TM

A New Software Estimating Framework

Presentation to ISPA-SCEA 2007 Conference

Mike Ross
President & CEO
r2ESTIMATING, LLC
7755 E. Evening Glow Drive
Scottsdale, Arizona 85262-1295
(o) 480.488.8382   (f) 480.488.8420
mike.ross@r2estimating.com

June 2007 Initial

TM

Fundamental Measures
Summary

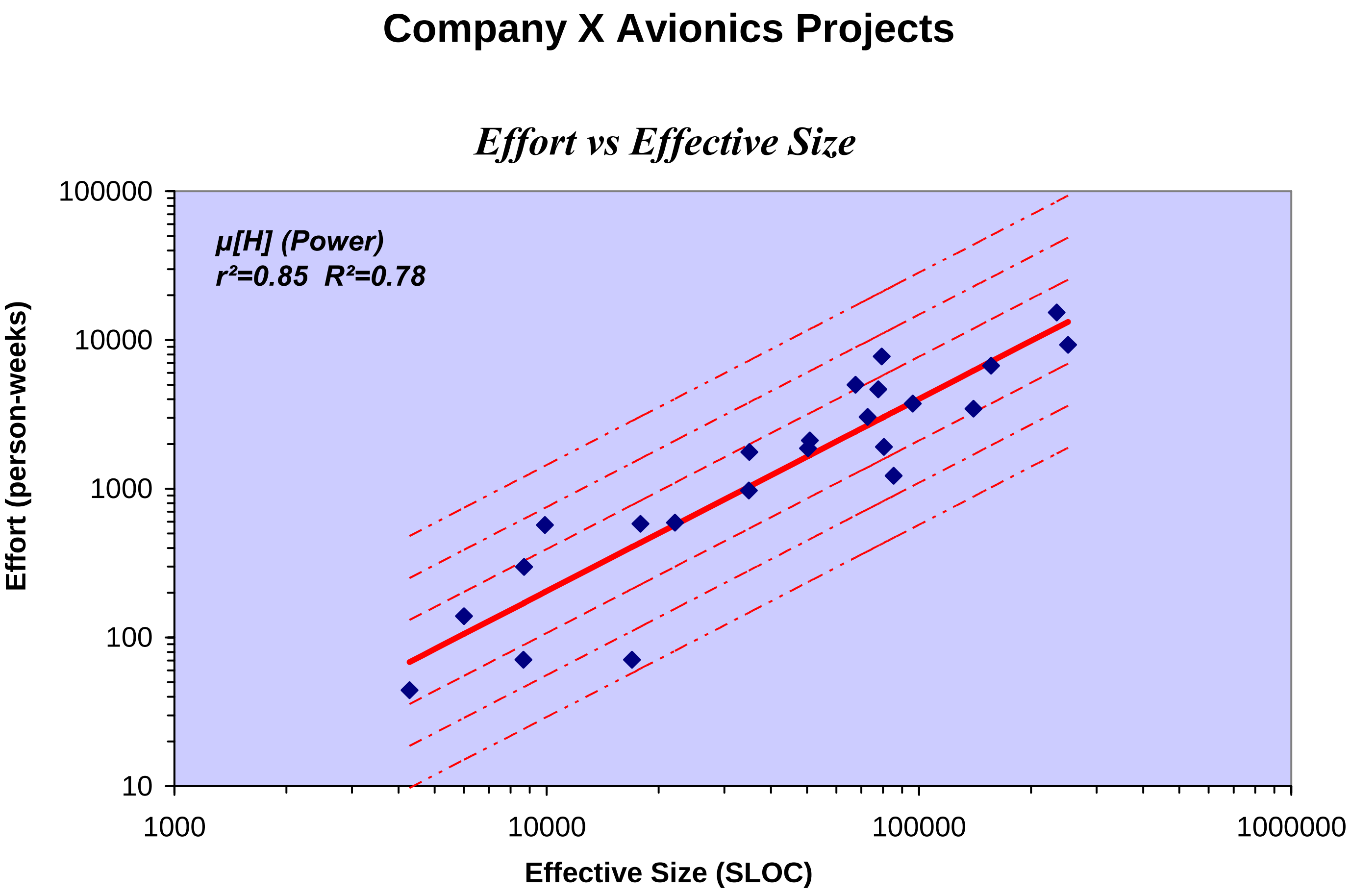
- ***What we assume / expect***
  - Effective Size
  - Efficiency
  - Defect Vulnerability
  - Management Stress

- ***What we want to know***
  - Duration
  - Effort
  - Cost
  - Staffing
  - Defects

TM

Fundamental Observations about Software Development

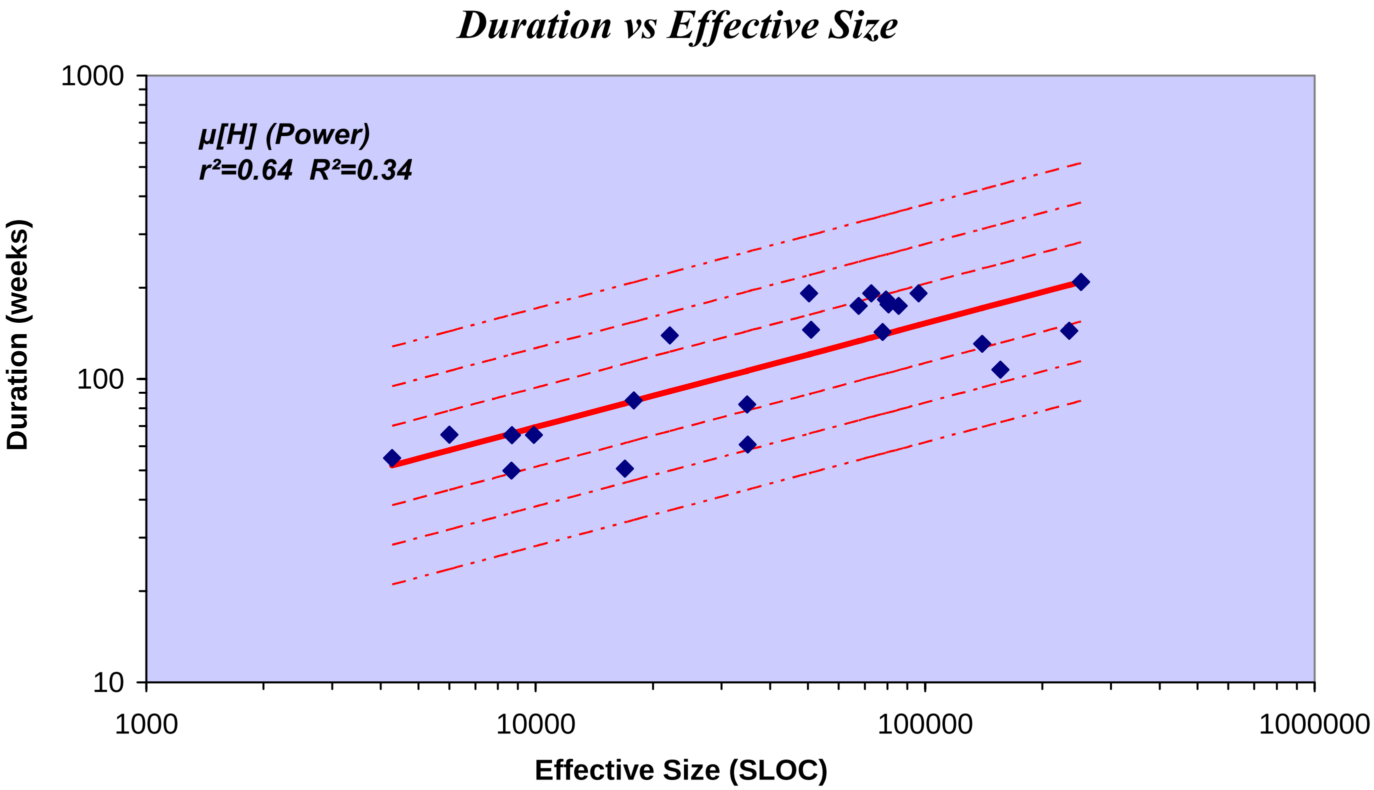
- ***No free software***
  - Effort (and hence cost) increases as effective size increases



