



TRUE PROGRAM SUCCESS '06

PRICE Systems U.S. Symposium
April 19-21
Sheraton San Diego Hotel & Marina
San Diego, CA

**To Level V and Beyond:
Keys to Successful Implementation
of a Comprehensive Parametric-
Based Software Development
Estimating System**

JEFF MURPHEY
Executive Consultant
PRICE Systems, LLC

PRICE

Agenda

- > Goals and Objectives
- > “Best of Breed” Ingredients
- > Elements of an Effective System
- > Steps in Building a Parametric-Based Estimating System
- > Conclusion – Considerations and Suggestions

Mission Statement

- > “Develop an improved estimating process that enables organization to achieve a higher CMM-I certification level”
- > Automate the process as much as possible
- > Employ a commercial estimating tool that is comprehensive, yet flexible
- > Create a “closed loop” system

**To Level 5
and Beyond!**



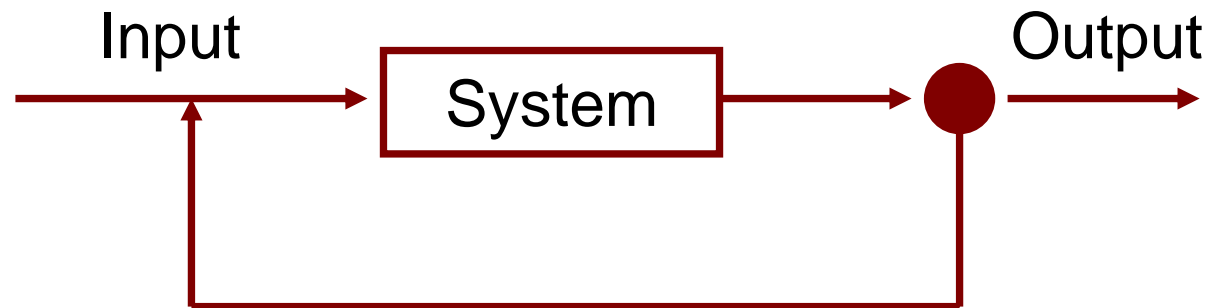
Objective

- > Close the gap between estimates and actuals
 - Work both sides of the accuracy equation
- > Actuals: implement “feedback control system” relationship between estimates and actuals
 - Manage projects to estimates throughout project life
 - Move software engineering from “art” to “science”
- > Estimates: improve estimates over time
 - Better historical basis should yield better projections

Open-Loop Control System



Closed-Loop Control System



Output of system modifies
inputs to reduce variability

“Best of Breed” Ingredients

- > Estimating system that can support both prime estimating and design trades
- > A direct automated link to “design” and/or pricing tools and environment
- > A defined standard input/output that can be applied to any estimating scenario
- > A commercial cost tool (which can be calibrated and validated) that covers all program activities and resources
- > A well documented process

Major System Elements

- > Key personnel buy-in and involvement
- > Process control matrix
- > Seamless program data collection
 - Parametric database
- > Electronic tool set
 - Parametric model (True S) as estimating “engine”
 - Adaptation of model to database & organization
- > Calibration process
- > Self-creating product

System Elements

Key Player Involvement

- > Process change can be unsettling to many
- > Process change requires acceptance from everyone (communicate benefits)
- > Who are the key players?
 - Software engineering community
 - Mid- and upper-level management
 - Cost analysts/estimators, parametricians
 - Customer
 - DCAA

System Elements

Process Control Matrix

- > Proper planning is crucial
 - Don't rush
 - Take “baby steps” (start with side-by-side estimating)
 - “Rome wasn't built in a day”
- > Process must be documented, properly communicated and maintain configuration control
 - Who does what? When? How?
- > Personnel must be adequately “trained”
- > Process is not static; should be refined periodically



System Elements

Seamless Program Data Collection

- > Automate as much as possible
 - WBS
 - Software Size
- > Standardize format
- > Data dictionary (definitions)
- > Link to estimating tool

System Elements

Electronic Tool Set

- > Parametric model (e.g., True S) as estimating “engine”
- > Data Library
 - Guidelines, policies, and procedures
 - Calibration database
 - Technical notes
 - Technical papers
- > Adaptation of model to database and organization (“mapping”)
- > Link to traditional estimating system

