Over 30 years of empowering informed decision making so that organizations can achieve their goals with greater confidence.
Bad Estimates Are A Root Cause of Project Failure

- An estimate is the most knowledgeable statement you can make at a particular point in time regarding:
  - Effort / Cost
  - Schedule
  - Staffing
  - Risk
  - Reliability

- Estimates more precise with progress

- **A WELL FORMED ESTIMATE IS A DISTRIBUTION**
Vision of Parametrics Over the Next 20 Years Presented 2002

- Parametrics will be integrated into engineering processes and engineering decision making
  - For example: Cost of a system derived from simulation models of that system
- Parametrics will lose its “magic” reputation
  - Improved processes will yield better data
    - Augmentation of parametrics with more data will increase believability among engineers and management
    - The nay-sayers who say that can make parametric models say anything they want will be replaced with belief
- More dynamic parametrics based on both historical and real time data
- Parametric models will be available to use as “objects” in financial and engineering analysis
Add Performance Measurement (Enhanced Earned Value) methods to your toolbox.

**SEE Big Picture**

More dynamic parametrics based on both historical and real time data.
COST IQ - OBTAINING COST FROM REQUIREMENTS
A Case Based Reasoning system that can transform high level requirements and specifications into a cost modeling workup within a sophisticated cost estimating model.

- DOMAIN: UAV, Missile, Ship, Etc.
- CBR
- NOTIONAL DESIGN
- COST MODEL

Parametrics will lose its “magic” reputation.
CAD TO COST CAPABILITIES

Parametrics will be integrated into engineering processes and engineering decision making.
COMPREHENSIVE SOLUTION SUITE

Predictive Analytics

SEER-SEM
Software/application development, maintenance, integration and testing for Total Ownership Cost

SEER-H
System, hardware and electronics development, production and support for Total Ownership Cost

SEER-IT
Total Ownership Costing for IT services and digital ecosystem strategies

SEER-MFG
Hardware manufacturing and assembly with automated CAD to Cost

SEER-SYS
Systems Engineering cost estimation for systems of all sizes and complexities

SEER-SPACE
Estimates lifecycle cost for key instruments and spacecraft subsystems
Businesses need to make decisions – either by Design or by Default

Many believe “If scope and deadline are not defined at the start it is not a program but operations”

Agile is excellent when actually planned

An Estimate is part of a plan, not a work product in itself
Human Nature: YOUR PEOPLE Are Optimism Biased

*Harvard Business Review explains this Nobel Prize Winning Phenomenon:*

- Humans seem hardwired to be optimists
- Routinely exaggerate benefits and discount costs
- Bias permeates opinions & decisions & causes waste & failure

**Solution - Temper with “outside view”:**
Past Measurement Results, traditional forecasting, risk analysis and statistical parametrics can help

Don’t remove optimism, but balance optimism and realism
Anchoring Biases Estimates

(Source: myweb.liu.edu/~uroy/eco23psy23/ppt/04-anchoring.pptx)

01 Observe number when a wheel of fortune is spun

02 Ask if U.N. African countries greater than or less than fixed number

03 Asked to guess the number of African countries in the U.N.

Result: Giving a number creates biases those given higher numbers on the wheel guessed bigger numbers in Step 3
Averages Gone Wrong

Example

Actual Crest above 50 feet

Dikes breached flood forced 50,000 people to evacuate

$2 billion property damage

Weather Service forecast North Dakota’s rising Red River would crest at 49 feet.

Made flood management plans based on that average
Gunning for Models
(Adapted from Hubbard)

George E. P. Box: “Essentially, all models are wrong, but some are useful.”

Many try to substitute biased opinions for viable models

Be careful of red herring arguments against models

“We cannot model that…it is too complex.”

“Models will have error and therefore we should not attempt it.”

“We don’t have sufficient data to use for a model.”

