1°International Conference on
IT Data collection, Analysis and Benchmarking
Rio de Janeiro (Brazil) - October 3, 2013

# Lessons Learned from the ISBSG Data Base

**Arlene F. Minkiewicz**
**Chief Scientist**
**PRICE Systems, LLC**

# Goals of the Presentation

- Present the evolution of a methodology for utilization of the ISBSG data in support of software estimation
- Provide a technique for performing data driven estimation using ISBSG data for calibration of a software estimating model
- Share lessons learned from analysis of ISBSG data

 | 

# Agenda

- About ISBSG
- Introduction and Motivation
- Methodology Overview
- Typical Software Estimation Model Cost Drivers
- Methodology for creating template scenarios
  - Preliminary Filters
  - Iterative Filters
- Calibration Process
- Building Scenario Templates
- Documentation
- Conclusions

# About ISBSG

- International Software Benchmark Standards Group (ISBSG)
  - Formed in 1997
  - Mission
    - *"To improve the management of IT resources by both business and government, through the provision and exploitation of public repositories of software engineering knowledge that are standardized, verified, recent and representative of current technologies"* ***www.isbsg.org***
  - Two repositories of historical data
    - *Software Development and Enhancement Projects (over 6000 data points)*
    - *Software Maintenance and Support Projects (over 500 projects)*
  - Regarded as one, if not THE, largest independent source of data and analysis for IT industry
  - Partners represent Australia, China, Finland, Germany, India, Japan, Netherlands, Spain, Switzerland and the US

 | ***Decades of Cost Management Excellence*** 

# Introduction

- PRICE Systems has recently partnered with ISBSG with licenses to both the data repositories
- Performing analysis with this data to determine ways we can use this data to help software estimators do their jobs better
- Performed a study focused on delivering calibrated software estimation templates for TruePlanning for Software® based on specific software scenarios
- This paper presents the methodology created for building these templates
- While the work done was focused on a specific estimation tool, the methodology for analysis can be tailored for and applied to any estimation tool, whether it be commercial or home grown.

# Methodology Overview

- Apply filters to eliminate suspect quality or inadequate information
- Identify scenarios through iterative application of filters
  - E.g New Development for the Communications Industry developing an Information System using a Fourth Generation Language
- Translate data from ISBSG format into cost model inputs
- Perform calibration to identify values not directly provided with ISBSGs data.
- Use ISBSGs and derived values to develop cost estimating templates for each scenario
- Document the analysis and risks ranges for the cost estimating inputs provided

# Typical Software Estimation Model Cost Drivers

- Most all software estimation tools have one or more inputs quantifying the following project characteristics
  - Application Type
  - Operating Platform
  - Code Size including possibly….
    - *New Code*
    - *Modified Code*
    - *Reused Code*
    - *Deleted Code*
    - *AutoGenerated Code*
    - *Percent of changes made to design, code and test*
  - Team Experience
  - Language
  - Calibration Constant
    - *Relevant only to general purpose estimating models, home grown models generally have this built in by nature of the data they are based on*

# Methodology for creating template Scenarios

- Started with over 6000 data points, each with up to 120 different attributes or metrics.  Many data points did not contain enough data for this type of analysis
- Filter the entire data set for
  - Quality Rating – Only A's and B's selected
  - IFPUG or NESMA Counting approach
  - Functional Size > 0 and != blank
  - Resource Level = 1 (software development activities only)
  - Normalized Level 1 Work Effort > 0 and !=blank
- Left with a set of 2586 data points

# Methodology for creating template Scenarios

- Began iterative filtering to identify scenarios (Column indicators are relevant to the November 2011 release)
- Industry Sector e.g. Banking (Column E)
- Application Type e.g. Billing (Column L)
  - *Required interpretation and assumptions as this column was rather freef this release*
- Development Type – either New Development or Enhancement (Co..... AM)
- Language Type – either 3GL or 4GL (Column J)
  - *Would have liked to work at individual programming language but this did not provide robust enough data sets*

