Integrating Planning, Control, Measurement & Analysis CMMI™ Process Areas To Achieve More Successful Software Projects

Daniel D. Galorath
galorath@galorath.com
310 414-3222 x614

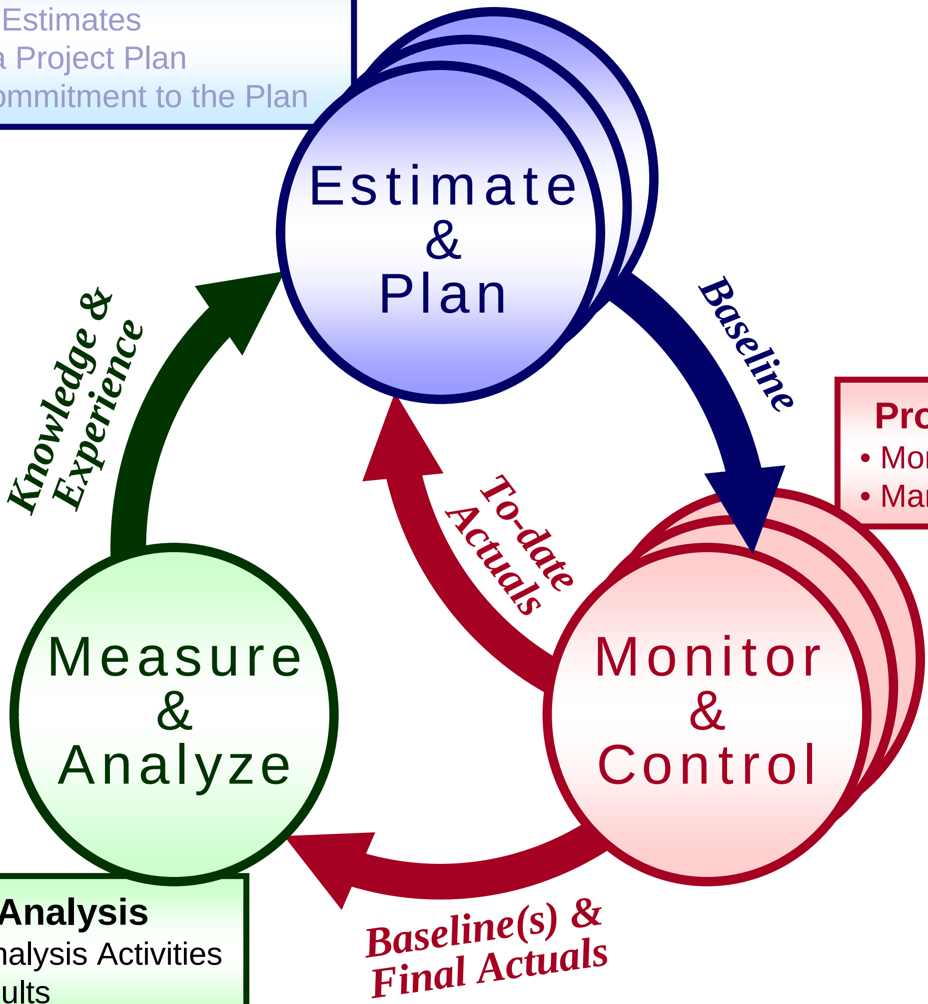
- Integrating Planning, Control, Measurement & Analysis CMMI™ Process Areas To Achieve More Successful Software Projects

- Project Planning and Project Monitoring and Control, two CMMI® Level 2 Process Areas, are essential to Software Project Management. Prior to project initiation, the relationship between these two process areas tends to be strong. Unfortunately, the status quo tends to exhibit a breakdown in this relationship once the project has begun. The result is an open-loop situation: we know where we planned to be and where we are; however, we don’t know where and when we will likely end up.

- This paper first reviews the fundamentals of Parametric Estimating and Performance Measurement
- and then proposes an integrated process uniting Project Planning, Project Monitoring and Control, and Measurement and Analysis. The benefits of this integration are an increased probability that: 1) a project’s plan is achievable (i.e., it honors the laws of software project dynamics and keeps from “death marches”); and 2) actual performance matches reasonably-expected (planned) performance; (i.e., that a project is delivered on time and within budget); and 3) if there are troubles that there is an early warning of trouble and possibilities for managers to remediate. Finally, the paper chronicles, through the use of selected artifacts, an actual project employing this integrated approach.

CMMI Key Process Areas Require Estimation With Feedback

Project Planning
- Establish Estimates
- Develop a Project Plan
- Obtain Commitment to the Plan



Project Monitoring and Control
- Monitor Project Against Plan
- Manage Corrective Action to Closure

Measurement and Analysis
- Align Measurement and Analysis Activities
- Provide Measurement Results

✓**Foundation of Risk Management &**
✓**Quantitative Project Management**

Fundamentals of Software Estimation

Poor Estimates Effects on Projects

- **Inaccurate estimates can have 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
  - 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

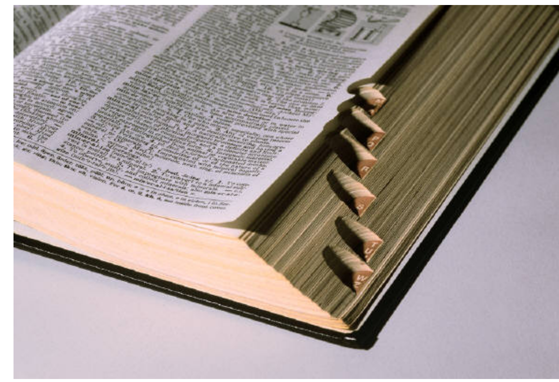
We Can't Afford Not To Spend Time and Money On Viable Estimates and Project Plans

Adapted From Aspire S. Hunt

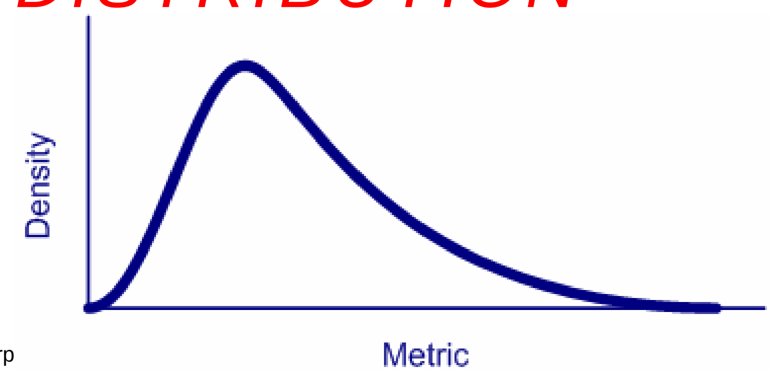
- **Apollo**
  - Most Apollo program faults were software related
- **Loss of NASA Spacecraft**
  - FORTRAN statement error caused loss of 1st NASA probe to Venus
- **After success with Ariane 4, the maiden flight of Ariane 5 ended in destruction when design defects in the control software were revealed by the faster horizontal drifting speed of the new rocket.**
- **Financial & Program Impacts**
  - Airbus A310 45,000 hours expended against software development (20,000 hours budgeted!)
  - American Airlines lost nearly $1billion due to software faults
  - In 1991 after 3 lines of code (in millions of lines of code in a signaling program) were changed, telephone systems in California and the Eastern Seaboard went off line
- **Loss of Life**
  - Gulf War 1 Patriot Missile tracking failure due to 0.36 second software driven clock.
  - And so on …….

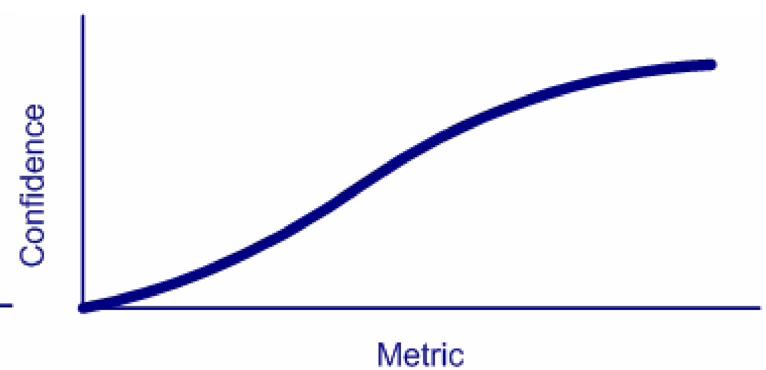
An Estimate Defined

- **An *estimate* is the most knowledgeable statement you can make *at a particular point in time* regarding:**
  - Effort / Cost
  - Schedule
  - Staffing
  - Risk
  - Reliability

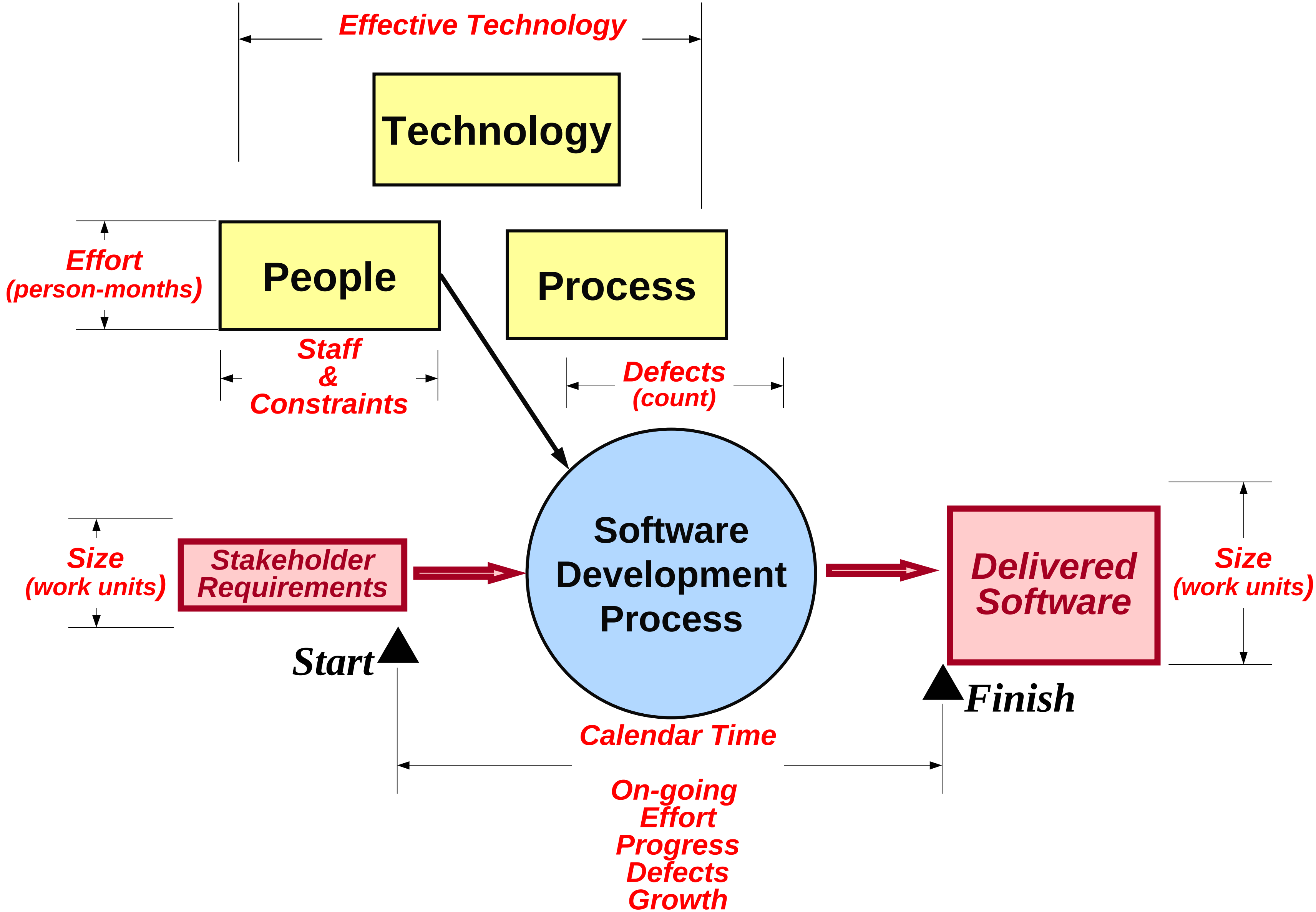


- **Estimates more precise with progress**
- *A WELL FORMED ESTIMATE IS A DISTRIBUTION*

