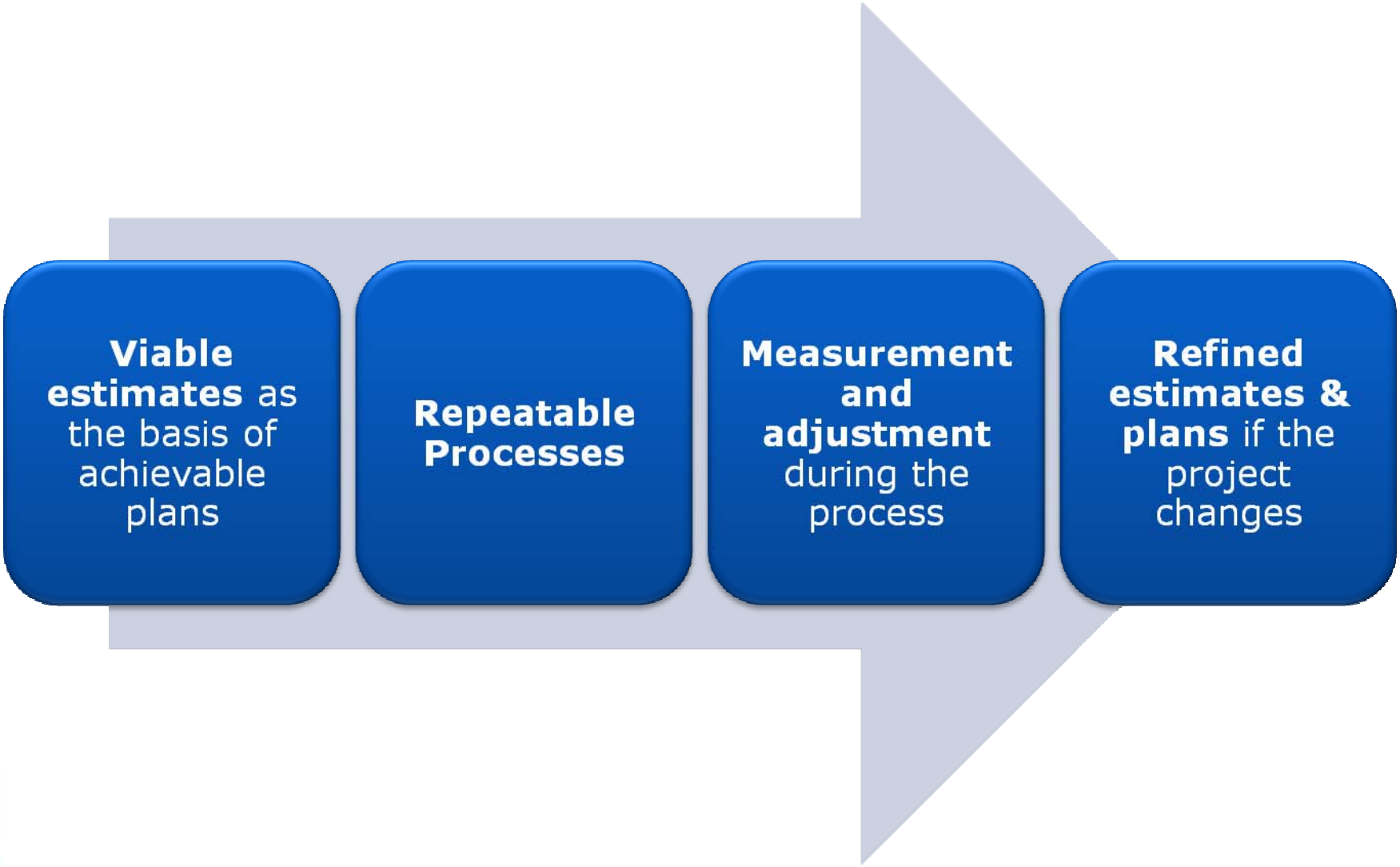
Improving Project Planning and Control: A 10-Step Process Within CMMI or other Process Orientations

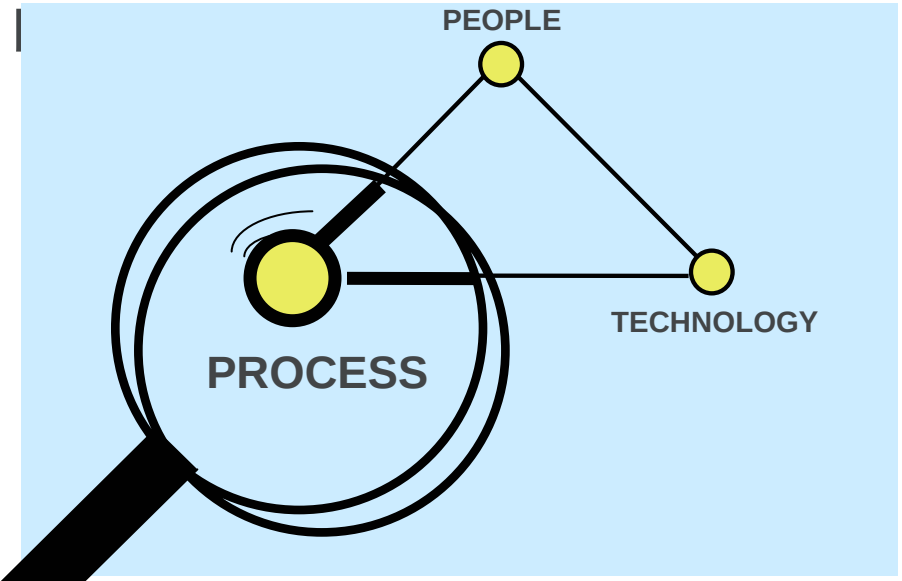
Daniel D. Galorath – galorath@galorath.com

Keys to project planning and control are



People, Process, Technology Are Keys Source CMMI Tutorial

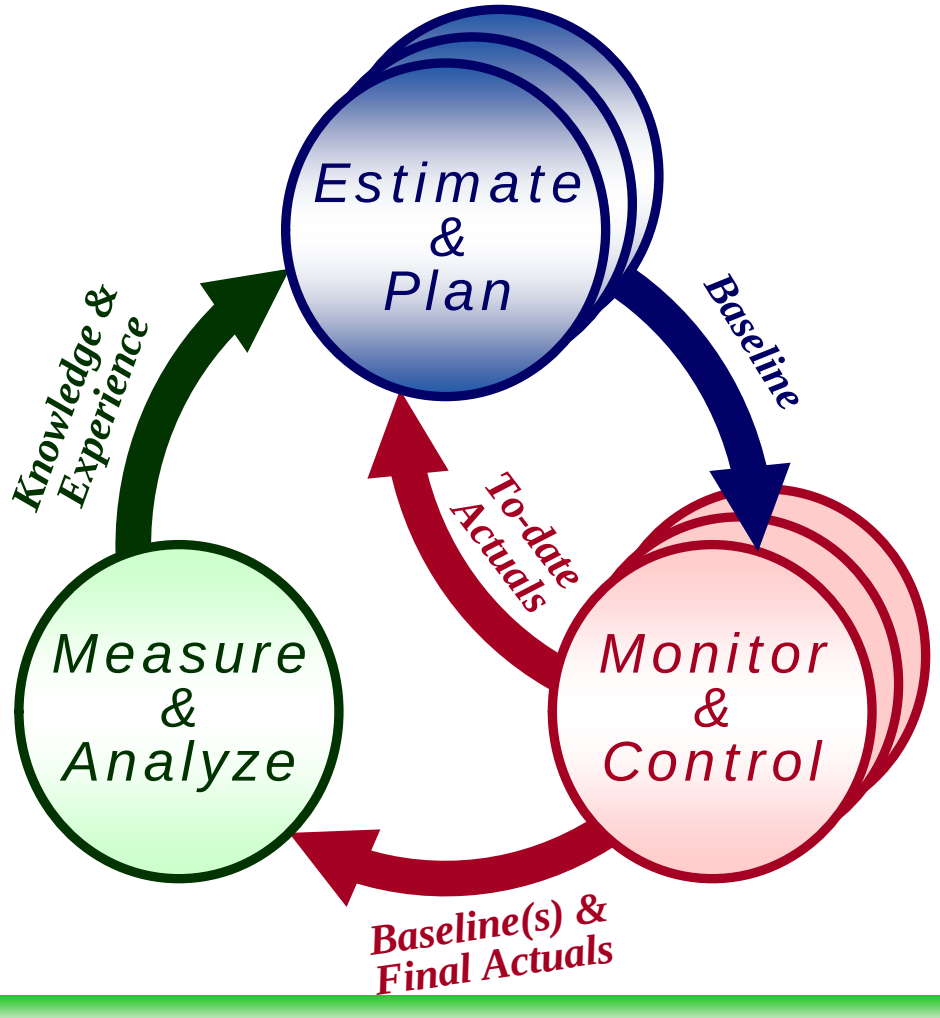
- Everyone realizes the importance of having a motivated, quality work force but…
- …even our finest people can't perform at their best when the process is not understood or operating "at its