- Once these filters had been applied the following steps were taken
  - Review Normalized Level 1 Product Delivery Rate (PDR ) – this level includes on effort from actual software development activities
    - *Examine statistics and identify, question and potentially eliminate outliers*
- If at least 5 data points remain consider this adequate for creating a template

# Template Scenarios - Examples

| Development Type | Industry | Language Type | Application Type |
|---|---|---|---|
| New | | | |
| | Banking | | |
| | | Third Generation | |
| | | | Financial Transaction Processing |
| | | | Transaction Production System |
| | | Fourth Generation | |
| | | | Transaction Production System |
| | Communication | | |
| | | Fourth Generation | |
| | | | Information System |
| | Financial | | |
| | | Third Generation | |
| | | | Transaction Production System |
| Enhancement | | | |
| | Banking | | |
| | | Third Generation | |
| | | | Financial Transaction Processing |
| | | | Information System |
| | | | Sales Contact Management |
| | Government | | |
| | | Third Generation | |
| | | | Billing Software |
| | | | Electronic Data Interchange |
| | | | Management of licesnes and permits |
| | | | Transaction Production System |
| | | | Workflow Support and Management |
| | | Fourth Generation | |
| | | | Personal Productivity Software |

 | *Decades of Cost Management Excellence*

# Calibration Process – What is Calibration

- Based on historical data collected from a project
- Technical data is used to determine cost drivers for the model
  - Size information
  - Development Team information
  - Programming Language
  - Schedule
- Actual effort data (or cost) is used to determine cost drivers that are not directly aligned with the data collection
  - Application Type (although it is an column in the ISBSG data base – it needs to be quantified numerically)
  - Calibration Constant

# Calibration Process

- Selected data points are used to build software component models for the estimation tool by filling in relevant inputs. Process and assumptions follow:
  - Application Type – leave as default as this is calibration target
  - Operating Platform - select value indicating a commercial application
  - Functional Size
    - Size Units – select Function Points (or IFPUG Function Points)
    - New Code Size = 75% of Functional Size if New Development
    - Adapted Code Size = 75% of Functional Size if Enhancement
    - Reused Code Size = 25% of Function Size
    - Percent Design, Code and Test Adapted = 25%


  - Language – select January 1st of Start Year
  - Team Experience – A quantification based on analysis of experience of BA's and IT personnel (columns FF through FN). Select nominal value in the absence of values in these columns

# Calibration Process

- Size Assumptions
  - May seem somewhat arbitrary but it we've found it very unusual that new developments are all new – especially in markets and for applications where there's a lot of code already out there. While the Functions being added may be new, some of the implementation is being borrowed from elsewhere
    - While not strictly relevant in a benchmarking exercised – this information is pretty important for a estimation exercise.
  - Assumption partially validated by the fact that calibrations within Application Types were much more consistent than analysis conducted without this assumption
  - The risks associated with this should be understood and where possible there should be an effort to ascertain more information
    - Some of the risk may be mitigated by the fact that the templates follow the same assumptions as the analysis

# Calibration Process

- The actual process applied was slightly different than your typical calibration
- Focused on simultaneous calibration of two cost drivers
- Goal was not to find the best Calibration Constant or Application Type value for each data point but rather to find the best fit pair of these two for each scenario
- Average values were considered
  - The non linear behavior of the Calibration Constant in this effort made the results of using an average undesirable
- Automation was created that iterated through thousands of combinations and created statistics for the data set in the form of r-Square and Pred(50)
- This process was applied to each of the scenarios identified in Table 1.

