estimate • analyze • plan • control

Issues in Software Measurement & Estimation ISMA 2010

Dan Galorath + 1 310 414-3222 x614

galorath@galorath.com

SEER by GALORATH

Abstract

- **STANDARDS**:
  - Measurement requires clear agreement on definitions & standards and on how terms are applied
  - Uniform values for units of measure are fundamental
- **SIZE**:
  - Function points **can** provide a uniform gauge of software functionality… lots of issue exist
- **EFFORT**:
  - Subject to uniqueness of individual companies and users
  - Allocation of costs incurred is equally subjective between and WITHIN most companies
- **DELIVERABLES**:
  - Deliverables & work products are variable. Inconsistent management record keeping make analysis and prediction of component values suspect.
  - Major decisions are made every day based on this sketchy information
- **CONSEQUENCES**:
  - Enormous consequence from inconsistent terms in software measurement
- What can be done to reduce variability and produce more reliable business information

Standards: Software Metrics Needed By All

- Software Development relatively young in comparison to other professions – accounting, law, medicine and even manufacturing
- Software measurement & standards maturing & slowly becoming part of the process
- Must continue to raise awareness from software practitioners to senior management
- Howard Rubin 2006: The average life of a software metrics program is 3 years

Decision Makers	Focus	Measurements
Project Manager	Project Control	Effort, schedule, defects
Middle Management	Department Project Performance	Productivity, cost, performance
Senior Management	IT Performance	Quality, savings, time to market, business contribution

Size: Sizing Pitfalls

Sizing Mistake	Consequence
Wrong sizing metric chosen for level of detail desired	Large variance in estimates
Not enough time/effort spent on software sizing in general	Unbelievable estimates – results don't match the program and are too optimistic or pessimistic
No clear definition of size	Inconsistent estimates – results don't pass the sanity check, unreliable output, blame the model
Size growth not considered OR size estimates reduced to achieve desired cost	Inaccurate estimates – results are too optimistic, programs will overrun cost / schedule estimates

Why should we care: Bad sizing yields bad estimates

Effort: Packaged Applications Are Costly To Organizations

- "Commercial application program or collection of programs developed to meet the needs of a variety of users, rather than custom designed for a specific organization"
- Many are enterprise applications
- Often allows / requires customization
- Examples: SAP; Rational PPM, SEER for Software; Microsoft Excel, CA Clarity, Oracle Business Suite

"One-third [of the budget] has to go to testing. Don't ever short change testing. Everyone always underestimates it, and says it's the last thing to worry about. Don't do that!"

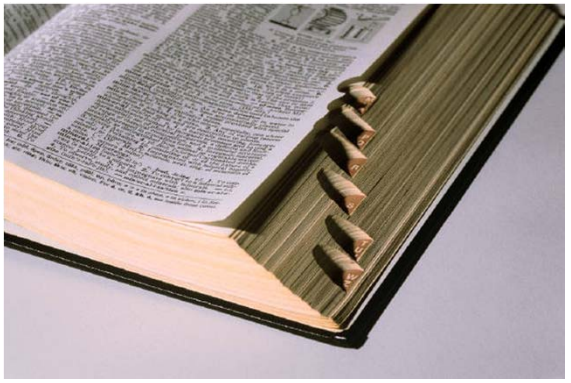
- Jim Larson, consultant for communications solutions provider

Why should we care: Packages sometimes comprise solutions for parts of complicated systems and can be trouble

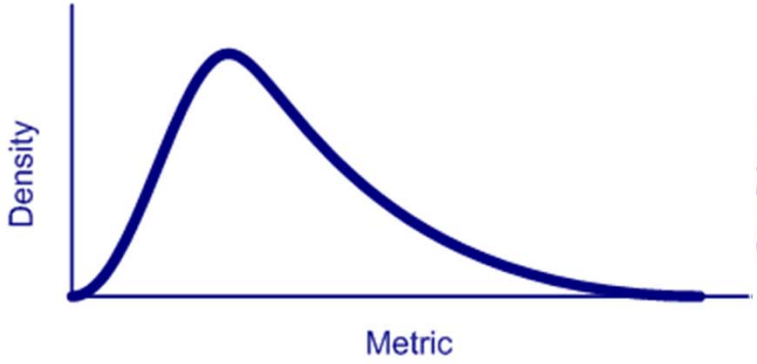
Deliverables: Estimation & Planning

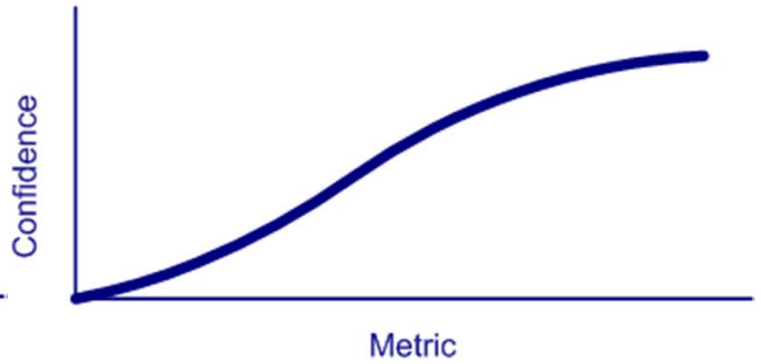
An Estimate Defined

- 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*





Consequences: Poor Estimates - Effect on Projects

- Inaccurate estimates can reduce project success:
  - Poor implementations
  - Critical processes don't scale
  - Emergency staffing
  - Cost overruns caused by underestimating project needs
  - Forever changing project goals
  - Frustration
  - Customer dissatisfaction
  - Cost overruns and missed schedules
  - Project Failures
- Important project business decisions made early with minimum knowledge & maximum uncertainty