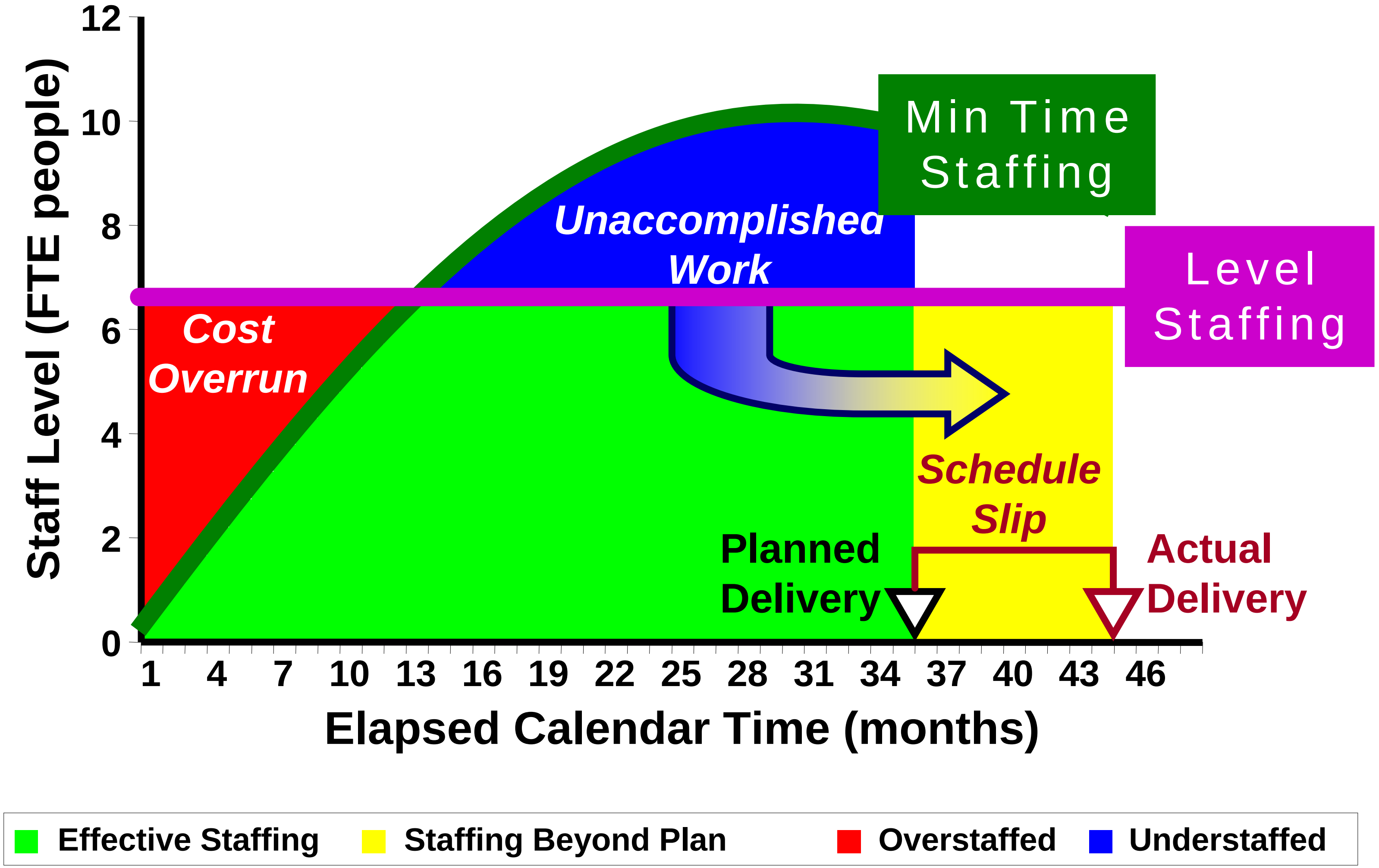




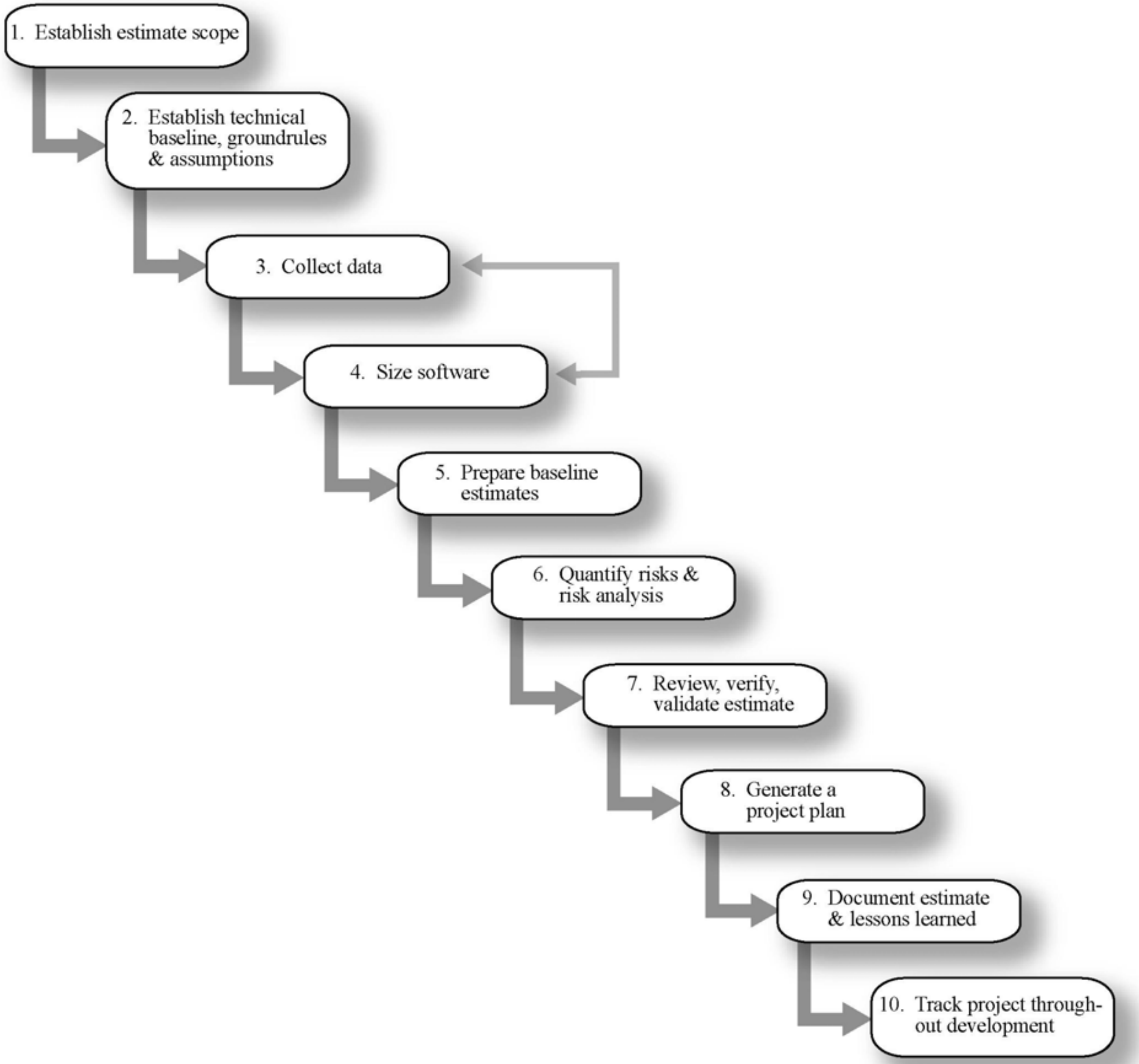
Software Development Core Metrics



SEER-SEM Provides Staffing Plans that Avoid "Death Marches" and Failed Projects Applying "Brooks Law"



10 Step Software Estimation Process Supports Software Best Practices



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

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 BOE**

Ground Rules & 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

Data Collection

- **Begin with available documentation**
  - SDP, TRD, SRS, COO, ORD, CARD, BAFO, BOE, SOW
  - A-level Spec, Program Schedule, Contractor Presentations
- **Perform interviews where & when possible**
  - Development contractor(s)
  - Program office personnel, FFRDC engineers
  - Preparation is paramount!
  - Use a questionnaire, sent in advance
  - Phraseology determines answers
  - Confirm key definitions
  - Be willing to challenge, point out inconsistencies

SEER® by GALORATH

Step Four: Software Sizing

- **Spend Time on sizing**
  - Include rework that will be required to develop the product,
- **Estimate *least likely most* range.**
- **Some widely used methods of estimating product size are:**
  - Expert opinion
  - Analogy
  - Formalized methodology
  - Statistical sizing — Provides a range of potential sizes that is characterized by *least*, *likely*, and *most*

Steps to Estimating Software Size

- **Use a software sizing method that is consistent and repeatable**
  - Reestimate as specs change
  - Establish baseline definition of the size metric & identify normalization process if different from the definition chosen
1. **Define sizing objectives**
   - Are you trying to describe the size of individual computer programs, plan major milestones in the estimation process, or adjust for project replanning? Varying granularities of sizing detail will be appropriate
2. **Plan data and resource requirements of the sizing activity**
3. **Identify and evaluate software requirements, a set of software specifications that are as unambiguous as possible**
4. **Use several independent techniques and sources and put findings in a table**
5. **Track the accuracy of the estimate versus actuals as the project progresses, and reestimate the product size periodically with actual data.**

SEER® by GALORATH

Galorath Size Estimation Methodology

Apply Size Estimation Methodology

Evaluate all sources of software size...

New size

Generated code

Glue code

Pre-existing size (rework)

COTS/GOTS

Integrated code

Multiple size estimates

Total Size Estimates	Least	Likely	Most
Expert Judgement	**12000**	15500	17000
Relevant Range by Analogy	19850	24750	32540
Sizing Database	8000		**46000**
Functional Analysis		27540	
SEER-AccuScope	15450	**22650**	29850
Delphi Analysis	16788	19750	22713
Composite	**12000**	**22650**	**46000**

Viable size range

Expert judgement

SEER-AccuScope analysis

Sizing databases

Functional analysis

Counts for pre-existing

Analogies

...Using multiple methods

Step Five: Prepare Baseline Estimate

1. Trained, experienced, and skilled people should be assigned to size the software and prepare the estimates

2. it is critically important that they be given the proper technology and tools

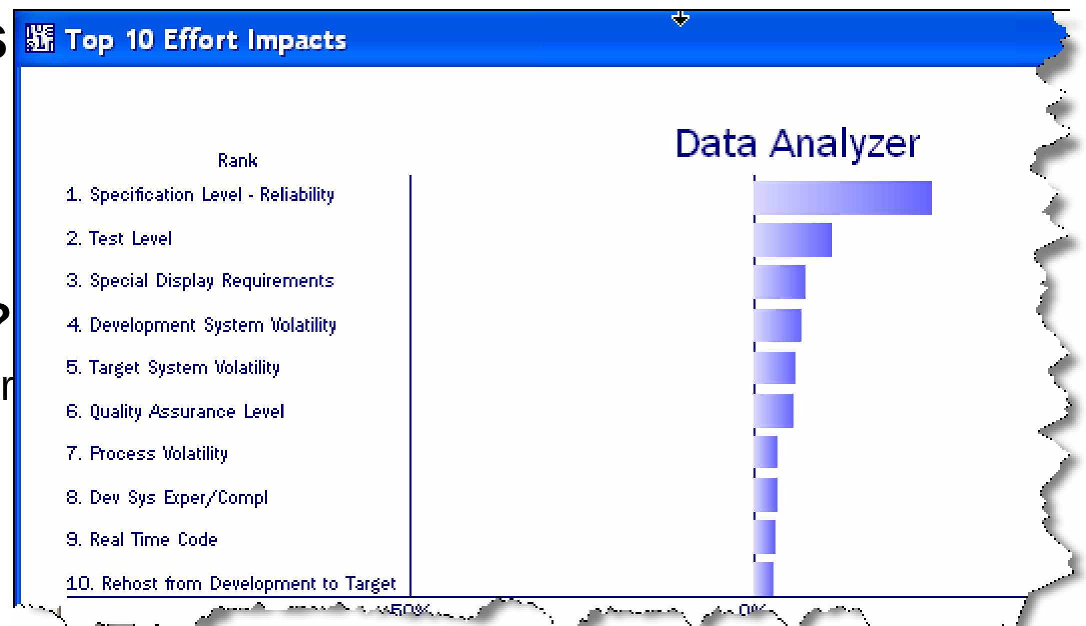
3. Project manager must define and implement a mature, documented, and repeatable estimation process