Fundamental Observations about Software Development

- ***No instant software***
  - **Duration increases as effective size increases**

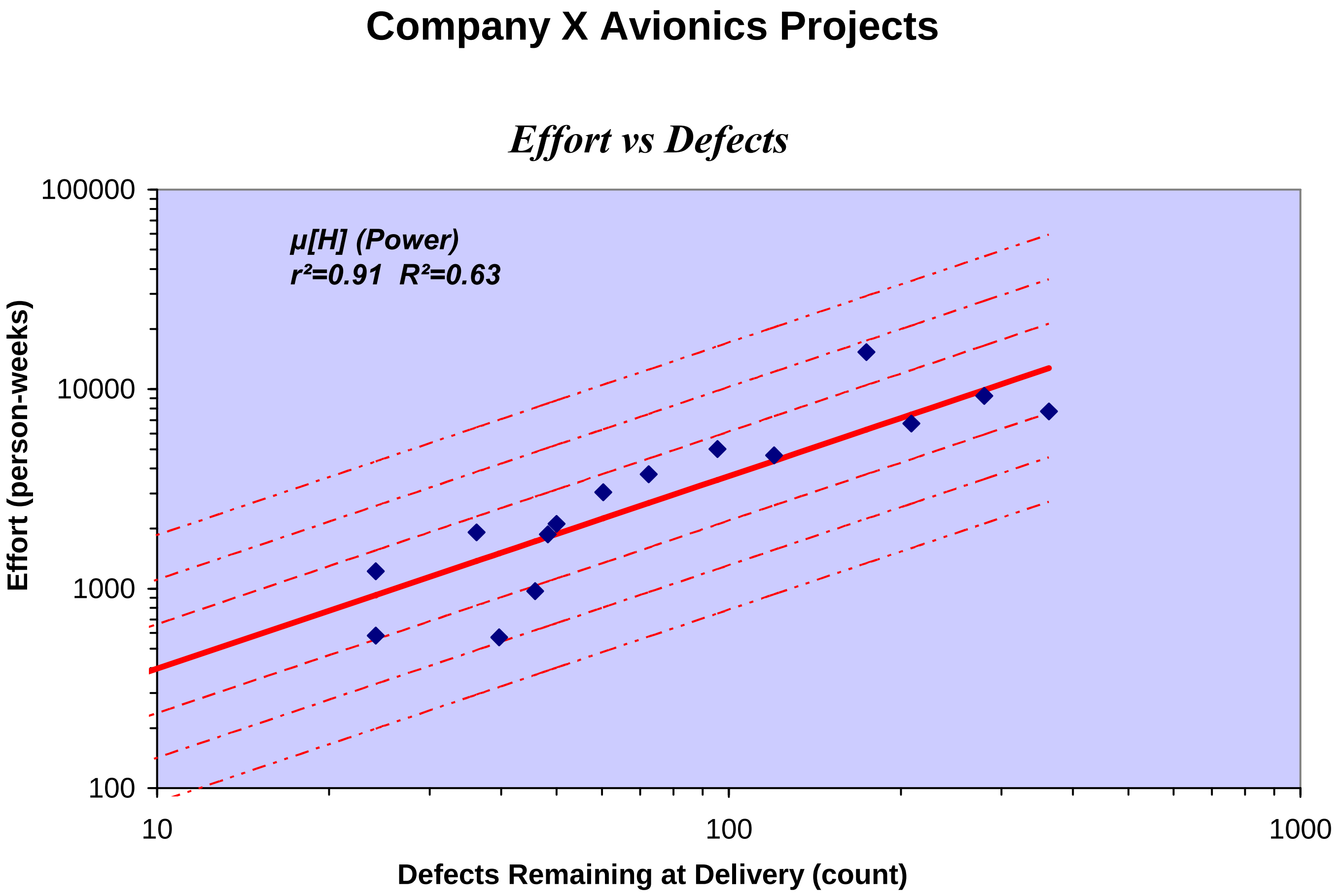
Company X Avionics Projects



TM

Fundamental Observations about Software Development

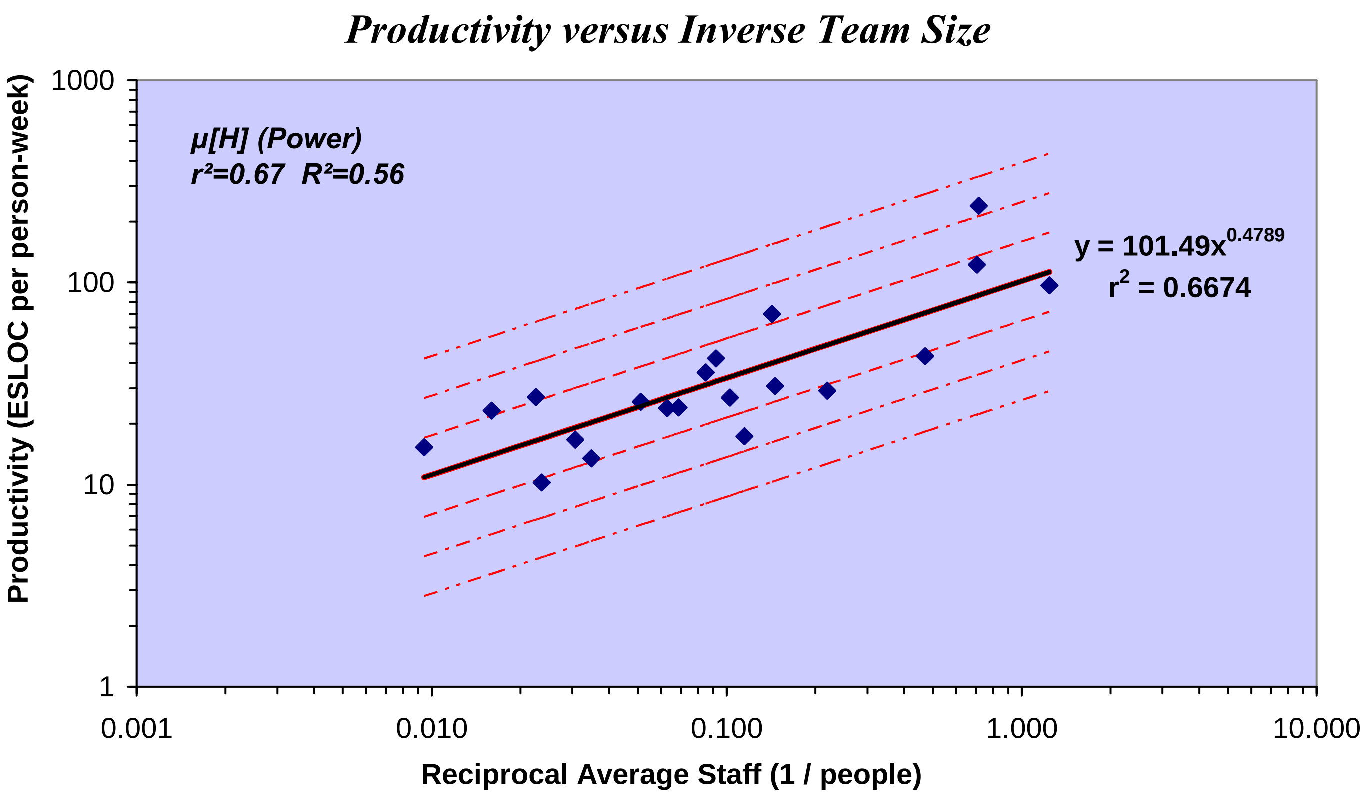
- ***No perfect software***
  - Defect count increases as effort increases