System Elements

Calibration Process

- > Determine what data (and format) to collect
- > Explicitly define data input parameters (data dictionary)
- > Prepare “footprint” map
- > Create standard calibration library format and reference system
- > Collect, analyze, and normalize data
- > Review results with prime customer(s), DCAA, and/or accounting department
- > Define application & implementation standards (Excel, Access, other)
- > Determine update frequency

System Elements

Self-Creating Project

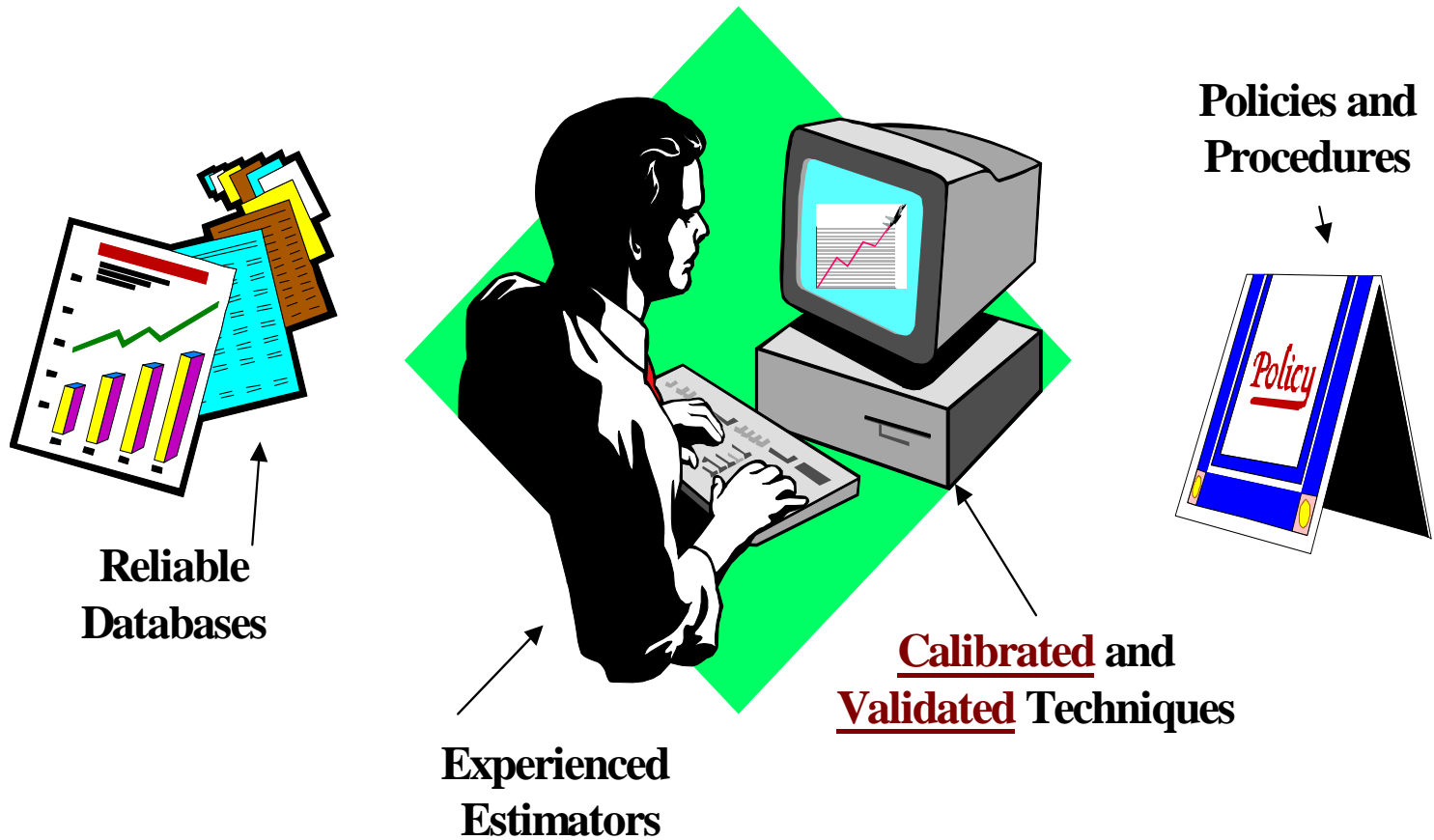
- > Standard cost/price output convention
- > WBS-oriented model input & output
- > Quality determination (review process)
- > Link to Proposal Pricing System (PPS)
- > Basis of Estimate (BOE) development

