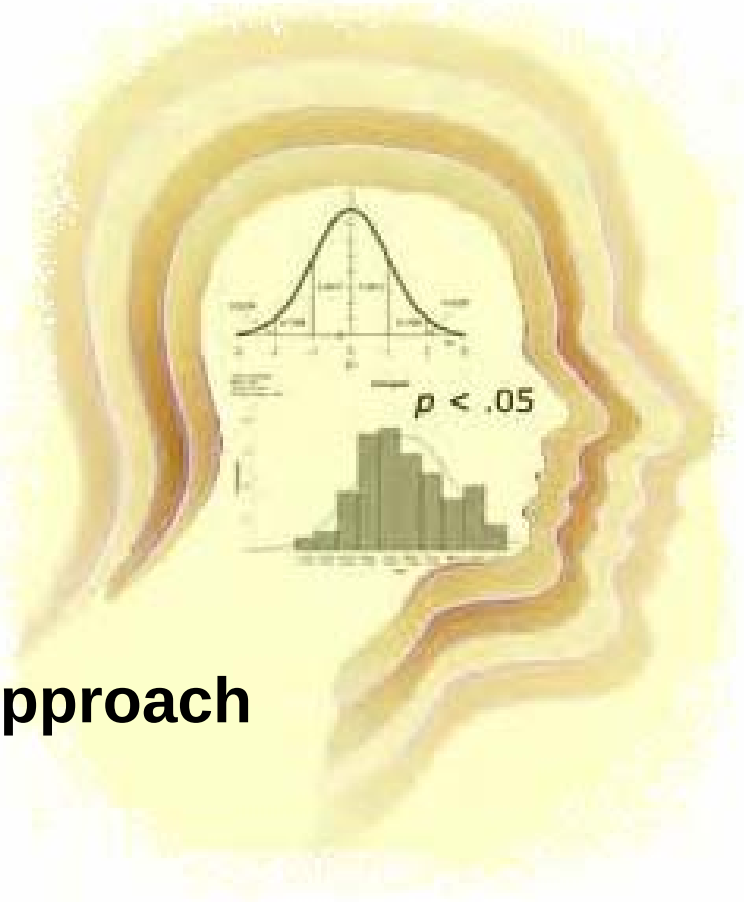
October 2006
PMI DC Chapter
Knowledge Exchange Forum



Software Estimation

A Quantitative, Repeatable Approach

Booz | Allen | Hamilton

Three key components to a quantitative software estimation methodology

Defined Estimation Process

Standardized Size Measure

Structured Estimation Tool

It's All About the Process

Defined Estimation Process

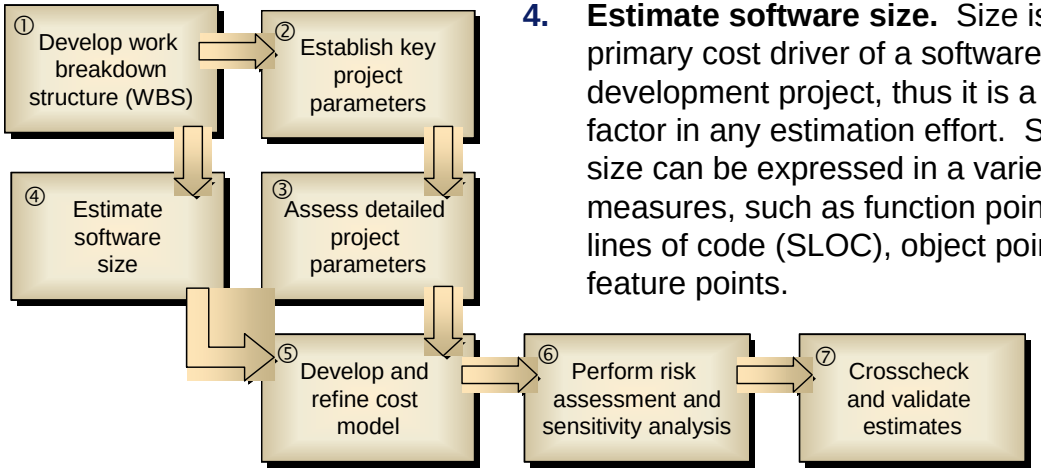
- In order to improve, you first must understand where you are
- In order to understand where you are, you need to set a baseline
- In order to baseline, you should have a repeatable, standard process
- The process should be documented and followed