Major determinants of product cost, schedule, and quality

CMMI Repeatable Process Areas of Interest

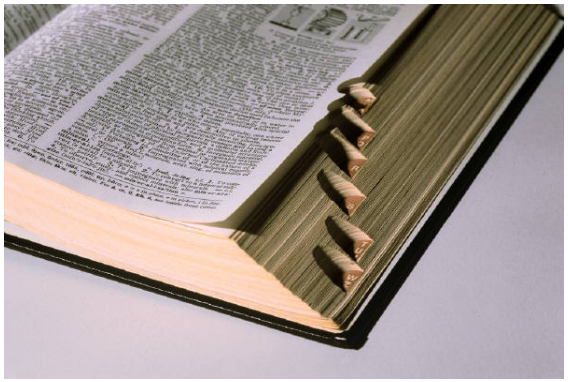
Quantitative Project Management



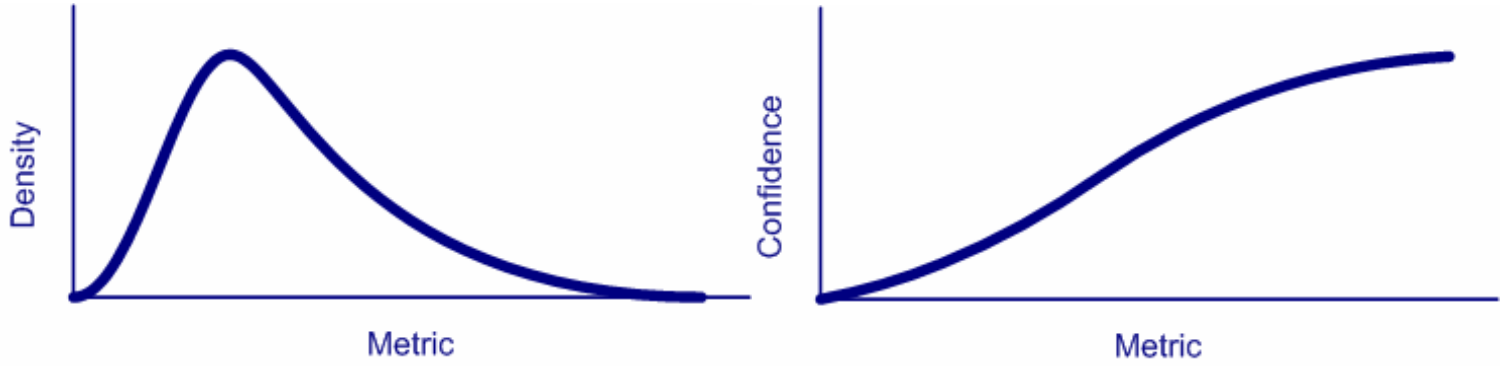
A Foundation of Risk Management

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*



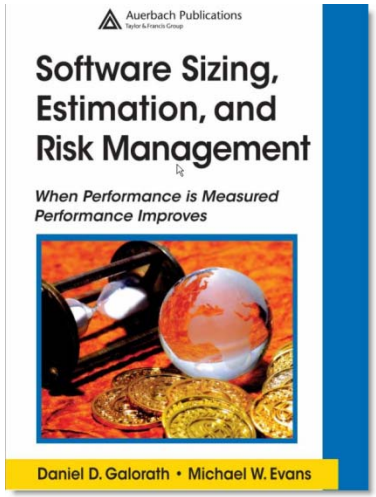
Poor Estimates Effects on Projects

- Inaccurate estimates significant impact on project success:
  - Poor implementations
  - Critical processes don't scale
  - Emergency staffing
  - Cost overruns caused by underestimating project needs
- Lack of well defined objectives, requirements, & specifications, results in creeping scope resulting in:
  - Forever changing project goals
  - Frustration: Death Marches
  - Customer dissatisfaction
  - Cost overruns and missed schedules
  - Project Failures
- Incorrect estimates / bad plans are a root cause of subsequent program risk

Estimating & Planning are key to software project success

10 Step Software Estimation Process:

Consistent Processes = Reliable Estimates



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

Step One: Establish Estimate Scope and Purpose

- Define and document estimate expectations, scope & Purpose
  - Provides a baseline against which to gauge the effect of future changes
  - Reduces misunderstandings & contradictory assumptions
- Estimate should be considered a living document
  - As projects change, data changes or new information becomes available, it should be documented and factored into the estimate in order to maintain the project's integrity

Define What's Included in the Estimate

- Development effort breaks down into…
- What functionality will be developed
- How will it be done?
  - Activities & phases
- Who will do the work?
  - Personnel labor categories

The Importance of Scope

- Nature of estimate
  - Rough or precision?
  - Most likely, worst case, or “should cost” scenario?
  - Cross-check or “target costing” exercise?
  - Acquisition or life-cycle?
- Nature of system
  - Number of subsystems: flight, ground
  - Number of WBS elements
  - Level of indenture
  - Number of releases / builds
- Nature of code
  - Waterfall Vs Agile Vs Incremental Vs…
  - All new, any reuse?
  - Effort applied to reused code: modify, I&T, etc.
  - Any COTS or GOTS to be integrated?
  - Any code generators to be used?

Know what you need to do before you need to do it

Step Two: Establish Technical Baseline, Groundrules, & Assumptions

- Functionality included in the estimate or range must be established
  - If detailed functionality is not known, groundrules and assumptions state what is and isn't included in the estimate.
  - Issues of COTS, reuse, and other assumptions should be documented as well.
- Groundrules and assumptions form the foundation of the estimate
  - although early at early stages they are preliminary and therefore rife with uncertainty, they must be credible and documented
  - Review and redefine these assumptions regularly as the estimate moves forward.

Technical Baseline

- Describe each work element
  - Core functionality
  - Key qualitative modeling inputs
  - Size, if available
- Identify multiple builds/releases (if any)
- Reference or excerpt available documentation
- Provide rationale & justification for Basis of Estimate

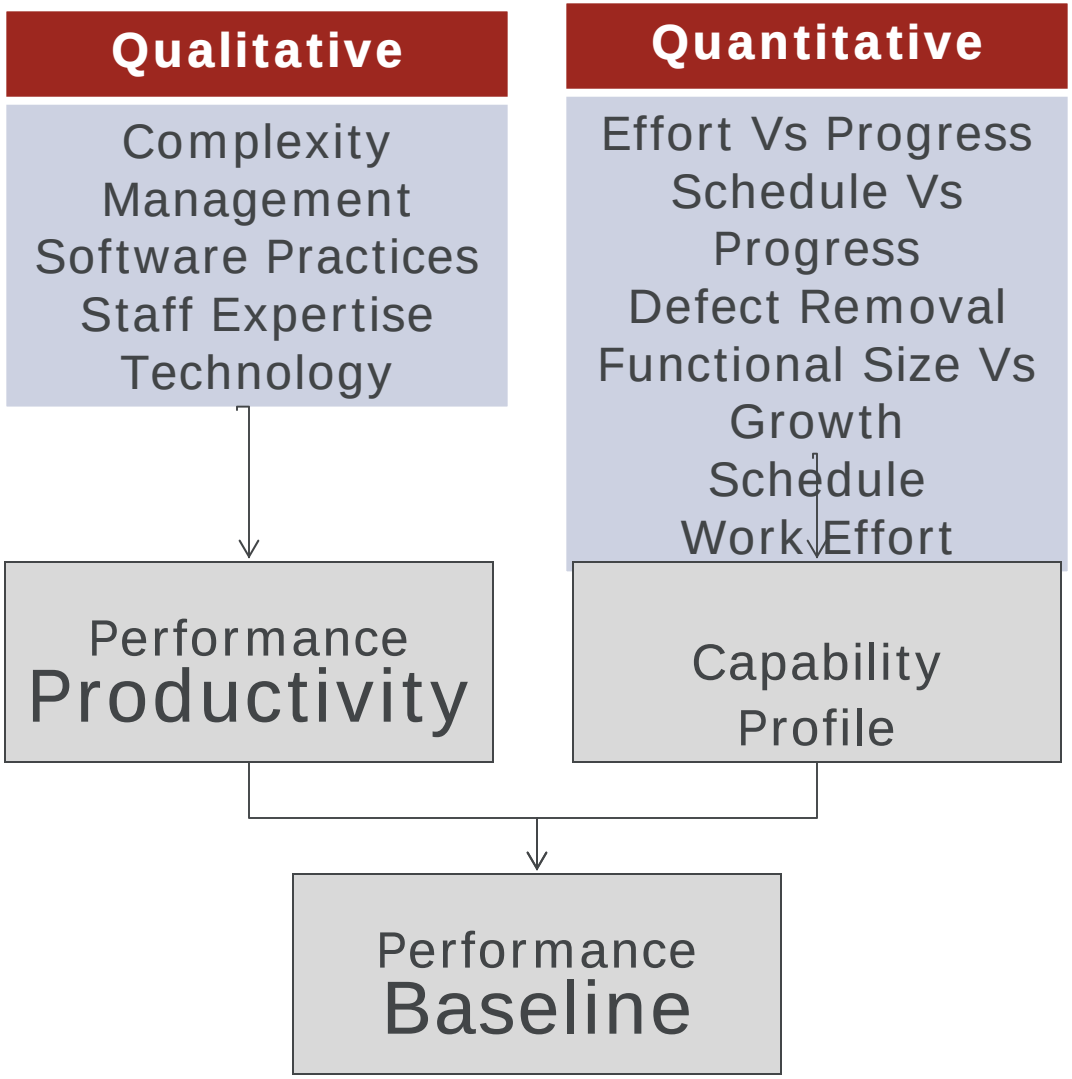
Groundrules & Assumptions

- Groundrule: given requirement of the estimate (e.g. software must support windows and Linux)
  - Be sure to provide source
- Assumption: assumed to scope estimate
  - Be sure to provide source and substantiation
- What's known, what's unknown
- Anything relating to scope
  - What's included, what's excluded
- Anything relating to modeling inputs
  - Who you interviewed and when
  - What you learned

Step Three: Collect Data

- Software Data Collection Process key considerations
  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
  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