Parametric-Based Estimating System

Parametric Estimating System – Overview

- > A collection of databases, techniques, processes, and tools
- > Not a single commercial model
 - Parametric cost model is only a PART of an ***entire estimating system***
 - Estimating and non-estimating (design, engineering, business management, risk, etc.) tools
 - Full potential of a parametric cost model is realized only with proper training and effort

Basic Elements of a Parametric Estimating System



Parametric Cost Model

- > A cost model must **consistently** produce **reliable** and **credible** cost estimates
- > Effectiveness of a cost model (whether in-house or commercially available) hinges on two processes – calibration and validation

Calibration: “the process of indexing a parametric model or system to local cost and product history. The calibration process turns a commercially developed or public domain model into the equivalent of one developed locally.”



Extracted from Parametric Estimating Initiative (PEI) newsletter, dated August 1996.



Validation: “the process of demonstrating the calibrated model’s ability to function as a credible forward estimating tool or system.”

Steps in Building a Parametric-Based Estimating System

1. Conduct an internal evaluation
2. Survey data sets available for use
3. Extract pertinent information from surveyed sets
4. Construct parametric database using extracted information
5. Calibrate parametric tool(s) to parametric database
6. Implement, use, monitor, and maintain (update) system

Step #1 – Internal Evaluation

- > Establish goals and objectives
- > Cost/Benefit/Effectiveness analysis
- > Identify strengths and weaknesses of approach/system
- > Develop a project plan
 - Establish roles and responsibilities
 - Produce schedule for completing steps 2-6

Step #2 – Survey Data Sets

- > Schedule information
- > Financial reports
- > Project status reports
- > Engineering data
- > Project experience databases (metrics)

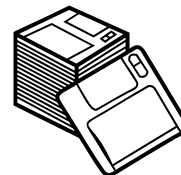
Step # 3 – Extract Pertinent Information

- > Identify analogous program and collect historical data
 - How to determine relevancy?
 - Time (how recent is the project/program?)
 - Similarity in...
 - Technology
 - Product line
 - Operating environment
 - Application
 - Customer



Step # 3 – Extract Pertinent Information (con't)

- > Establish scope of data collection (evaluate data requirements)
 - Cost data
 - Product data (e.g., size, complexity, quality)
 - Programmatic (e.g., schedule)
- > Identify potential sources of data and POC
- > Establish data selection criteria
 - For example: all completed programs between certain years
 - Data Collection Checklist
- > Build database



Database Issues

- > Maintaining the data quality
- > Maintaining the data quantity
- > Augmenting the data
- > Manual sorting of data
- > Understanding the data
 - Normalization?

Step #4 – Build Parametric Database

- > Define fields/structures for access and sorting
- > Make it easy to maintain
- > Optimize – only include as much information as needed for project differentiation

Software Size and Type (new, adapted, reused, deleted, auto-generated)
Code complexity
Heritage (amount adapted, design repeat)
Specification level
Integration requirements
Team Experience Level; Software Tools

Parametric Database Caveats

- > No single standard or format
- > Most likely contains more information than needed for existing projects, but not enough for future projects
- > Parametric databases often constructed with pieces from other databases

Software Metrics Collected

- > Size
- > Effort
 - Labor hours
 - Dollars
 - Staff size
- > Productivity
- > Requirements stability
- > Schedule
 - Start, finish, and milestones
- > Environment
- > Quality
 - Number of Defects
 - Average time spent to correct a defect
 - Amount of rework
- > Earned value
 - Cost performance index

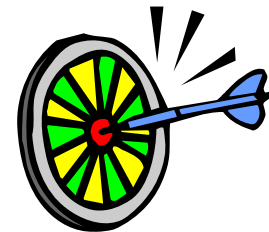
Step #5 – Calibrate Tool to Database

- > Parametric tool should incorporate the results of your calibration
 - Relationship must be symbiotic
- > Don't let tool become stagnant
 - Update whenever you perform new calibration