Estimation Approaches

- **Guessing (which is not recommended)**
- **Using existing productivity data exclusively (also not recommended)**
- ***Bottom-Up Estimating***
  - Bottom-up (a.k.a. grassroots) estimating, decomposes the software to its lowest levels by function or task and then sums the results
  - Can be effective for smaller systems and where estimates are carefully produced
- ***Parametric Software cost models***
  - Provide means for applying a consistent method for subjecting uncertain situations to rigorous mathematical and statistical analysis
  - When used properly they are more comprehensive than other estimating techniques and help to reduce the amount of bias that goes into estimating software projects
- ***Delphi Based Estimates***
  - Several expert teams or individuals, each with an equal voice and an understanding up front that there are no correct answers, start with the same description of the task at hand and generate estimates anonymously, repeating the process until consensus is reached
- ***Activity-Based Estimates***
  - Begin with requirements & size then define the required tasks, which will serve to identify the overall effort that will be required
  - Typically Major cost drivers focused on non-coding tasks included in effort estimate
  - Build a task hierarchy (WBS) that represents how the work will be organized and performed
  - This work breakdown structure can be the backbone of the project plan

Generate the Estimate

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



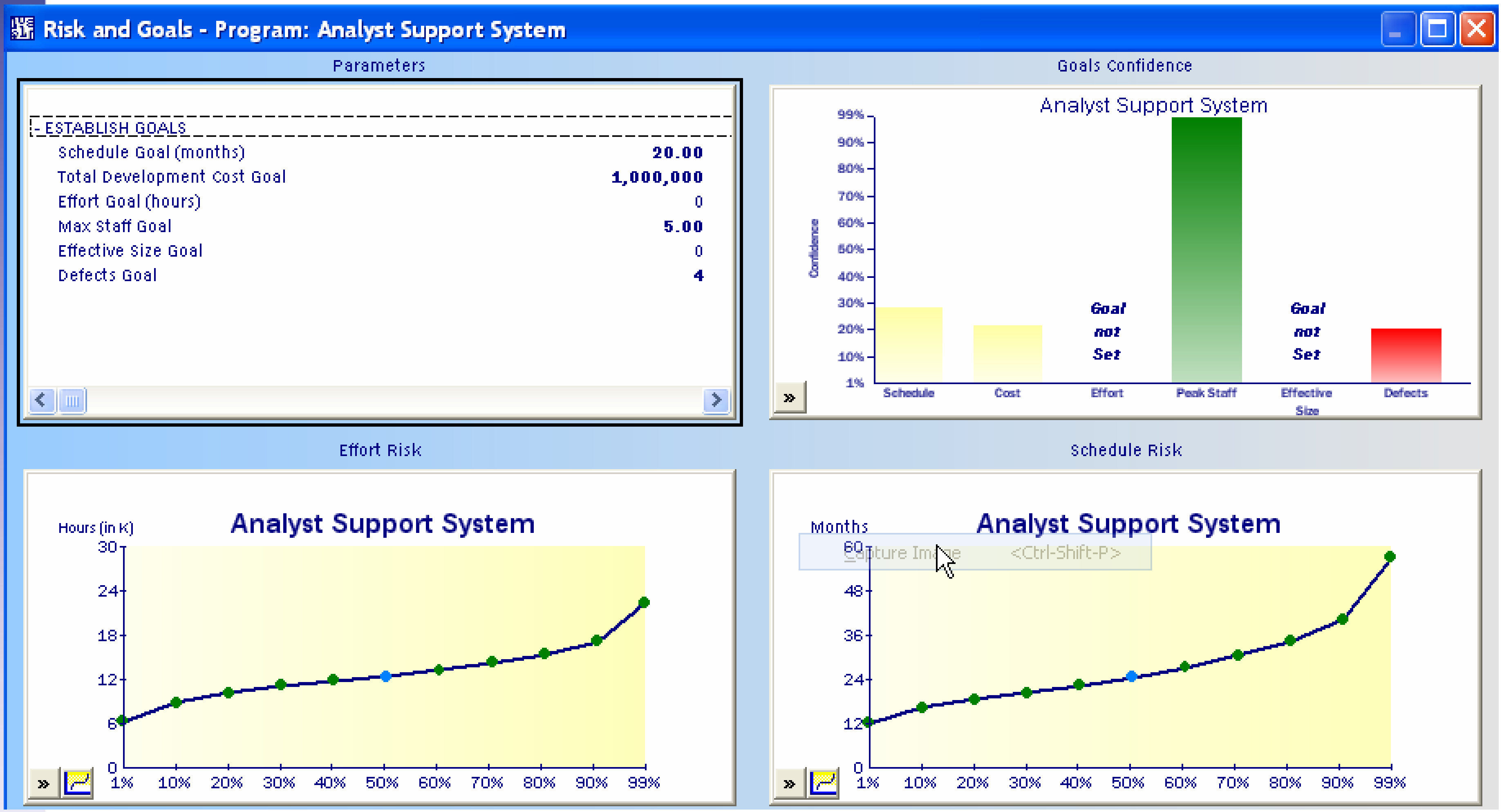
Update/Refine the Estimate

- **Optional or more specific information may become available in the future**
- **Follow configuration control guidelines to keep estimates straight**
  - Change log
  - Naming conventions
  - File storage
- **Maintain estimating integrity – eschew artificially driving the estimate to a predetermined result**

Step Six: Quantify Risks and Risk Analysis

- **Risk can produce loss of time, or quality, money, control, understanding, and so on.**
  - Loss associated with a risk is called the risk impact
- **Approximate the probability that the event will occur**
  - Risk probability: Likelihood the risk, measured from 0 (impossible) to 1 (certainty) When the risk probability is 1, the risk is a problem since it is certain to happen.
  - 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

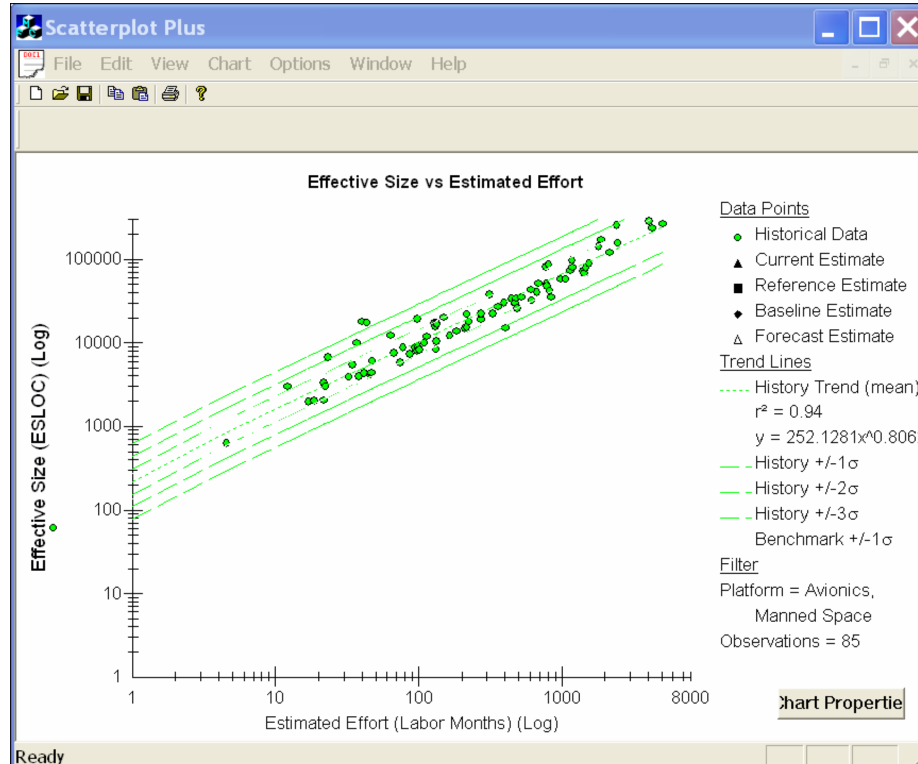


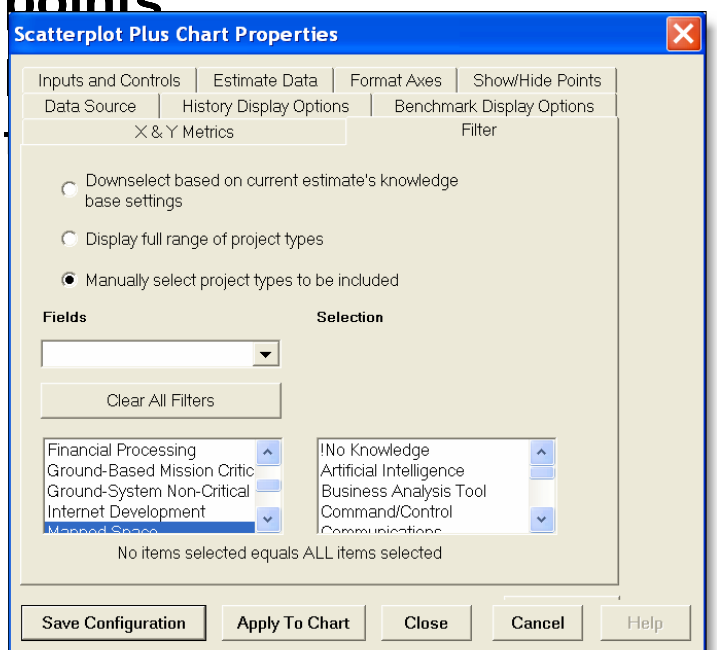
Step Seven: Estimate Validation and Review

- **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 the risks inherent in your 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

- Works with common repository
- Shows actual data, ranges, and correlations
- Plots SEER-SEM estimates and contrasts with data points





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

Step Eight: Generate A Project Plan

- **includes allocating estimate cost & schedule & allocating the cost and schedule to a function and task-oriented work breakdown structure**
- **Issues**
  - Inexperienced managers may not properly evaluate decisions effects over the long run.
    - lack necessary information
    - believe time to develop the information will make the project suffer
  - Other managers make decisions based on what they think higher management wants to hear
  - Good manager understands project realities even if it is not what higher management wants.
    - Job is to explain the reality in language his managers can understand
    - Problem managers although they may mean well, either lead a project to an unintended conclusion or, worse, drift down the road to disaster.
- **Software management and planning problems have been recognized for decades as the leading causes of software project failures**
  - bad management decisions
  - Incorrect focus
  - destructive politics