Measurement Elements: You Get What You Measure



KEY
Both elements collected, with quality

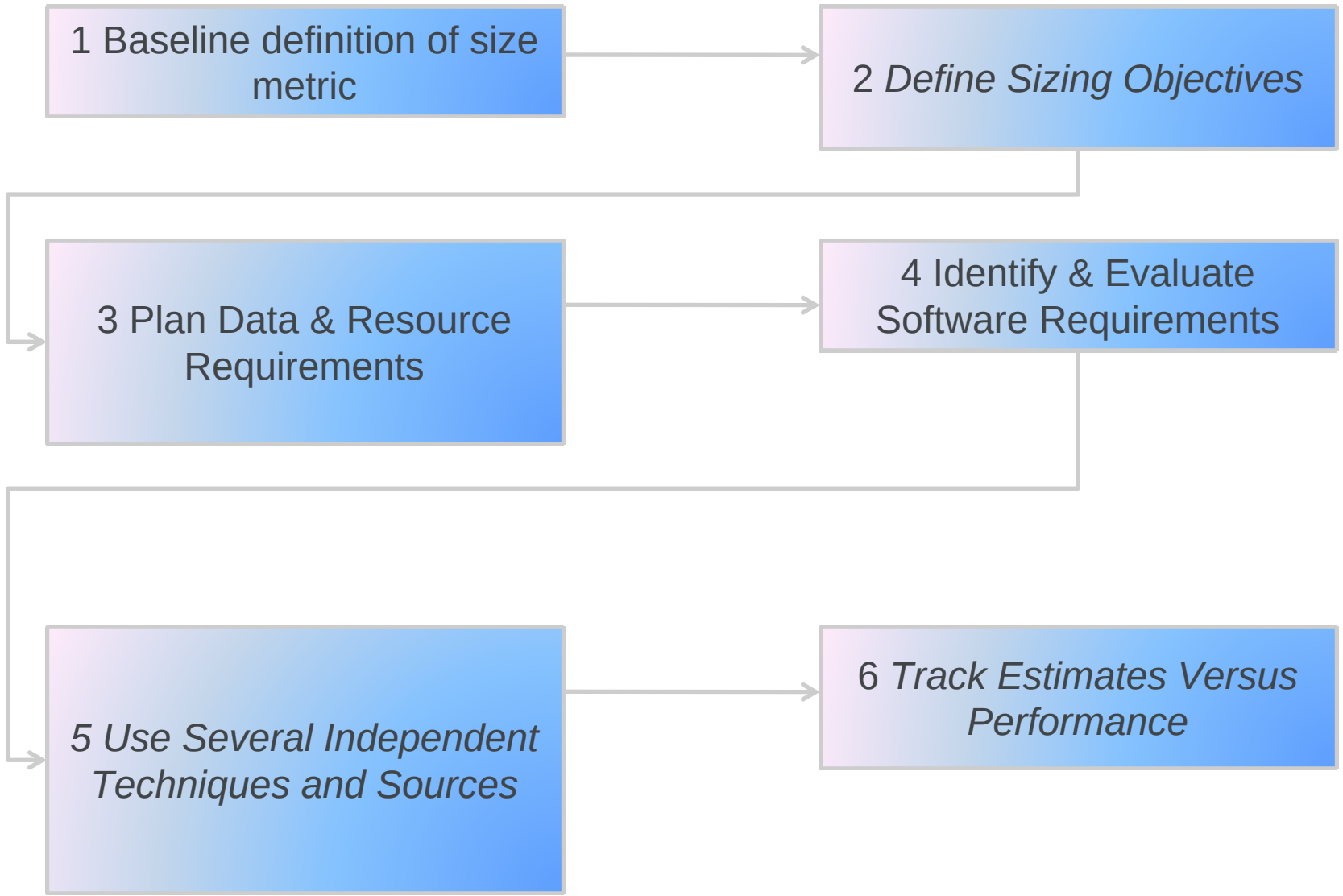
Organization Profile

Internal and External Comparator

Step Four: Software Sizing

- Spend Time on sizing
  - Include rework that will be required to develop the product
- Size defines scope
  - Function Points, Lines of Code and Use Cases are most used
- Estimate *least, likely, most* range.
- Common methods of estimating product size:
  - Expert opinion
  - Analogy
  - Formalized methodology
  - Statistical sizing — Provides a range of potential sizes that is characterized by *least*, *likely*, and *most*

Ideal Size Projection Takes Time



Step Five: Prepare Baseline Estimate

- Trained, experienced, and skilled people should be assigned to size the software and prepare the estimates
- Critical that they be given the proper technology & tools
- Project manager must define and implement a mature, documented, and repeatable estimation process

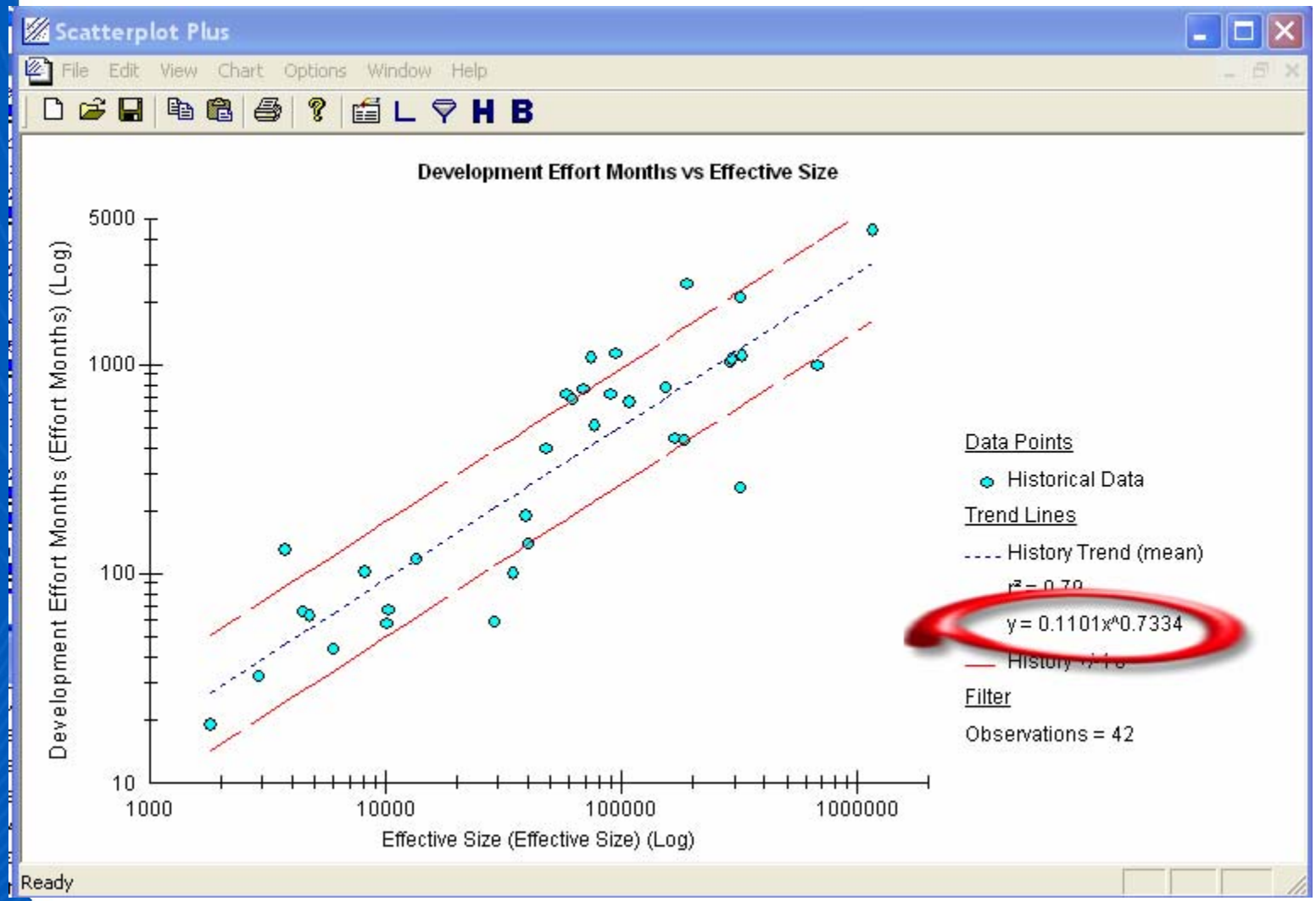
Estimation Methods 1 of 2

Model Category	Description	Advantages	Limitations
Guessing	Off the cuff estimates	Quick Can obtain any answer desired	No Basis or substantiation No Process Usually Wrong
Analogy	Compare project with past similar projects.	Estimates are based on actual experience.	Truly similar projects must exist.
Expert Judgment	Consult with one or more experts.	Little or no historical data is needed; good for new or unique projects.	Experts tend to be biased; knowledge level is sometimes questionable; may not be consistent.
Top Down Estimation	A hierarchical decomposition of the system into progressively smaller components is used to estimate the size of a software component.	Provides an estimate linked to requirements and allows common libraries to size lower level components.	Need valid requirements. Difficult to track architecture; engineering bias may lead to underestimation.

Estimation Methods 2 of 2

Model Category	Description	Advantages	Limitations
Design To Cost	Uses expert judgment to determine how much functionality can be provided for given budget.	Easy to get under stakeholder number	Little or no engineering basis.
Simple CER's	Equation with one or more unknowns that provides cost / schedule estimate	Some basis in data	Simple relationships may not tell the whole story Historical data may not tell the whole story
Comprehensive Parametric Models	Perform overall estimate using design parameters and mathematical algorithms.	Models are usually fast and easy to use, and useful early in a program; they are also objective and repeatable.	Models can be inaccurate if not properly calibrated and validated; historical data may not be relevant to new programs; optimism in parameters may lead to underestimation.