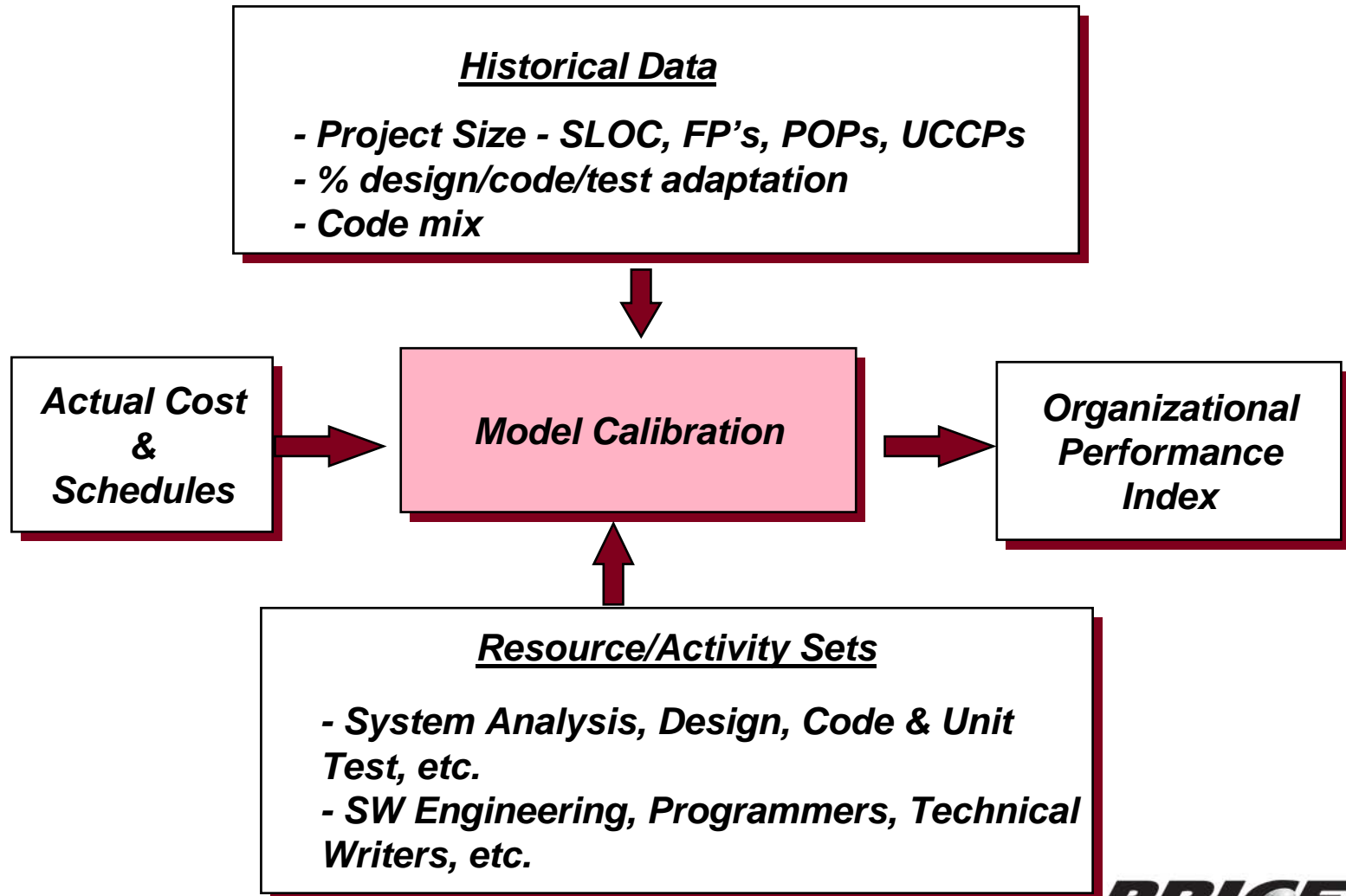


Calibration

- > What is calibration?
 - Fine tuning the parametric cost model to specific company experience
- > Calibration is **critical** if a parametric model is to be used for bid & proposals
 - Cost proposals require a much greater degree of accuracy that can only be achieved through calibration



Calibrating True S



Calibrating True S

- > What input parameters should I calibrate to?
 - Organizational productivity index (reflects how efficiently the organization develops software)

- > How many data points (projects) are necessary?

- > What are the correct data points?