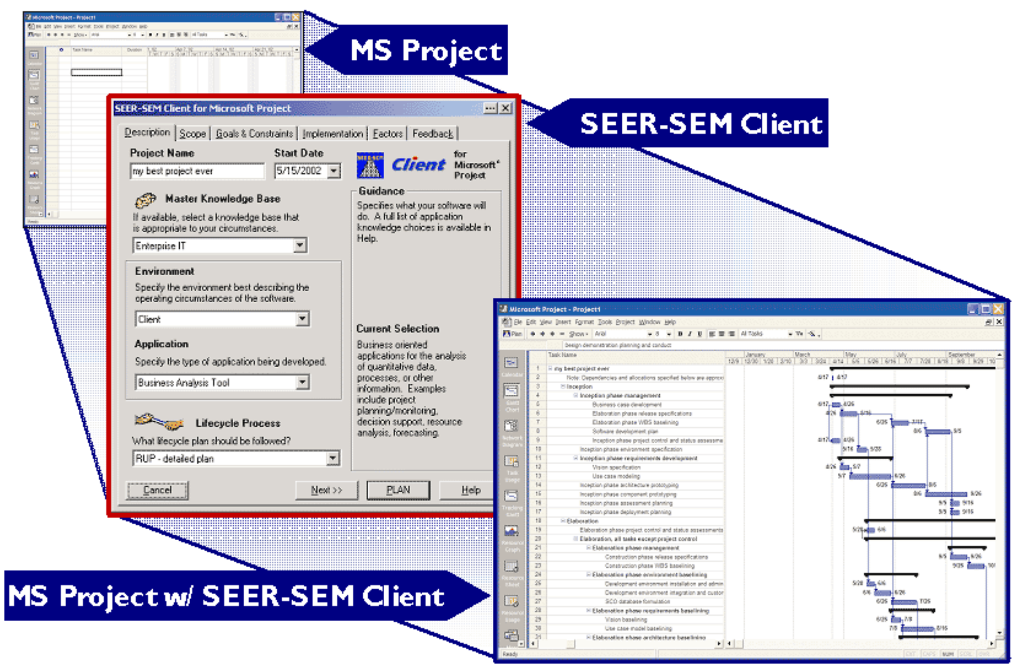
Revitalizing the Software Aspects of Systems Engineering

- **Memorandum of 13 June 2004 from Undersecretary of the Air Force: "Revitalizing the Software Aspects of Systems Engineering"**
- **Number 1 (of 10) recommendation: High Confidence Estimates**
  - "Estimate the software development and integration effort (staff hours), cost and schedule at high (80-90%) confidence."
- **•Number 2 recommendation: Realistic Program Baselines**
  - "Ensure cost, schedule and performance baselines are realistic and compatible. … The program budget must support the high confidence estimates for effort (staff hours), cost, and schedule."
- **Number 3 recommendation: Risk Management**
  - "Continuously identify and manage risks specific to computer systems and software as an integral part of the program risk management process. Ensure the risks, impact, and mitigation plans are appropriately addressed during program and portfolio reviews."

Expand Product WBS to Task level Plan

Transforms Microsoft Project into a powerful tool for planning software development projects.

- Automatically constructs a complete project plan
  - ***with relatively few inputs***
  - ***or directly from your SEER-SEM 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. By doing so, you will have evidence that your process was valid and that you generated the estimate in good faith, and you will have actual results with which to substantiate or calibrate your estimation models.
- **Be sure to document any missing or incomplete information and the 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 the completion of a project while the participants' memories are still fresh.**
- **Every software project should be used as an opportunity to improve the estimating process**

Step Ten: Track Project Throughout Development

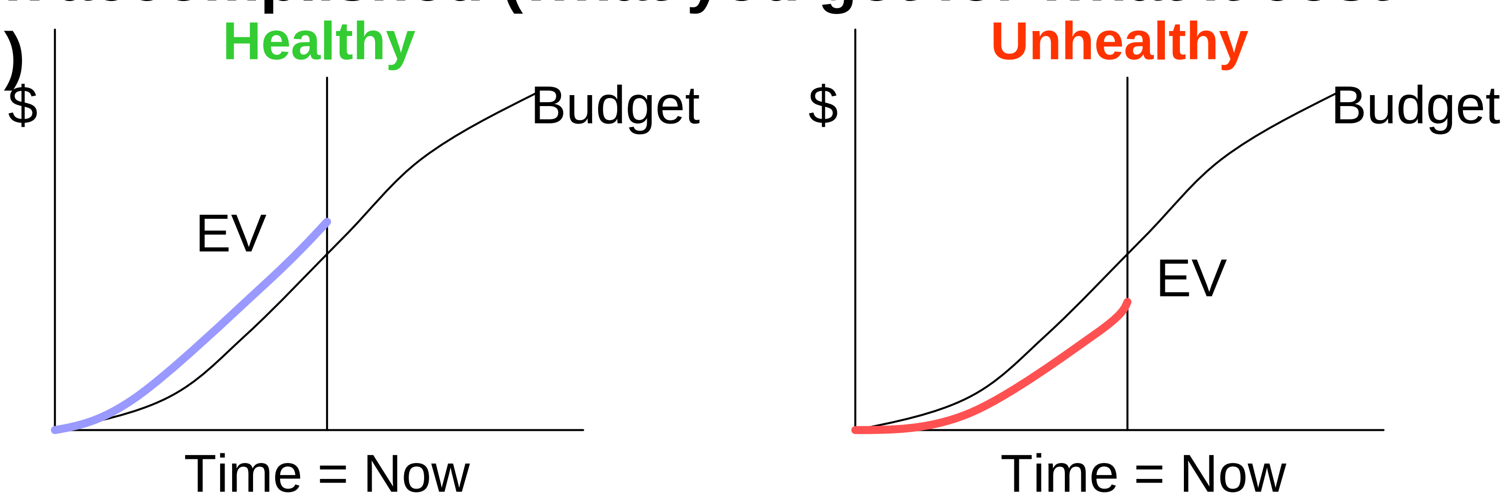
- ***Refining Estimates throughout Project***
- **Once a project has started, use estimates as a basis for performance measurement & project control**
- **Monitor actual effort & duration of tasks and/or phases against planned values to ensure you have the project under control**
- **Applying earned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

Earned Value as a Root Level Software Management Technique

- **Many think of earned value as a high level program control metric**
- **But Earned value analysis is most valuable when the units of work are small (1-2 week activities)**
- **Use binary earned value…**
  - Either not done (earned 0 value) or
  - Complete (earned full value) for the task
- **Galorath has used earned value at the root level as a core technique for saving failing software projects**

Use Earned Value To Quantify Progress Versus Effort

- **The main concern of EVM is 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)**

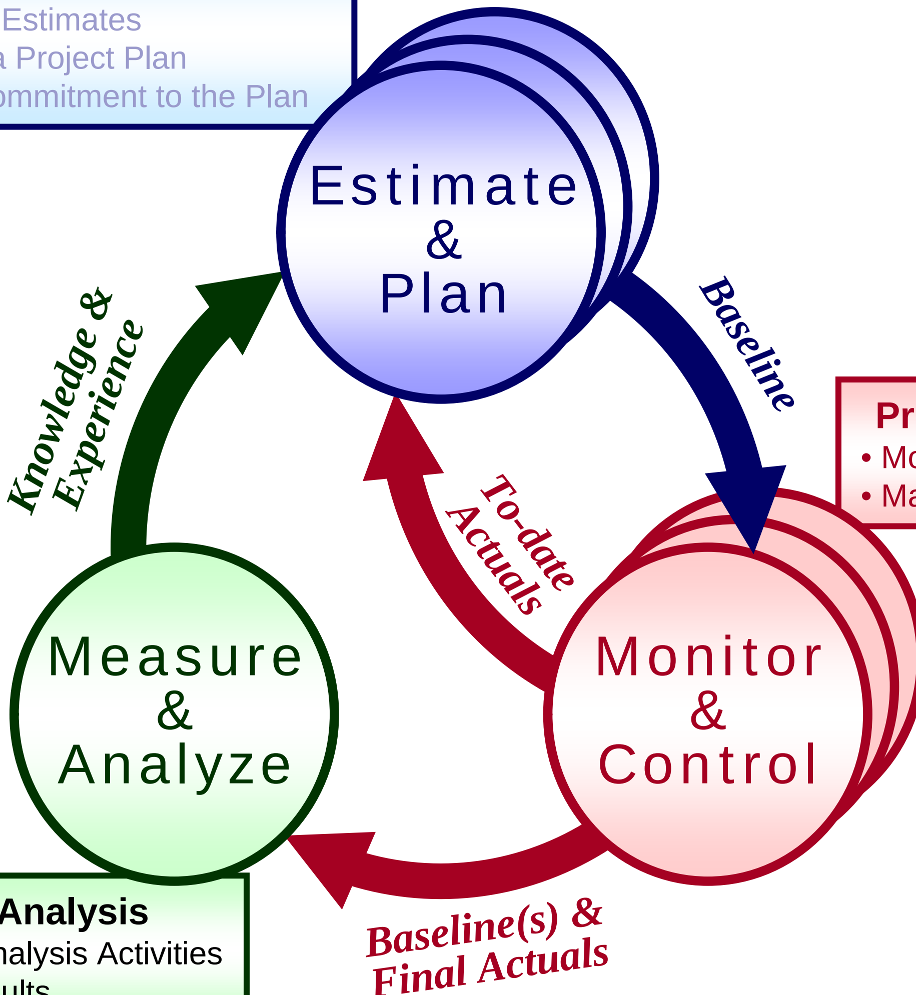


CMMI Process Areas For Planning, Monitoring & Control

SEER by GALORATH

CMMI Key Process Areas Require Estimation With Feedback

Project Planning
- Establish Estimates
- Develop a Project Plan
- Obtain Commitment to the Plan



Project Monitoring and Control
- Monitor Project Against Plan
- Manage Corrective Action to Closure

Measurement and Analysis
- Align Measurement and Analysis Activities
- Provide Measurement Results

✓**Foundation of Risk Management &**
✓**Quantitative Project Management**

The CMMI Process Areas
Source: CMMI Tutorial

Category	Process Area
Project Management	1. Project Planning 2. Project Monitoring and Control Supplier Agreement Management Integrated Project Management(IPPD) Integrated Supplier Management (SS) Integrated Teaming (IPPD) 3. Risk Management 4. Quantitative Project Management
Support	Configuration Management Process and Product Quality Assurance 5. Measurement and Analysis Causal Analysis and Resolution Decision Analysis and Resolution Organizational Environment for Integration (IPPD)
Engineering	Requirements Management Requirements Development Technical Solution Product Integration Verification Validation
Process Management	Organizational Process Focus Organizational Process Definition Organizational Training Organizational Process Performance Organizational Innovation and Deployment

We will concentrate on:

1. Project Planning
2. Project Monitoring & Control
3. Measurement & Analysis
4. Risk Management
5. Quantitative Project Mgmt

SEER-PPMC Parametrics Combined With Traditional EVM Captures Software Project Trouble More Completely

SEER-PPMC: Performance Mode Significant Overrun Aparrent

Data Value	Baseline Estima		
Development Schedule Months		17.97	12%
Development Effort Months		108.57	25%
Development Effort Hours		16,502	25%
Development Base Year Cost	1,2	1,595,906	25%
Start Date	2,	2/02/04	
Completion Date	6,	7/31/05	
Effective Lines Only			
Effective Functions Only			
Effective Size			
PERFORMANCE INDICES			
Cost Performance Index (CPI)	0.80		
Schedule Performance Index (SPI)	0.81		
Composite Performance Index (XPI)	0.81		
Time Performance Index (TPI)	0.90		
VARIANCE			
Cost Variance (CV) (Hours)	-1,038		
Schedule Variance (SV) (Hours)	-968		
Time Variance (TV) (Months)	-0.97		
ACCOMPLISHMENTS			
Hours thru Latest Snapshot	5,200		
Cost thru Latest Snapshot	502,895		
Earned Value thru Latest Snapshot	4,162		
Earned Value % thru Latest Snapshot	31.51%		
Planned Hours thru Latest Snapshot	5,131		
Projected Productivity Lines/PM	61.94		
Projected Productivity Functions/PM	0.7		