Example Simple Parametric Model (Inadequate for Estimation or Management By Itself)

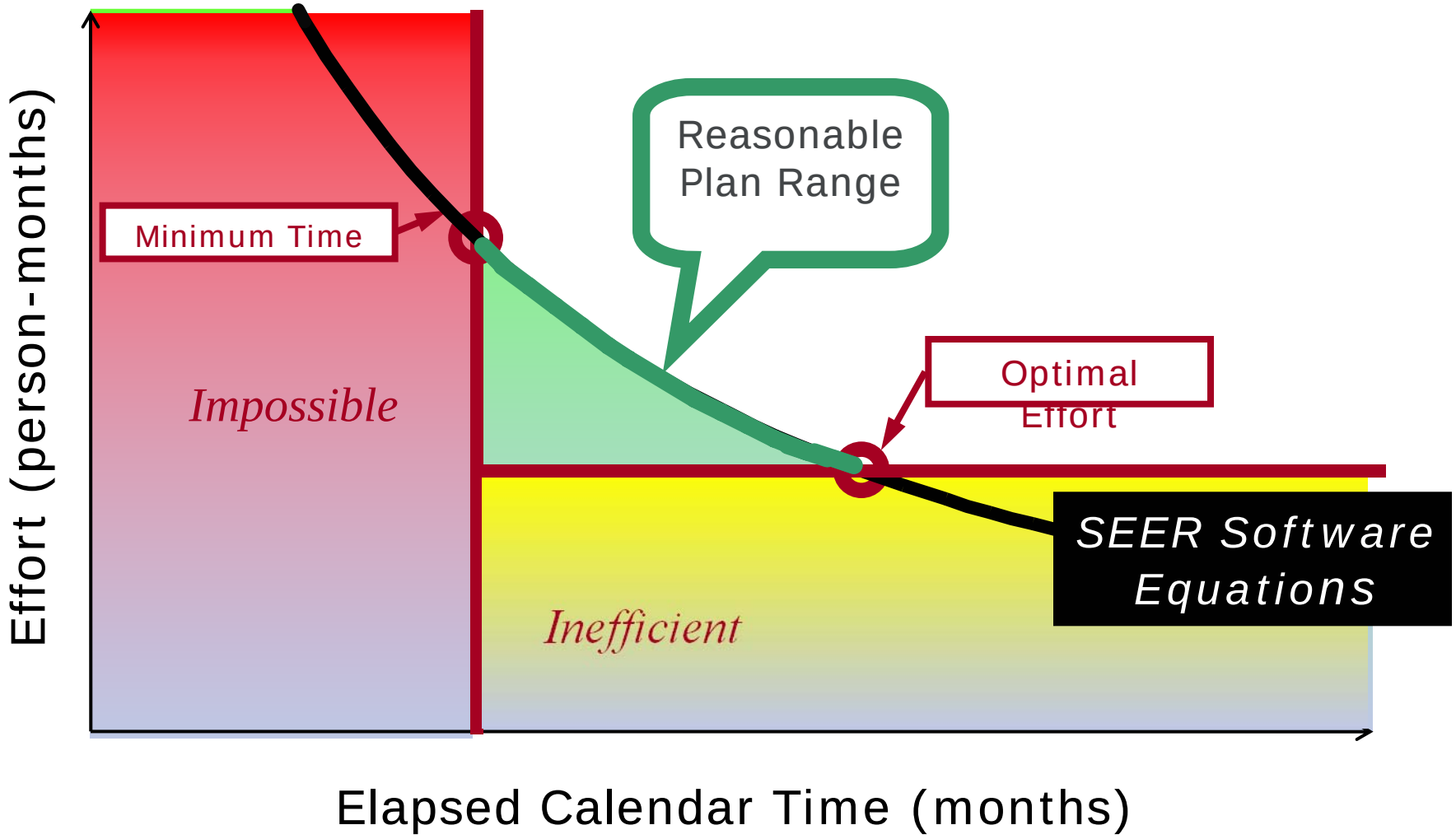


Manual Estimates Human Reasons For Error

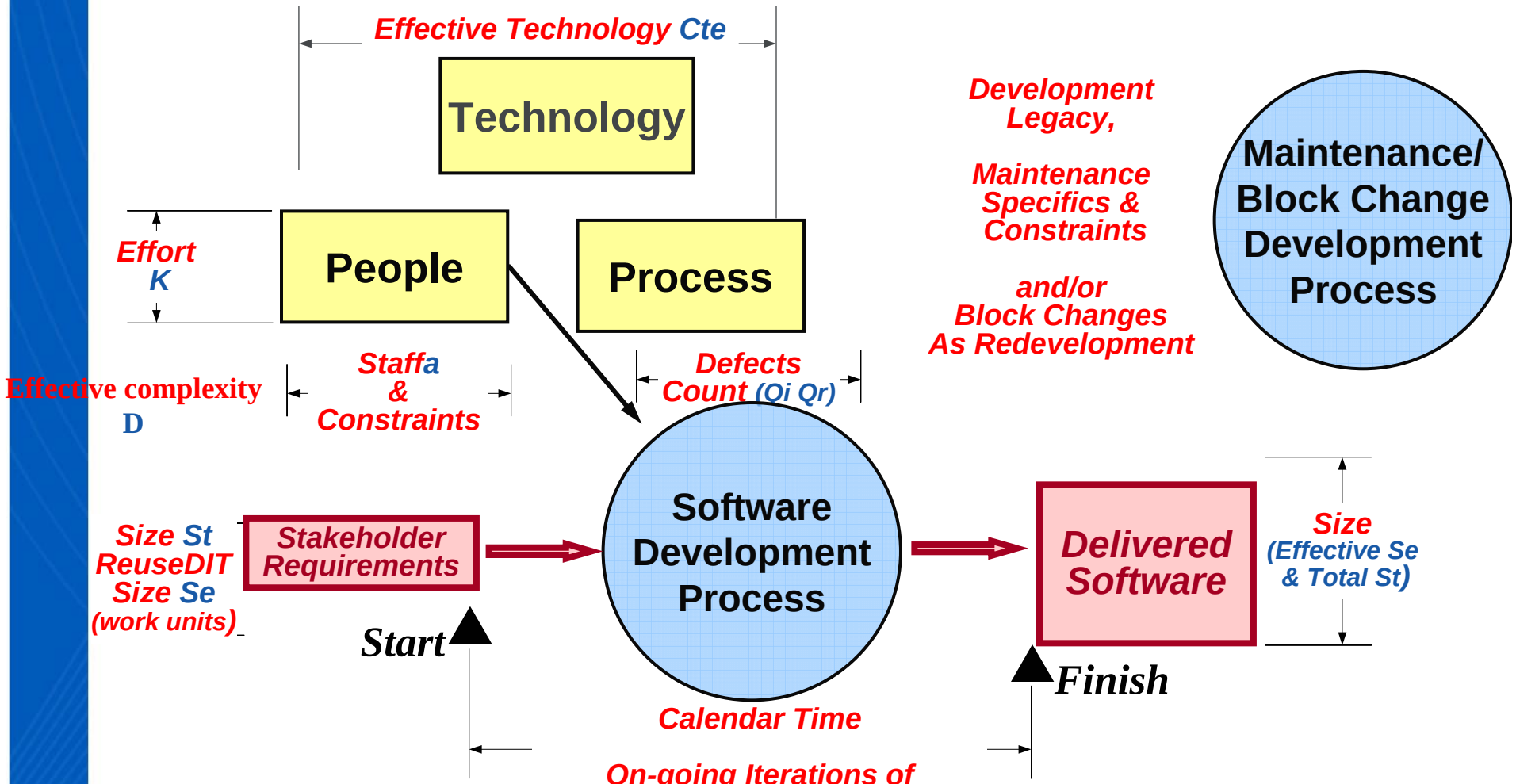
- Desire for “credibility” motivates overestimate behavior (80% probability?)
  - So must spend all the time to be “reliable”
  - Better approach force 50% probability & have “buffer” for overruns
- Technical pride causes underestimates
- Buy-in causes underestimates