# Building Scenario Templates

- Templates were developed for each scenario as follows:
  - Application Type = value determined by the calibration
  - Operating Platform = value indicating commercial software
  - Calibration Constant = value determined by the calibration
  - Size Units = IFPUG Function Points
  - New Code Size = 75% of average for Functional Size for the data set if Development Type is New
  - Adapted Code Size = 75% of average for Functional Size for the data set if Development Type is Enhancement
  - Percent Design, Code and Test Adapted = 25%
  - Language = most frequently occurring language in the data set
  - Team Complexity = nominal value for the tool

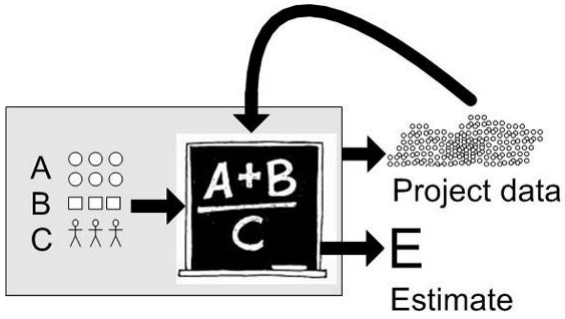
# Building Scenario Templates

- Within the cost model template, a risk range was established for each of the following inputs determined through this analysis
  - Calibration Constant
  - Application Type
  - New or Adapted Size


- The statistics from the calibration analysis were used to determine the distribution for the risk curve
  - R-Square >= 0.8 **and** Pred (50) > 70%, narrow curve as when a program is in the Detailed Design phase with much known about the project
  - R-Square >= 0.8 **or** Pred (50) > 70%, moderate distribution as might apply when a program is in the Preliminary Design Phase where much can still be expected to change
  - Other cases – wide distribution as might be applied in the Early Concept phase of a project

# Delivering and Using the Results

- For each scenario, a template was creating containing the cost model inputs from analysis representing the following project constraints
  - Size
  - Calibration Constant
  - Operating Platform
  - Application Type