If an estimation process is not defined and followed, it is impossible to tell where improvements are even necessary or possible

Example Estimation Process

Defined Estimation Process

1. **Develop WBS.** The WBS provides an overview of the project that will be analyzed. It maps out the scope of the application, identifies interfaces, and breaks the project down into components to ensure that all required functionality is captured
2. **Establish key project parameters.** Key project parameters put the project into context and set the framework for the rest of the analysis. Key project parameters are primary mission or operating environment, primary application function, acquisition method, development method, and development standard.
3. **Assess detailed project parameters.** Detailed project parameters focus on the specific characteristics of a particular project. These details include the development environment, personnel capabilities and experience, product requirements, target environment, and labor rates. Estimates can be developed with as little or as much project-specific detail as is available at the time of the analysis.

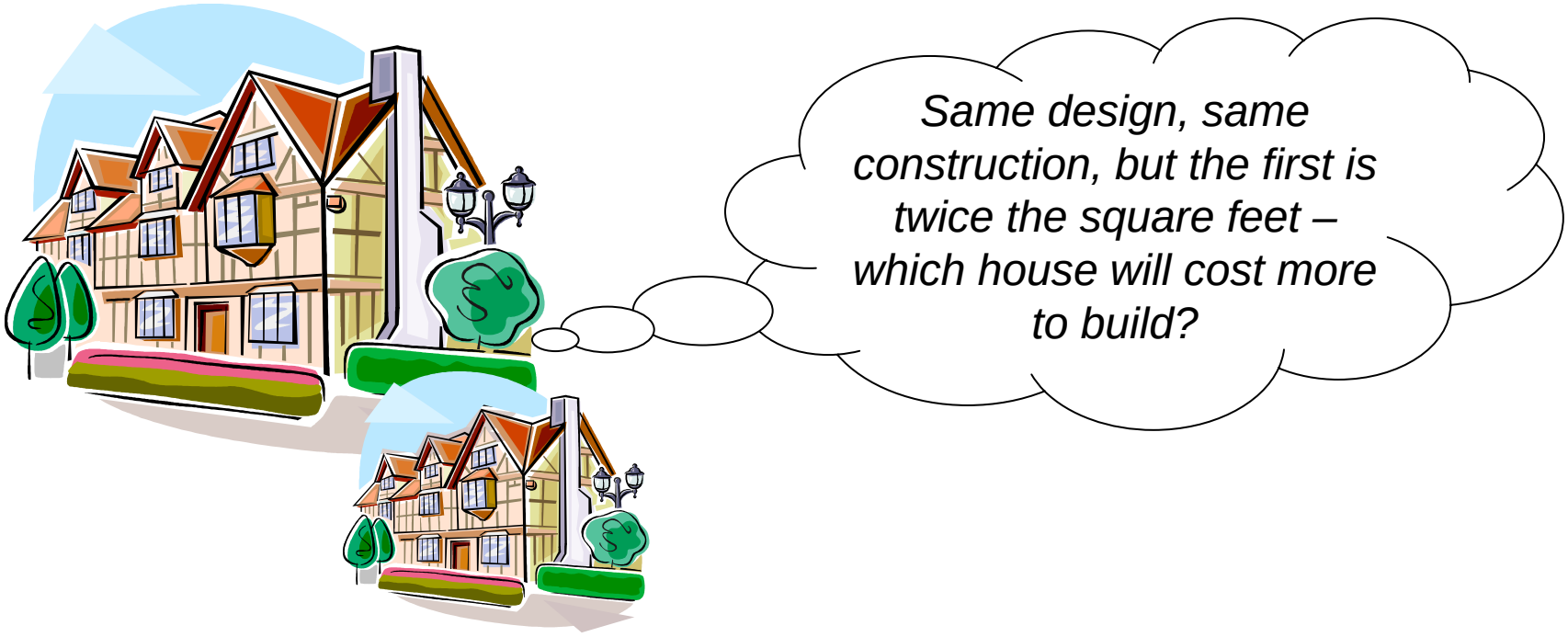