Fundamental Observations about Software Development

- ***Smaller teams are more productive***
  - **Productivity increases as team size decreases**

Company X Avionics Projects

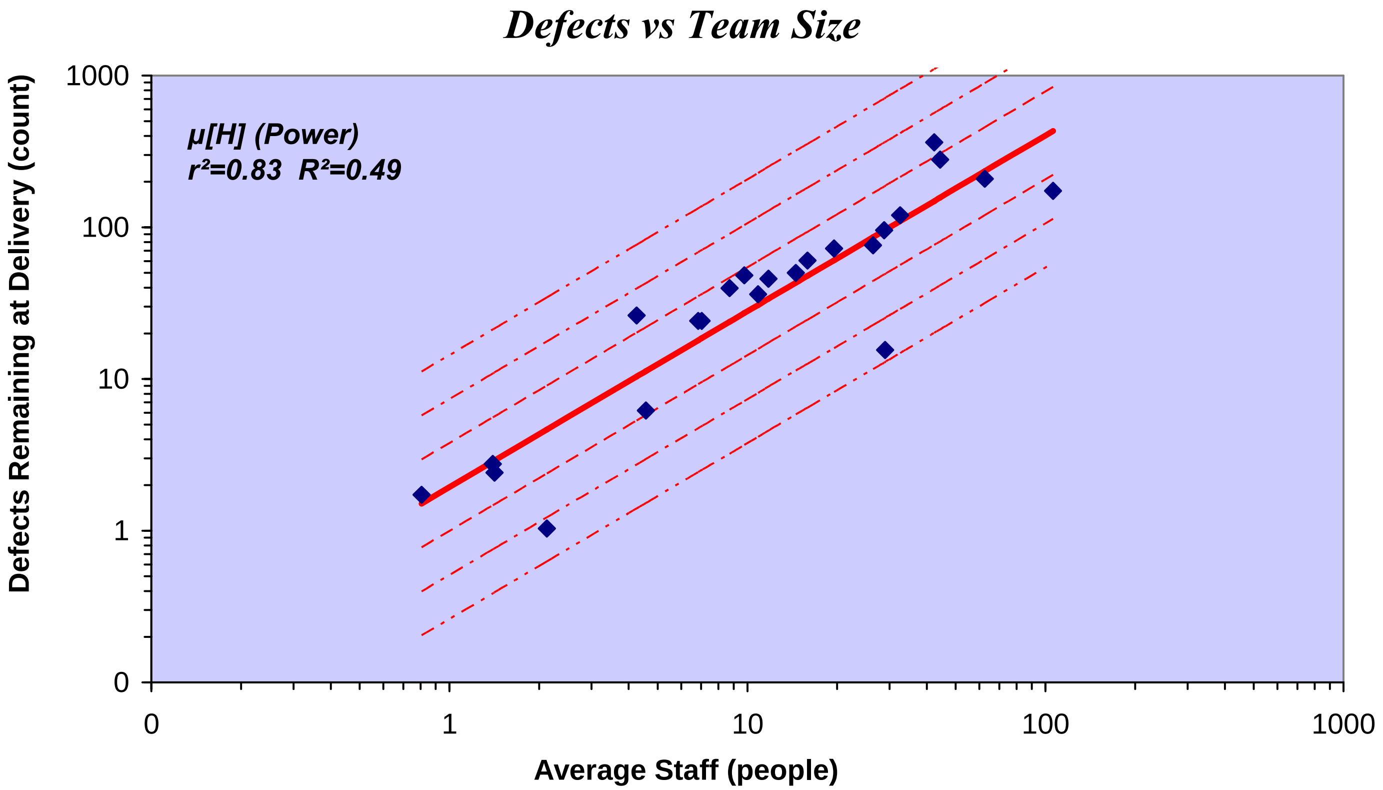


™

Fundamental Observations about Software Development

- ***Smaller teams produce fewer defects***
  - **Defects increase as team size increases**

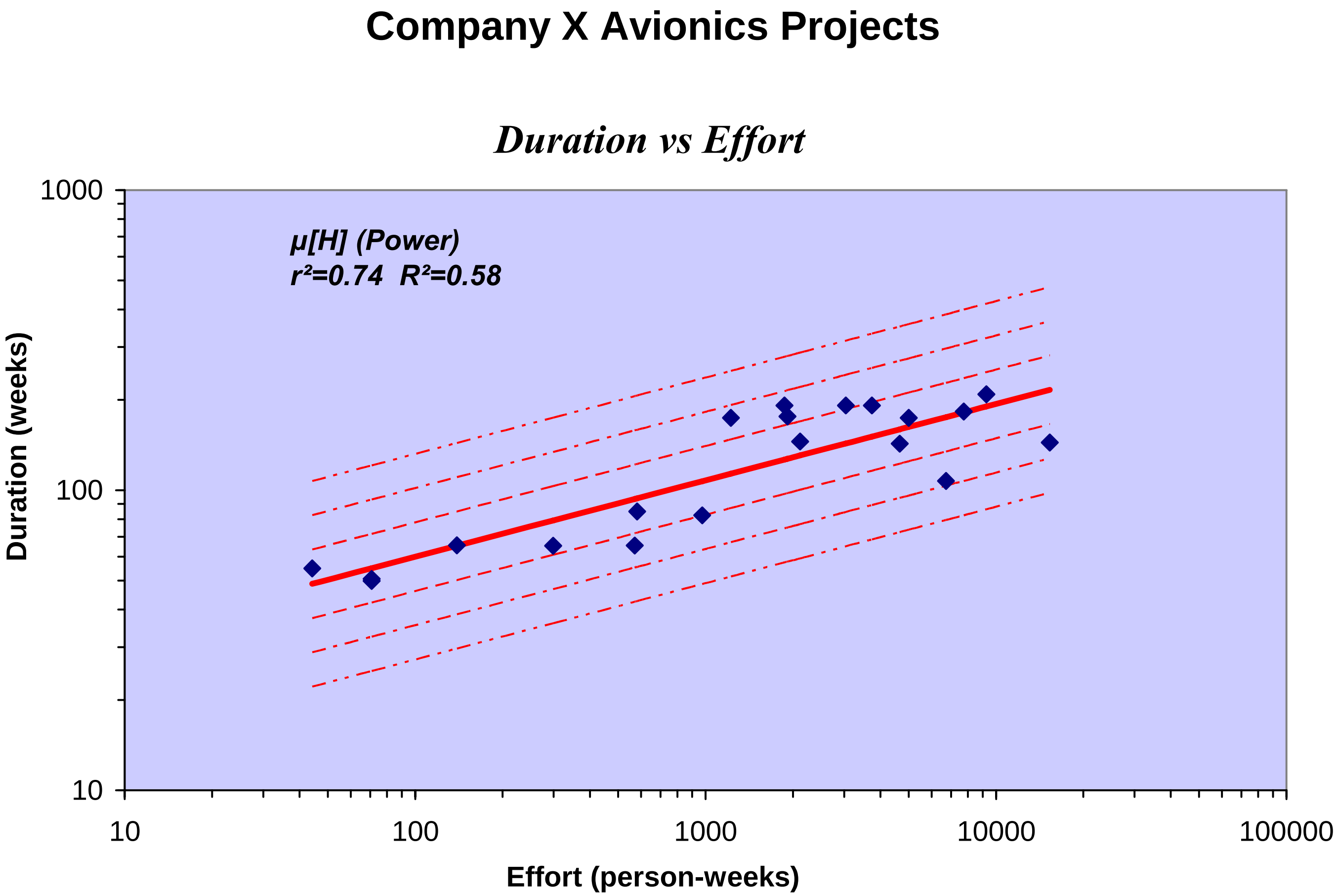
Company X Avionics Projects



TM

Fundamental Observations about Software Development

- ***Projects seek balance***
  - **Inherent equilibrium between effort and duration**
  - **Potential for concurrency**