PPMC Options

Terminology: Standard

Forecasting Method: Performance

Show Baseline WBS Elements

Show Current Estimate in PPMC Charts

PPMC Report Options | OK | Cancel

SEER-PPMC: Basic (EVM) Mode Overrun Not as Apparent

Data Value	Baseline Estimat		
Development Schedule Months		16.77	4%
Development Effort Months		93.73	8%
Development Effort Hours	1	14,247	8%
Development Base Year Cost	1,27	1,377,824	8%
Start Date	2/	2/02/04	
Completion Date	6/		
Effective Lines Only	1,602		
Effective Functions Only	78		
Effective Size	6,675		
PERFORMANCE INDICES			
Cost Performance Index (CPI)	0.80		
Schedule Performance Index (SPI)	0.81		
Composite Performance Index (XPI)	0.81		
Time Performance Index (TPI)	0.90		
VARIANCE			
Cost Variance (CV) (Hours)	-1,038		
Schedule Variance (SV) (Hours)	-968		
Time Variance (TV) (Months)	-0.97		
ACCOMPLISHMENTS			
Hours thru Latest Snapshot	5,200		
Cost thru Latest Snapshot	502,895		
Earned Value thru Latest Snapshot	4,162		
Earned Value % thru Latest Snapshot	31.51%		
Planned Hours thru Latest Snapshot			
Projected Productivity Lines/PM			
Projected Productivity Functions/PM			

PPMC Options

Terminology: Standard

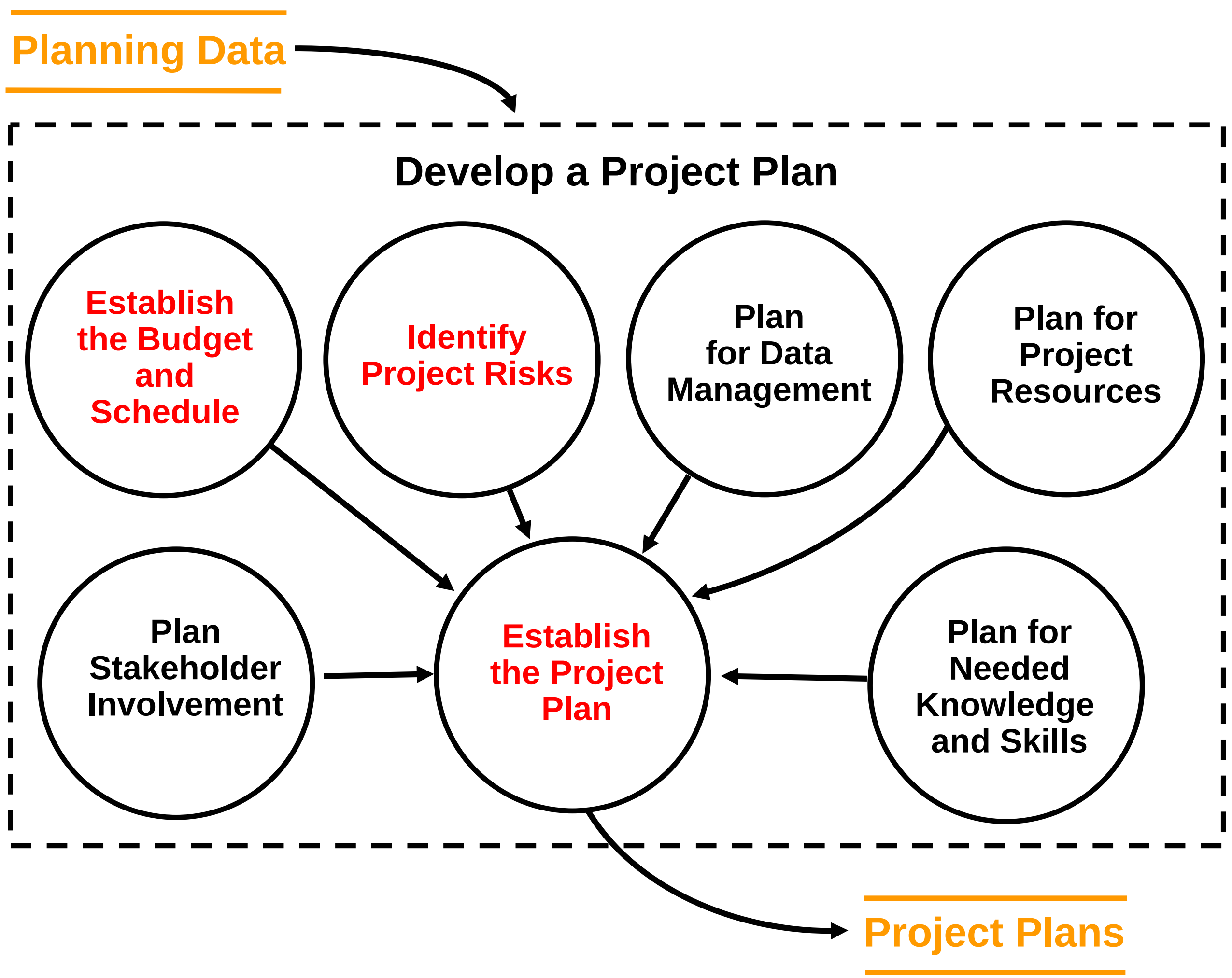
Forecasting Method: Basic

Show Baseline WBS Elements

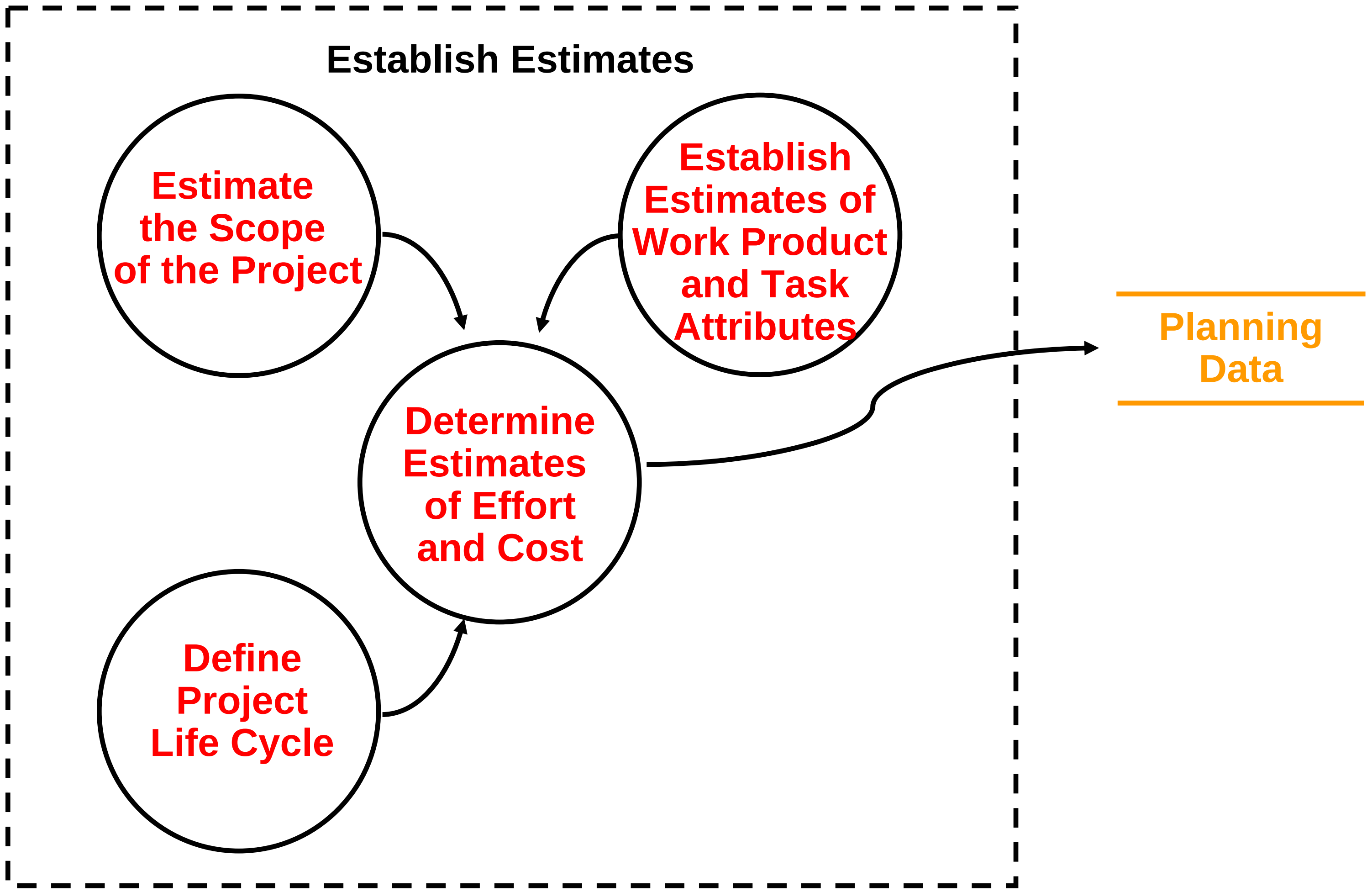
Show Current Estimate in PPMC Charts

PPMC Report Options | OK | Cancel

CMMI Project Planning Requires Schedule & Effort Estimation & Planning – Context Source CMMI Tutorial

