}}}\right)^{\left(\frac{1}{A}\right)} t_C{}^{\left(\frac{1}{A}\right)}$$

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 (92) as an equality).

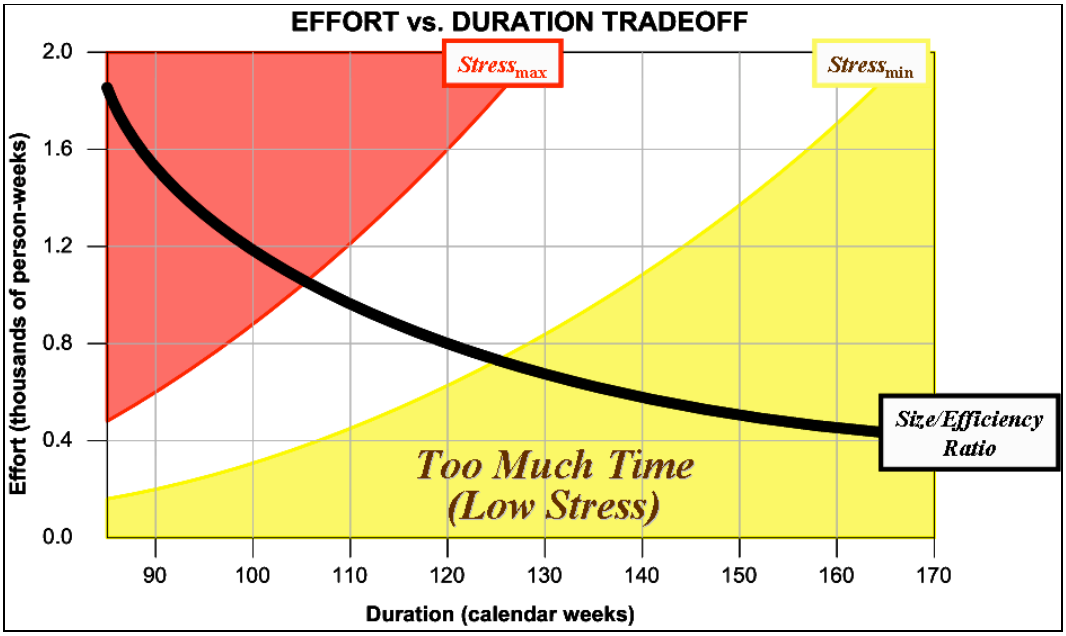

Figure 3. Minimum Effort Limit

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

Goals

Most software development projects are governed by management constraints. We define a ***management constraint***, within this context, to be a two-parameter vector associated with some management measure or metric; these two parameters being a ***goal*** value and a ***desired confidence probability*** of success value. For our evolving example, we assume there exists a constraint for each of effort and duration.

Figure 4 shows the goal values associated with each constraint as interactive dynamic goal symbols (blue diamonds) that traverse each axis and represent the goal value associated with the corresponding variable (metric). In our evolving example, the effort goal is displayed as 1,200 person-weeks and the duration goal is displayed as 120 calendar weeks.

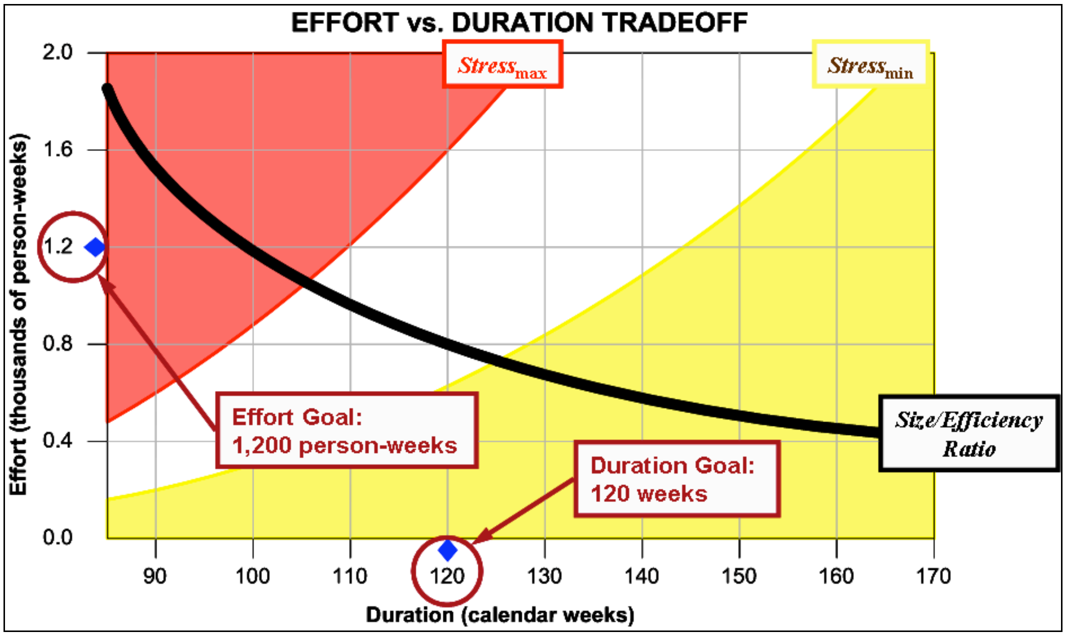

Figure 4. Goal Symbols

Reasonable Solutions

Figure 5, Figure 6, and Figure 7 each show the addition of an interactive dynamic solution symbol (blue circle) with projection lines to each axis, each of which represents a specific instance (effort-duration solution) on the correlation 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 correlation curve and the minimum duration limiting function. Mathematically, the x (duration) coordinate can be found by substituting Equation (90) instantiated with $t_{C\min}$ and $E_{t_{C\min}}$ into Equation (91) as an equality, it also instantiated with $t_{C\min}$ and $E_{t_{C\min}}$, and then solving for duration $t_{C\min}$.

$$\left(\frac{\left(\frac{84}{365.25}\right)^{(1-A)}}{\left(P_{\min}{}^{\left(\frac{1}{1-A}\right)} C_{t\,\mathrm{nom}}\right)}\right)^{\left(\frac{1}{A}\right)} t_{C\min}{}^{\left(\frac{1}{A}\right)} = \psi^{\left(\frac{B}{1-A}\right)} t_{C\min}{}^{-\left(\frac{1}{1-A}\right)}$$

$$t_{C\min} = \left(\frac{P_{\min}{}^{\left(\frac{1}{1-A}\right)} C_{t\,\mathrm{nom}}}{\left(\frac{84}{365.25}\right)^{(1-A)}}\right)^{(1-A)} \psi^{AB} \tag{93}$$

The y (effort) coordinate can then be found from Equation (91) expressed as an equality; 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.

$$E_{t_{C\min}} = \left(\frac{\left(\frac{84}{365.25}\right)^{(1-A)}}{P_{\min}{}^{\left(\frac{1}{1-A}\right)} C_{t\,\mathrm{nom}}}\right)^{\left(\frac{1}{A}\right)} t_{C\min}{}^{\left(\frac{1}{A}\right)} \tag{94}$$

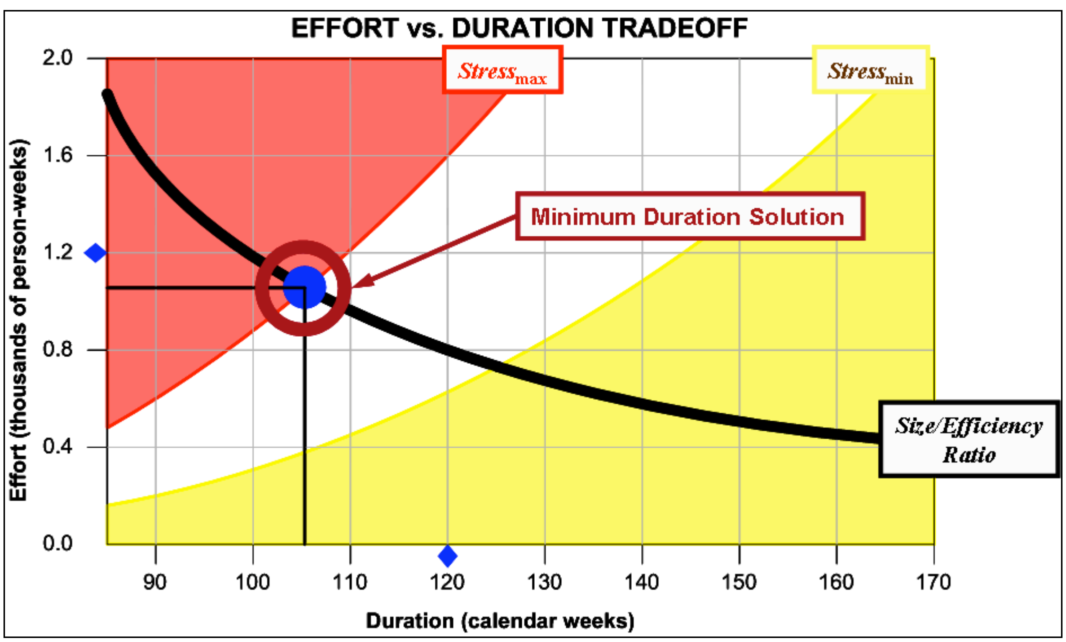

Figure 5. Minimum Duration Solution

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 correlation curve and the minimum effort limiting function. Mathematically, the x (duration) coordinate can be found by substituting Equation (90) instantiated with $t_{E_{C\min}}$ and $E_{C\min}$ into Equation (92), it also being instantiated with $t_{E_{C\min}}$ and $E_{C\min}$, and then solving for duration $t_{E_{C\min}}$.