Knowing Software Planning Possibilities Is Critical To Success

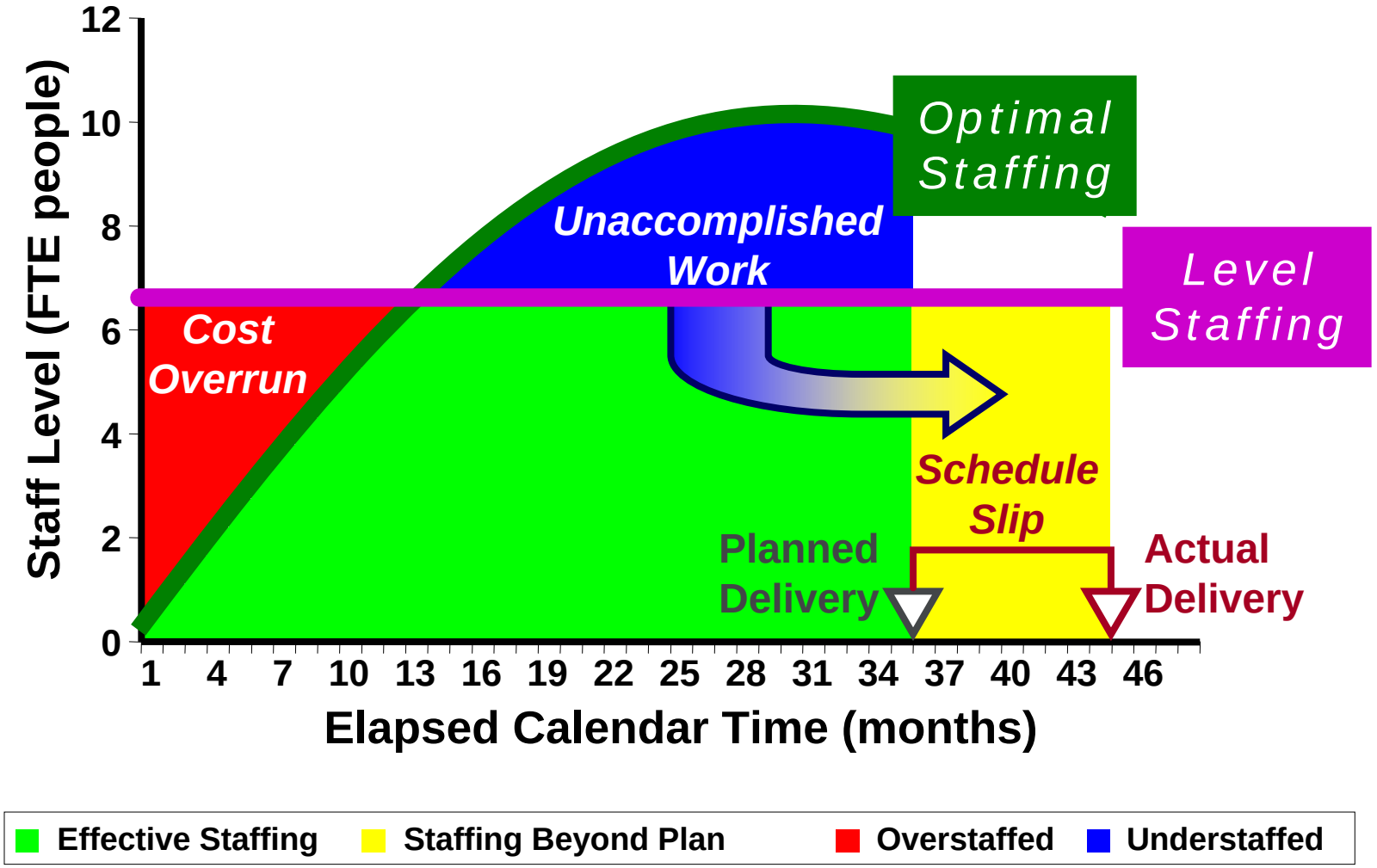
For a given Size,Technology, Complexity & Probability



Software Estimation Basic Model & Associated Metrics

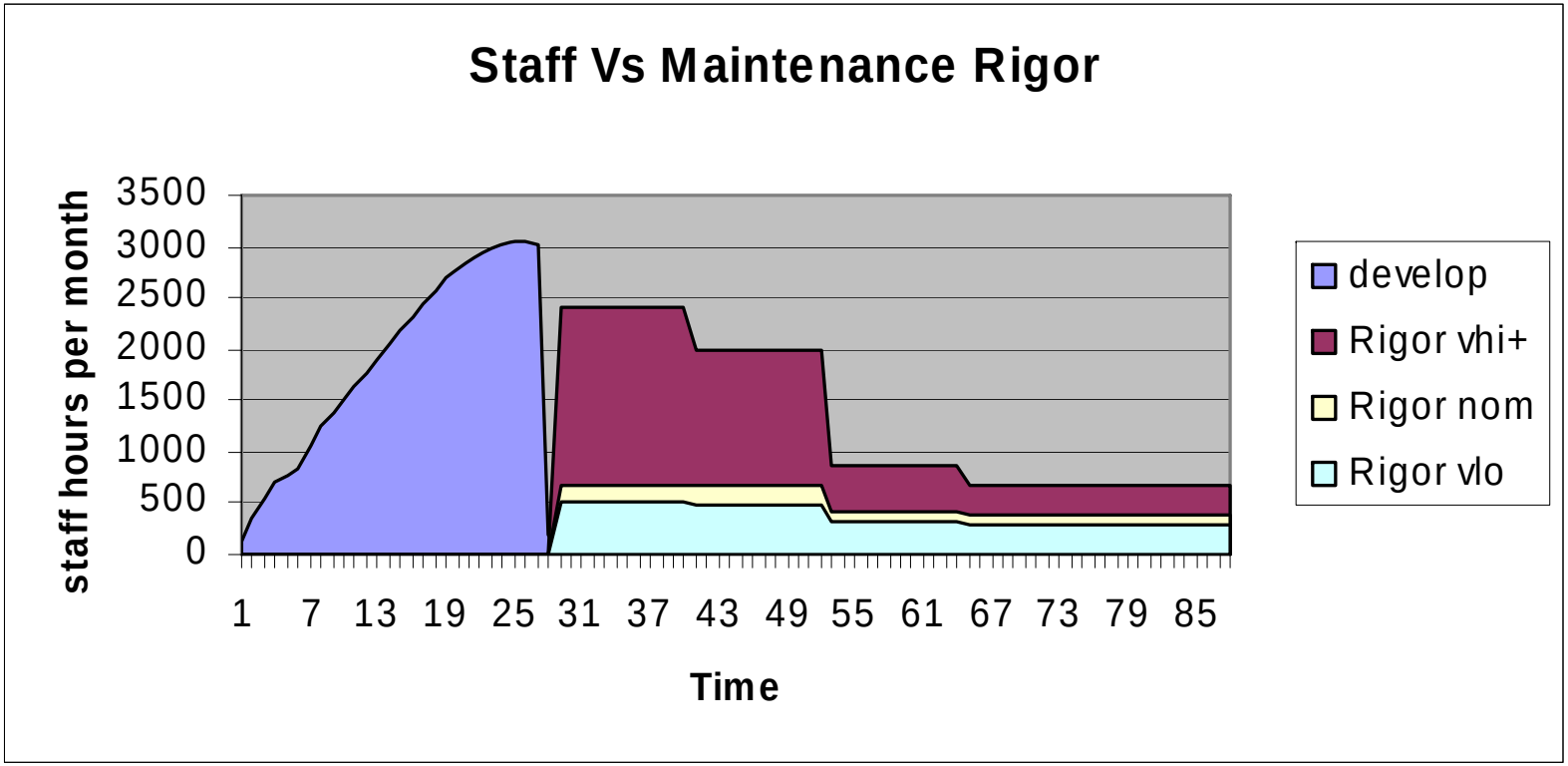


Avoid "Death Marches" and Failed Projects By Applying "Brooks Law"



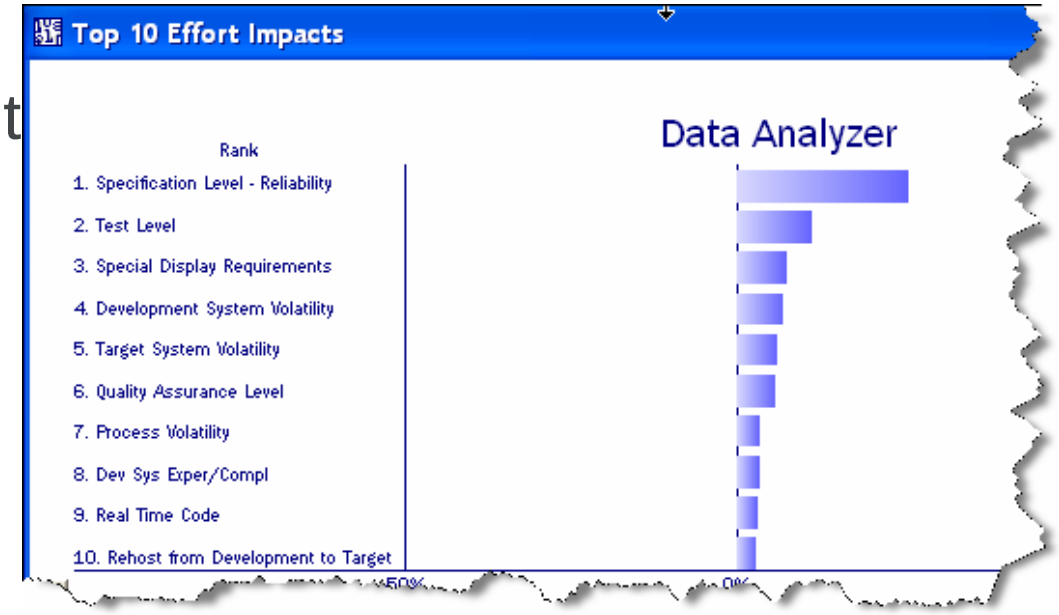
Estimate and Plan Project Total Ownership Costs Up Front