4. **Estimate software size.** Size is the primary cost driver of a software development project, thus it is a crucial factor in any estimation effort. Software size can be expressed in a variety of measures, such as function points, source lines of code (SLOC), object points, and feature points.
5. **Develop and refine cost model.** This step utilizes a parametric tool to combine the WBS, project parameters, and size estimates to generate cost, schedule, and effort estimates.
6. **Perform risk assessment and sensitivity analysis.** The parametric models produce not only a “most likely” point estimate, but also a range of possible values and corresponding probabilities. Reviews and assess model inputs to underscore areas that might be crucial to the success of the project, then conducts sensitivity analyses to understand the potential impacts on cost and schedule.
7. **Crosscheck and validate estimates.** This final step places the project estimates into context. Compare model output and resulting metrics to industry benchmarks, actual data from similar projects, and actual data from the organization (when available).

Size Matters

Standardized Size Measure

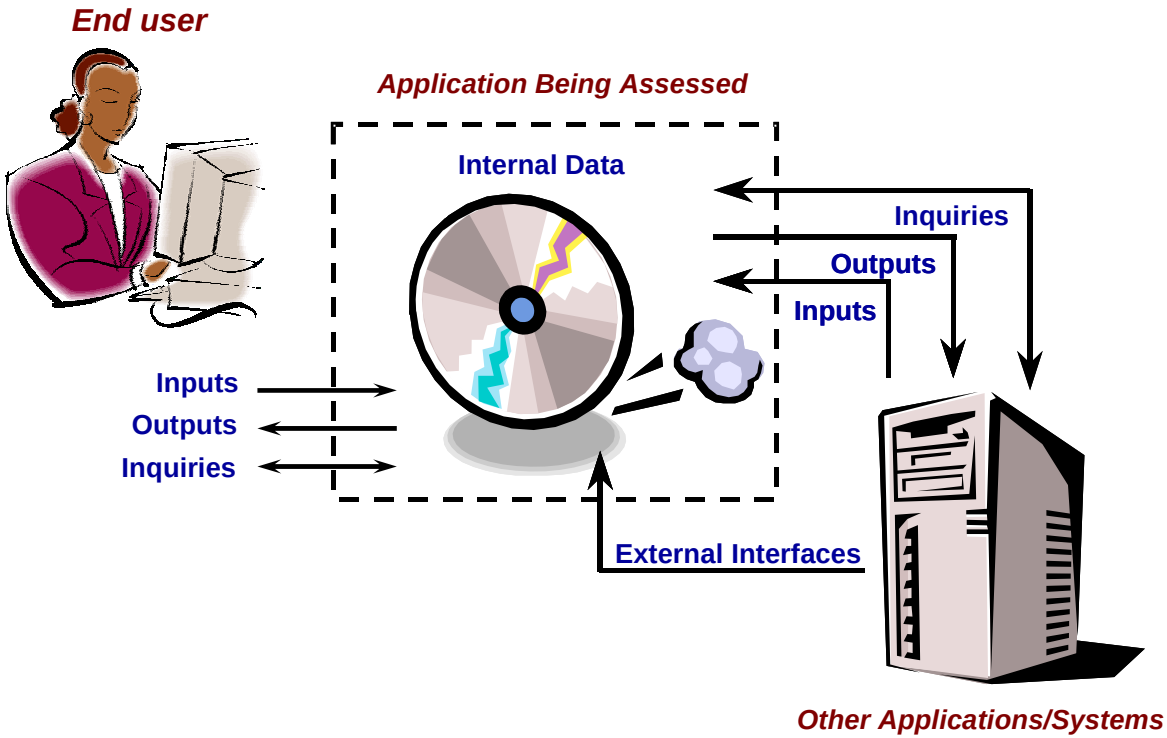
- Size is the most critical driver of cost and schedule on a software project.
  - All other things equal, the larger the size, the greater the effort and cost and the longer the schedule