True S Calibration – Suggestions

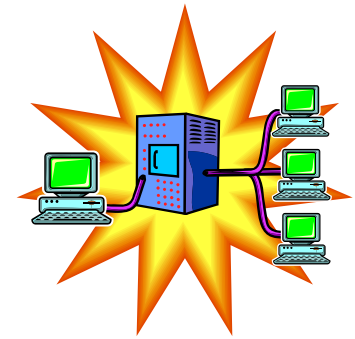
- > Define what is included in actual “cost”
 - Are any activities excluded?
 - Does the cost entered capture all resources or some subset?
- > Conduct calibration for “product” groupings
 - Application, language, customer, etc.
- > Enter as much schedule information as possible (dates)
- > Relevancy: add new projects; delete older programs

True S Calibration – Suggestions (con't)

- > Refine True S definitions
 - Adapt to corporate usage
- > Collect data and calibrate tool
- > Dedicated cost analyst/estimator, software engineer, or parametrician that understands...
 - The organization's software process
 - The organization's methodologies

Calibration Database

- > Define fields/structures for access and sorting
- > Make it easy to maintain
- > Develop a plan to preserve configuration control
 - Who will have access? Who will enter initial data?
- > Expect and plan for growth
- > Optimize – only use as much information as needed for project differentiation



Step #6 – Implement, Use, Monitor and Maintain System

- > Footprint mapping
 - Template representing organization's performance, structure, accounting & business practices
 - Consistent and rational mapping is more important than air-tight precision mapping

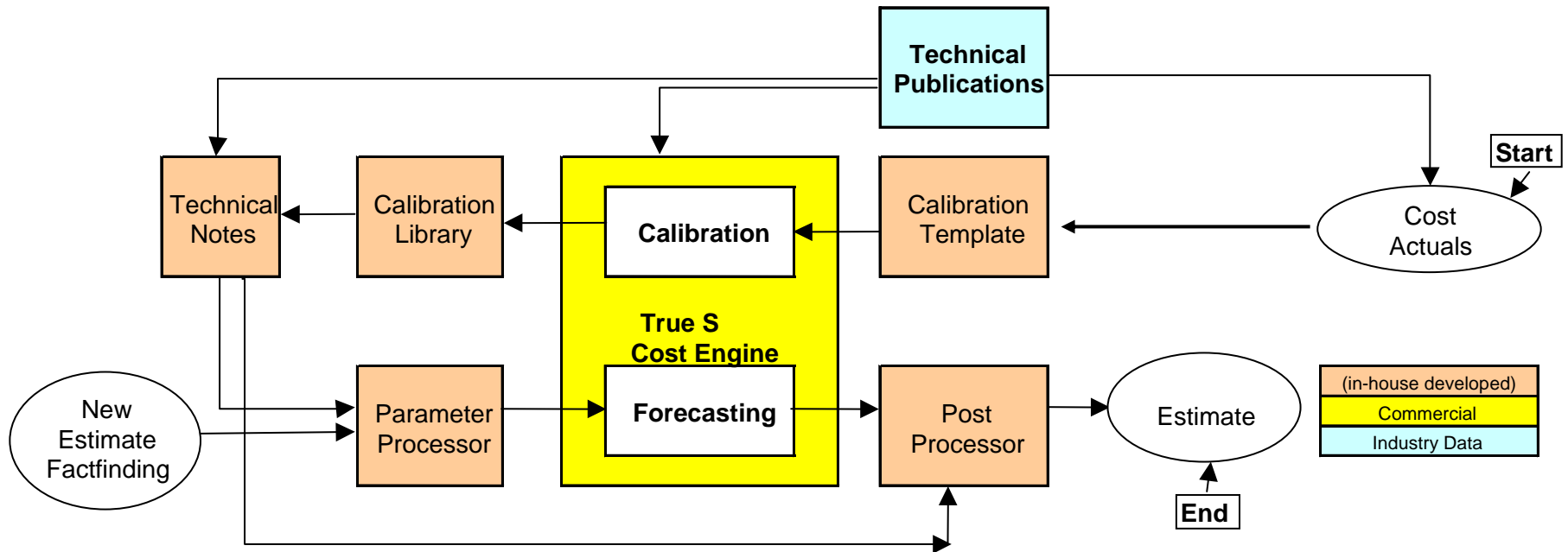
- > Link to traditional estimating system
 - Export parametrically-estimated costs to external tool (PPS)