Most Projects Spend Low During Maintenance



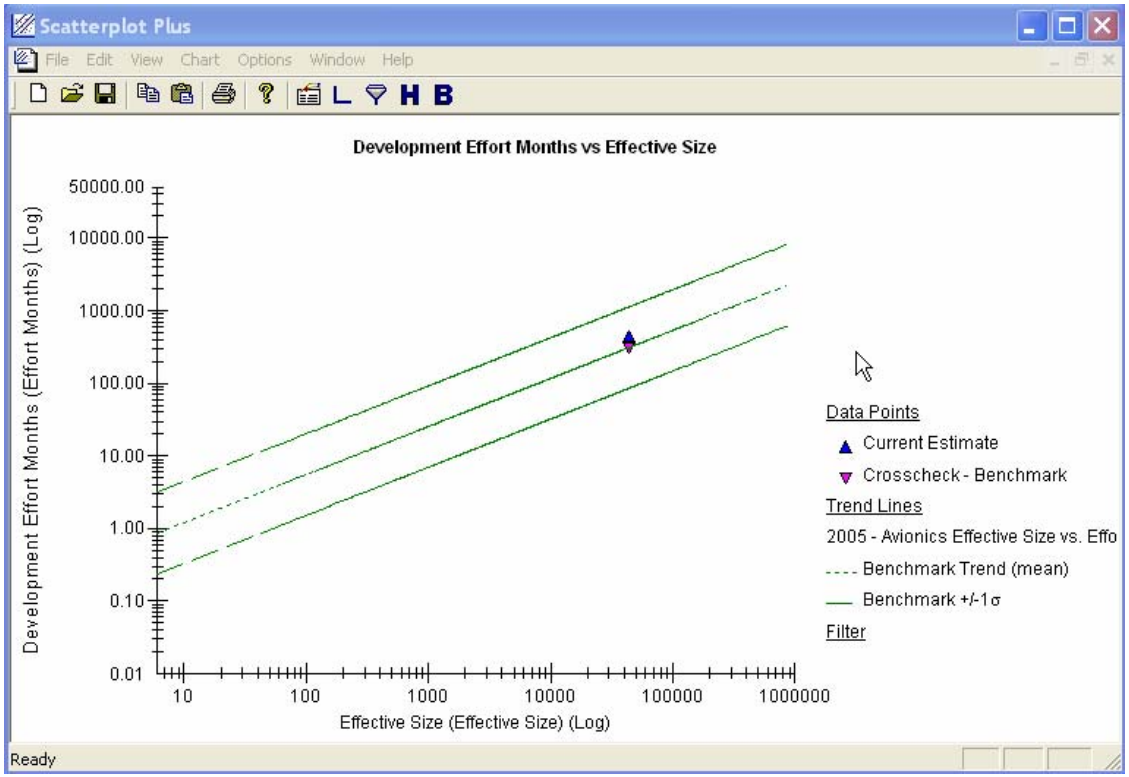
Generate the Estimate

- Using chosen methodology and tool, do a first run
- Never report preliminary results!
- Focus on the inputs
  - Verify completeness
  - Verify accuracy
- Focus on the output
  - Sanity check for reasonableness, completeness
- What's driving the estimate?
  - Top ten parameters
- Use "fresh eyes" to review
  - Ask a colleague for help
  - Set aside overnight



Cross Checks

- A cross check estimate prepared using a different methodology and/or tool may be required
  - Understand the usage of the cross check methodology and/or tool
  - Keep the focus on the estimate, not on the tool



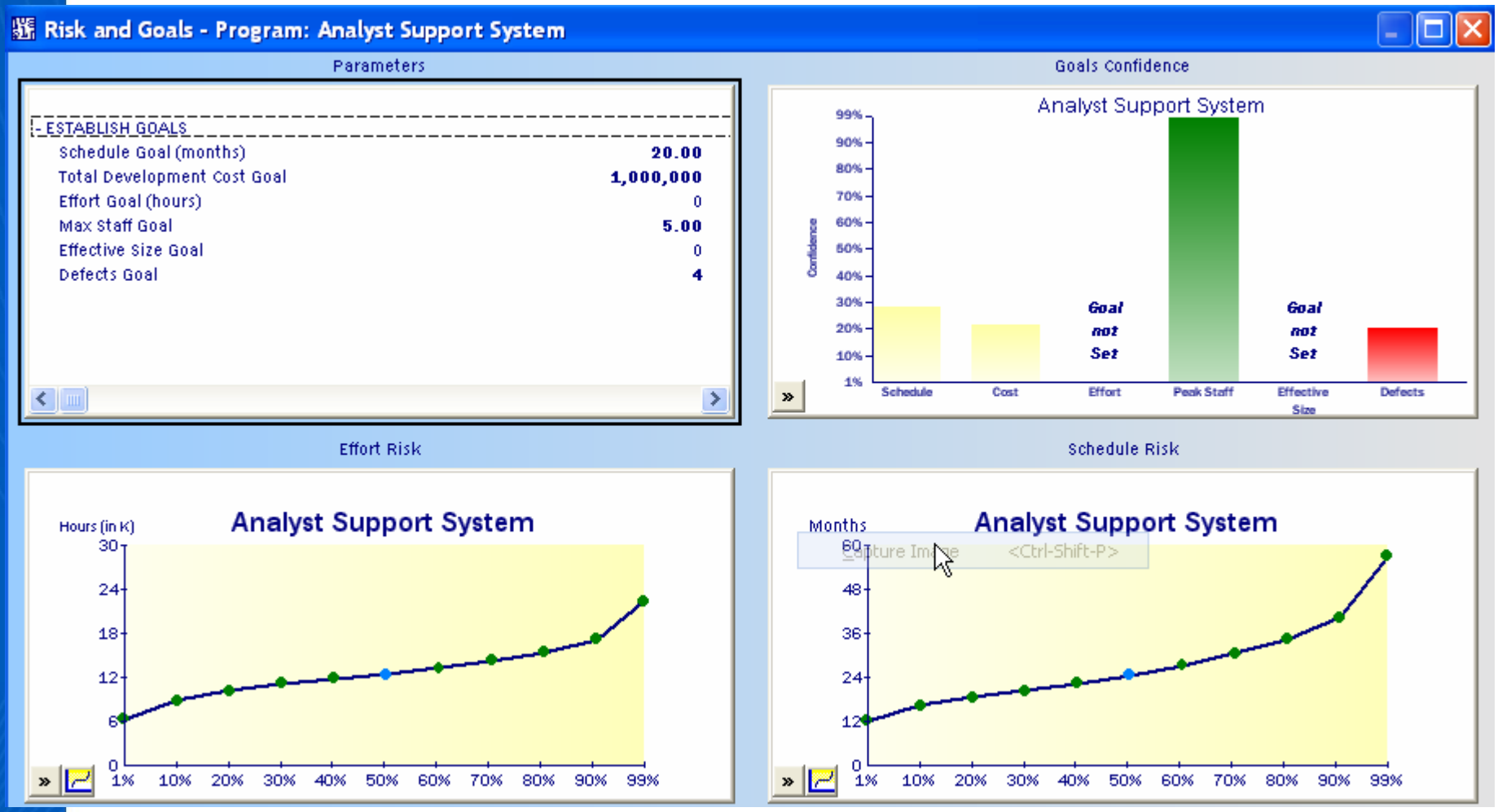
Estimate Process Checklist

- Scope project
  - Depth & breadth
  - Use project plan
- Data collection
  - Spend time on sizing
  - Document sources
- Tech baseline
  - Document all critical inputs
  - Ground rules & Assumptions, estimating process
- Prepare estimate
  - Verify inputs & Use “fresh eyes”
  - Sanity check outputs
  - Never deliver preliminary results
  - Benchmark

Step Six: Quantify Risks and Risk Analysis

- Risk can produce loss of time, or quality, money, control, understanding, etc.
- Approximate the probability that the event will occur
  - Determine how risk can be mitigated
  - Risk control involves a set of actions taken to reduce or eliminate a risk.
- Risk management identifies & addresses internal & external potential threats
  - Problems associated with sizing and estimating software potentially can have dramatic negative effects.
  - If problems can be foreseen & causes acted upon in time, effects can be mitigated
- Although cost, schedule, and product performance risks are interrelated, they can also be analyzed independently
  - Risks must be identified as specific instances in order to be manageable
  - Statistical risk/uncertainty analysis should be a part of schedule & effort estimation process