Fundamental Empirically-Verified Hypotheses

- Software can be estimated as a multiplicative relationship between labor and time
  - More Size ➔ *More Effort and/or More Duration*
  - More Effort ➔ *More Size and/or Less Duration*
  - More Duration ➔ *More Size and/or Less Effort*
- Defects can be estimated as a ratio relationship between labor and time
  - More Effort ➔ *More Defects*
  - More Duration ➔ *Less Defects*

TM

Reasonable Corollary Truisms

- **Bigger Software → *More Defects***
- **Shorter Schedule with More People →**
  Higher Cost* and *More Defects
- **Longer Schedule with Fewer People →**
  Lower Cost* and *Fewer Defects

Three Laws of Software Project Dynamics

- **Software Construction Process Law**
  - *Software is made by people doing work* (effort) *over some period of time* (duration)*; the result being neither free nor perfect.*
  - **Increasing the number of people that work on a project dramatically increases communication overhead, which dramatically decreases productivity and dramatically increases defect propensity.**
- **Brooks' Law (limit) – *too many people* → ☹**
  - *Adding manpower to a late software project makes it later.*
  - **Every project, by its nature (divisibility or potential for concurrency), can effectively handle only so much management stress (only so many people); therefore, there exists, for every project, some *minimum achievable development time*.**
- **Parkinson's Law (limit) – *too much time* → ☹**
  - *Work expands so as to fill the time available for its completion.*
  - **Every project, by its nature (divisibility or potential for concurrency), has some point of maximum productivity; therefore, there exists, for every project, some *minimum achievable development effort*.**

Software Construction Process Law Mathematical Relationships

- **Software Productivity Law**

  Software can be estimated as a multiplicative relationship between labor and time.

$$Effort^{(\alpha_E)} \times Duration^{(\alpha_t)} = \frac{Size}{Efficiency}$$

- **Defect Propensity Law**

  Defects can be estimated as a ratio relationship between labor and time.

$$\frac{Effort^{(\varphi_E)}}{Duration^{(-\varphi_t)}} = \frac{Defects}{Defect\ \ Vulnerability}$$

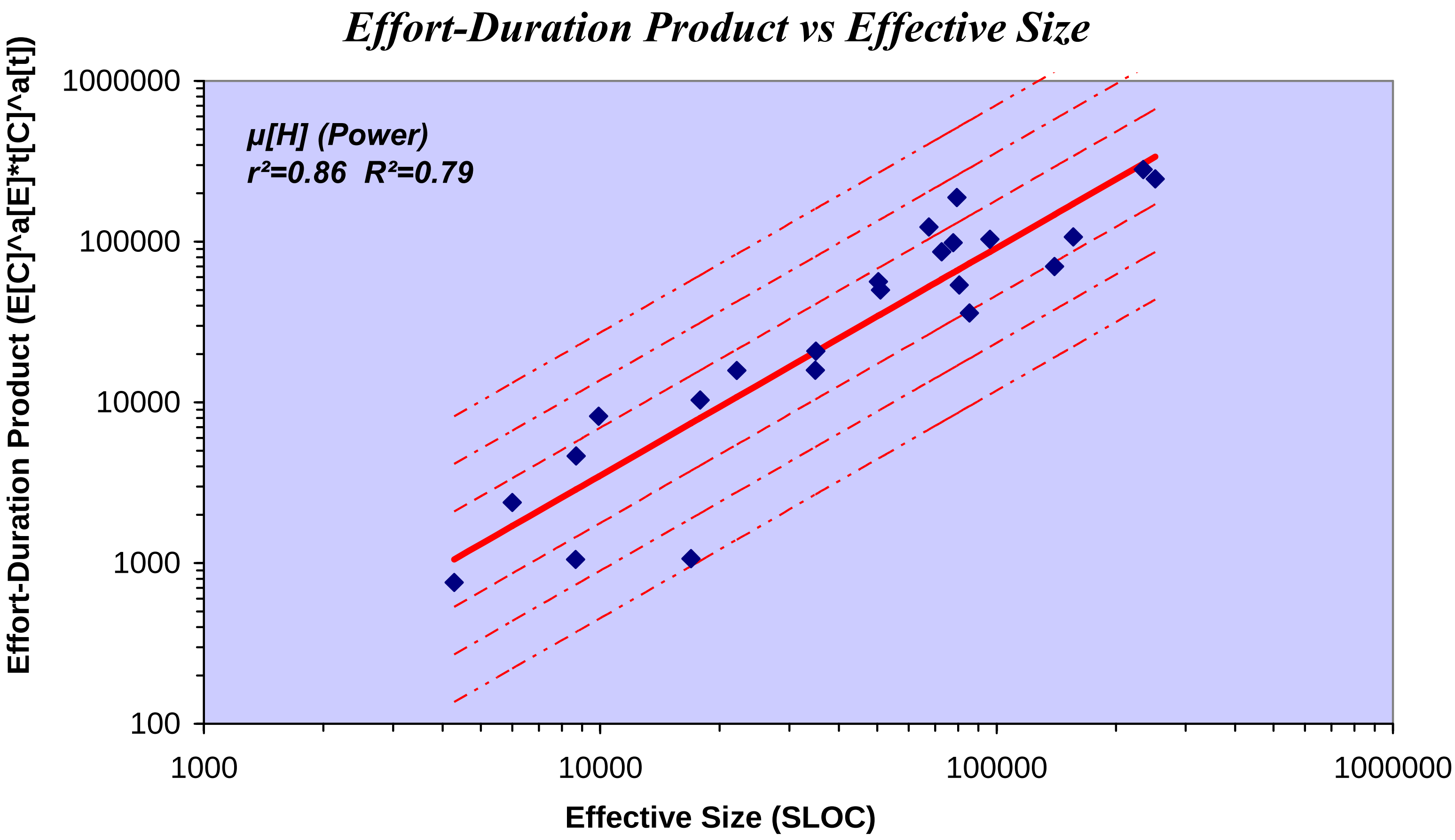
- **Management Stress Law**

  Management stress quantifies the balance (or imbalance) between effort and duration.