Why should we care: Poor estimates and plans are a root cause of program failure

estimate

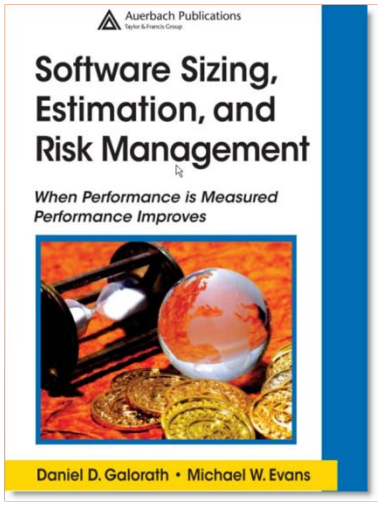
estimate • analyze • plan • control

What can be done to reduce variability and produce more reliable business information

SEER by GALORATH

10 Step Software Estimation Process

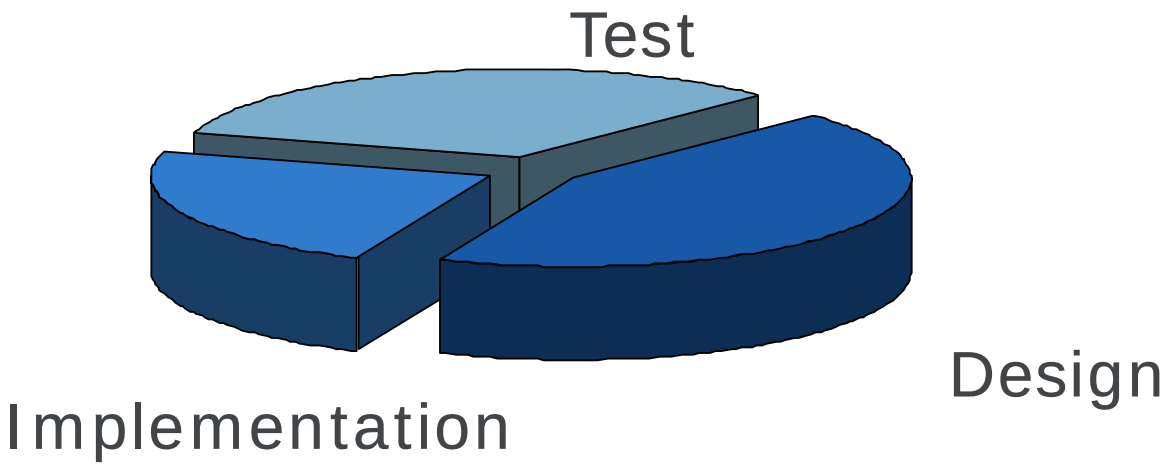
Consistent Processes = Reliable Estimates = Successful Projects



1. Establish Estimate Scope
2. Establish Technical Baseline, Ground Rules, Assumptions
3. Collect Data
4. Estimate and Validate Software Size
5. Prepare Baseline Estimates
6. Review, Verify and Validate Estimate
7. Quantify Risks and Risk Analysis
8. Generate a Project Plan
9. Document Estimate and Lessons Learned
10. Track Project Throughout Development

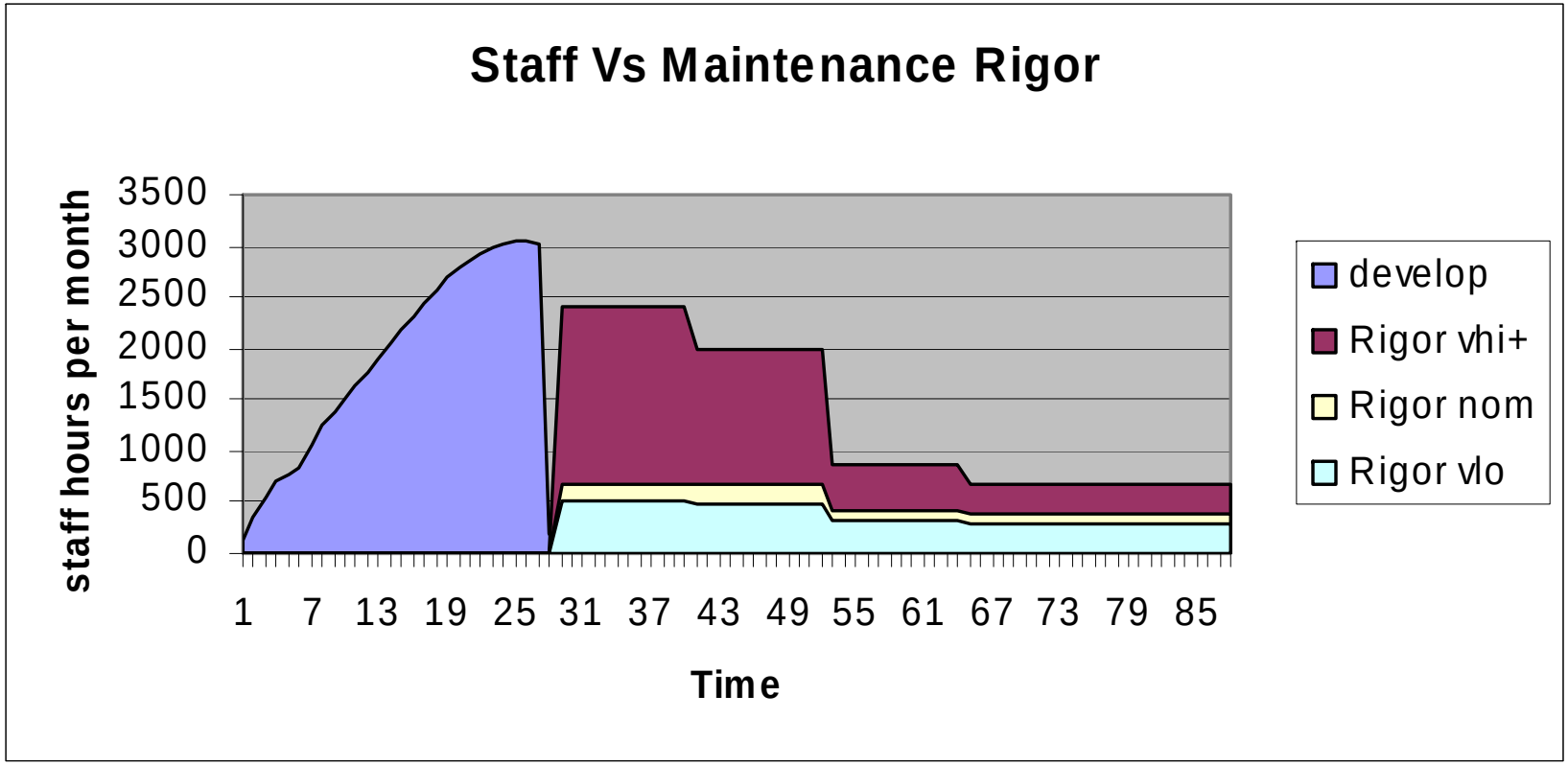
Reuse: Watch Out For Low Cost Assumptions on “Heritage”