Size Matters (cont'd)

Standardized Size Measure

- Having a consistent, well-defined, standard size measure as part of the estimation process is critical
- IFPUG function points are an example of this



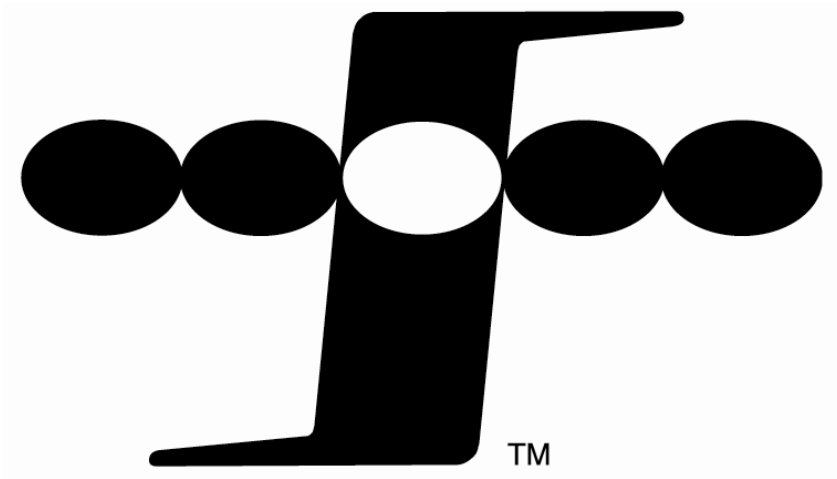
Function point counting resources

- ✓ *User/analyst interviews*
- ✓ *Requirements documents*
- ✓ *Design documents*
- ✓ *Data dictionaries*
- ✓ *Use cases*
- ✓ *User guides*
- ✓ *Screen captures*
- ✓ *Actual software*
- ✓ *Entity-relationship models*
- ✓ *Semantic object models*

Function point analysis (FPA) provides a consistent, documentable, repeatable size measure

Standardized Size Measure

- Standards are established and managed by International Function Point Users Group (IFPUG)
- Function points accepted as a standard size measure by ISO (ISO 20926:2003)
- Certified Function Point Specialist (CPFS) professional certification program recognizes trained experts
- Because it is linked directly to system requirements and functionality, FPA puts size analysis into terms that a client or end user can understand
  - Function points can help with communications between the end user community and the developer
    - A client would never say, “I need a system that is 20,000 lines of code”
    - A client says, “Build me a system that does…and supports these processes”



Automated COTS estimation tools leverage large historical data sets and flexible input parameters