Step #6 – Implement, Use, Monitor and Maintain System

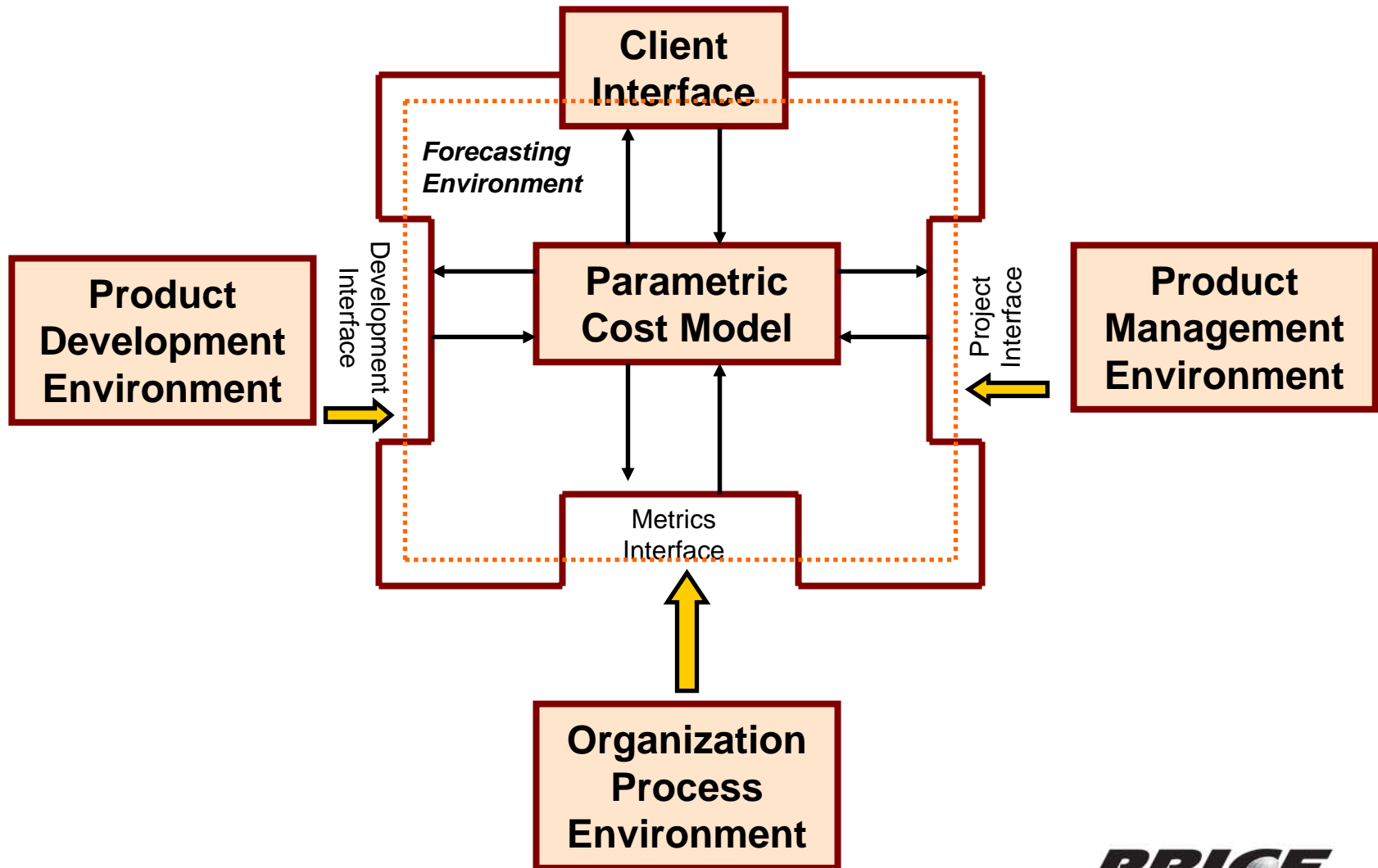
- > Parametric tool(s)
 - Evolves with database
 - Train and monitor users

- > Implementation/Use
 - Ensure database and model are “synchronized”

Example of a Parametric Cost Estimating System



The Future – Collaborative Estimating



Conclusion – Considerations and Suggestions

- > Define goal – ability to generate quality estimates within required time period
- > Without justification, model can be made to support any number
- > What should the organizations goal be – within 20%, 10%, 5% of actuals?
- > Model should be able to predict schedule and perform risk analysis
- > Requires very good understanding of parameter settings and usage

Conclusion – Considerations and Suggestions (con't)

- > Guidelines for model structure
- > Guidelines for parameter settings
- > Adequate training for model user