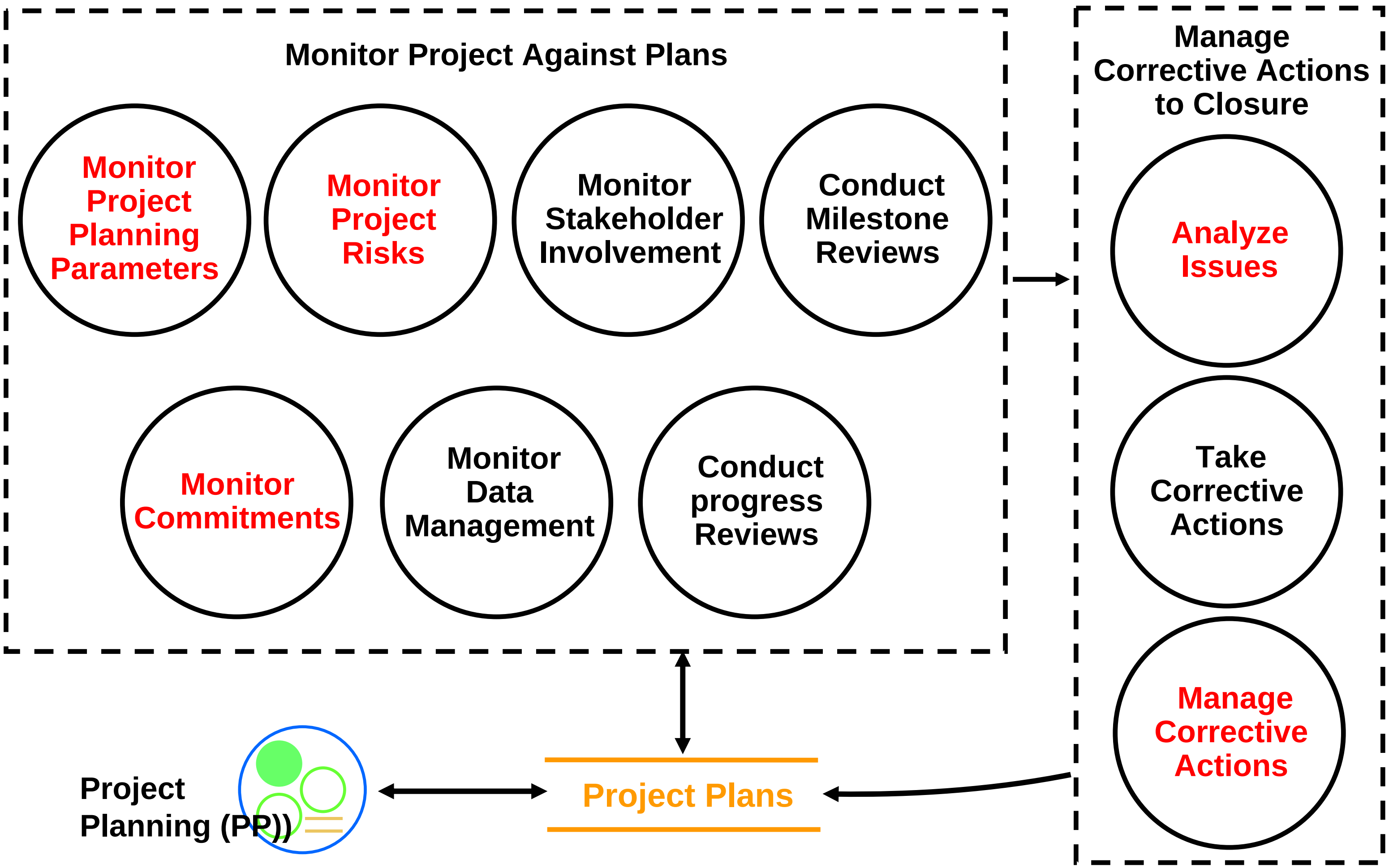


CMMI Estimation Requires Effort, Schedule, Scope – Context Source CMMI Tutorial



CMMI Project Monitoring & Control – Context

Source CMMI Tutorial



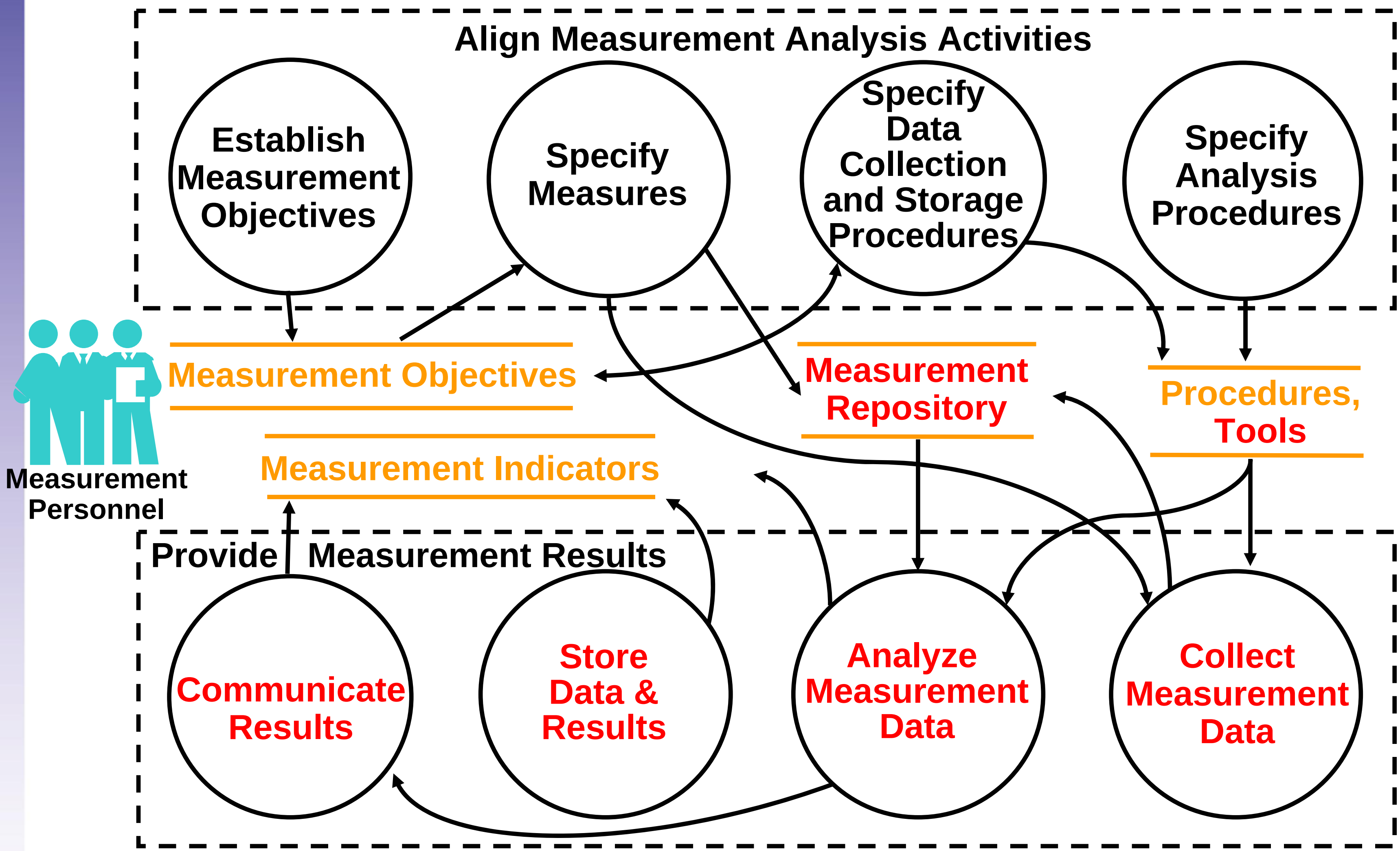
The First Step To EVM Is A Workable Plan for Cost & Schedule

"…Using EVM properly puts a considerable burden on team members to plan and schedule projects -- skills often in short supply at federal agencies"

… Norman Enger, OMB

http://www.fcw.com/article91403-11-14-05-Print

CMMI Measurement & Analysis – Context: Source CMMI Tutorial



SEER® by GALORATH

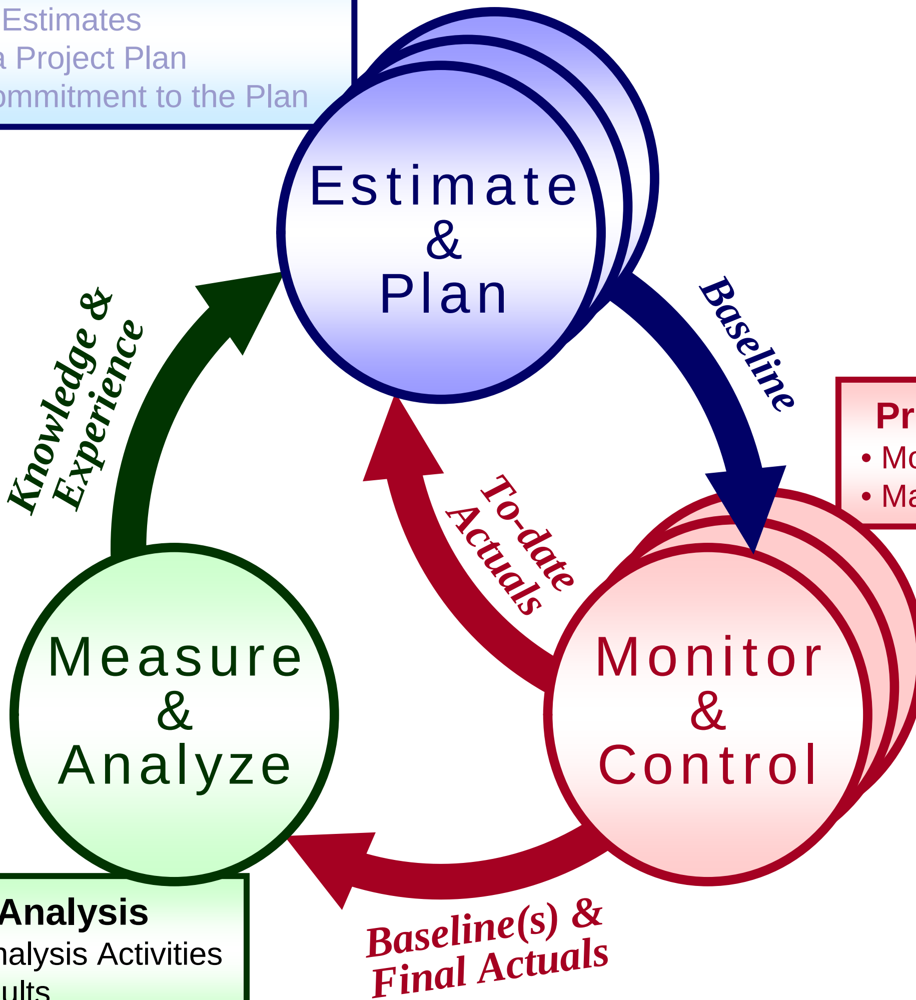
Risk Management Source: CMMI Tutorial

- **Purpose:**
  - Identify potential problems before they occur, so that risk handling activities may be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives
- **SEER Provides the cost & schedule impacts of such risk**
- **According to Mike Evans "We can tell them they have a problem. SEER can quantify what the impact is"**

SEER® by GALORATH

CMMI Key Process Areas Require Estimation With Feedback

Project Planning
- Establish Estimates
- Develop a Project Plan
- Obtain Commitment to the Plan



Project Monitoring and Control
- Monitor Project Against Plan
- Manage Corrective Action to Closure

Measurement and Analysis
- Align Measurement and Analysis Activities
- Provide Measurement Results

✓**Foundation of Risk Management &**
✓**Quantitative Project Management**

Performance Measurement: Earned Value Analysis Can Make Projects More Successful

- **"Earned Value Analysis" is an industry standard way to:**
  1. measure a project's progress
  2. forecast its completion date and final cost
  3. provide schedule and budget variances along the way.

- **Provides consistent, numerical indicators with which you can evaluate and compare projects**

- **Earned value (Performance Measurement) techniques are just as viable using effort for root level project management**
  - Effort Value – units of labor; e.g.:
    - person-hours, staff-hours, effort-hours, labor-hours
    - person-months, staff-months, effort-months, labor-months

Earned Value Measures Support CMMI Tracking & Control

Source: PMBOK & Galorath

- *Baseline Budget* **– Budget at Completion (BAC)**
- *Planned Value* – **Budgeted Cost of Work Scheduled (BCWS aka PV)**
  - Planned Value: Budgeted cost for work scheduled to be completed up to a given point in time
- *Earned Value* – **Budgeted Cost of Work Performed (BCWP aka EV)**
  - Progress: Budgeted value for work actually completed during a given time period.
- *Spent* – **Actual Cost of the Work Performed (ACWP aka AC)**
  - Actual cost (AC) - Total cost accomplishing work during a given time period.

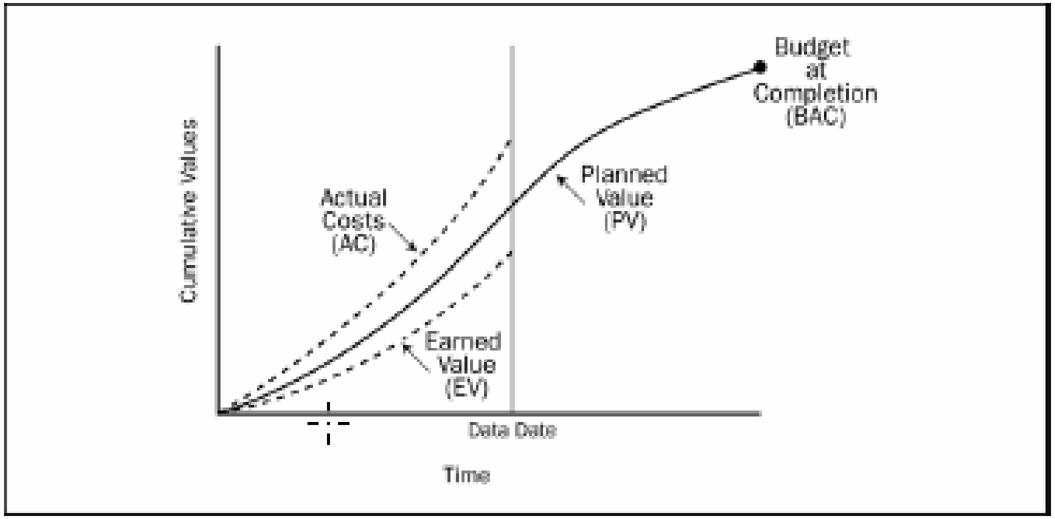
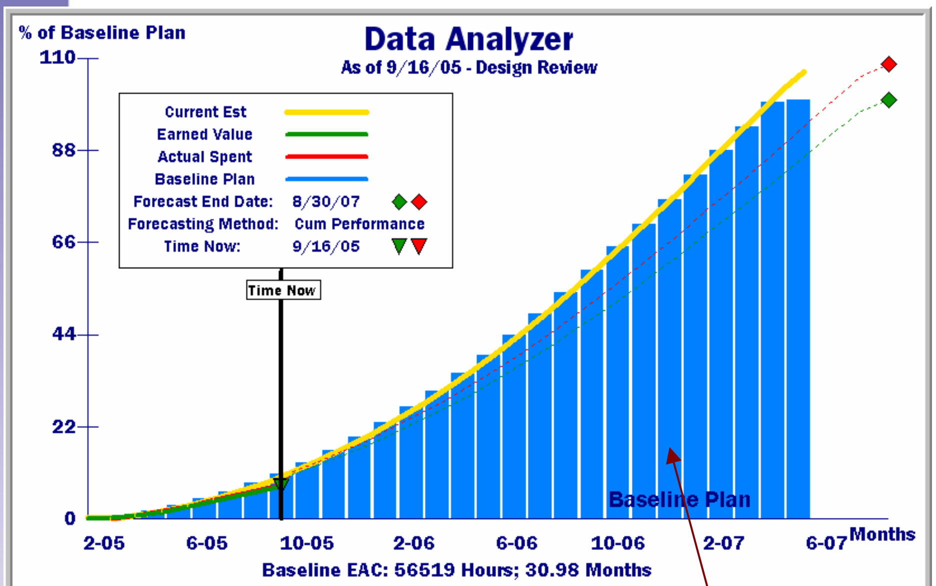