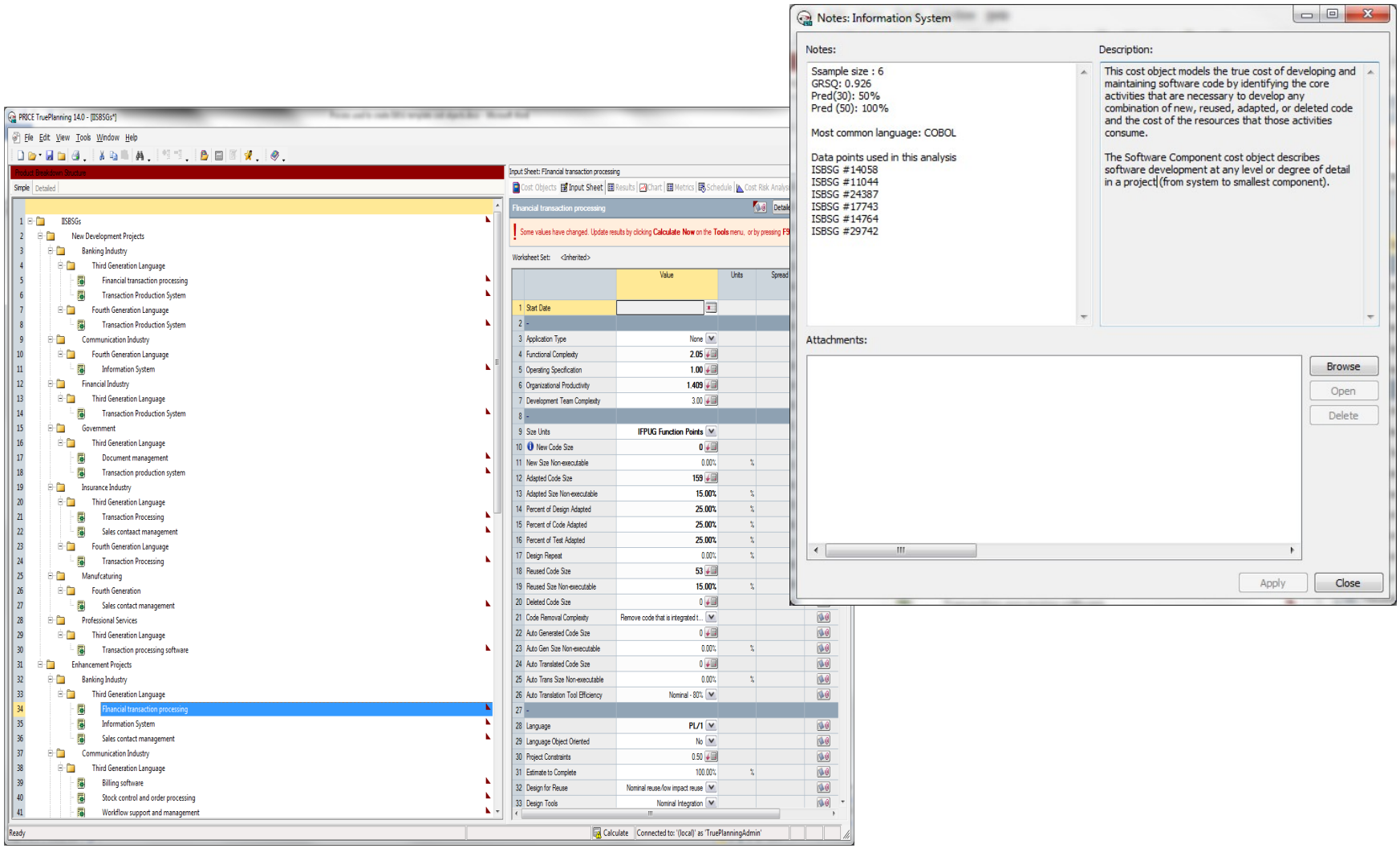

- Other cost drivers which weren't adequate targets for analysis from this particular study are left at nominal values as defined by the cost model
- The expectation is that software estimators use these templates as a starting point applying thoughtful analysis as to how they do or do not make sense within the context of the project currently being estimated

# Documenting Results

- Along with each scenario, documentation is delivered that gives the estimator an idea of the data behind the template to facilitate decisions as to whether it is relevant or not to their current circumstances.
- The following documentation is included
  - Sample Size
  - Stats associated with the analysis for that scenario
    - R-Square
    - Pred (30)
    - Pred (50)
  - Most frequently occurring programming language
  - List of the ISBSGS ID Numbers of the data items used (so subscribers can access all the information relative to the scenario

# Templates in TruePlanning for Software




 | ***Decades of Cost Management Excellence***

# Conclusions

- While there is a lot of variation found in the ISBSG Data, it certainly can be used as a basis to support better estimates, particularly in situations where no historical data is available.
- ISBSG Data can be used to develop data driven estimates for a wide range of application types across many industries
  - The analysis is not always clean
  - Assumptions need to be made around the data for several reasons
    - ISBSG data collection, while comprehensive, does not completely line up with the inputs to every estimation model
    - ISBSG data sets are often not complete
  - Admittedly assumptions make the analysis less 'accurate' however…..
  - An estimator aware of these assumptions and familiar with the analysis results can use such templates as a …
    - Support of a Rough Order of Magnitude (ROM) estimate
    - Starting place for an estimate where no historical data is available
    - Sanity check for values they select for input variables for their estimation model

# Steps Forward

- More work needs to be done to improve assumptions in this analysis
- Analysis would be improved if:
  - More submitters filled in the Added Count and Changed Count Columns (of the 6000+ only about 2000 have these fields)
  - The Application Type Field was less freeform with discrete choices (even if some of the discrete choices encompassed multiple application types – allowing for granularity where possible but not requiring it). This would at least allow for stratification of like application types.
- Analysis is going to be repeated with the following changes:
  - Use the latest version of the ISBSG data and review new attributes to see what more we can use.
  - Work from averages for Application type within Scenarios
  - Use automation to optimize Calibration Constant within the scenario

 | *Decades of Cost Management Excellence*


Arlene.minkieiwcz@pricesystems.com