- Reuse or Heritage: applying existing software to a new mission (or additional innovation in its current mission)
- Effort to reuse software is routinely under estimated



Why should we care: Heritage is often underestimated and causes major schedule / cost overruns

Measure Both Development And Total Ownership Costs



Why should we care: Maintenance is costly and usually not well considered in measurement, estimation & planning

Measuring & Controlling Rework Can Pay for a Measurement Program

- Rework: Doing the same work over again because it was incorrect the first time
  - Prototyping is not rework
  - Refactoring (tuning to make software better) is not rework
- Between 40% and 50% of the total effort on software projects is spent on rework.. Barry Boehm
- Initiatives to reduce rework can save significantly

Why should we care: Rework measurement and mitigation can help reduce wasted effort, freeing resources for work with an ROI to the business

When & Why To Collect Estimation & Measurement Data

When To Collect

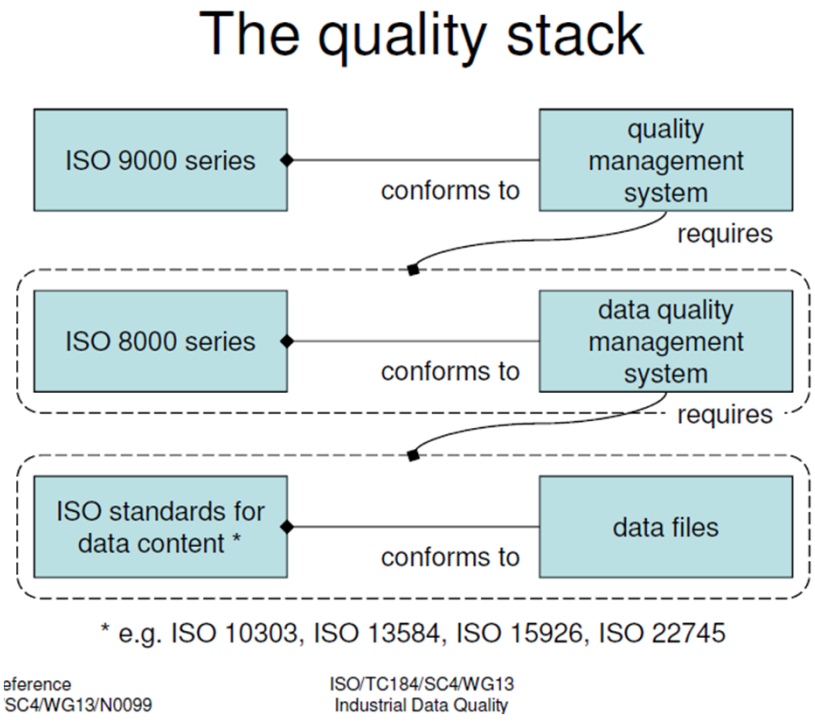
1. Up-front: when scoping new project data from completed projects
2. In-Process: During development for management, to identify issues and progress
3. Post Mortem: Upon development completion to improve corporate history repository
4. In Service: During maintenance to continue learning & improving

Why People Don't Want To Provide Data

- They could be proven wrong
- It could be used against them
- Data often doesn't exist
  - Even if processes dictate data requirements
- If it exists it may not be clean
- It may give away corporate productivity & bid strategy

Standards Can Help But Which Ones?