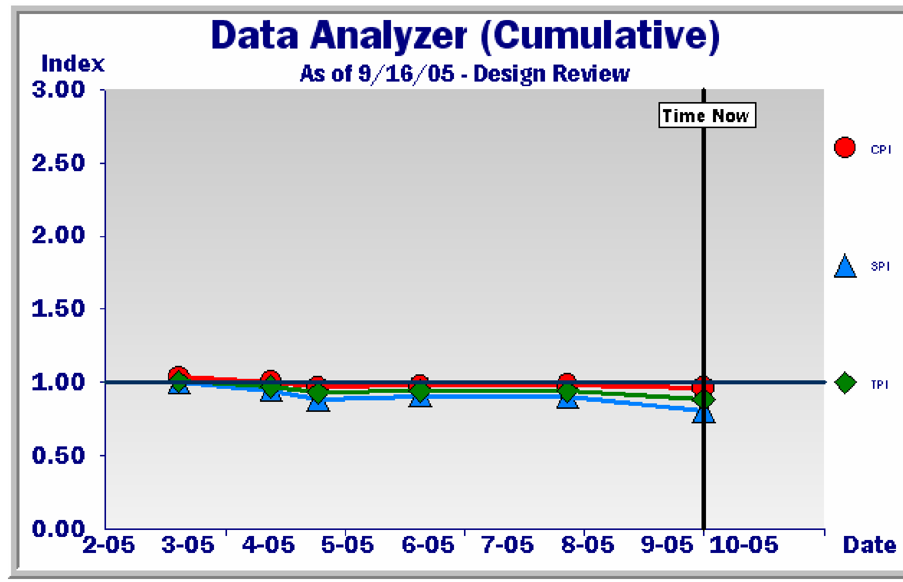
Figure 7-7. Illustrative Graphic Performance Report

Looking at Progress Over Time

See how actual and earned effort measure up to the baseline plan

See CPI, SPI and TPI trends over time





Month	19	
Date	Jan-06	
METRIC	BASELINE	CURRENT
Plan % comp	54.6 %	54.6 %
Earned Value %	47.1 %	47.1 %
Actual (spent) %	56.7 %	56.7 %
Total Hours	51,165.4	
Schedule	26.9	
Constraint	MIN TIME	

Click on any month to see specific information

CPI = Cost Performance Index;

The ratio of earned to actual cost of work performed

SPI = Schedule Performance Index;

The percent of the investment that has been completed

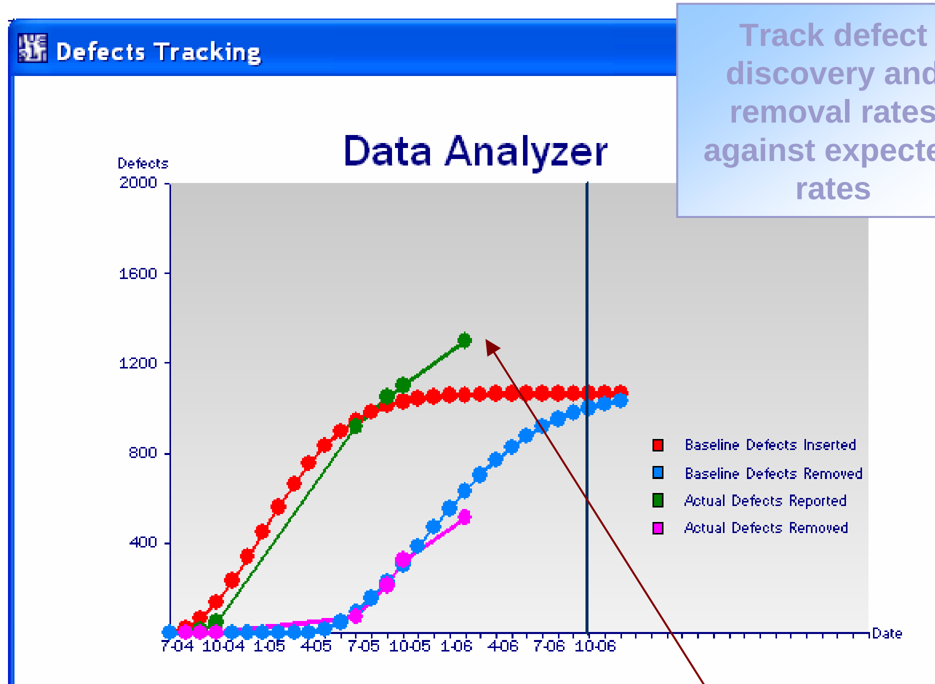
TPI = Time Performance Index;

The ratio of elapsed time from start to plan & start to time-now

Defects and Growth Impact Software Process

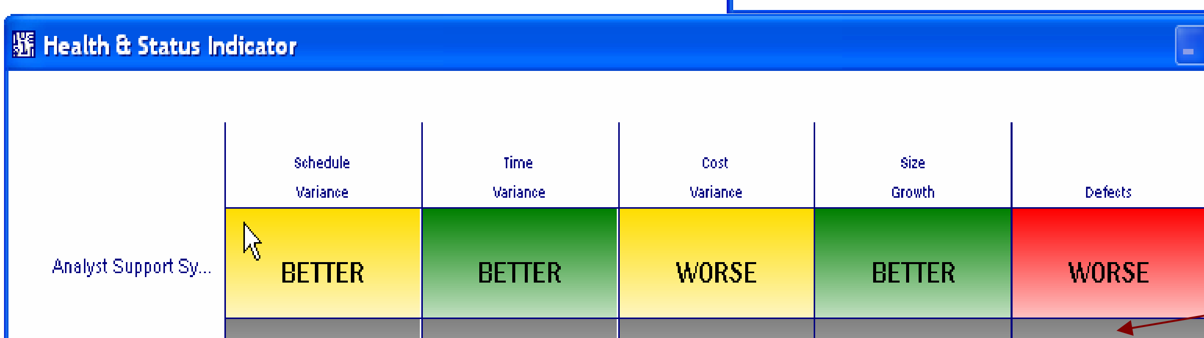
SEER® by GALORATH

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

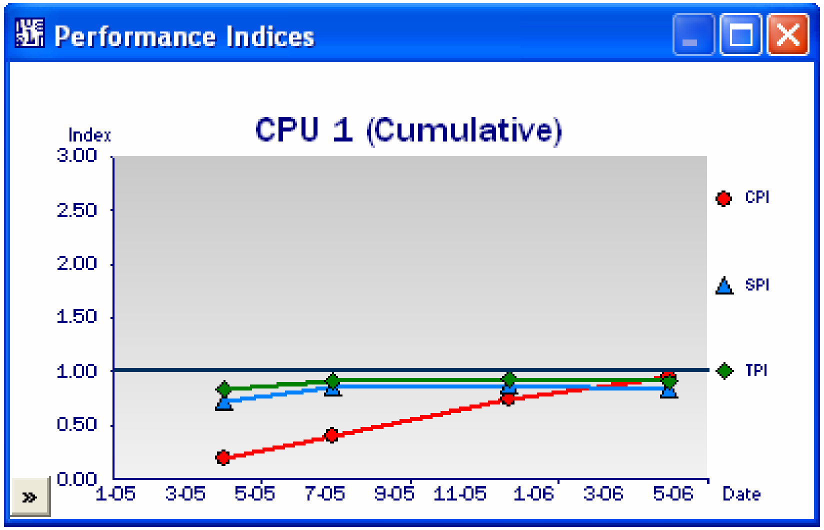
SEER® by GALORATH

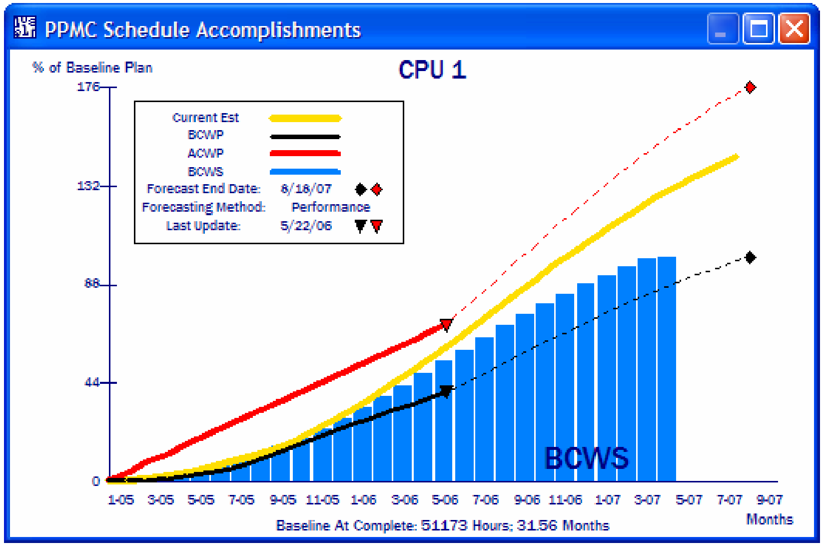
Parametric Project Monitoring & Control

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

- Adds Performance Measurement (*Earned Value*) methods to SEER-SEM
- 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

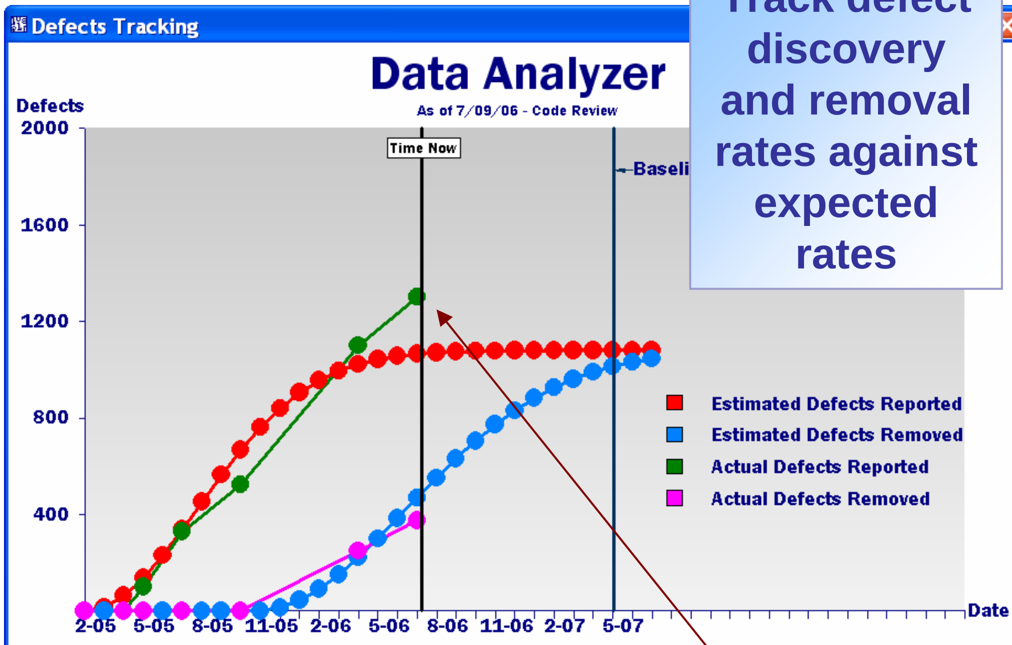
	Earned Value	Schedule	Cost Variance	Size Growth	Defects Outstanding
CPU 1	WORSE	WORSE	BETTER	NO CHANGE	NO CHANGE
	N/A	N/A	N/A	N/A	N/A
	N/A	N/A	N/A	N/A	N/A
	N/A	N/A	N/A	N/A	N/A