$$\left(\frac{\left(\frac{84}{365.25}\right)^{(1-A)}}{\left(P_{\max}{}^{\left(\frac{1}{1-A}\right)}C_{t\,\mathrm{nom}}\right)}\right)^{\left(\frac{1}{A}\right)} t_{E_{C\min}}{}^{\left(\frac{1}{A}\right)} = \psi^{\left(\frac{B}{1-A}\right)} t_{E_{C\min}}{}^{-\left(\frac{1}{1-A}\right)}$$

$$t_{E_{C\min}} = \left(\frac{P_{\max}{}^{\left(\frac{1}{1-A}\right)}C_{t\,\mathrm{nom}}}{\left(\frac{84}{365.25}\right)^{(1-A)}}\right)^{(1-A)} \psi^{AB} \tag{95}$$

The y (effort) coordinate can then be found from Equation (92) expressed as an equality; 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.

$$E_{C\min} = \left(\frac{\left(\frac{84}{365.25}\right)^{(1-A)}}{P_{\max}{}^{\left(\frac{1}{1-A}\right)}C_{t\,\mathrm{nom}}}\right)^{\left(\frac{1}{A}\right)} t_{E_{C\min}}{}^{\left(\frac{1}{A}\right)} \tag{96}$$

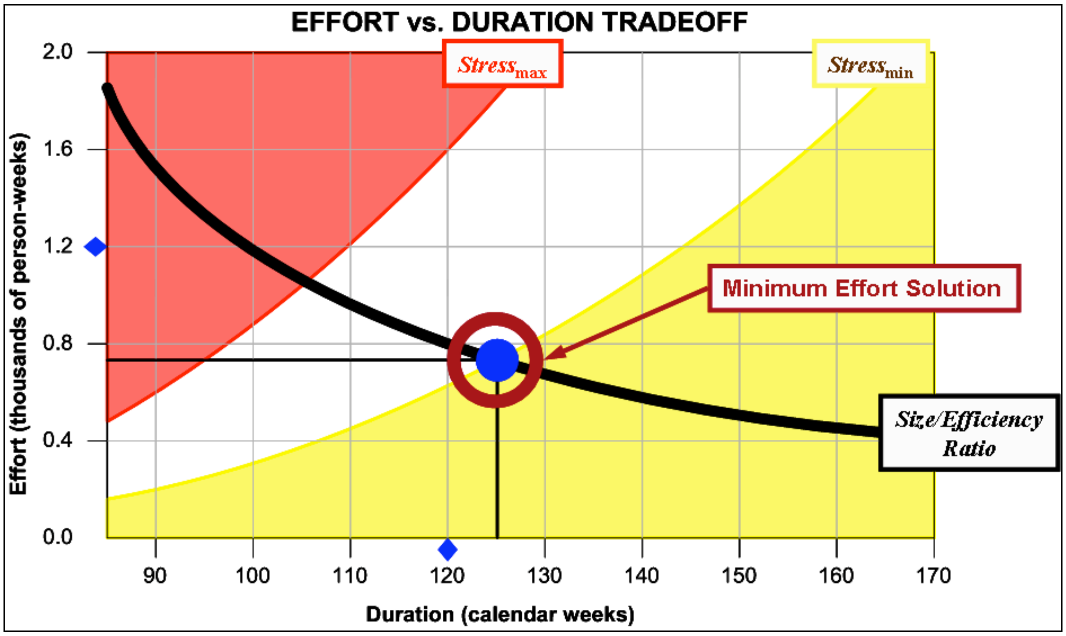

Figure 6. Minimum Effort Solution

Typical (Nominal Stress) Solution— The r2SEF includes a third (and somewhat arbitrary) management stress curve referred to as the typical (nominal stress) curve that is described by Equation (4) where M_{nom} quantifies mean or default management stress. [8] Equation (84) solved for M_{nom} instantiates this curve for COCOMO II. Therefore, substituting Equation (84) solved for M_{nom} into Equation (4) and solving for effort $E_{C\text{nom}}$ yields

$$M_{\text{nom}} = \frac{E_{C\text{nom}}}{k_M t_{C\text{nom}}{}^{\gamma}}$$

$$M_{\text{nom}} = \frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t - \alpha_E}\right)}}{P_{\text{nom}}{}^{\left(\frac{\gamma\alpha_E + \alpha_t}{\alpha_E}\right)} C_{t\text{nom}}{}^{\gamma}}$$

$$E_{C\text{nom}} = \left(\frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\alpha_E}{\alpha_t - \alpha_E}\right)}}{P_{\text{nom}}{}^{\left(\frac{\gamma\alpha_E + \alpha_t}{\alpha_E}\right)} C_{t\text{nom}}{}^{\gamma}}\right) t_{C\text{nom}}{}^{\gamma}$$

$$E_{C\text{nom}} = \left(\frac{\left(\frac{84}{365.25}\right)^{\left(\frac{\left(\frac{1-A}{B}\right)}{\left(\frac{1}{B}\right)-\left(\frac{1-A}{B}\right)}\right)}}{P_{\text{nom}}{}^{\left(\frac{\left(\frac{1}{A}\right)\left(\frac{1-A}{B}\right)+\left(\frac{1}{B}\right)}{\left(\frac{1-A}{B}\right)}\right)} C_{t\text{nom}}{}^{\left(\frac{1}{A}\right)}}\right) t_{C\text{nom}}{}^{\left(\frac{1}{A}\right)}$$

$$E_{C\text{nom}} = \left(\frac{\left(\frac{84}{365.25}\right)^{(1-A)}}{P_{\text{nom}}{}^{\left(\frac{1}{1-A}\right)} C_{t\text{nom}}}\right)^{\left(\frac{1}{A}\right)} t_{C\text{nom}}{}^{\left(\frac{1}{A}\right)} \tag{97}$$

Figure 7 shows the blue solution circle positioned on the typical or nominal stress solution. Note that the solution occurs at the intersection of the correlation curve and the nominal stress curve. Mathematically, the x (duration) coordinate can be found by substituting Equation (90) instantiated with $t_{C\text{nom}}$ and $E_{C\text{nom}}$ into Equation (97) and then solving for duration $t_{C\text{nom}}$.

$$\left(\frac{\left(\frac{84}{365.25}\right)^{(1-A)}}{\left(P_{\text{nom}}{}^{\left(\frac{1}{1-A}\right)} C_{t\text{nom}}\right)}\right)^{\left(\frac{1}{A}\right)} t_{C\text{nom}}{}^{\left(\frac{1}{A}\right)} = \psi^{\left(\frac{B}{1-A}\right)} t_{C\text{nom}}{}^{-\left(\frac{1}{1-A}\right)}$$

$$t_{C\text{nom}} = \left(\frac{P_{\text{nom}}{}^{\left(\frac{1}{1-A}\right)} C_{t\text{nom}}}{\left(\frac{84}{365.25}\right)^{(1-A)}}\right)^{(1-A)} \psi^{AB} \tag{98}$$

The y (effort) coordinate can then be found from Equation (97) ; i.e., projecting duration off of the nominal stress curve onto the y (effort) axis.

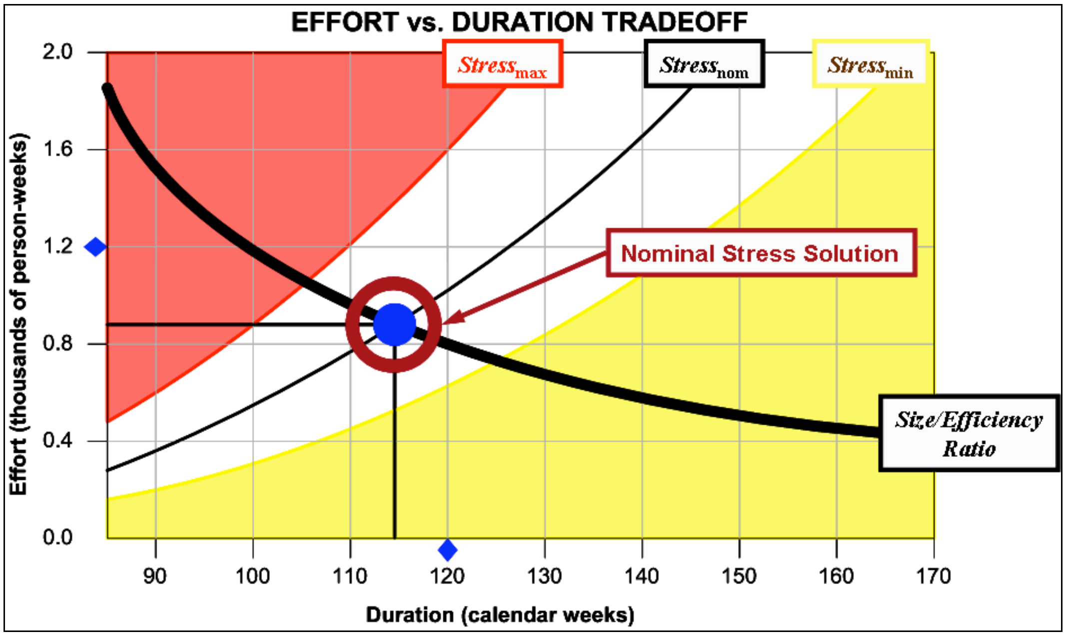

Figure 7. Typical (Nominal Stress) Solution

Size/Efficiency Ratio as a Random Variable

Up until this point, we have been treating the independent variables effective software size S_e, the effort multipliers that determine specific efficiency η, and the size/efficiency ratio ψ as certain; i.e., single-point values. Unfortunately, until project completion, we have exact values for neither effective software size nor specific efficiency; these values are ***uncertain***; i.e., they have many different possible outcomes. We, therefore, choose to represent effective software size and specific efficiency as random variables $\boldsymbol{S_e}$ and $\boldsymbol{ç}$, and instantiation becomes selecting an appropriate distribution function for each.[35] From this we write the ratio of these two random variables $\boldsymbol{Ø}$, itself a random variable, as

$$\boldsymbol{Ø} = \frac{\boldsymbol{S_e}}{\boldsymbol{ç}} \tag{99}$$

The choice of specific distributions for $\boldsymbol{S_e}$ and for $\boldsymbol{ç}$ 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) $\boldsymbol{D_ø}(x)$ of our size/efficiency ratio $\boldsymbol{Ø}$. Finding a neat

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

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 size/efficiency ratio. One approach to this process is summarized as follows:

1. Create a randomly-ordered n-element vector of distributed (triangularly or otherwise) possible outcomes $\mathbf{S_e}$ and $\mathbf{ç}$ for each of effective software size $\boldsymbol{S_e}$ and specific efficiency $\boldsymbol{ç}$.