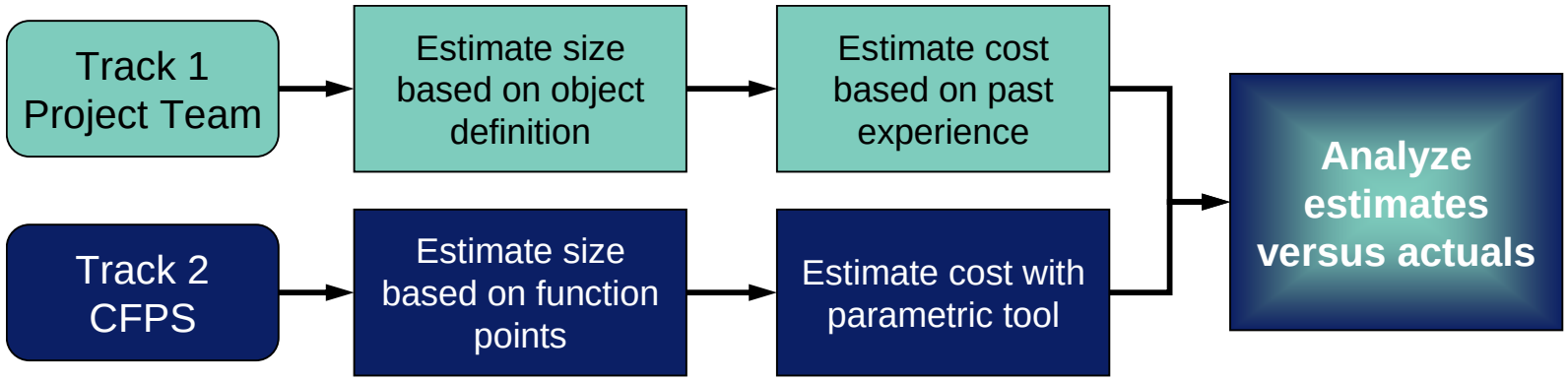
Structured Estimation Tool

Examples

- Tools are parametric in nature, meaning calculations are based on complex statistical algorithms
- Outputs from model are based on input assumptions
  - Size
  - Personnel skills and experience
  - Development environment
  - Productivity factors
  - Labor rates
- Estimates can be generated with as much or as little information as is available
- Tools typically estimate all development life cycle activities, including various levels of testing
- Tools can handle the complex factors that can impact cost and schedule on a software project

Examples

Recent pilot study in Booz Allen to evaluate function points versus objects as a size measure



- Average FP estimation variance of 2.83% demonstrates highly reliable estimation methodology
- Correlation between FP size and actual cost: **0.9977**

Release	Function Points	Estimated Cost	Actual Cost	Variance
CAERS 2.07	652	$361,537	$359,500	0.56%
CAERS 3.0 It 1	241	$175,534	$162,377	7.50%
CAERS 3.0 It 2	302	$205,703	$204,800	0.43%

  - This strong correlation indicates a strong predictive relationship between FPs and actual cost
  - It also demonstrates the consistency of the FP sizing methodology over multiple releases

Project: SPAWAR PMW-151 eNTCSS Baseline Analysis

Issues

- SSCN estimated it would cost $12M to migrate RADM, RSupply, and OMMS-NG to web-based platform
- It eventually cost around $32M
- PMW-151 wanted an audit of the software development effort to identify points of failure and potential areas of improvement