$$Management\ Stress = \frac{Effort}{Duration^{(\gamma)}}$$

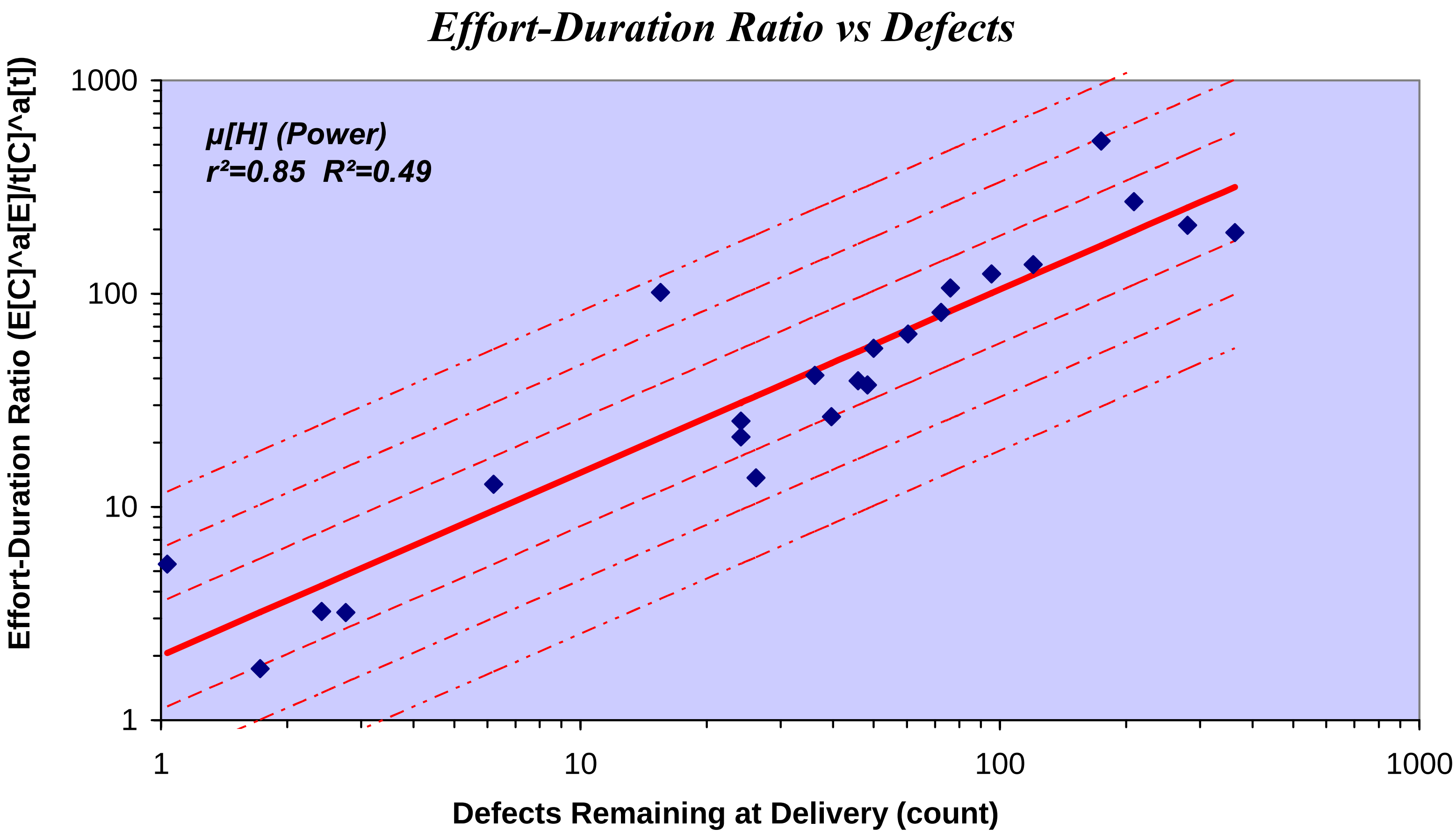
Empirically Verifying an Exponent-Calibrated Software Productivity Equation

Company X Avionics Projects



Empirically Verifying an Exponent-Calibrated Defect Propensity Equation

Company X Avionics Projects



Brooks' and Parkinson's Laws Mathematical Relationships

- **Brooks' Law (Limit)**

  For a given size and efficiency, there exists maximum achievable management stress (potential for concurrency) that limits, on the low side, the time necessary to complete the project.

$$\text{Management Stress}_{\max} \geq \frac{\text{Effort}}{\text{Duration}^{(\gamma)}}$$

$$\therefore \text{Management Stress}_{\max} = \frac{\text{Effort}_{t\min}}{\text{Duration}_{\min}^{(\gamma)}}$$

Too Little Time

- **Parkinson's Law (Limit)**

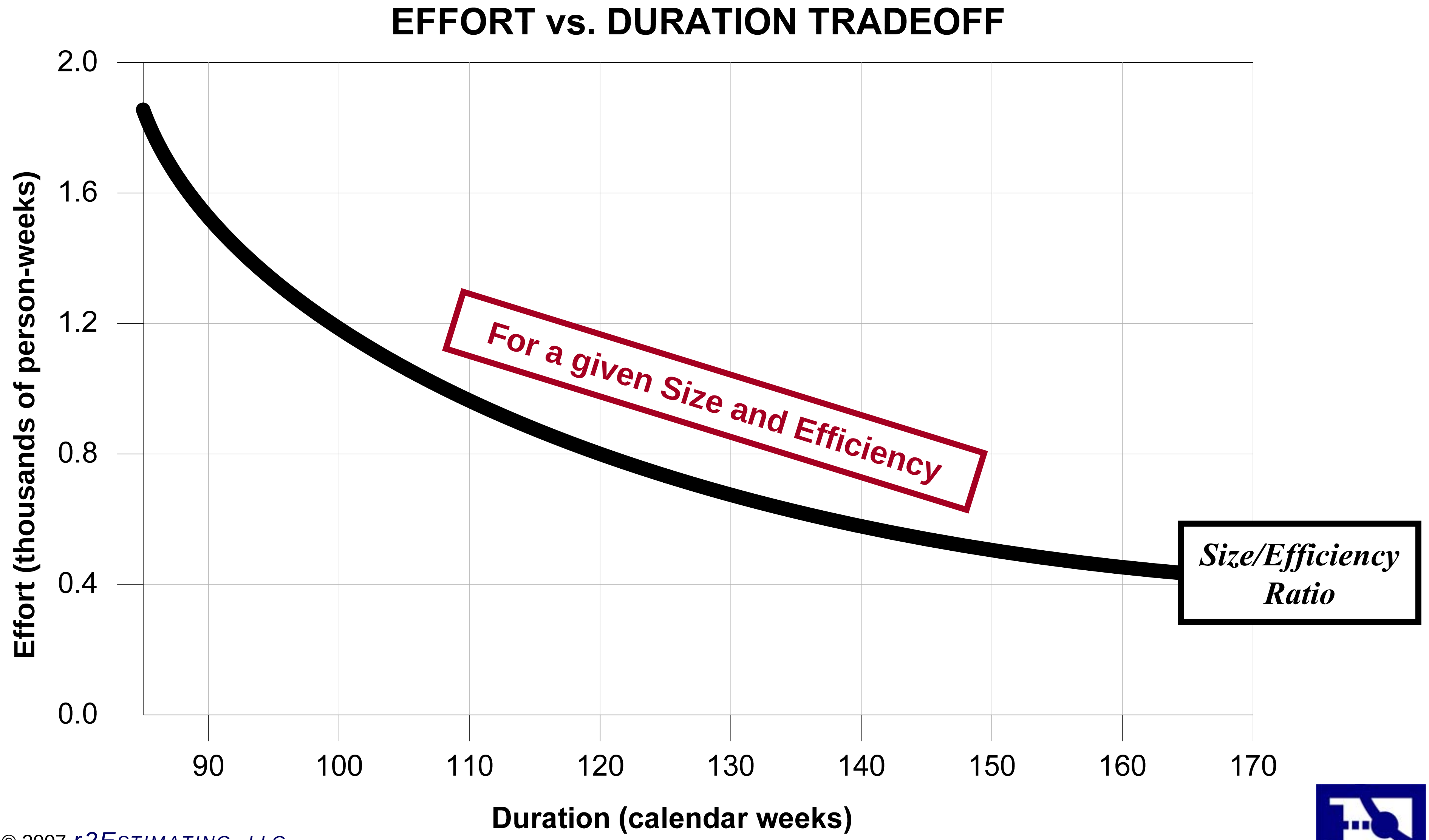
  For a given size and efficiency, there exists minimum practical management stress (potential for concurrency) that limits, on the low side, the effort necessary to complete the project.

$$\text{Management Stress}_{\min} \leq \frac{\text{Effort}}{\text{Duration}^{(\gamma)}}$$

$$\therefore \text{Management Stress}_{\min} = \frac{\text{Effort}_{\min}}{\text{Duration}_{E\min}^{(\gamma)}}$$