2. Compute an n-element vector $\mathbf{ø}$ for the size/efficiency ratio $\boldsymbol{Ø}$ that is the vector quotient of $\mathbf{S_e}$ and $\mathbf{ç}$ where the vector quotient is defined as

$$\mathbf{ø} = \frac{\mathbf{S_e}}{\mathbf{ç}} \equiv \begin{bmatrix} S_{e1}/\eta_1 \\ S_{e2}/\eta_2 \\ \vdots \\ S_{en}/\eta_n \end{bmatrix} \qquad (100)$$

3. Use quantization (binning) and hit counting to produce a frequency distribution vector $\mathbf{F_ø}$ associated with the size/efficiency ratio $\boldsymbol{Ø}$ where each vector element consists of the bin lower and upper boundary values and a hit count value.

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

Analyzing tradeoff 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 stochastic 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 tradeoff relationships as described in the following paragraphs; these features being well-suited for implementation as part of a dedicated software application.

Ross Charts and Uncertainty

Size/Efficiency Ratio Confidence Probability

Recall that the correlation between our two dependent variables effort and duration is based on the expected size/efficiency ratio which is a random variable. We begin by postulating that the correlation curve shown in all of the figures thus far is based on the 50% probability value for the size/efficiency ratio. In other words, each point on the curve represents an effort-duration solution where there is a 50% probability that the corresponding effort value and duration value will be achieved or bettered. We therefore rewrite Equation (90), the equation of the correlation curve, to reflect this 50% confidence probability assumption

$$E_{C50\%} = \left(\mathbf{D}_{\mathbf{ø}}^{\mathbf{-1}}\left(50\%\right) t_{C50\%}{}^{-\alpha_t}\right)^{\left(\frac{1}{\alpha_E}\right)} \tag{101}$$

where

$\mathbf{D}_{\mathbf{ø}}^{\mathbf{-1}}\left(50\%\right)$ ::= Inverse-indexing the size/efficiency ratio CDF vector with a probability of 50% to yield the associated size/efficiency ratio value.

Focusing now on Figure 7, 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 correlation, there is actually a family of correlation curves, each curve corresponding to a different confidence probability of the size/efficiency ratio.

$$E_{CProbability} = \left(\mathbf{D}_{\mathbf{ø}}^{\mathbf{-1}}\left(Probability\right) t_{CProbability}{}^{-\alpha_t}\right)^{\left(\frac{1}{\alpha_E}\right)} \tag{102}$$

Figure 8 illustrates three members of this family; the 50% probability curve (solid black line) and two additional member curves (dashed black lines); the curve at 1% confidence probability and the curve at 99% confidence probability.

$$E_{C1\%} = \left(\mathbf{D}_{\mathbf{ø}}^{\mathbf{-1}}\left(1\%\right) t_{C1\%}{}^{-\alpha_t}\right)^{\left(\frac{1}{\alpha_E}\right)} \tag{103}$$

and

$$E_{C99\%} = \left(\mathbf{D}_{\mathbf{ø}}^{\mathbf{-1}}\left(99\%\right) t_{C99\%}{}^{-\alpha_t}\right)^{\left(\frac{1}{\alpha_E}\right)} \tag{104}$$

We choose these particular confidence probabilities to provide a reasonable (for estimating purposes) range of confidence probabilities that we will use later in the paper.

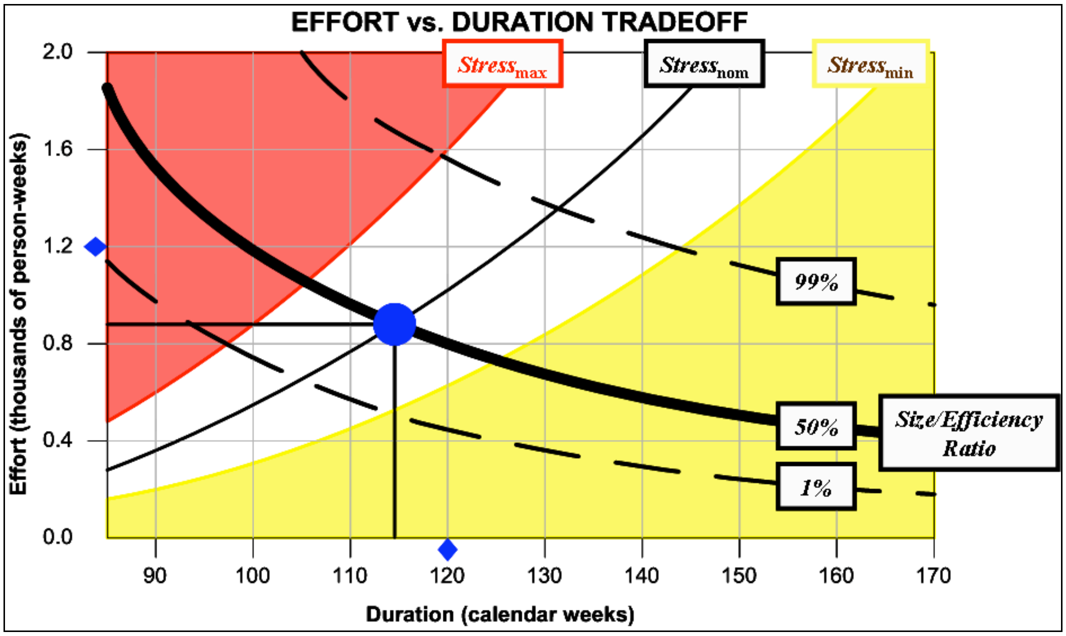

Figure 8: Family of Size/Efficiency Correlation Curves

Projecting Uncertainty

We can generalize the previously-described projection process to map the size/efficiency ratio (input) uncertainty to duration and effort (output) uncertainties. The x (duration) coordinate of a solution with management stress M for any confidence probability of the size/efficiency ratio can be found by substituting Equation (102) into Equation (4) and solving for duration.

$$M = \frac{\left(\mathbf{D}_{\boldsymbol{\theta}}^{-1}\left(Probability\right) t_{CProbability}{}^{-\alpha_t}\right)^{\left(\frac{1}{\alpha_E}\right)}}{t_{CProbability}{}^{\gamma}}$$
$$t_{CProbability} = \left(\frac{\mathbf{D}_{\boldsymbol{\theta}}^{-1}\left(Probability\right)}{M^{\alpha_E}}\right)^{\left(\frac{1}{\gamma\alpha_E + \alpha_t}\right)} \tag{105}$$

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.

$$M = \frac{E_{CProbability}}{t_{CProbability}{}^{\gamma}}$$
$$E_{CProbability} = M t_{CProbability}{}^{\gamma} \tag{106}$$

Focusing on the situation illustrated in Figure 8, we can use the projection process defined by Equation (105) and Equation (106) to map size/efficiency ratio (input) uncertainty to duration and effort (output) uncertainty. 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 (black bar on each axis); dynamic since they move and change width as the blue solution circle is moved along the correlation curve.[36]

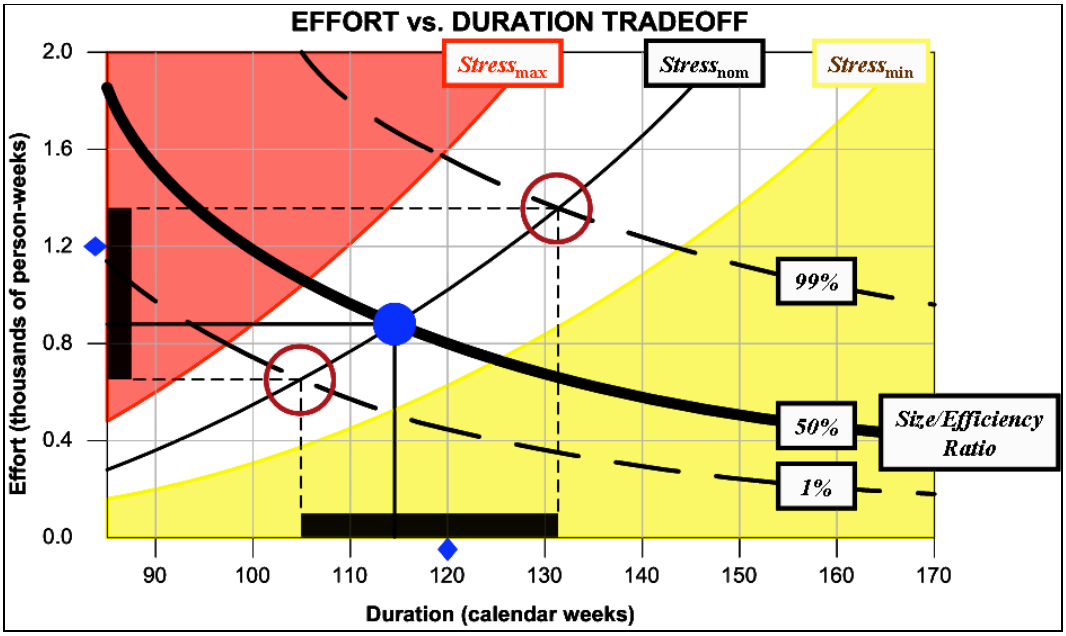

Figure 9: Cumulative Distribution Range Bars

Desired Probability

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 probability. It specifies the desired probability that the final outcome of the associated dependent variable will be less than or equal to the goal value.

Figure 10 shows the desired confidence probability values associated with each constraint as interactive dynamic confidence probability 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 evolving example, the desired confidence probabilities 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 the same value for both metrics.