Understanding Risk and Uncertainty is Essential To Project Management

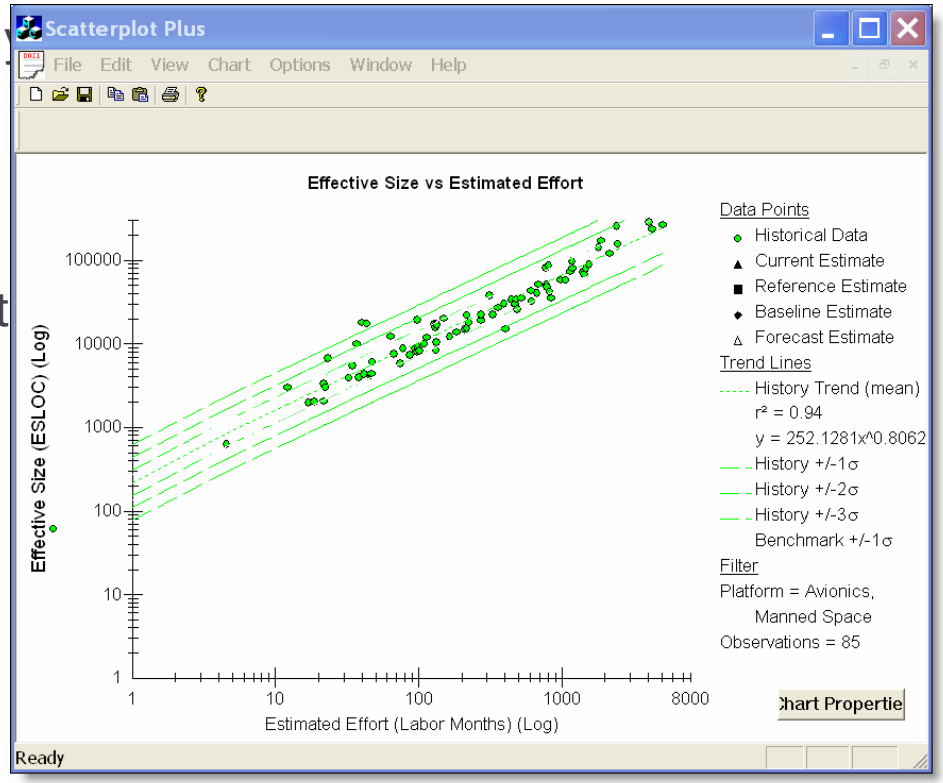


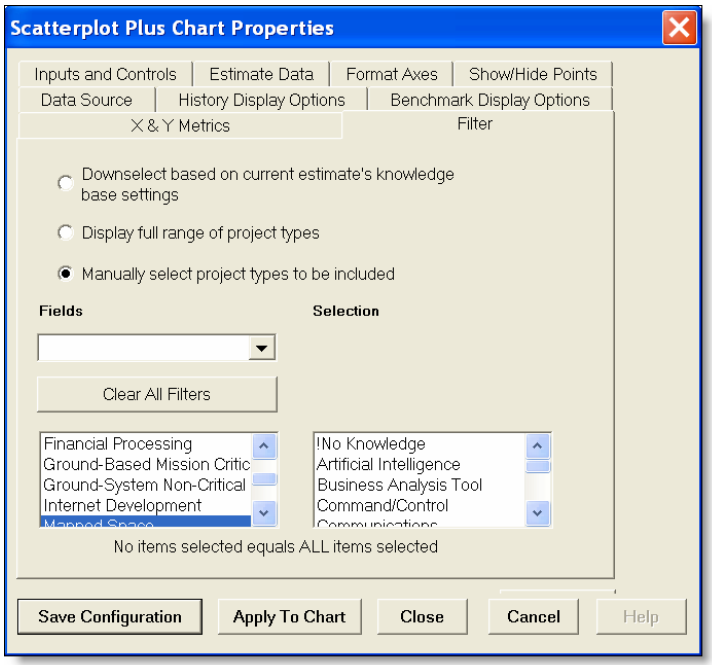
Step Seven: Estimate Validation and Review

- Ideally, validation performed by one who was not involved in generating the estimate
- Assess estimate assumptions
- Ensure groundrules are consistent applied
- Rigorous validation process exposes faulty assumptions, unreliable data and estimator bias
  - Provides clearer understanding of risks inherent in projections
  - Having isolated problems at their source, you can take steps to contain the risks associated with them, and you will have a more realistic picture of what your project will actually require to succeed
- Failing to validate the estimate may result in much greater downstream costs, or even a failed project

Compare Parametrics With Metrics and Sanity Checks

- Work with common repository
- Shows actual data, ranges, and correlations
- Plots parametric estimates and contrasts with data point
- Plots actual data and / or trends





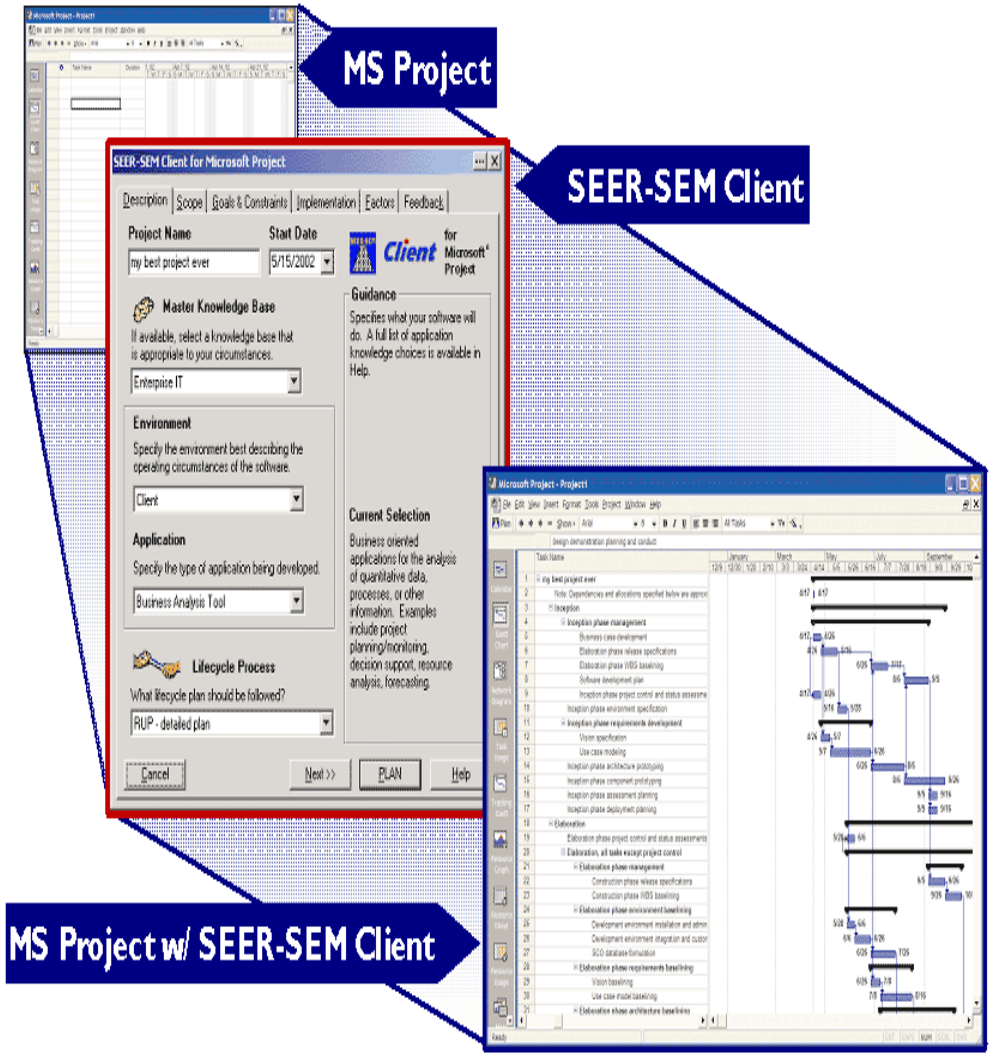
"In God we trust,
all others bring data."
- W. Edwards Deming

Step Eight: Generate A Project Plan

- Allocating estimate cost & schedule & allocating to function and task-oriented work breakdown structure
- Issues
  - Inexperience evaluating decisions long term impacts
    - Lack necessary information
    - Unwillingness to spend the time
  - Decisions based on what management wants to hear
    - Good manager understands project realities
      - Explain the reality in language his managers can understand
      - Problem managers either lead a project to an unintended conclusion or, worse, drift down the road to disaster
- Software management / planning problems long recognized as leading causes of project failures
  - Bad management decisions
  - Incorrect focus
  - Destructive politics

Expand Product WBS to Task level Plan

- Automatically constructs a complete project plan
  - ***with relatively few inputs***
  - ***or directly from your parametric project estimate.***
- You can create custom life cycle templates.
- You can customize labor categories to reflect the way that your organization assigns tasks to departments or labor categories to accurately plan staff allocation for a project.



Step Nine: Document Estimate and Lessons Learned

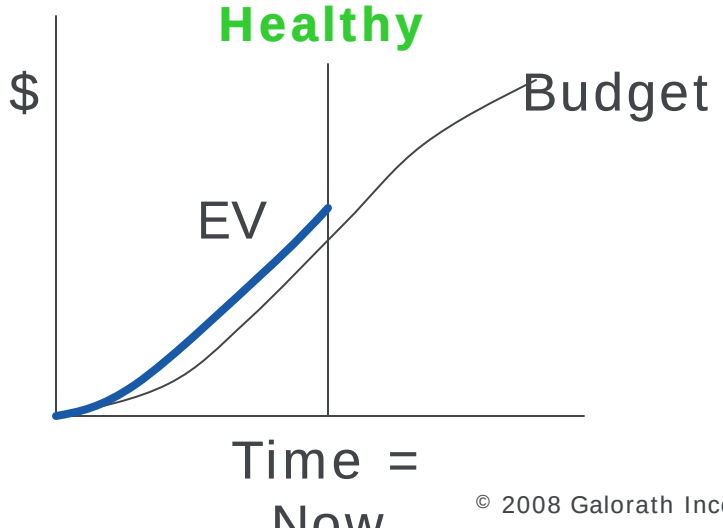
- Document upon estimate complete AND project complete
  - document the pertinent information
  - record the lessons you learned
  - Provide evidence of valid process was valid
  - that you generated the estimate in good faith
  - Collect results to substantiate or calibrate estimation models
- Document any missing or incomplete information
- Capture risks, issues, and problems that the process addressed and any complications that arose
- Document key decisions made during the estimate & results
- Document dynamics that occurred during the process e.g.
  - interactions of your estimation team
  - interfaces with clients
  - trade-offs made to address issues identified during the process
- Conduct a lessons-learned session as soon as possible after project completion
- Every software project should be used as an opportunity to improve the estimating process

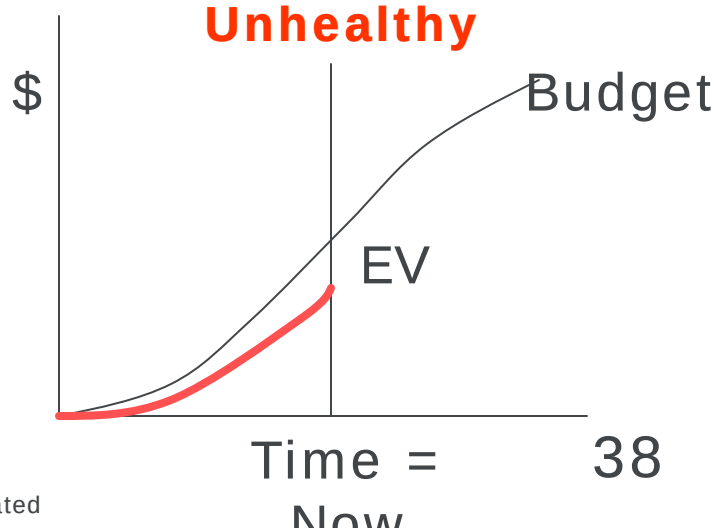
Step Ten: Track Project Throughout Development