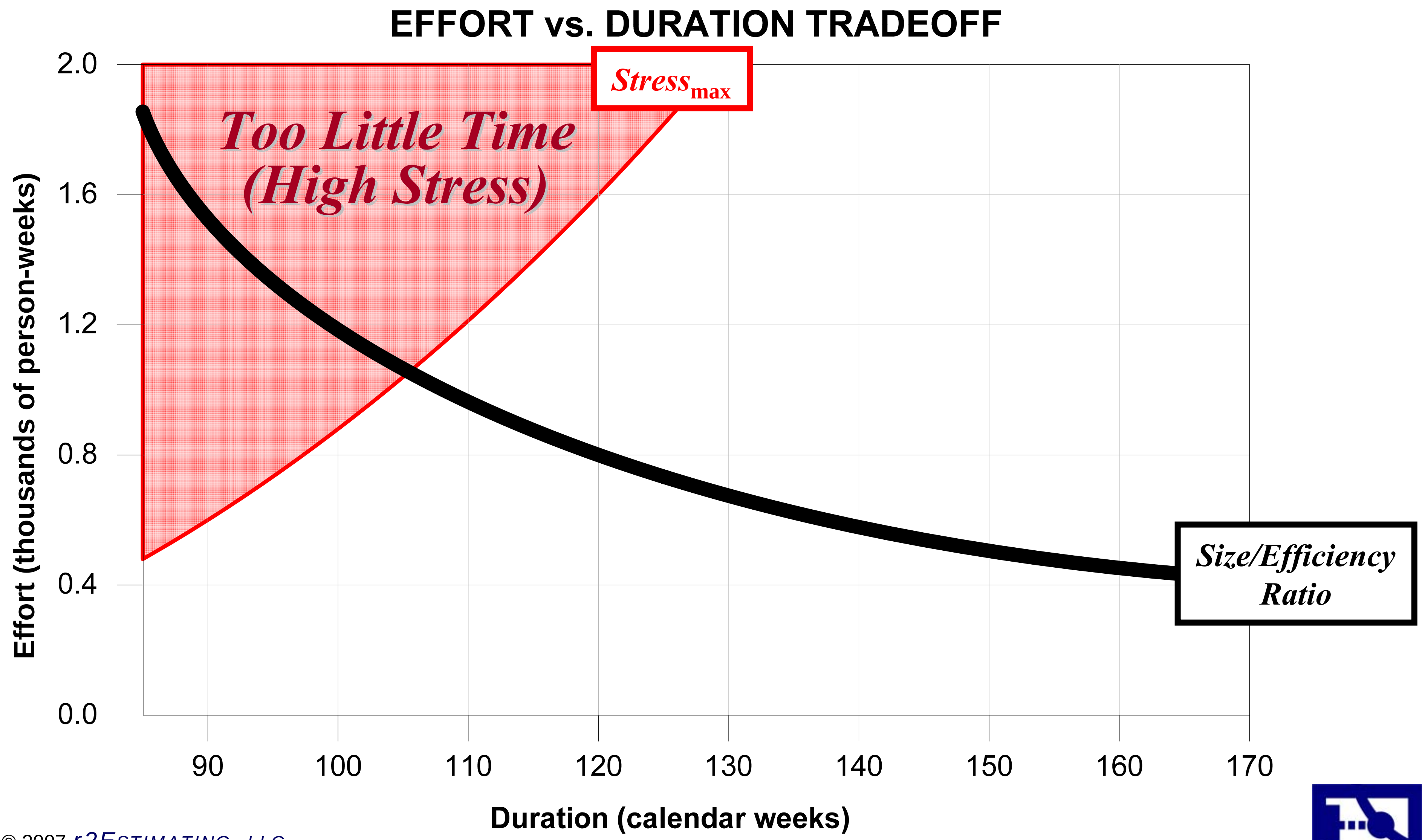
Too Much Time

Software Productivity Law



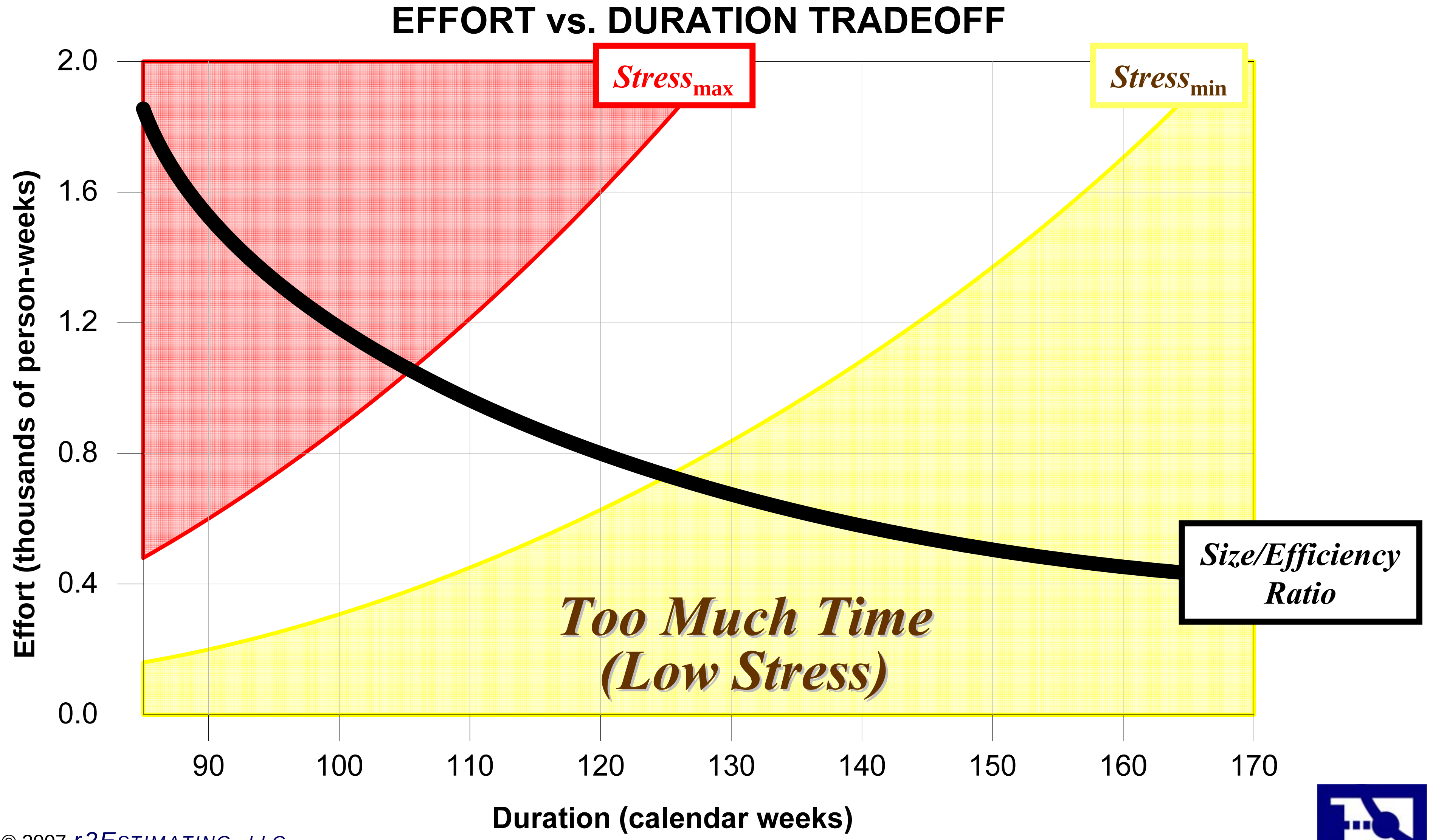
Presented at the 2007 ISPA/SCEA Joint Annual International Conference and Workshop - www.iceaaonline.com