[36] The projection of uncertainty process used in this paper is derived from Ref. [1] pp. 6-10.

Interpreting the Ross chart in Figure 10, we conclude that the typical (nominal stress) solution does satisfy the effort goal of 1,200 person-weeks with a desired confidence probability of 70%; however, it does not satisfy the duration goal of 120 weeks with a desired confidence probability of 70%.

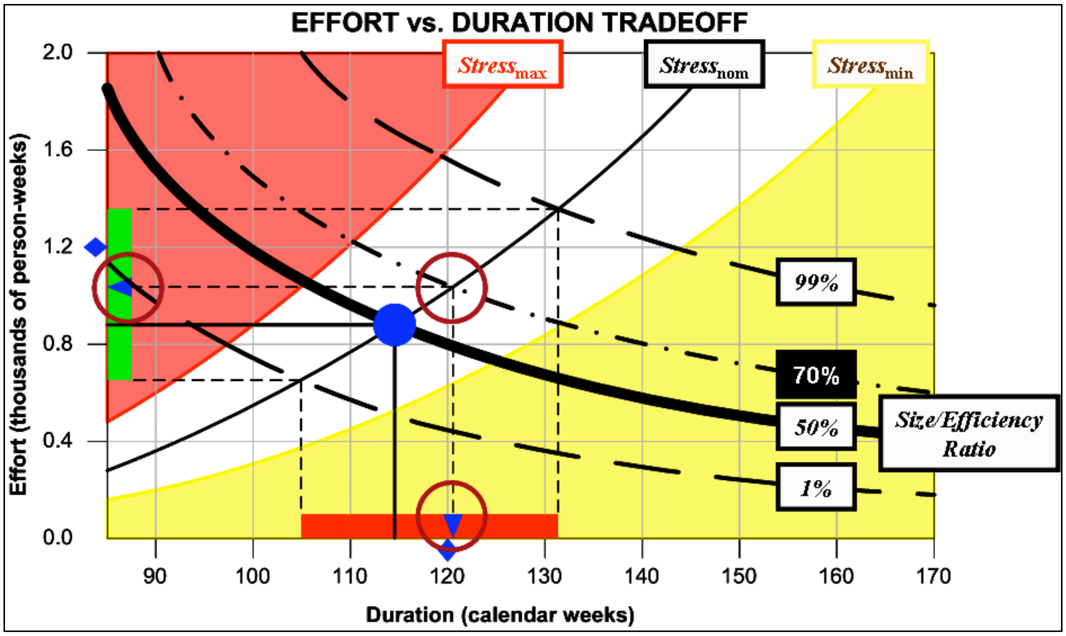

Figure 10. Desired Confidence Probability Symbols

It is important to note here that the distributions of probabilities on the range bars are neither symmetrical nor linear due to the nonlinear and skewed nature of the distributions involved in this particular example. This explains why the projection lines from the current solution circle, which represent median (50% probability) values, do not intersect their associated cumulative distribution range bars at the center of the range.

Analyzing Alternative Situations

Interpreting the Ross chart in Figure 11, we conclude that the minimum duration (maximum stress) solution does not satisfy the effort goal of 1,200 person-weeks with a desired confidence probability of 70%; however, it does satisfy the duration goal of 120 weeks with a desired confidence probability of 70%.

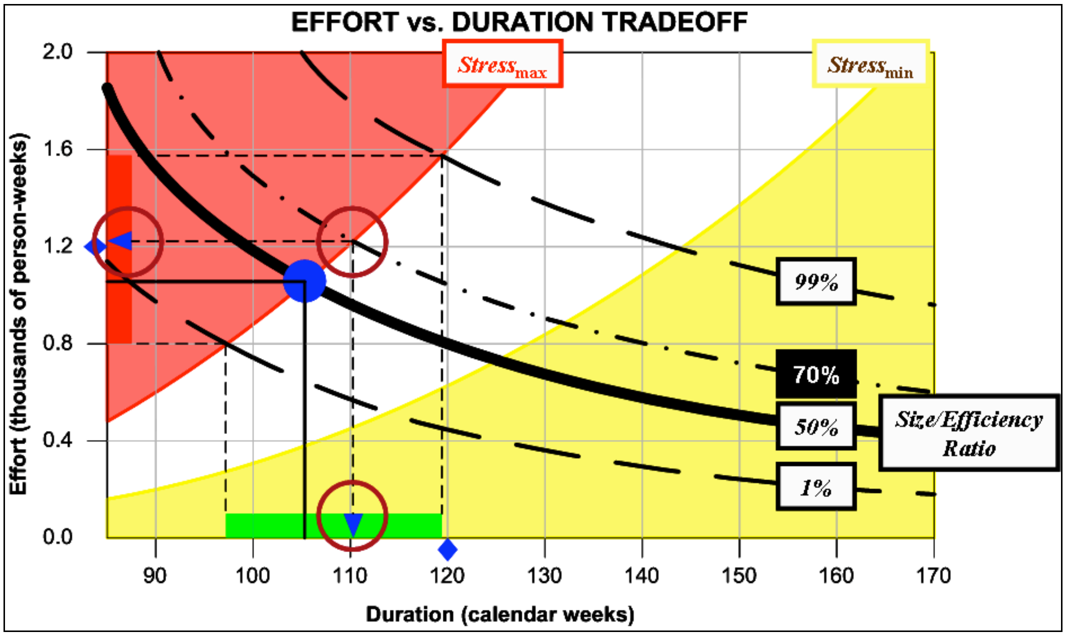

Figure 11: Minimum Duration Solution with Probabilities

Interpreting the Ross chart in Figure 12, we conclude that the minimum effort (minimum stress) solution does satisfy the effort goal of 1,200 person-weeks with a desired confidence probability of 70%; however, it does not satisfy the duration goal of 120 weeks with a desired confidence probability of 70%.

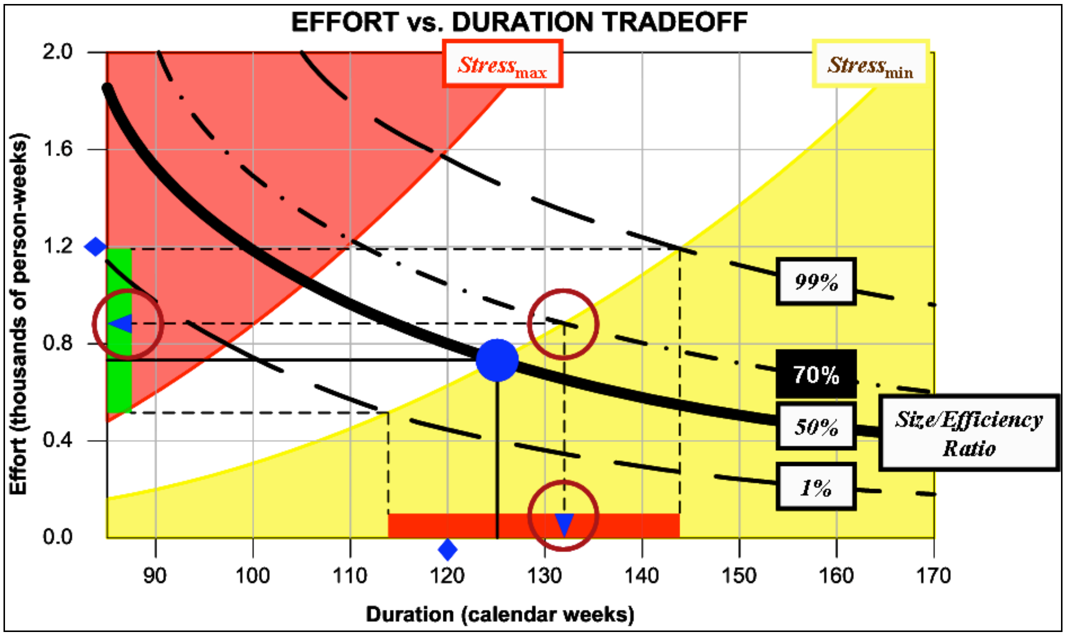

Figure 12: Minimum Effort Solution with Probabilities

The Ross chart in Figure 13 shows a solution where we have examined various points on the correlation curve (varied the management stress M) until we found a point that would satisfy both constraints (goals with their respective desired probabilities). Note that now both cumulative distribution range bars are green indicating the solution to be satisfactory in both dimensions (duration and effort).

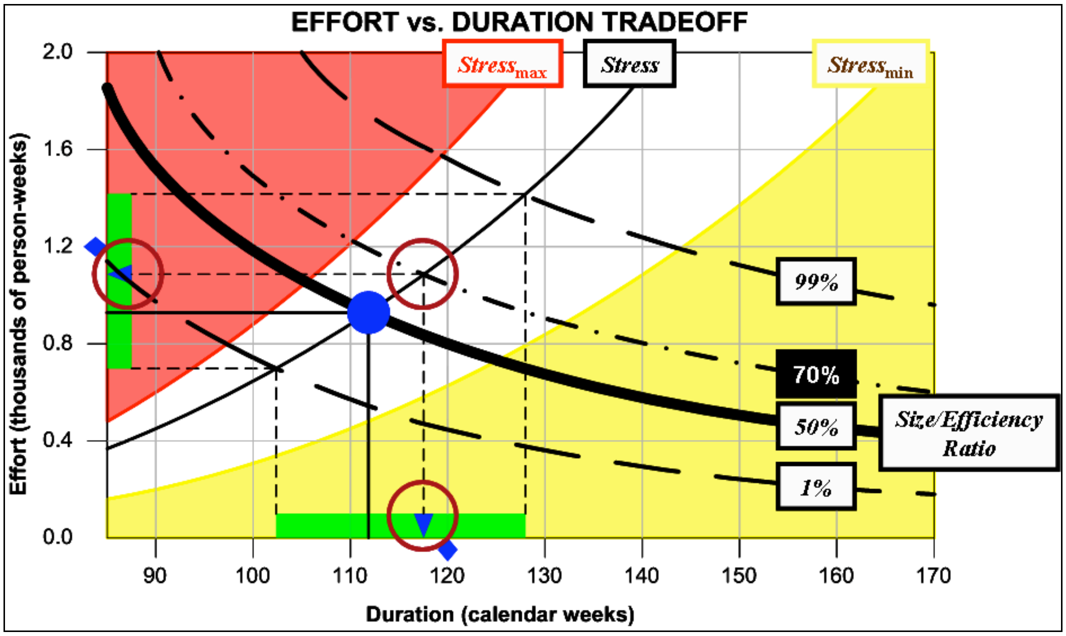

Figure 13: Acceptable Solution

6. EXAMPLE SOFTWARE PROJECT ESTIMATING SCENARIO

Assumptions and Constraints

To demonstrate the application of our COCOMO II instantiation of the r2SEF with its associated Ross chart capabilities, we present a proposed project with the following assumptions and constraints:

- COCOMO II scale driver and effort multiplier settings and associated uncertainty consistent with a typical aerospace contractor developing real-time embedded avionics software for commercial air transport application (see Figure 14).
- Nominal defect density assumption: [1.06; 1.48; 2.07] defects per KSLOC (triangular).
- 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 (2 years); Confidence ≥ 80%.
- Effort constraint: Goal ≤ 2,000 person-weeks; Confidence ≥ 50%.
- Cost constraint: Goal ≤ $10,000,000; Confidence ≥ 80%.
- Defects constraint: Goal ≤ 75 defects remaining at delivery; Confidence ≥ 90%.

We use the ***r2ESTIMATOR***™ estimating tool[37] and its Ross chart capabilities to show various estimation alternatives for the above-described project. Note that the ***r2ESTIMATOR***™ blue solution circles, blue confidence triangles, and blue goal diamonds are all mouse-moveable.

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

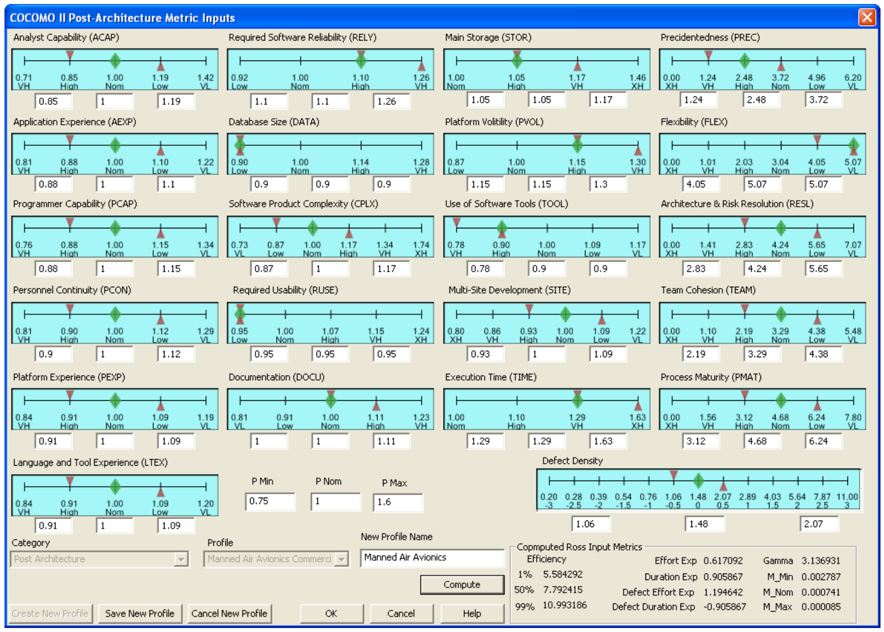

Figure 14: COCOMO II Scale Driver and Effort Multiplier Settings

Initial Examination of the Solution Space

Typical (Nominal Management Stress) Solution— Figure 15 is a synchronized set of three Ross charts that are displaying the nominal stress solution for the above-described scenario. Note that *neither* the duration constraint nor the cost constraint are satisfied. On the other hand, the effort and defects constraints *are* satisfied. Accepting this solution implies accepting a duration confidence probability of 39.0% and a cost confidence probability of 62.5%. Note that the Ross chart illustrating the cost versus duration tradeoff shows both confidence probability range bars as red. This is an indication there is no solution that will simultaneously satisfy both of these constraints.

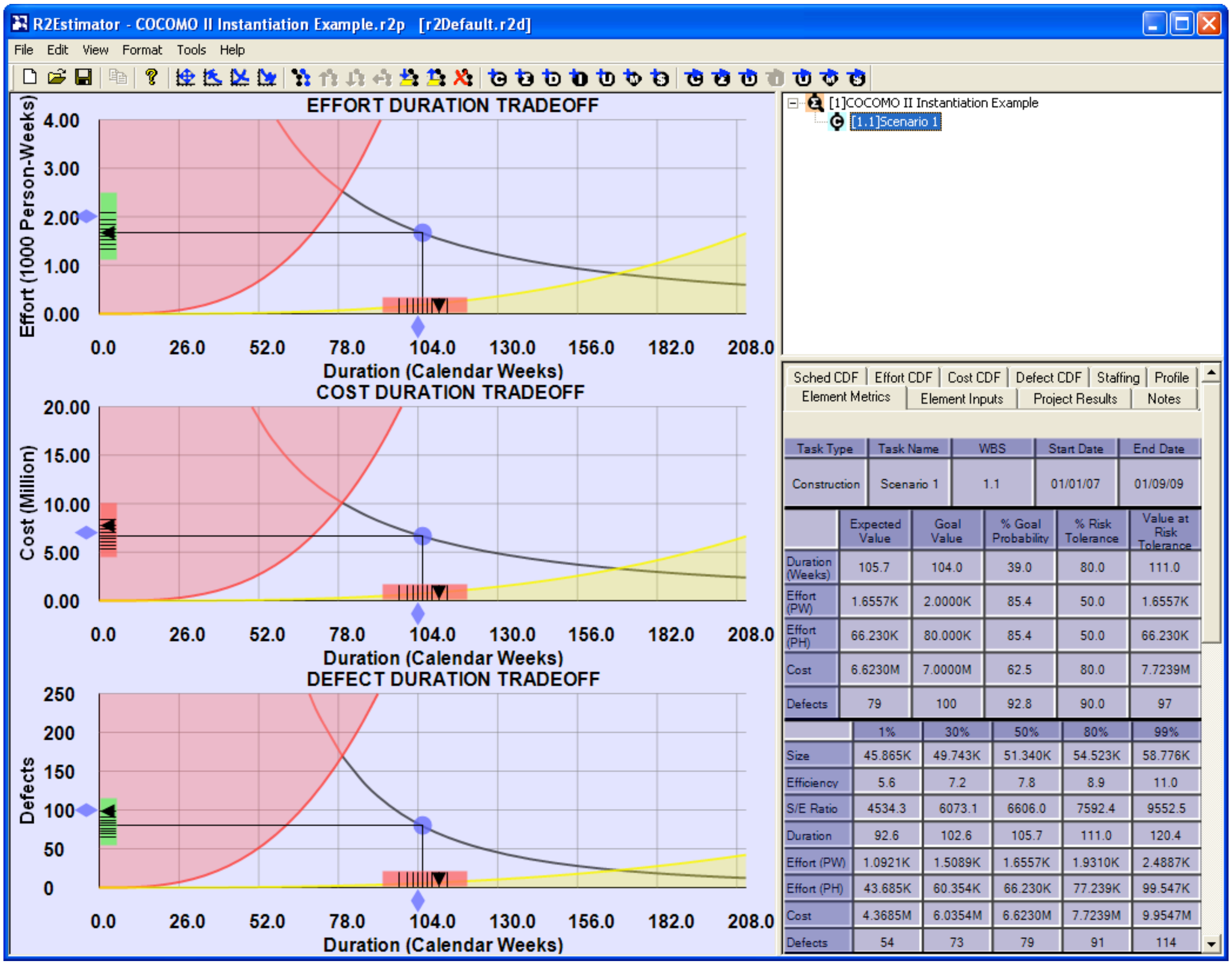

Figure 15: Synchronized Ross Charts:
Typical (Nominal Management Stress)

Minimum Acceptable Duration Solution— Figure 16 is a synchronized set of three Ross charts that are displaying the minimum acceptable duration solution for the above-described scenario. This is achieved by finding the solution that exactly meets the duration goal with the desired confidence probability (i.e., placing the duration confidence probability triangle directly on the duration goal value diamond). Note that this is the green to red transition point for the duration confidence probability range bar. In this scenario, barely satisfying the duration constraint satisfies the effort constraint; however, neither the cost constraint nor the defects constraint is satisfied. Accepting this solution implies accepting a cost confidence probability of 41.1% and a defects confidence probability of 65.3%.