“It works but we can’t see all data so we should not use it”
Correlation Doesn’t Always Mean Causation
(Source: www.memolition.com)
<table>
<thead>
<tr>
<th>Category</th>
<th>Sub Category</th>
<th>Application</th>
<th>Human/Technical/Email (1:1)</th>
<th>Identity</th>
<th>Prevent</th>
<th>Detect</th>
<th>Respond</th>
<th>Recover</th>
<th>Least</th>
<th>Likely</th>
<th>Most</th>
<th>Unit (cost per)</th>
<th>Unit (cost per)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Security</td>
<td>Data Encryption</td>
<td>Total</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>$180.00</td>
<td>$1,105.60</td>
<td>$17.00</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Email Security</td>
<td>Email Encryption</td>
<td>Total</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>$35,009.02</td>
<td>$35,009.02</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Identity Governance</td>
<td>Federated identity</td>
<td>Single Sign on Service</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Identity Governance</td>
<td>LDAP repository</td>
<td>LDAP-Process</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Identity Governance</td>
<td>LDAP repository</td>
<td>LDAP-Process</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Identity Governance</td>
<td>LDAP repository</td>
<td>LDAP-Process</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Identity Governance</td>
<td>Identity Access Management</td>
<td>Lifetime</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Identity Governance</td>
<td>Identity Access Management</td>
<td>Lifetime</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Identity Governance</td>
<td>Identity Access Management</td>
<td>Lifetime</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Identity Governance</td>
<td>Identity Access Management</td>
<td>Lifetime</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Identity Governance</td>
<td>Identity Access Management</td>
<td>Lifetime</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Most people are significantly **overconfident** about their estimates ... especially educated professionals
Example: Project Cost Alone Is not The Cost of IT Failure (Source: HBR)

• Case Study: Levi Strauss
  ▪ $5M ERP deployment contracted
  ▪ Risks seemed small
  ▪ Difficulty interfacing with customer’s systems
  ▪ Had to shut down production
  ▪ Unable to fill orders for 3 weeks
  ▪ $192.5M charge against earnings on a $5M IT project failure

“IT projects touch so many aspects of an organization they pose a new singular risk”
AGILE ESTIMATES
Are they necessary?

#NOESTIMATES
- Estimates are difficult to produce
- Provide little to no value
- Estimation is overhead and should be minimized

#ESTIMATES
- Organizations need to do budget planning
- Estimates are needed to make informed decisions
- Managers need estimates for accountability to shareholders

UNDERLYING TRUTHS
Software estimation is challenging. Agile developers see estimates as committing them to a schedule and therefore they are antithetical to the Agile Manifesto.

Software estimates drive decision making - they are not just for the developers.

Total ownership cost should be considered for the immortal systems and as a result, more emphasis and research should be and is being applied to the area of software maintenance.
# NoEstimates Makes Sense From Developer View

- No missed deadlines
- No cost overruns
- Minimum Viable Product may reduce excess functionality
- Some kind of progress made every few weeks
- Do what the customer wants as they change their mind
Agile PMI View

- Agile methods deal with projects that are complex
- Involving many design unknowns
- Traditional project management requires a clear scope and defined parameters at the onset
- Traditional project management deals with uncertainty when the “how” is not defined
- Agile methods are meant to deal with both uncertainty and ambiguity, when the “what” is not defined

"Agile management is a development method, not a project method"
DEFENSE SCIENCE BOARD*
Recommendation

Recommendation 1: Implement the “Software Factory”

Recommendation 2: Adopt continuous iterative development best practices (continuing through sustainment) for software.

Recommendation 3: Implement: Multiple vendors to begin work with down select; Service cost estimators should modernize cost/schedule estimation processes; Project manager should build program-appropriate frameworks for status metrics; examples include: sprint burndown, epic and release burndown, velocity trending, control chart, line of balance and cumulative flow diagrams.

Recommendation 4: Current and legacy programs should plan transition to a software factory and continuous iterative development

Recommendation 5: Develop a modern software development expertise

Recommendation 6: Software is Immortal

Recommendation 7: Implement Independent Validation and Verification (IV&V)

Predictions for the next 20 years (2019)

Data Science will rule
We will learn from data but sometimes we will learn what is rather than what can be.

Artificial intelligence will make decisions
Some good, some really bad

There will be disasters
SIZE CONTINUES TO BE MAIN DRIVER

FUNCTION POINTS

USER STORIES

USE CASES

SLOC

STORY POINTS

CHOOSE A METRIC AND BE CONSISTENT!