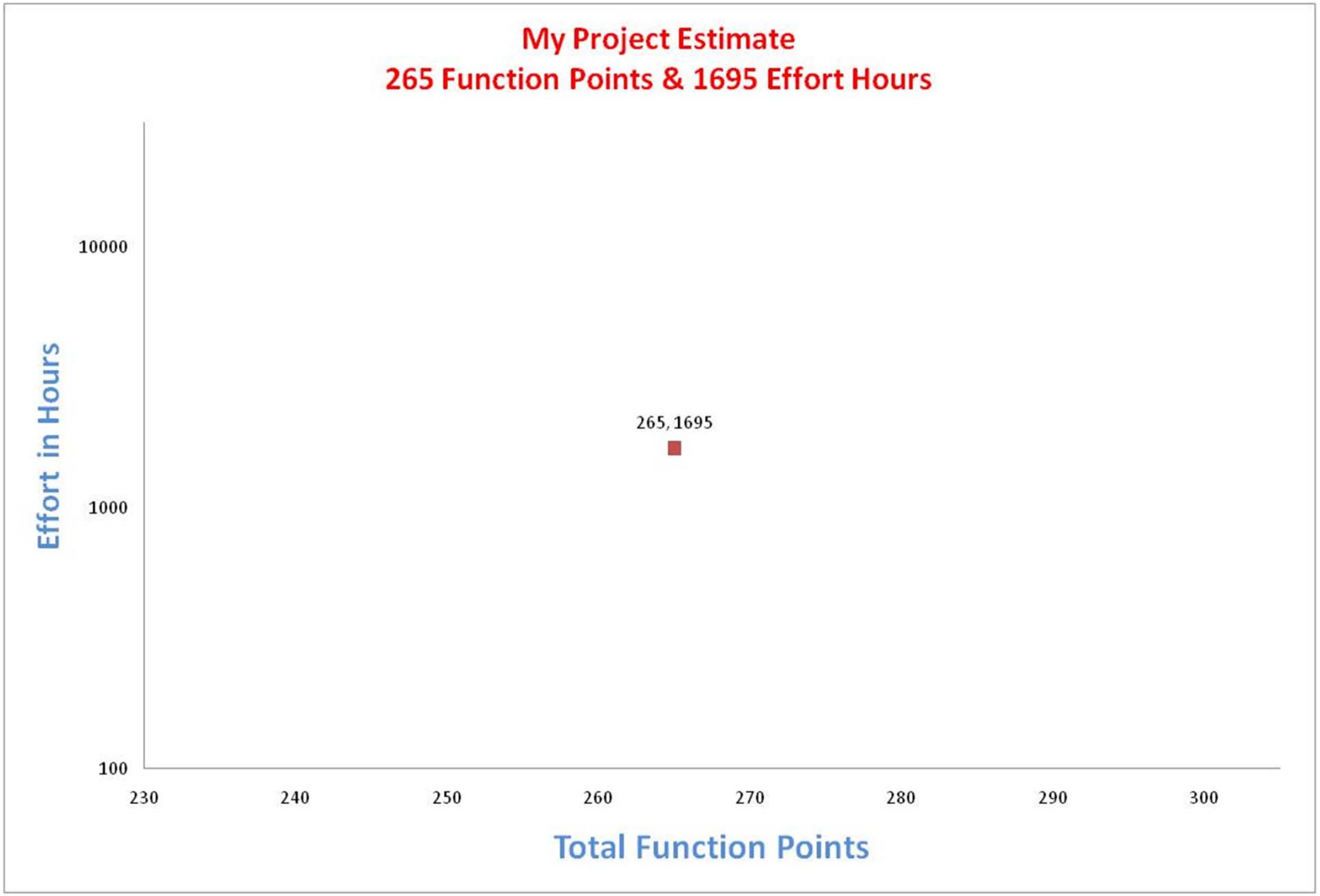
- IEEE Std 1045 Standard for Software Productivity Metrics provides
  - Framework for measuring & reporting software process productivity
- ISO/IEC 15939, *Software Engineering - Software Measurement Process* includes purpose & outcomes of compliant process & associated activities and tasks. It defines measurement information model and associated terminology
- Practical Software Measurement
- Home grown standard is ok except for benchmarking
- Others



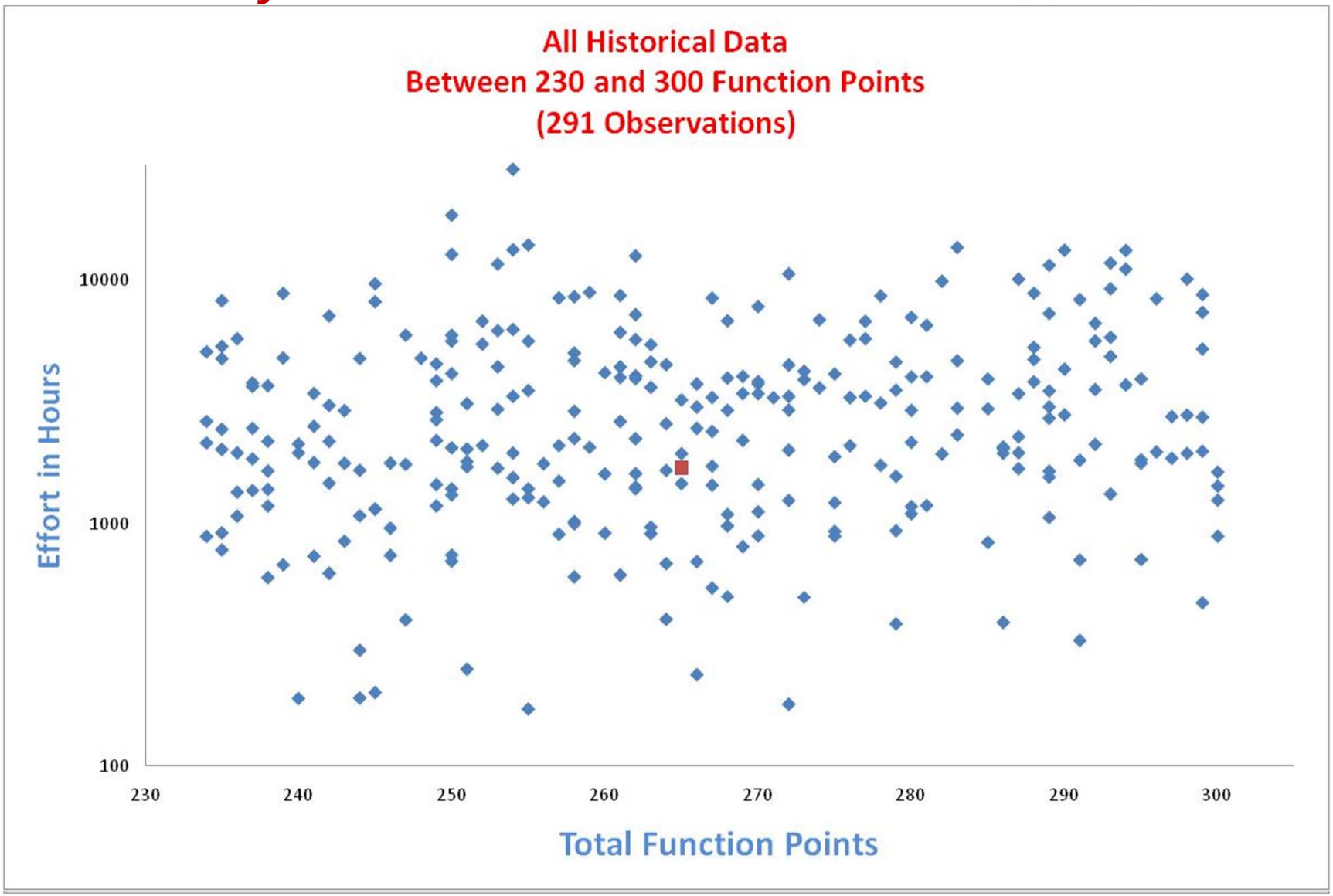
Data Collection Lessons Learned (Summarized)

1. Motivate potential data providers to participate
2. Avoid nondisclosure agreements containing clauses requiring exclusivity or destruction of data if you can
3. Provide data collection forms and instructions beforehand, in both hard copy and electronic formats
4. Provide clear definitions but recognize providers may not read them (allow providers to give definitions)
5. Identify which data are *required*, *highly desirable* or *desirable*
6. During the face-to-face interview confirm data is realistic and valid
7. Grade to indicate confidence
8. Normalize data via well-documented process & keep both the raw and normalized data

You have an estimate … Now what?