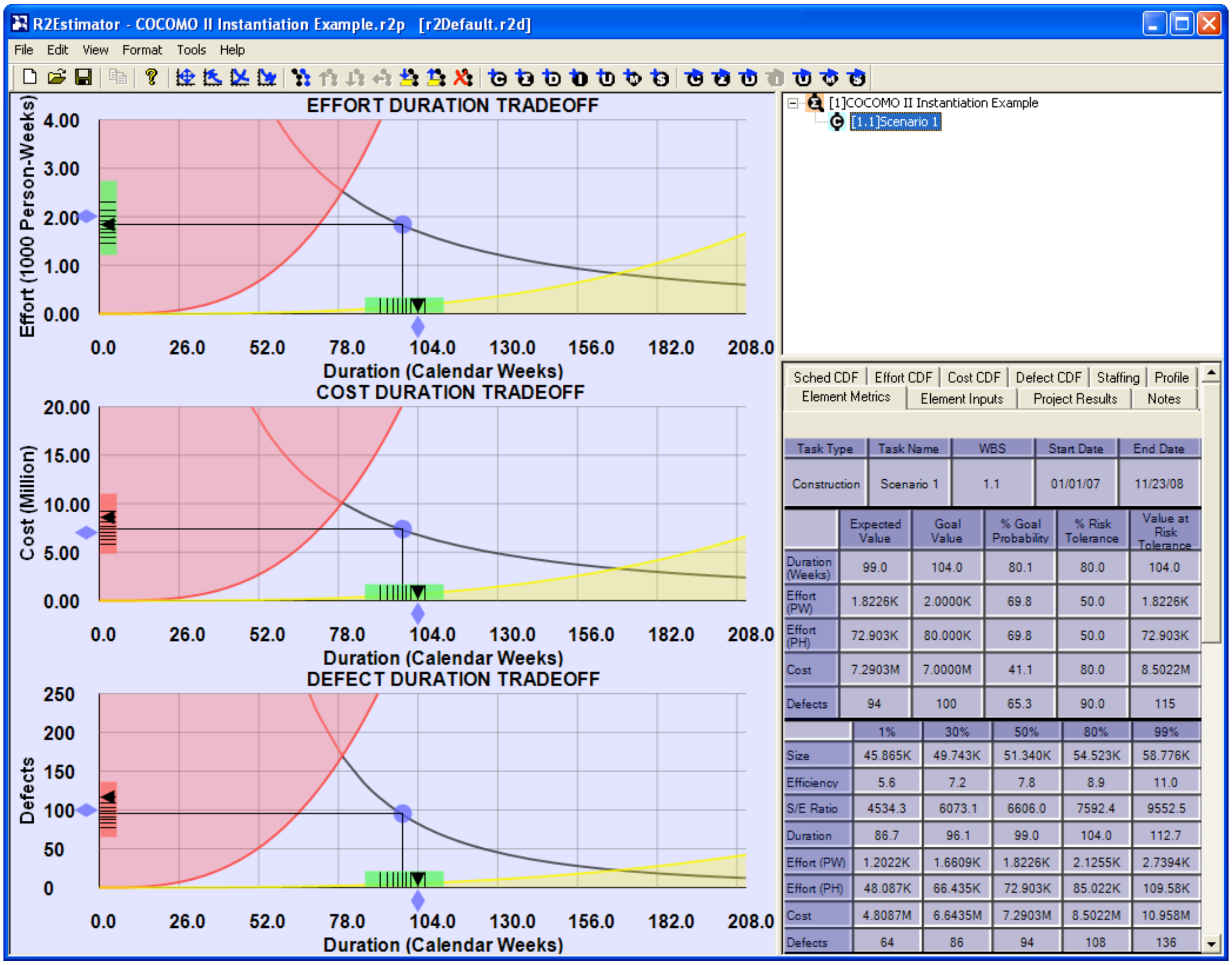

Figure 16. Synchronized Ross Charts:
Minimum Acceptable Duration Solution

Minimum Necessary Duration Solution— Figure 17 is a synchronized set of three Ross charts that are displaying the minimum necessary duration solution for the above-described scenario. This is achieved by finding the solution that exactly meets the most aggressive of the effort, cost, and defects goals with its desired confidence probability (i.e., placing the confidence probability triangle directly on the goal value diamond). One way to describe this solution is that it is the solution requiring the minimum amount of time necessary to cause the effort, cost, and defects confidence probability range bars to all turn green. We can see from Figure 17 that the cost constraint is the most aggressive of the three (i.e., is the last to turn green as the duration is increased). Accepting this solution implies accepting a duration confidence probability of 7.3%.

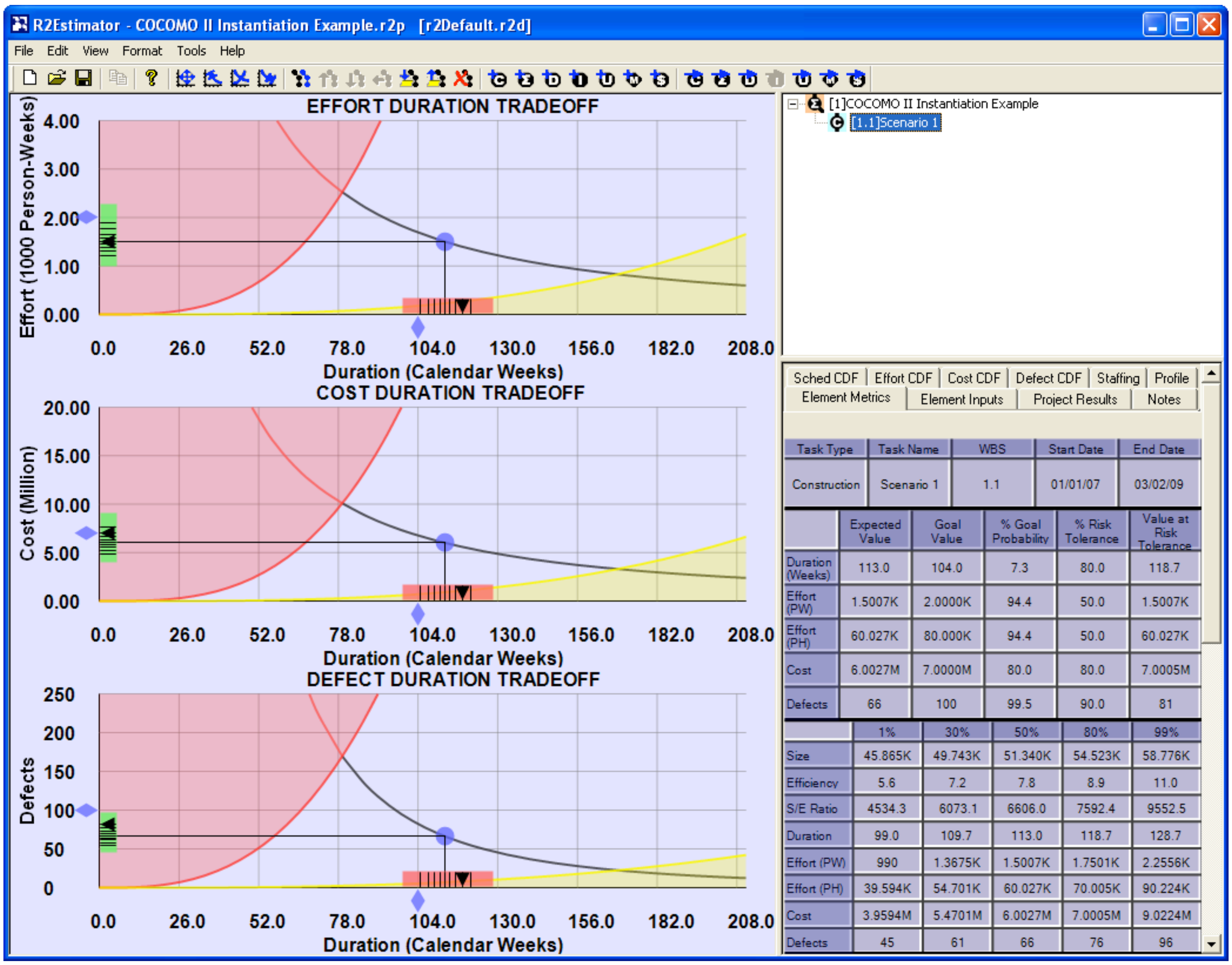

Figure 17: Synchronized Ross Charts: Minimum Necessary Duration Solution

Identifying and Analyzing Solution Alternatives

***Identifying Solution Alternatives*—** As was pointed out in the description of the typical (nominal management stress) solution, we have, with our example project, an all-too-common situation where, for the given size and efficiency, the goals and the desired probabilities are mutually impossible to achieve. We call this an over-constrained situation, one that offers 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 and allow the decision-makers to choose.

A reasonable way to list potentially satisfactory 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 Effective Software Size 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 18 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 11.5% of the original functionality (i.e., reducing the effective software size to [39,825; 44,250; 53,100] SLOC), we can satisfy all of the original constraints.