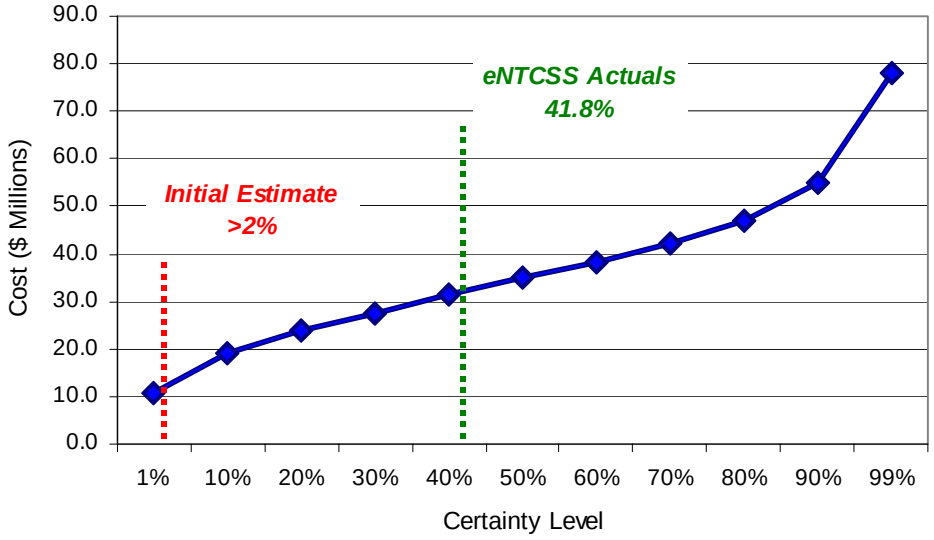
Methodology

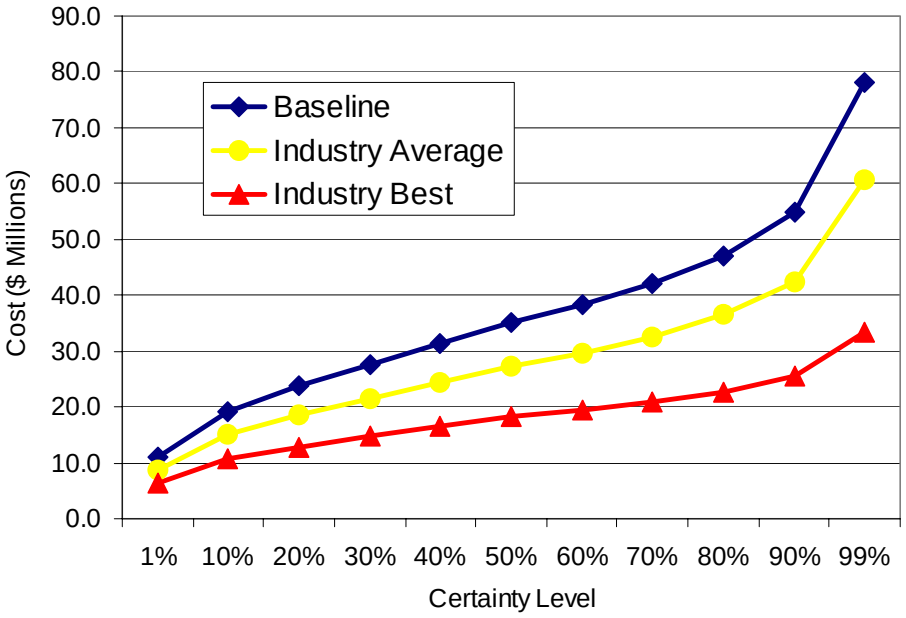
- Conducted function point analysis of 3 web-based NTCSS applications
- Modeled size in SEER-SEM, tailored model to system requirements and development environment
- Compared estimates to actuals, conducted cause and sensitivity analyses

Conclusions

- eNTCSS applications are very large
- SSCN delivered eNTCSS within reasonable cost and schedule
- The initial $12 million eNTCSS cost estimate was extremely optimistic
- With improved process stability and maturity SSCN could have developed eNTCSS for $17 to $25 million

Project: SPAWAR PMW-151 eNTCSS Baseline Analysis





Project: TRAC^{2}ES Measurement and Estimation Support

Issues

- Client frequently changes requirements
- Project team and client wants more insight into productivity metrics
- Project team wants to be able to demonstrate why cost/schedule estimates may change throughout development life cycle