Brooks' Law
Minimum Time Limit

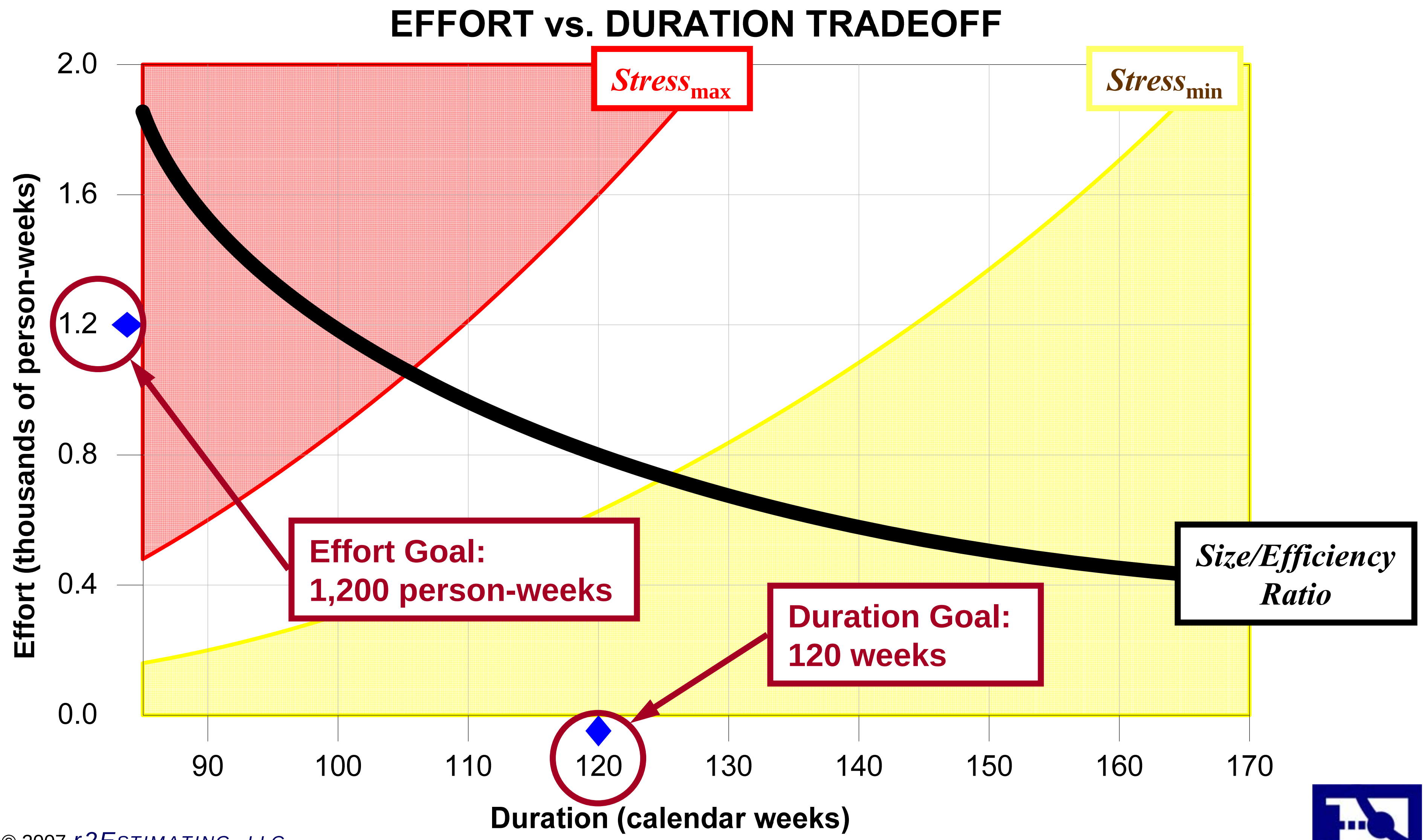


™

Parkinson's Law
Minimum Effort Limit



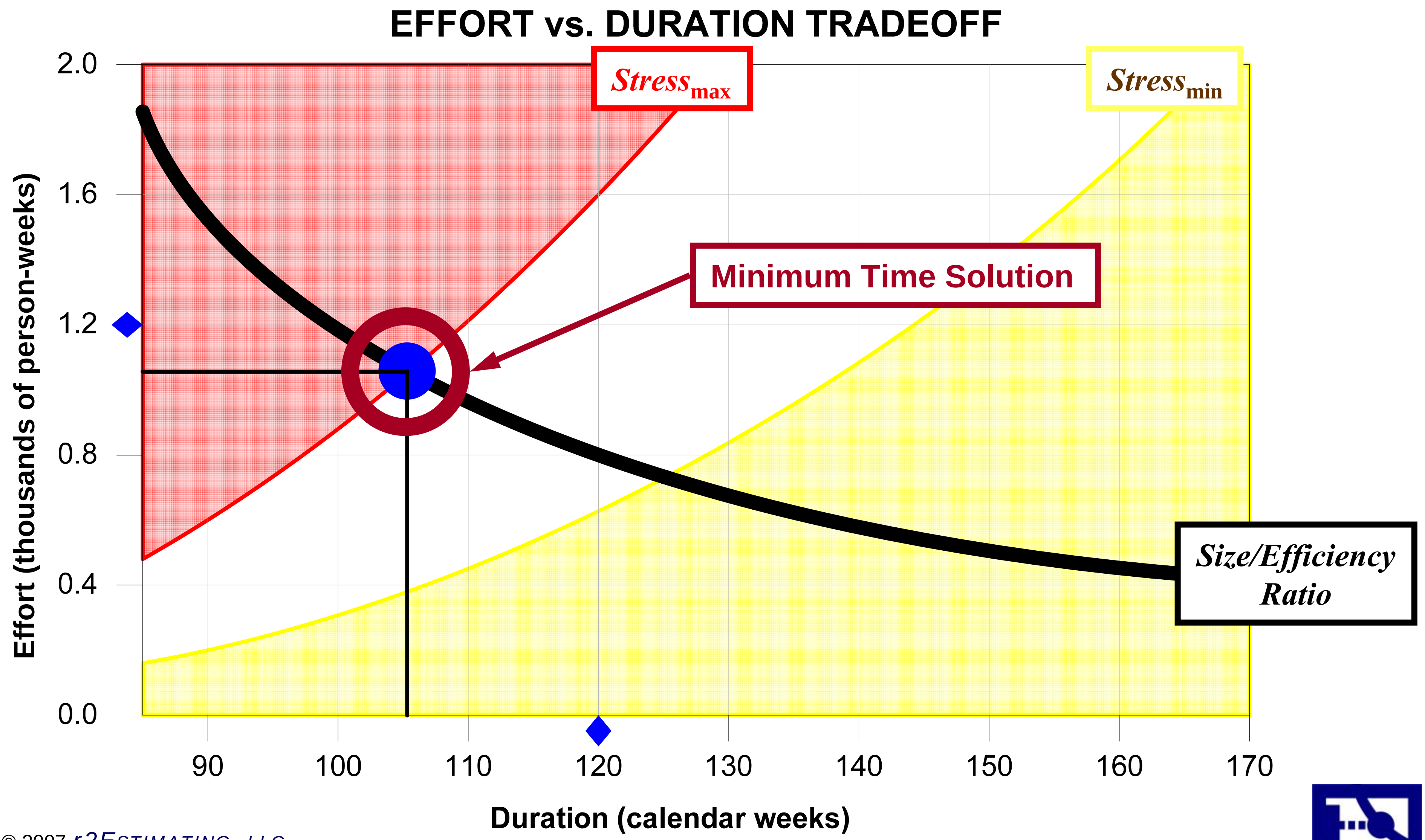
Goals (Constraints)