Use Historical Measurement to evaluate your estimate!



It's easy to dig deeper and deeper to justify an estimate!

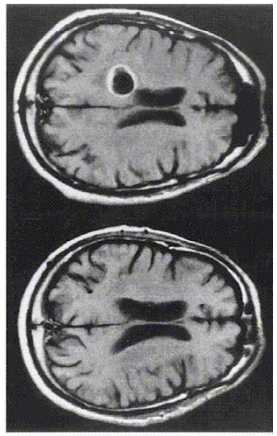
Parsing historical data looks logical
… but there are many potential/likely flaws

- The Error of Casual Analysis (False Association) - accumulating facts with minimal generalizations
- Narrative fallacy - when the set of connected and disconnected facts are picked to fit a story
- Fallacy of silent evidence - seeing only what has been recorded and remaining ignorant of the missing evidence
- Ludic Fallacy - assuming the data to be statistically analyzed is; complete, unaffected by small variations, and not intentionally corrupted

The Error of Causal Analysis

Creating a False Association

- Correlation does not imply causation
  - Just because two data points may sit side by side doesn't mean they are the same or will have the same outcome
- Casual analysis is a recognized error in medicine



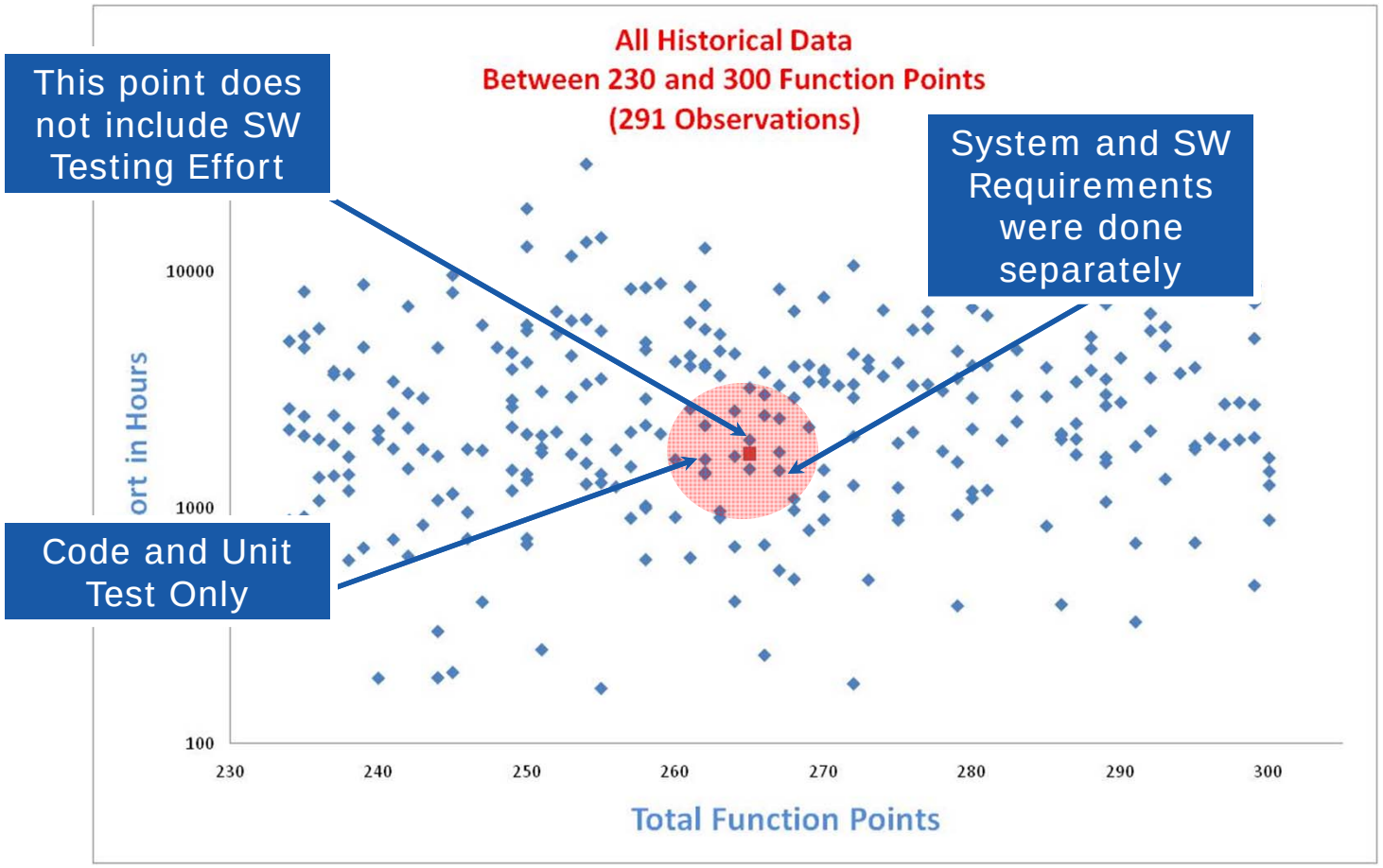
Tumor Can Cause Headache

Headache doesn't mean a tumor

Perhaps ???