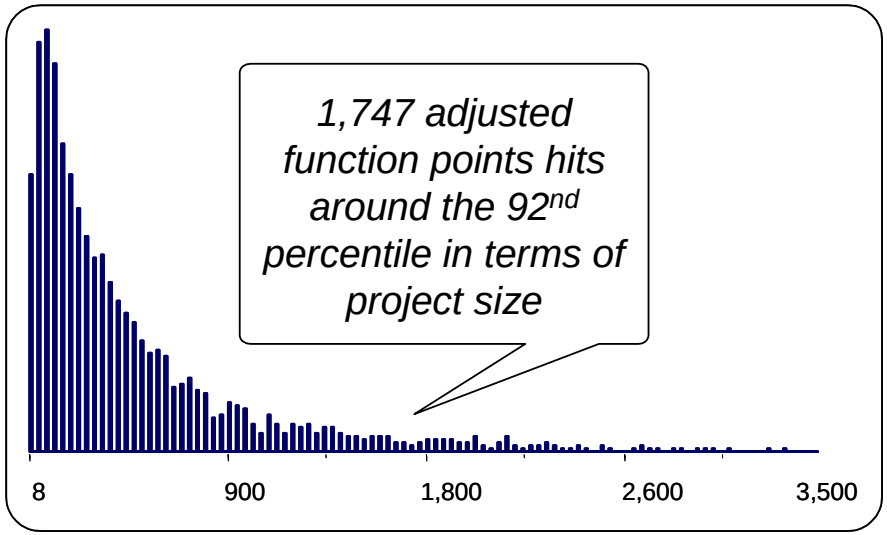
Methodology

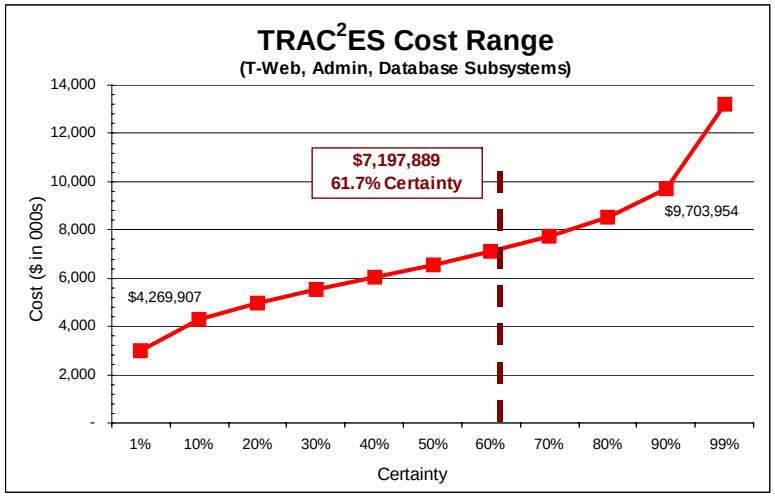
- Performed baseline FPA on entire system
- Perform enhancement/updated FPA on new requirements and requested changes to system
- Generate SEER-SEM range estimates and compare them to project engineering estimates

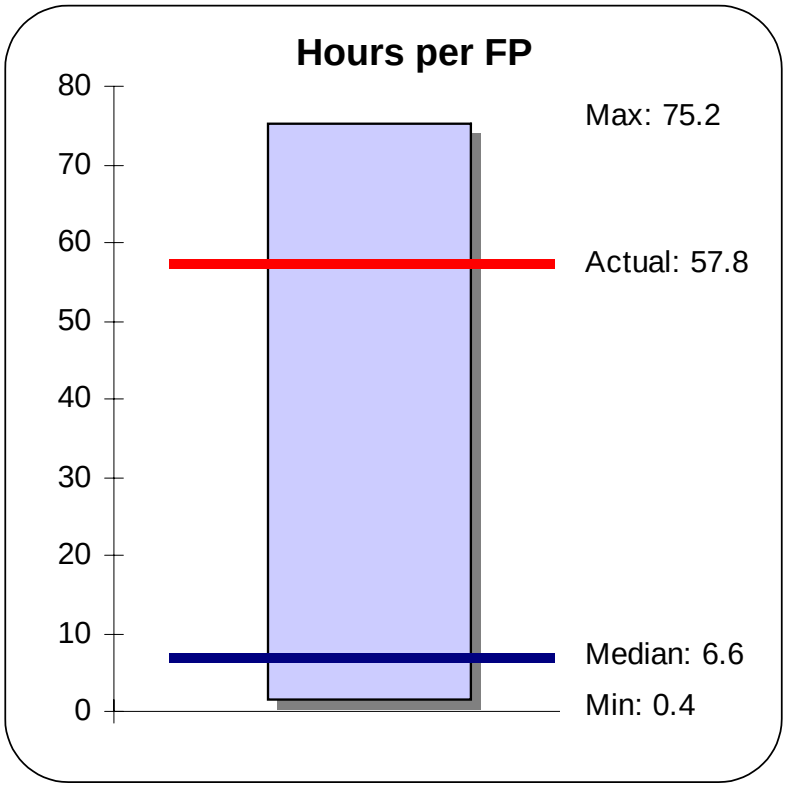
Results

- PM has increased confidence in cost and schedule estimates given to client
- Client has increased confidence in Booz Allen's estimates
- PM has been able to demonstrate impact of significant changes to requirements, resulting in more controlled scope

Project: TRAC²ES Measurement and Estimation Support







Based on analysis of projects 1,000 FP and larger

Contact Information

Ian Brown,
Certified Function Point Specialist
Senior Associate
McLean, VA
(703) 902-4971
brown_ian@bah.com