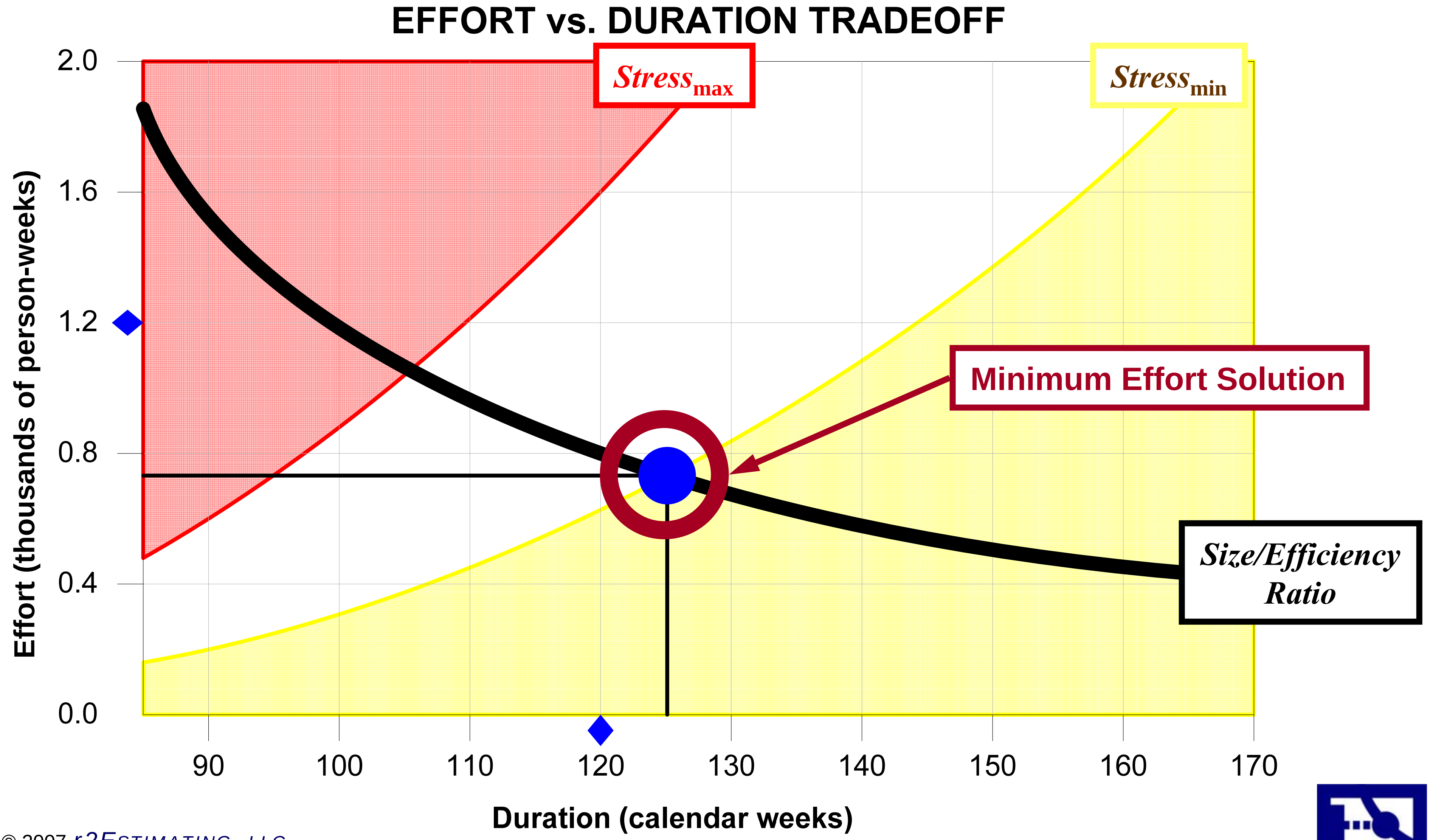
Effort Goal: 1,200 person-weeks

Duration Goal: 120 weeks

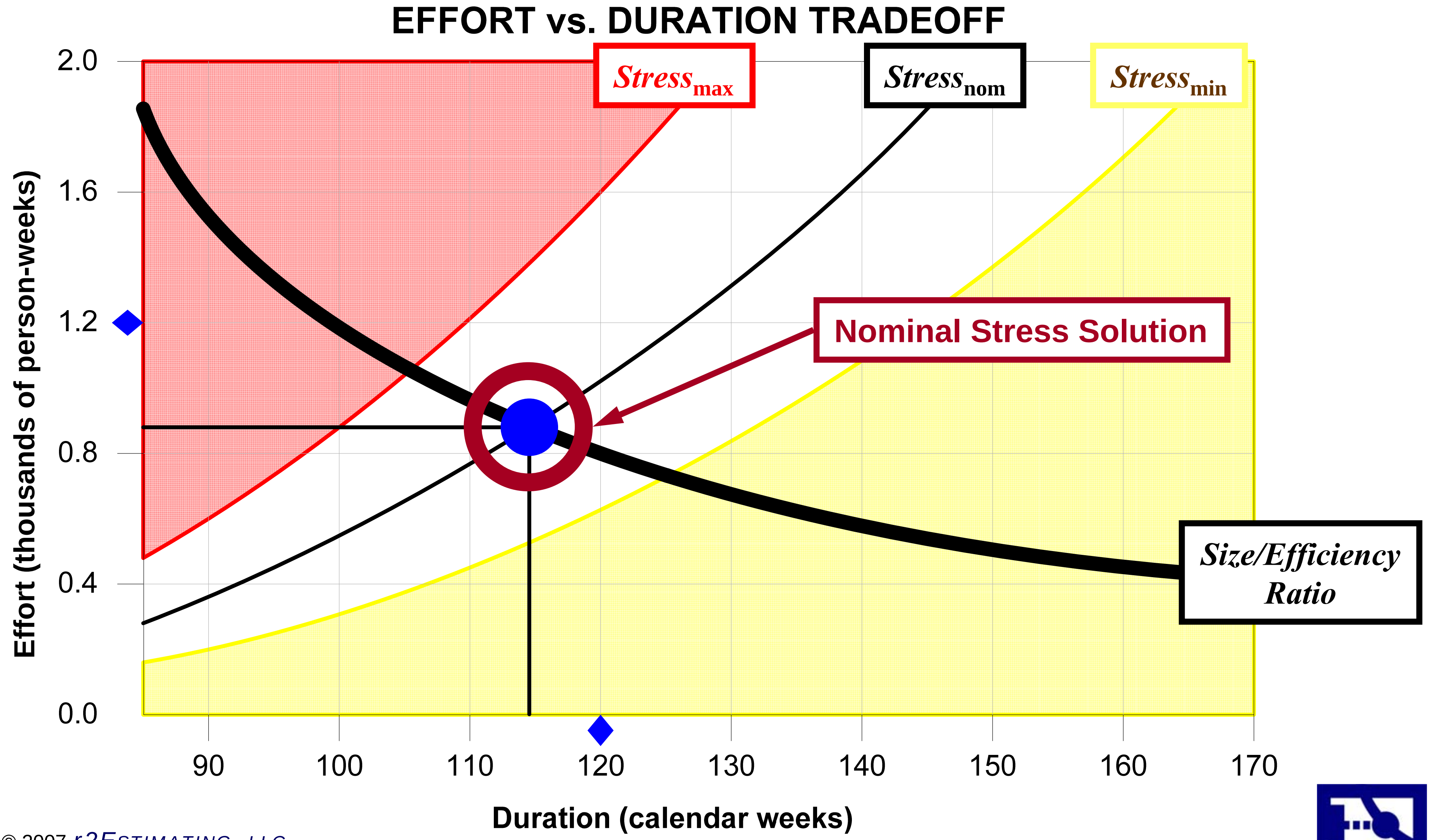
Brooks' Law & Minimum Time *(High Stress)* Solution



Parkinson's Law & Minimum Effort *(Low Stress)* Solution



Typical *(Nominal Stress)* Solution



TM

The Rest of the Story

- Calibrating to Historical Data
- Emulation of Existing Models
  - COCOMO 81
  - COCOMO II
  - Jensen (Seer)
  - NPR
- Input Uncertainty ➔ Output Confidence
- Integrating Estimating with Program Assessment