The Error of Causal Analysis

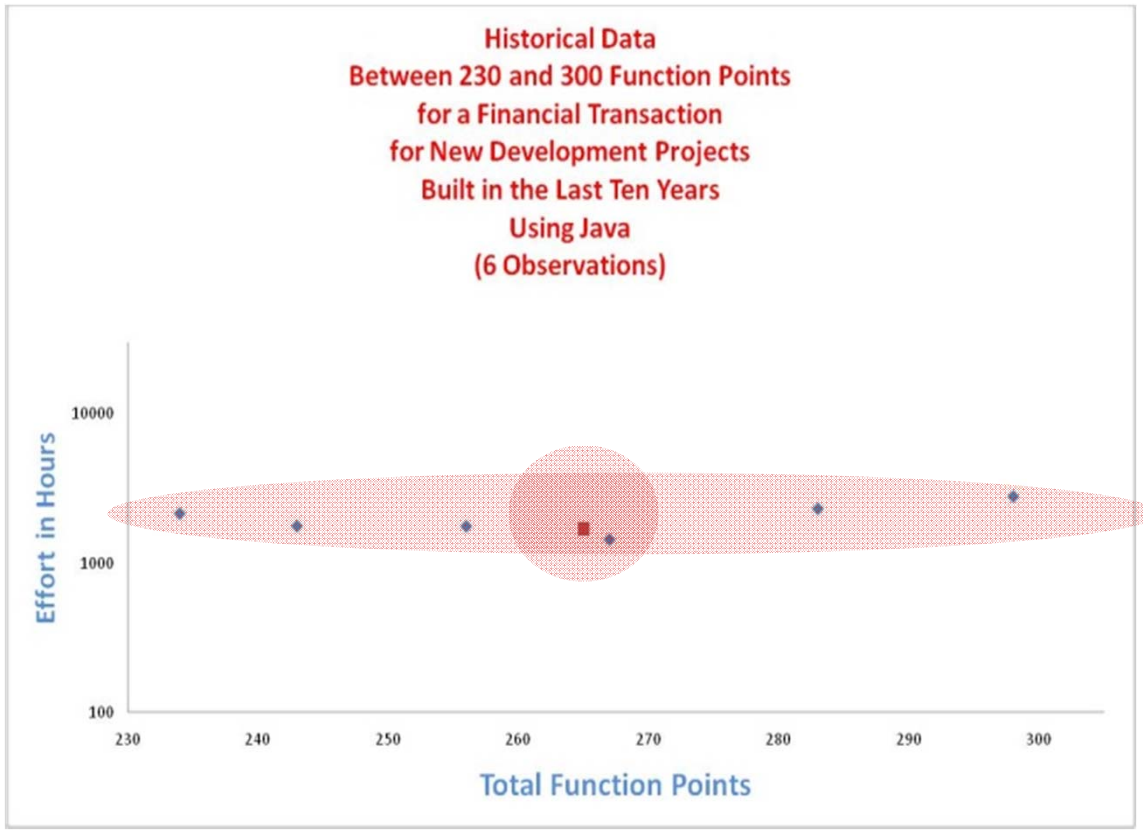
Is often encountered in data analysis!



There is no correlation that estimate of size ‘X’ is an equivalent of historical projects of about ‘X’.