- *Refining Estimates throughout Project*
- Once a project has started, use estimates as a basis for performance measurement & project control
- Monitor actual effort & duration of tasks and/or phases against planned values to ensure you have the project under control
- Applying earned value techniques along with parametric estimation can help ensure successful projects
- Evaluate defects & growth in addition to simple earned value

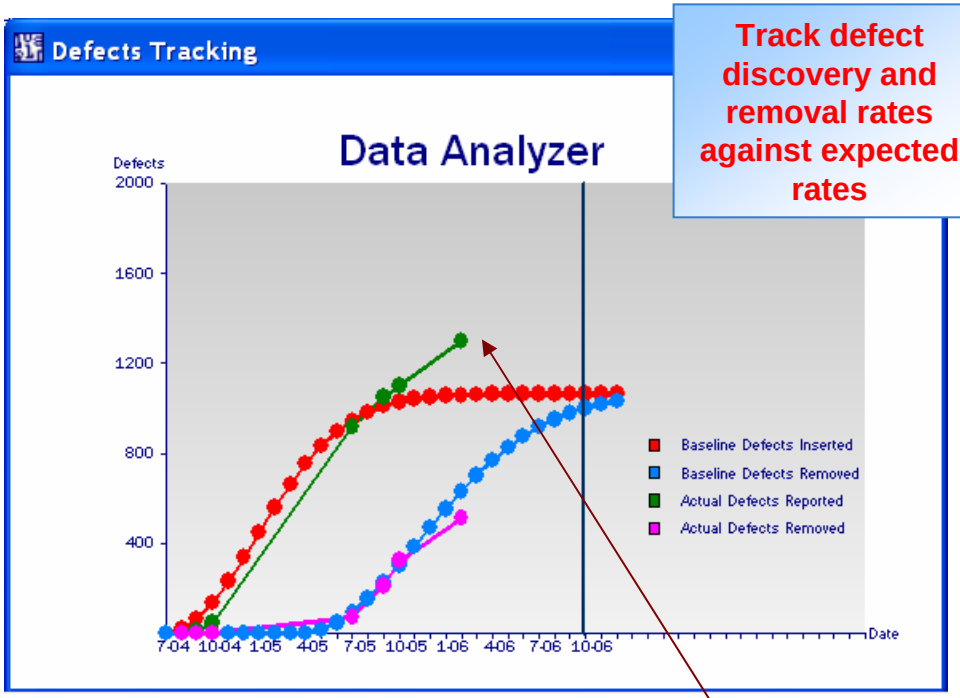
Use Earned Value TO Quantify Progress Versus Effort

- Main EVM concern: what has been accomplished in a given time and budget, versus what was planned for the same time and budget
  - A project is generally deemed healthy if what has been accomplished is what was planned, or more
  - A project is deemed unhealthy if accomplishment lags expectations
- Definition: Earned value = budgeted value for the work accomplished (what you got for what it cost you)





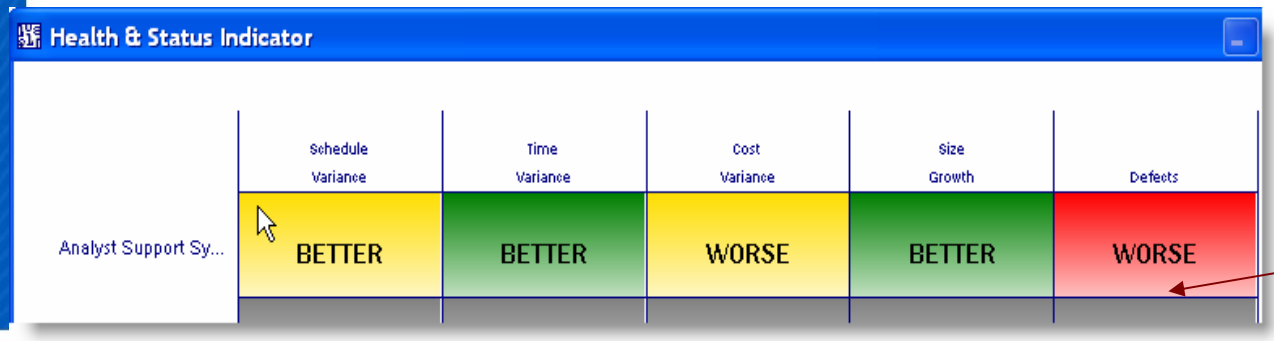
Defects and Growth Impact Software Process



Track defect discovery and removal rates against expected rates

Heath and Status Indicator shows status and trends from the previous snapshot

Thresholds are user definable

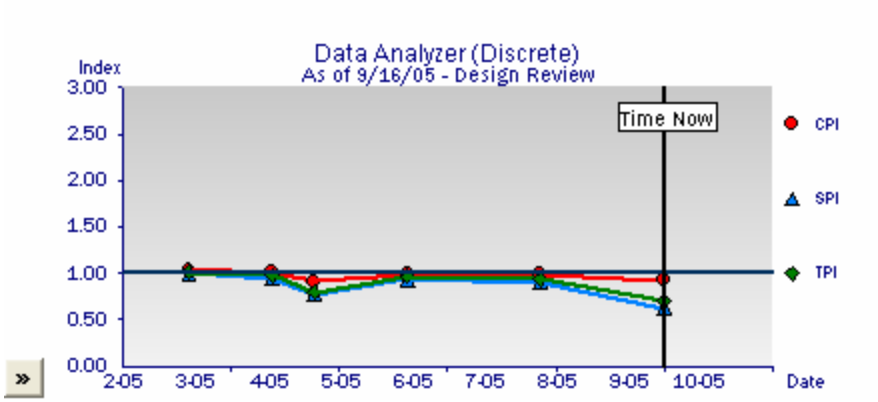


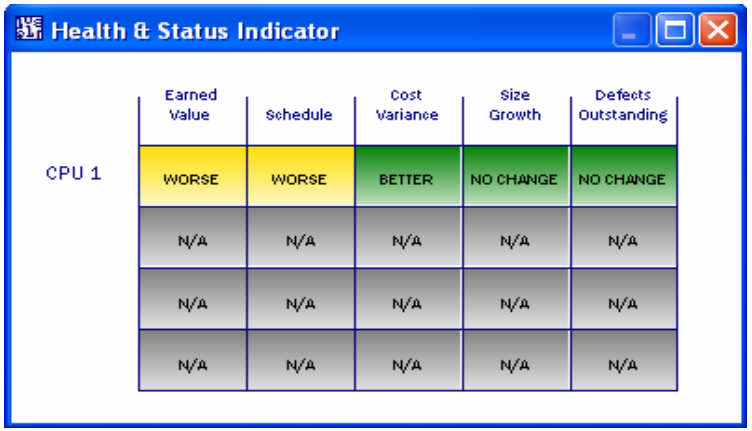
Increased defect reporting rate shows a worsening trend

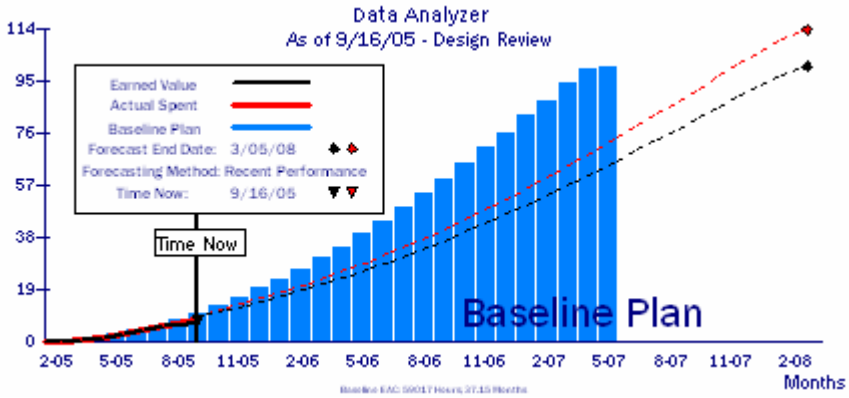
Parametric Project Monitoring & Control

Provides Performance Measurement aspects of ANSI/EIA-STD-748

- Adds Performance Measurement (*Earned Value*) methods to parametric estimation model
- Accepts progress & expenditure inputs
- Provides cost, schedule, and time variances
- Provides cost, schedule, & time indices
- Performance-based cost & schedule Estimate at Completion
- Displays health and status indicators







Estimation Lessons Learned

- Estimate should drive plan
- Re-estimate if the project changes
- Measure early
  - Before trouble
  - Early under-budget milestones not necessarily good
  - Skimping on upfront planning; requirements and design work, will most likely be in trouble later
- Estimation, planning, tracking, controlling – then using the information to do better next time
- EVM shouldn't be used alone
  - Other metrics are necessary to be kept in concert with the EVM metrics in order to keep a project on track
- Recognize that you have a problem

Summary

- Software projects are manageable
- Basis of management is a viable estimate
- The 10 step process provides a repeatable approach to estimation process
- Measurement is key to the estimation process
- Measurement, Monitoring & Control are keys to successful software projects
- Beware of using only simple metrics to drive estimates