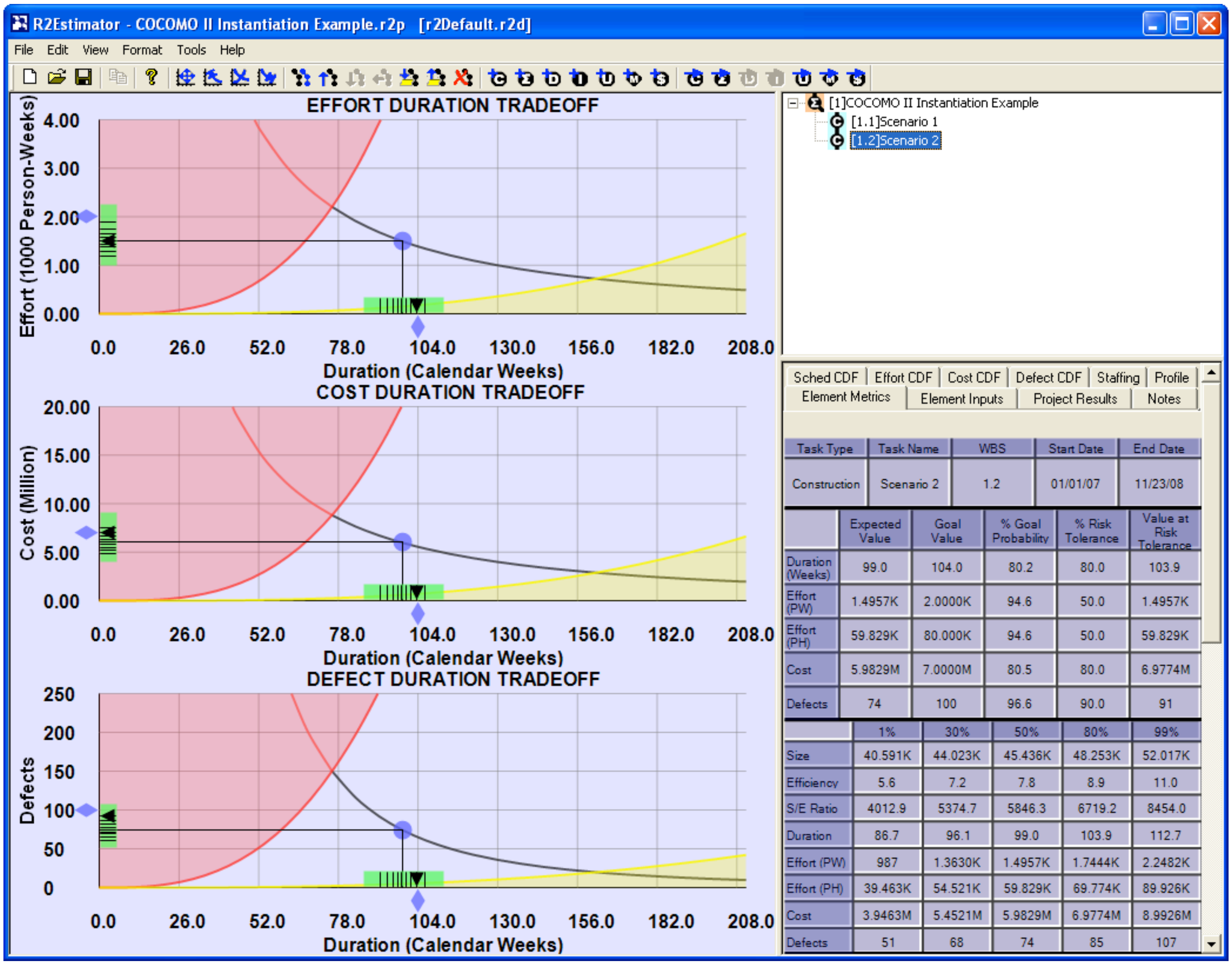

Figure 18: Synchronized Ross Charts:
Reduced Effective software size

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 19 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 14.8 weeks (a 14.2% schedule slip from 104 weeks to 118.8 weeks) while preserving its 80% confidence probability, we can satisfy all of the other original constraints.

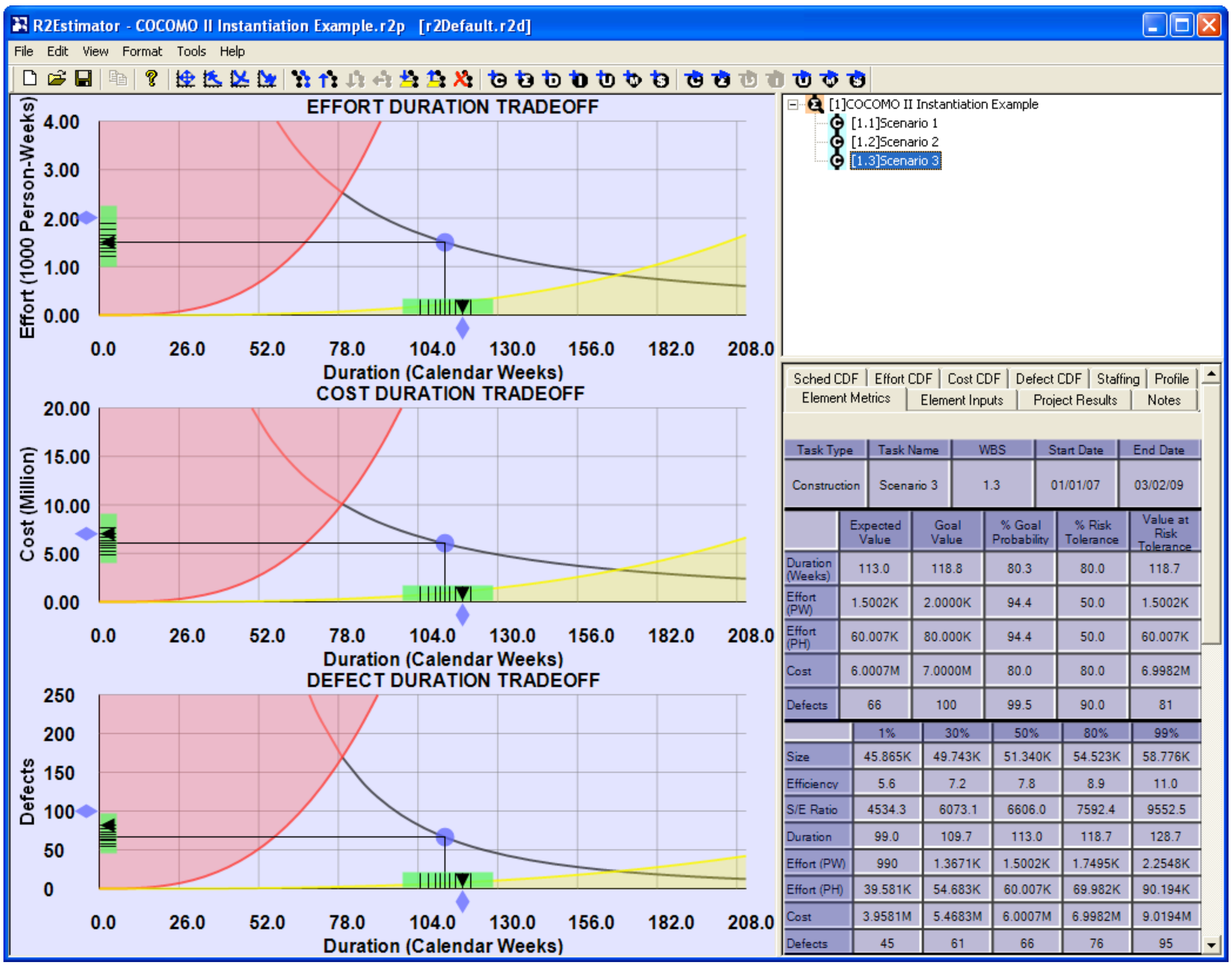

**Figure 19. Synchronized Ross Charts:
Relaxed Duration Goal**

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 20 is a synchronized set of three Ross charts that are displaying a 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 probability of only 7.2%.

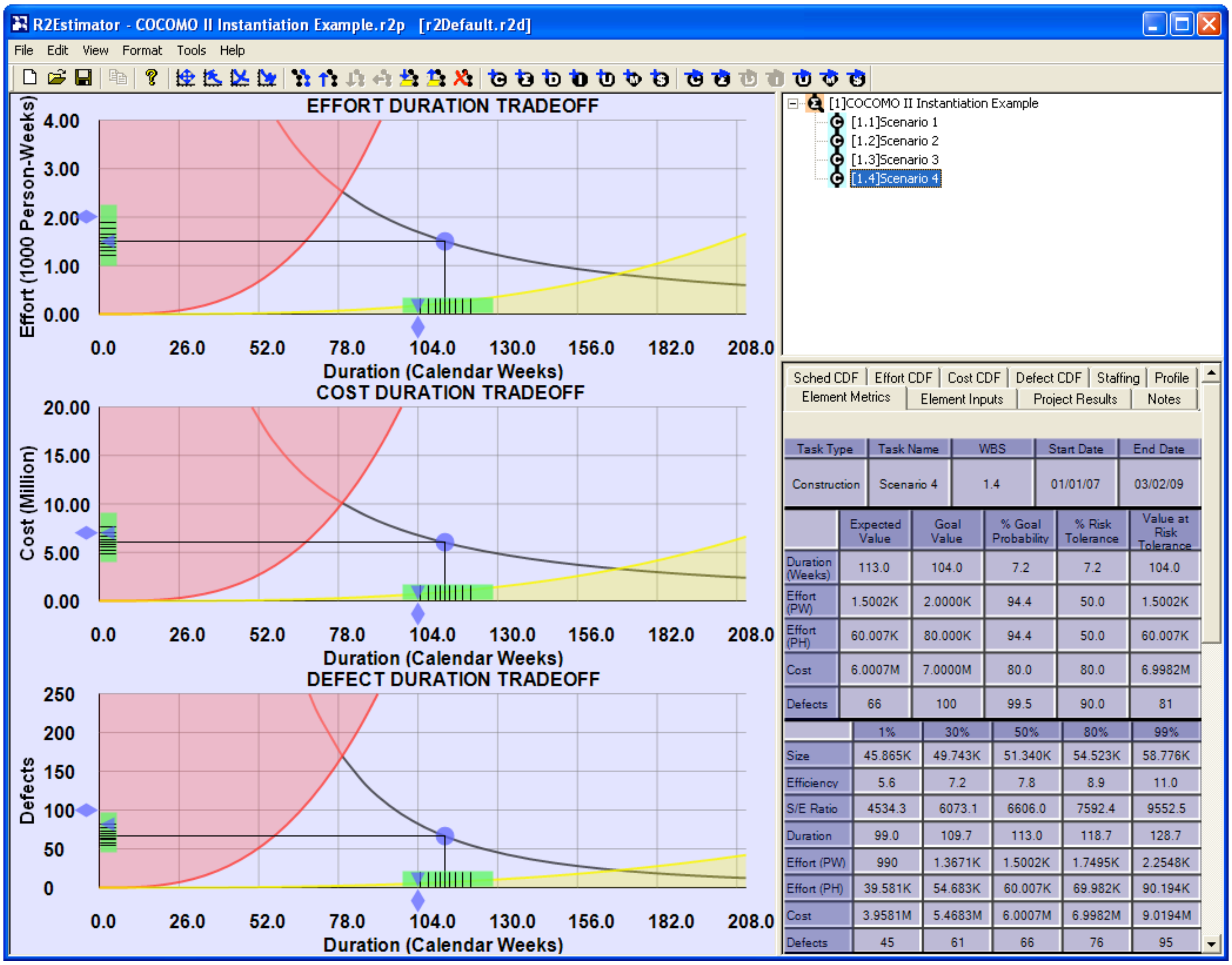

Figure 20. Synchronized Ross Charts:
High Duration Risk

Composite Solution #1— *"Show me a reasonable solution given that I really need to deliver something useful (at least 90% of the total functionality) and given that I can negotiate an additional 4 weeks into the schedule."* Figure 21 shows a composite solution with:

- Reduced (90%) size [40,500; 45,000; 54,000]
- 108-week duration goal with 80% confidence

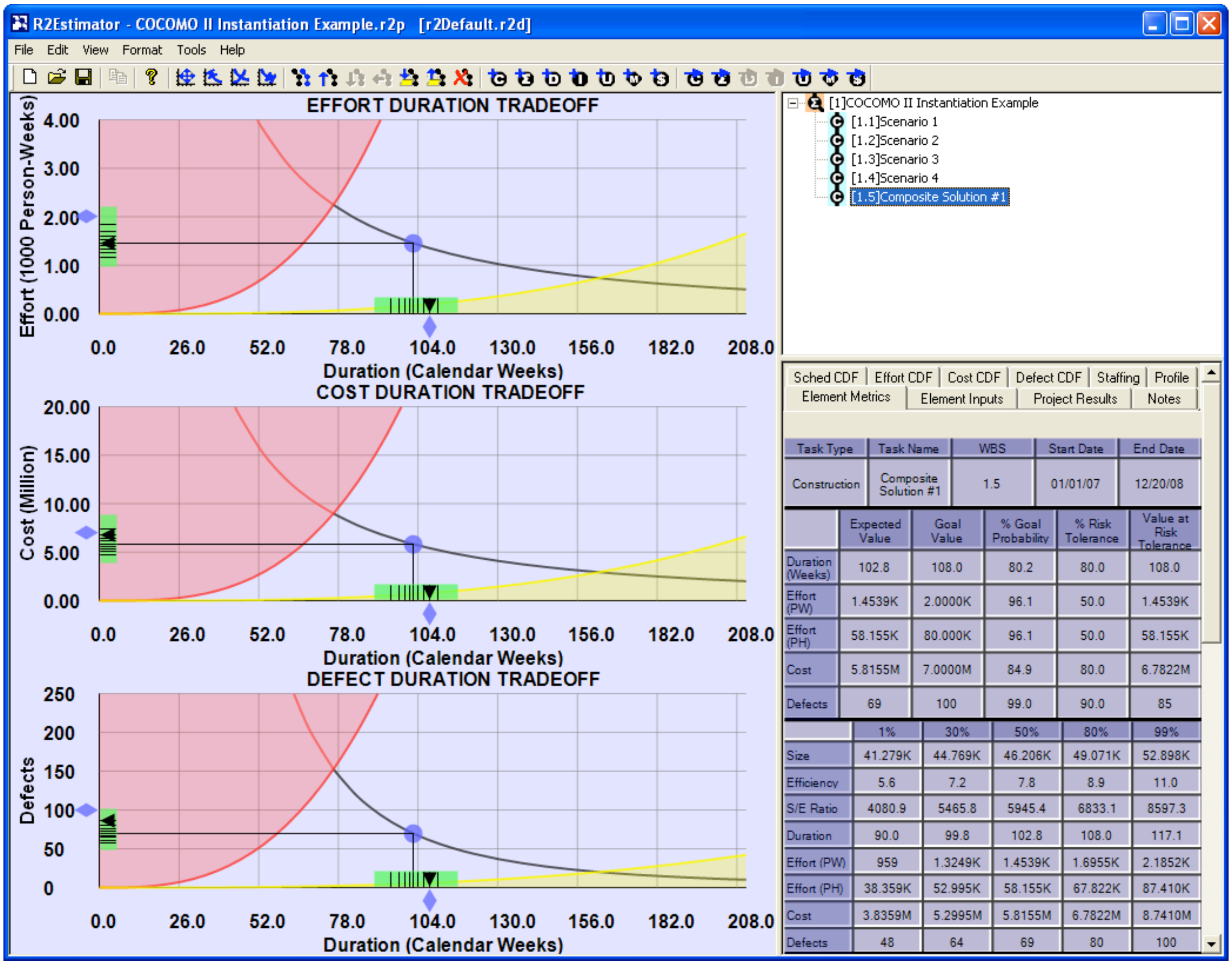

Figure 21. Synchronized Ross Charts: Composite Solution #1

Composite Solution #2— *"Given that I must deliver all of the functionality in 104 weeks with 80% certainty, to what values for effort, cost, and duration can I commit and maintain the associated original desired confidence probabilities?"* Figure 22 shows a composite solution with the goals:

- 104-week duration; 80% confidence
- 1,823 person-week effort; 50% confidence
- $8,502,200 cost; 80% confidence
- 115 delivered defects; 90% confidence

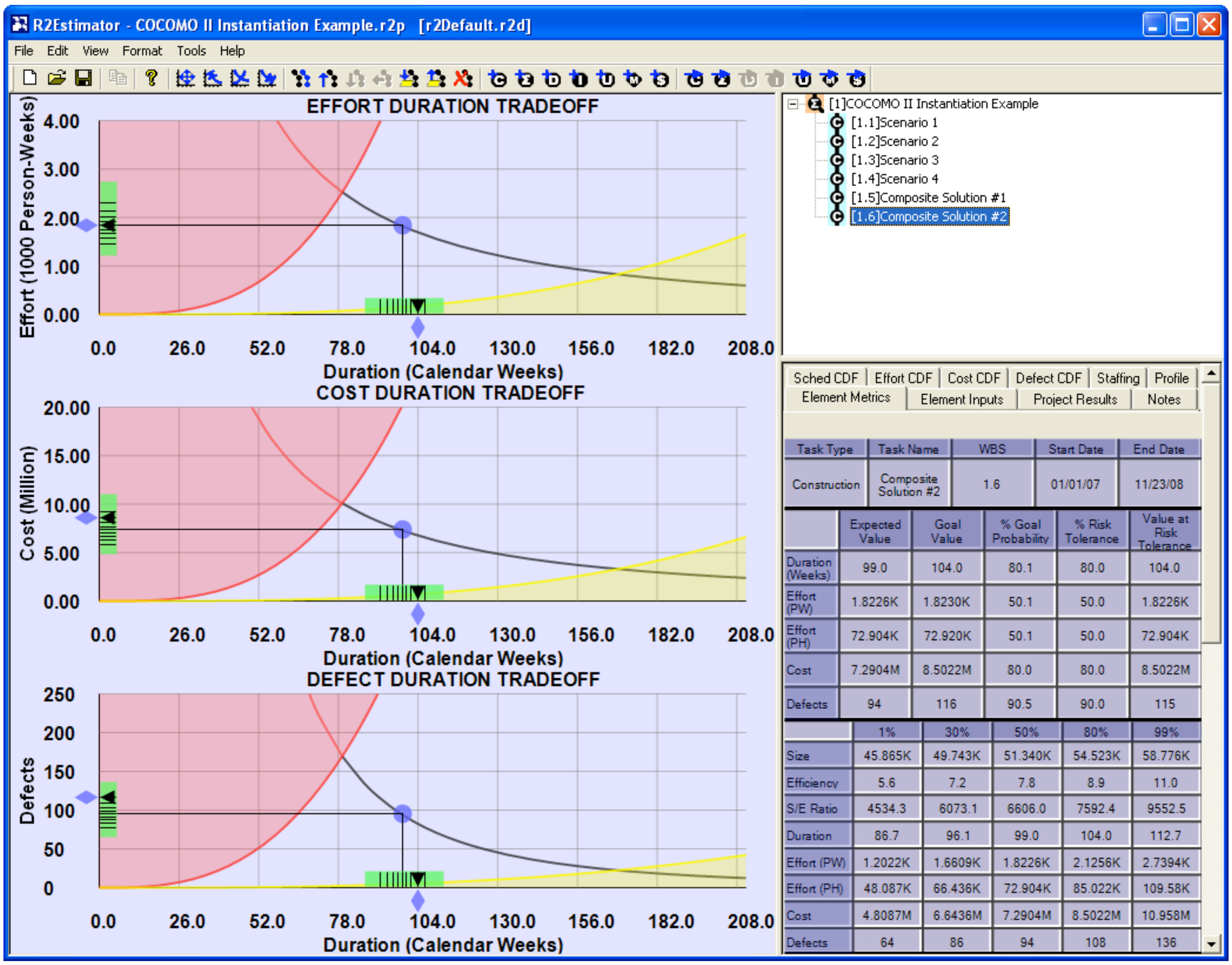

Figure 22. Synchronized Ross Charts: Composite Solution #2

REFERENCES

[1]Boehm, B., Software *Engineering Economics*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.

[2]Boehm, B. [et al.], *Software Cost Estimation with COCOMO II*, Prentice-Hall, Inc., Upper Saddle River, NJ, 2000.

[3]Browne, J., *Probabilistic Design: Course Notes*, Swinburne University of Technology, Melbourne, Australia, 2001.

[4]Jensen, R., "An Improved Macrolevel Software Development Resource Estimation Model," Software Engineering, Inc.

[5]Musa, John D., *Software Reliability Engineering: More Reliable Software Faster and Cheaper*, 2nd Edition, AuthorHouse, Bloomington, IN, 2004.

[6]Norden, P., "Project Life Modeling: Background and Application of Life Cycle Curves," *Software Life Cycle Management Workshop*, Sponsored by USACSC, Airlie, VA, August 1977.

[7]Putnam, L. *Software Cost Estimating and Life-Cycle Control: Getting the Software Numbers*, Computer Society Press, IEEE Computer Society, New York, NY, 1980.

[8]Ross, M., "Next Generation Software Project Estimating: 25 Years and Thousands of Projects Later", *Proc. Joint ISPA / SCEA 2007 Conference*, The International Society of Parametric Analysts and The Society of Cost Estimating and Analysis, New Orleans, LA, June 2007.

[9]Ross, M., "Next Generation Software Project Estimating: Know the Odds Before Placing Your Bet", *AIAA SPACE 2007 Conference and Exposition*, American Institute of Aeronautics and Astronautics, Long Beach, CA, September 2007.

BIOGRAPHY

Mike 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.*