Narrative fallacy

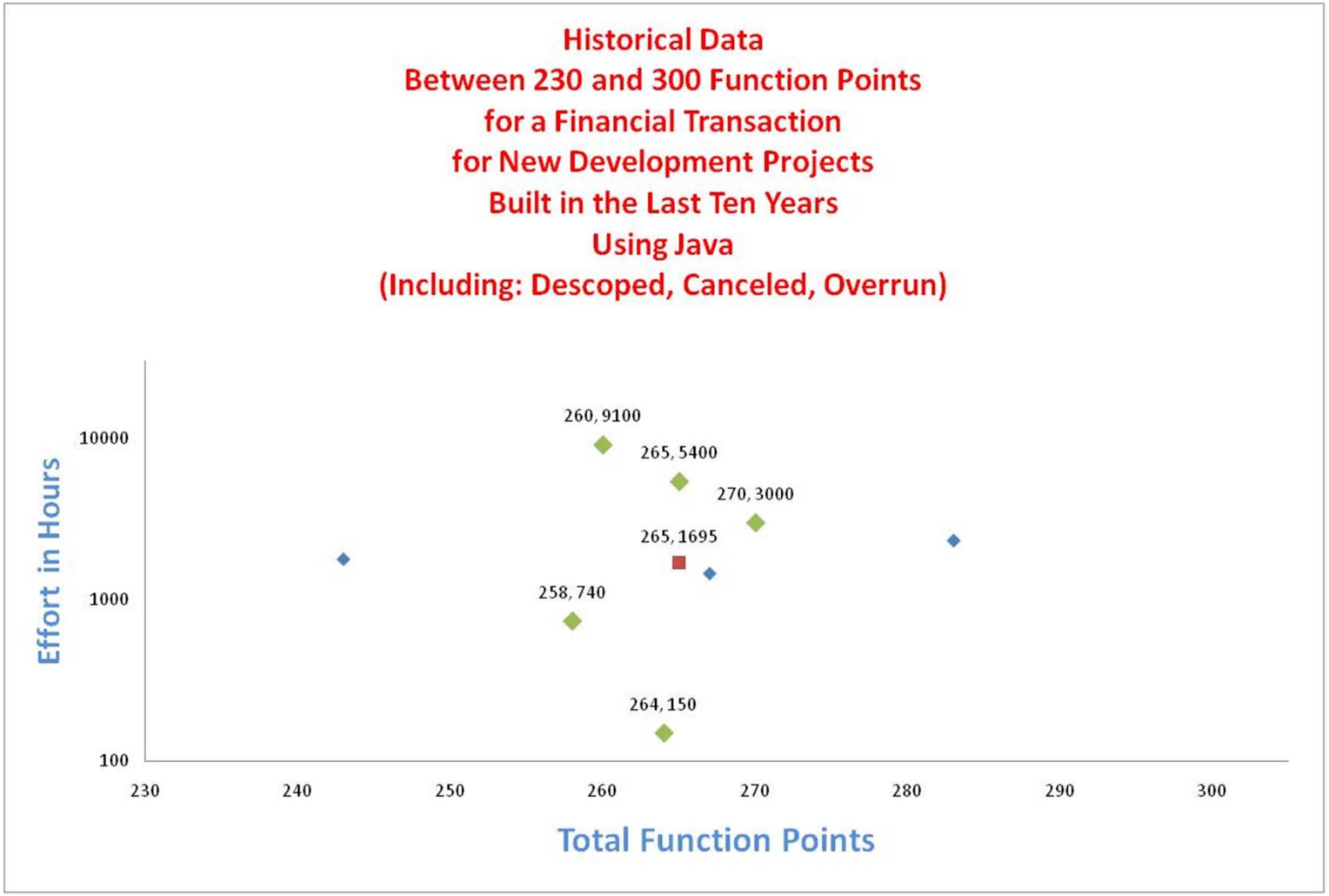
As our data becomes less cluttered…



It's almost automatic – and natural - to look for clumping and create relationships that likely have no historical correlation to one another.

Fallacy of Silent Evidence

What about what we don't know?



How confident would you feel if the Silent Evidence was visible?

Fallacy of Silent Evidence

Drives the conversation about data quality

- There are fundamental problems with historical data:
  - It's costly to obtain
  - Very difficult to catalog and store
  - Hard to manipulate, retrieve and review
  - Can often be wrong
  - Can be intentionally laced with inaccuracies
  - Incomplete

... Care to add to the list?

Note: Silent Evidence is different from Errors of Omission – Silent Evidence includes data that was intentionally omitted or not properly represented.

Ludic Fallacy

- "Ludic" is from the Latin *ludus*, meaning "play, game, sport, pastime"
- Are you now or has someone previously been playing games with historical data
- In this context, Ludic Fallacy is wrongly assuming historical data is complete, unaffected by small variations, and not intentionally corrupted

Ludic Fallacy

- Most common omissions from historical data ranked in order of significance

Sources of Cost Errors

1) Unpaid overtime by exempt staff
2) Charging time to the wrong project
3) User effort on software projects
4) Management effort on software projects
5) Specialist effort on software projects
    - Human factors specialists
    - Data base administration specialists
    - Integration specialists
    - Quality assurance specialists
    - Technical writing specialists
    - Education specialists
    - Hardware or engineering specialists
    - Marketing specialists
    - Metrics and function point specialists
6) Effort spent prior to cost tracking start up
7) Inclusion/exclusion of non-project tasks
    - Departmental meetings
    - Courses and education
    - Travel

Source: Capers Jones, *Errors And Omissions In Software Historical Data: Separating Fact From Fiction,* August 17, 2009

Ludic Fallacy

- Typical results reviewing customer historical data

Activities Performed	Completeness of historical data
01 Requirements	Missing or Incomplete
02 Prototyping	Missing or Incomplete
03 Architecture	Missing or Incomplete
04 Project planning	Missing or Incomplete
05 Initial analysis and design	Missing or Incomplete
06 Detail design	Incomplete
07 Design reviews	Missing or Incomplete
08 Coding	Complete
09 Reusable code acquisition	Missing or Incomplete
10 Purchased package acquisition	Missing or Incomplete
11 Code inspections	Missing or Incomplete
12 Independent verification and validation	Complete
13 Configuration management	Missing or Incomplete
14 Integration	Missing or Incomplete
15 User documentation	Missing or Incomplete
16 Unit testing	Incomplete
17 Function testing	Incomplete
18 Integration testing	Incomplete
19 System testing	Incomplete
20 Field testing	Missing or Incomplete
21 Acceptance testing	Missing or Incomplete
22 Independent testing	Complete
23 Quality assurance	Missing or Incomplete
24 Installation and training	Missing or Incomplete
25 Project management	Missing or Incomplete
26 Total project resources, costs	Incomplete

Source: Capers Jones, *Errors And Omissions In Software Historical Data: Separating Fact From Fiction,* August 17, 2009

Normalization
Adjustments for homogeneity

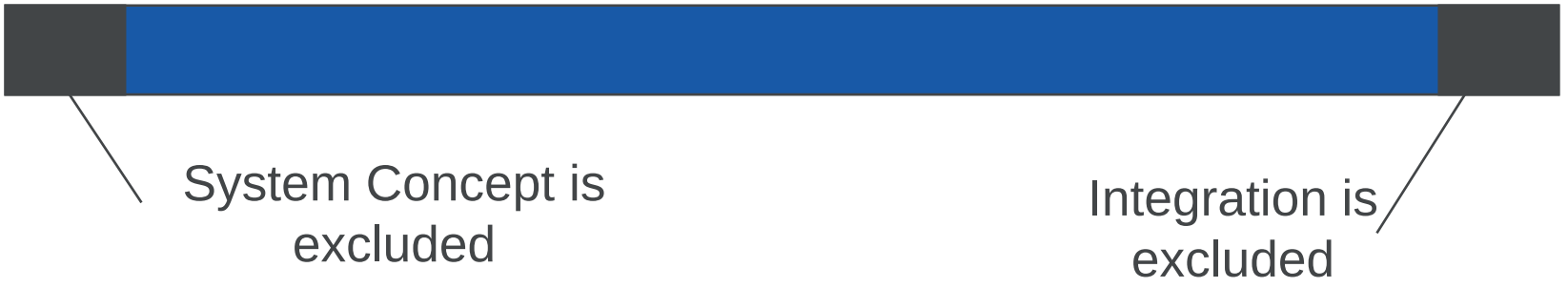
- Data comes in all “sizes and shapes”
- Development of CERs and calibration of models require that format and scale of data used be internally consistent
  - Inches versus centimeters
  - Full dollars versus “K” dollars versus Euros
  - Labor phases and activities
  - Handling of overhead
  - … and the list goes on

Normalization
Adjustments for homogeneity

- Time phased data can be particularly difficult to handle
  - As spent dollars are influenced by inflation
    - Requires knowledge of time period involved
    - May require converting to “fixed dollar” units
  - Process data
    - Need to know specific sub-phase of the process
    - May require allocation to sub-phase