YOUR SUBTITLE WILL GO HERE

USER STORIES

SLOC

STORY POINTS

T-SHIRT SIZING

25
Parametric Estimation for Agile Projects

Features

Project Characteristics

Team Dynamics

Development Team

<table>
<thead>
<tr>
<th>Description</th>
<th>Fraud Portal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform</td>
<td>Web Based Development</td>
</tr>
<tr>
<td>Application</td>
<td>Customer Relationship Management</td>
</tr>
<tr>
<td>Acquisition Method</td>
<td>*New Development</td>
</tr>
<tr>
<td>Agile Framework</td>
<td>Agile Full</td>
</tr>
<tr>
<td>Development Standard</td>
<td>Commercial</td>
</tr>
</tbody>
</table>

Collection of Functionality

Functionality Backing

Iteration to Build

Delivered Functionality

Working System

Fixes, Enhancements, Sustainment

Planning

Iterations

Warranty & Maintenance

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Team A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint Number</td>
<td>1</td>
</tr>
<tr>
<td>Sprint Duration Months</td>
<td>0.92</td>
</tr>
<tr>
<td>Sprint Effort Months</td>
<td>6.46</td>
</tr>
<tr>
<td>Sprint Effort Hours</td>
<td>981</td>
</tr>
<tr>
<td>Sprint Labor Cost</td>
<td>134,290</td>
</tr>
<tr>
<td>Sprint Velocity (UFP)</td>
<td>103.00</td>
</tr>
<tr>
<td>Team Size</td>
<td>7.00</td>
</tr>
<tr>
<td>Start Date</td>
<td>12/27/2018</td>
</tr>
<tr>
<td>End Date</td>
<td>1/24/2019</td>
</tr>
</tbody>
</table>
ESTIMATING METHODOLOGIES

Methodology 1: Many Agile programs are fixed price, it is often just a matter of labor rates times quantity

Methodology 2: Simple Build-up approach based on averages can be defined as: Sprint Team Size (SS) x Sprint length (Sp time) x Number of Sprints (# Sprints)

Methodology 3: Structured approach based on established “velocity” – most often used internally by the developer since detailed/sensitive data are available to them

Methodology 4: Automated Models approach based on a size metric – which may be difficult to quantify
  - There is a fixed relationship between size and effort, e.g. (Effort**n)*Time = Size/Technology
  - Results are then modified by current trends and analyses
  - Total effort can be distributed by a mathematical model; e.g. Weibull, Rayleigh

Methodology 5: Factor/Complexity approach based on data generated in early iterations

The Sprint work projections often follow the Weibull or Rayleigh distribution
SOFTWARE COST GROWTH

Significant Reasons for Software Cost Growth

01. Scope Creep: Requirements Growth
02. Poor Input to Estimate
03. Failure to Clearly Define the Initial Scope
04. Unrealistic Expectations and Assumption
05. Failure to Declare, Track & Reduce Risk & Uncertainties
06. Lack of Internal Peer Review
07. Lack of Estimation Experience
08. Failure to Consider Environmental Factors
09. Failure in the Estimation Tool/Process
10. Estimating to a Target Assumption
MANAGING MODERN SOFTWARE DEVELOPMENT PROGRAMS

Measure the Right Thing – Backlog – Velocity – Burndown Charts
Manage Expectations / Set Realistic Time Frames
Align the Work Streams
Seek Objectivity
TYPICAL HYBRID AGILE DEVELOPMENT

TWO CLASSES OF FEDERAL PROGRAMS

Incremental programs
- Follow the commercial Agile practices
  - Small user stories
  - Single sprint, or even multiple user stories being completed in a single sprint
- Generally not applying a full EVM process

Transformational programs
- Creating completely new capabilities
- “Hybrid-Agile” approach applied
  - Longer sprints
  - Larger conceptual stories/features
  - Full EVM process.

Testing and Sustainment (sometimes in the Sprint sometimes a separate activity)
An ROI Analysis of A New System: Should We Fund This

<table>
<thead>
<tr>
<th>Cost of capital</th>
<th>8.0%</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Initial Investment</th>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investment</td>
<td>$100,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Increase/dec. in revenue</td>
<td></td>
<td>($40,000)</td>
<td>$60,000</td>
</tr>
<tr>
<td>Increase/dec. in op. exp.</td>
<td>$90,000</td>
<td>$70,000</td>
<td>$70,000</td>
</tr>
<tr>
<td>Cash Flow</td>
<td>($100,000)</td>
<td>($130,000)</td>
<td>($10,000)</td>
</tr>
<tr>
<td>PV of Cash Flow</td>
<td>($100,000)</td>
<td>($120,370)</td>
<td>($8,572)</td>
</tr>
<tr>
<td>NPV</td>
<td>60,563</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRR</td>
<td>13.5%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROI</td>
<td>121%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Can we do better?
- Will stakeholders tolerate a loss for 3 years?
- What is the risk?
Dealing With the “Problem of Assumptions”

• Assumptions are essential but…
• Incorrect assumptions can drive an estimate to uselessness
• Use an assumption verification process