What To Look For In An Effort Measurement: Phases

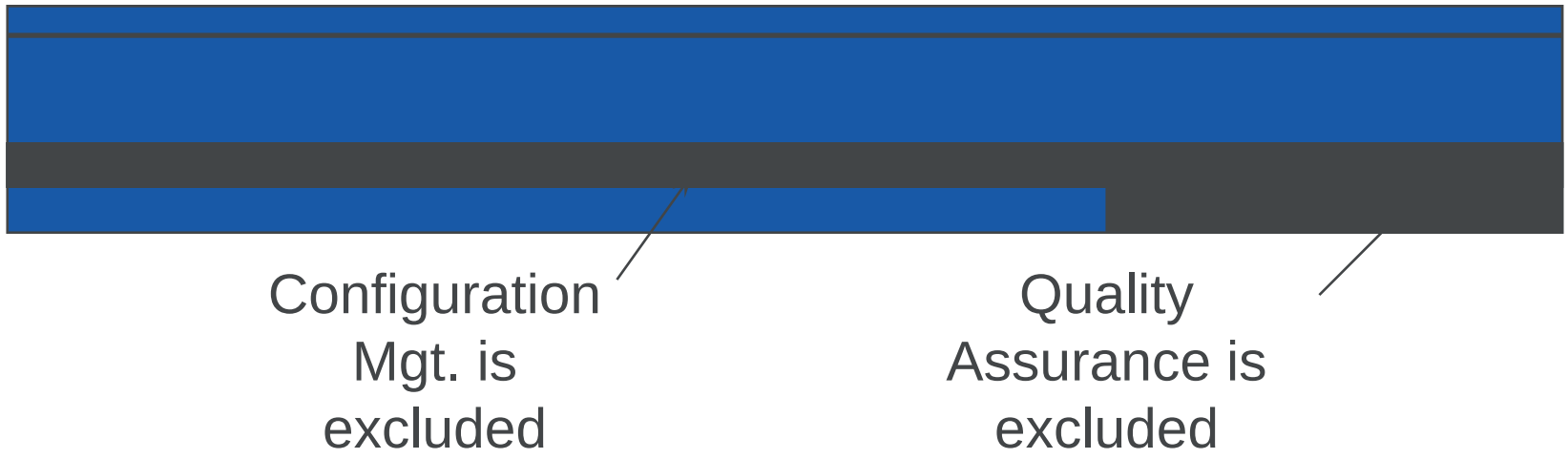
Phase : All *activities* may not be included.



Phases	
System Concepts	System Req & Design
System Req Analysis	Preliminary Design
Detailed Design	Code / Unit Testing
Software Test	System Integration / OT&E

What To Look For In An Effort Measurement: Labor

Labor : All *categories* may not be included.



Labor Categories	
Management	System Engineering
Design	Coding
Data Preparation	Test
Configuration Mgmt	Quality Assurance

What Else To Look For In An Actual

Forensics

Schedule. Was it "stop and start"? Were there schedule constraints?

Resources. Were there hard-hitting resource constraints?

Volatility. Did requirements undergo extraordinary evolution?

Manager's Objectives. Was it to complete the project in *minimum time* or at *least cost*?

Effort. Are effort figures actually derived from cost figures?

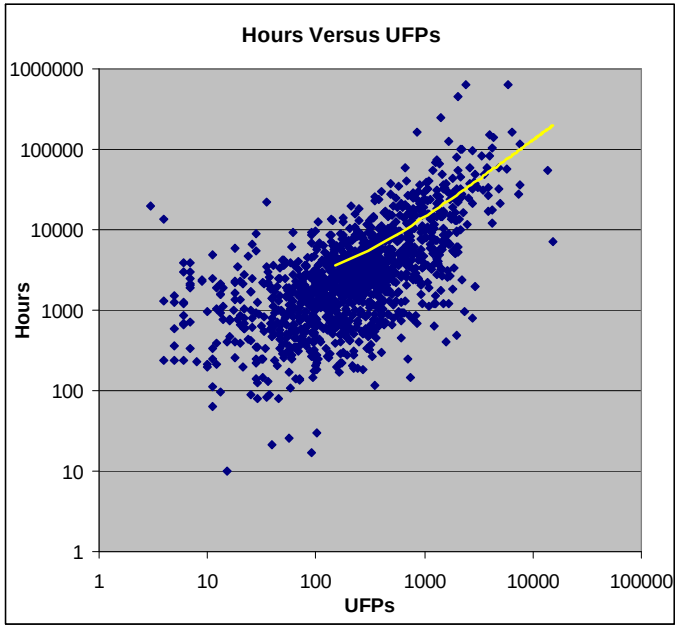
When creating the estimate --- adjustments for *extraordinary* conditions may be possible *within the software estimating model*.

Measurement & Data Evaluation Issues

- Sufficient data available?
- Established methodology to obtain relevant data on completed projects?
- Cost, technical, & programmatic data collected in consistent format?
- Data collection consistent with estimating practices?
- Was the source data adjusted?
- Did adjustments affect the logic, reasonableness, and defensibility of the data?
- Anomalies identified?

Key Items To Get Measurement Program "Right"

- Commitment from management
- Measure the right things right
- Clear standardized definitions
- Buy-in from troops
  - Don't use measurement to rate individual employees
- Measure at a level where something can be done about it
- Don't just measure top level data



Conclusions By Component

- **STANDARDS**
  - Adopt or create standards and definitions
- **SIZE**
  - Deal with counting standard and normalization among them (e.g. IFPUG & COSMIC)
- **EFFORT**
  - Standardize effort and provide normalization
- **DELIVERABLES**
  - Understand what is included and excluded & normalize where appropriate
- **CONSEQUENCES**
  - Reduce consequences of inconsistency AND provide viable ROI so IT can be controlled and understood by executives

Next Steps

- Evaluate alliance with PSM or other organization with measurement interests
- Choose or develop a data standard, checklists & processes
- Provide means to normalize from other forms to a standard format
- Reconcile
- Show successful measurement program ROI