Defects & Size Growth Are Additional Core Metrics

SEER® by GALORATH



Track defect discovery and removal rates against expected rates

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

- Including Size Growth and Defect Discovery/Removal Rate
- User defined control limits to control the transition between red-yellow-green

Increased defect reporting rate shows a worsening trend

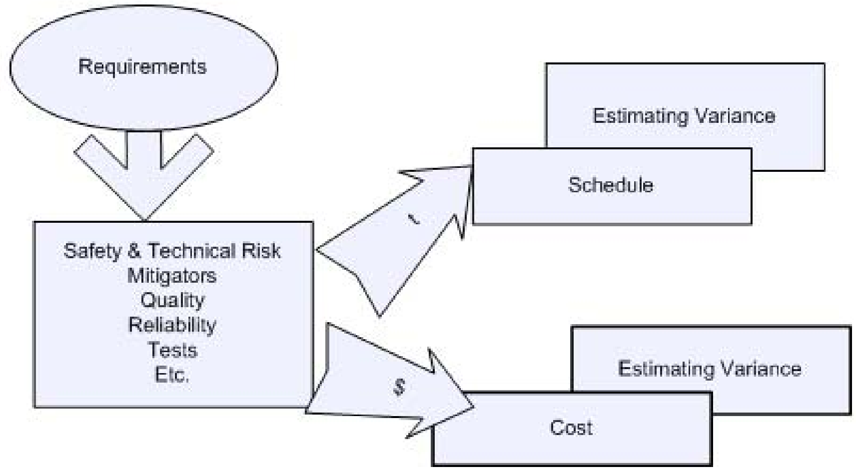
Health & Status Indicator

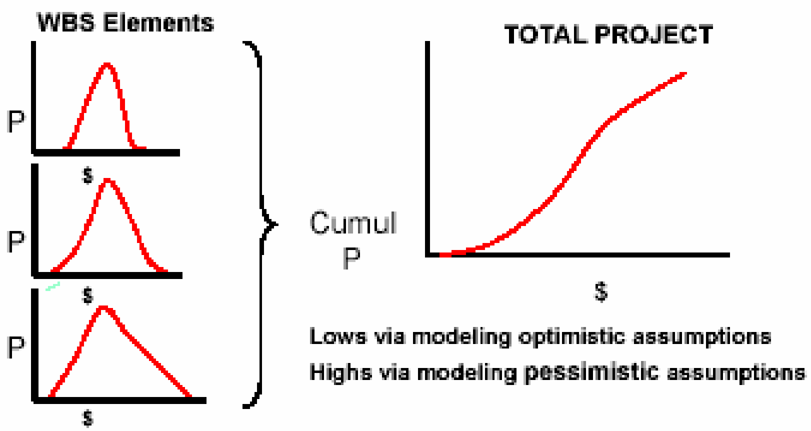
	Schedule Variance	Time Variance	Cost Variance	Size Growth	Defects
Data Analyzer	WORSE	WORSE	WORSE	WORSE	WORSE

"Dollarizing" Risks
Source J. Hamaker

Risk Areas

- Technical / Performance / Engineering
- Schedule
- Cost
- Safety



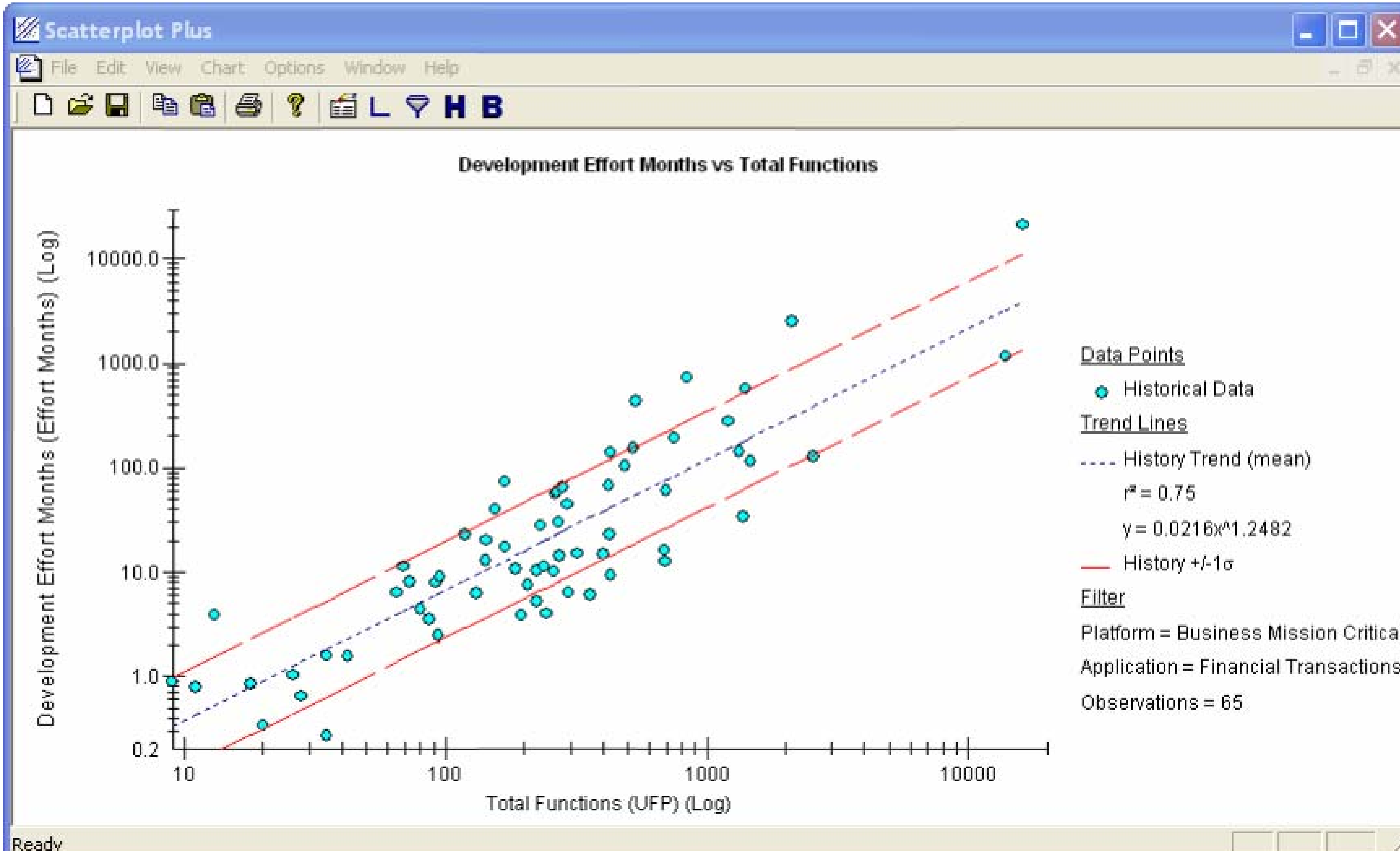


- **Estimates are really probability distributions**
- **Sources of uncertainty**
  - Project uncertainties
  - Model variance
  - Cost / risk analysis quantifies budget / plan with acceptable confidence levels

Parametric Modeling Supports CMMI

- **Parametric Cost Estimating Methods**
  - Estimating methods based on physical or performance characteristics and schedules of the end items
- **Parametric Estimation Model Definition**
  - Mathematical representation of parametric estimating relationships that provides logical, predictable correlation between the physical or functional characteristics of a system, and the resultant cost, schedule, and other attributes of the system.
    - Comprised of cost estimating relationships (CERs) and other parametric estimating functions, e.g., cost quantity relationships, inflation factors, staff skills, schedules, etc.
    - Provides logical repeatable relationship between input variables and resultant cost

Example Simple Cost Estimating Relationship (CER) Shows Effort Based On Size



Uncertainty and Risk
(Source Q. Redman)

- **A Point Estimate is never correct!**
- **There is always potential variability in inputs.**
- **But…**
  - 26% of program managers do not accept risk assessment at all, not even “slightly”.*
  - 44% of risk ranges are intuitive judgments, without historical data or guided-survey.*

	Low	Nominal	High
Advanced Development - Total	**68,348**	**97,132**	**257,492**
Government In-House Costs @ 35%	17,720	25,182	66,757
Advanced Development - Contractor	**50,628**	**71,950**	**190,735**
Program Management	2,448	5,103	19,100
System Engineering	4,805	7,731	33,425
Design Engineering	4,526	9,980	33,193
Hardware Manufacturing (Prototype)	24,150	**25,421**	27,963
Software Development	14,064	**14,805**	16,285
Tooling and Test Equipment	45	1,034	13,847
System Integration	0	1,880	11,937
System Test and Evaluation	453	3,541	20,532
Data	136	1,013	6,526
Training	0	288	2,547
Facilities	0	627	2,865
Spares	0	528	2,515
	Low	Nominal	High
System Develop. & Demo. - Total	**184,881**	**273,557**	**361,292**
Government In-House Costs @ 35%	47,932	70,922	93,668
System Develop. & Demo. - Contractor	**136,949**	**202,635**	**267,623**
Program Management	2,448	17,285	19,100
System Engineering	13,155	17,547	33,425
Design Engineering	16,093	21,952	33,193
Hardware Manufacturing (Prototype)	85,865	**90,385**	99,423
Software Development	18,753	**19,740**	21,714
Tooling and Test Equipment	45	2,866	13,847
System Integration	0	6,368	11,937
System Test and Evaluation	453	11,993	20,532
Data	136	3,432	6,526
Training	0	6,623	2,547
Facilities	0	2,123	2,865
Spares	0	2,323	2,515
	Low	Nominal	High
Procurement - Total	**2,149,874**	**3,162,081**	**6,722,475**
Government In-House Costs @ 25%	429,975	632,416	1,344,495
Procurement - Contractor	**1,719,899**	**2,529,665**	**5,377,980**
Program Management	36,007	116,424	551,182
System Engineering	18,003	98,129	367,455
Design Engineering	0	28,274	124,935
Hardware Manufacturing	1,497,878	**1,576,714**	1,734,385
Software Development	77,994	**82,099**	90,309
Tooling and Test Equipment	18,003	83,576	367,455
System Integration	18,003	107,276	367,455
System Test and Evaluation	23,404	144,698	492,389
Data	3,601	33,680	110,236
Training	9,002	30,769	80,840
Facilities	0	7,484	36,745
Spares	18,003	142,703	551,182
Operational Site Activation	0	77,838	503,413
	Low	Nominal	High
O&S Costs - Partial Total	4,241,190	5,173,716	8,377,451
Fielding	747,220	913,742	1,485,838
Sustainment (O&S)	2,689,990	3,289,472	5,349,016
Spares and Repair Parts	747,220	913,742	1,485,838
Personnel	56,760	56,760	56,760

*U.S. Aerospace Risk Analysis Survey, Hollis M. black, III, SCEA Estimator October 1999

7 Characteristics of a Dysfunctional Software Project (Source: Mike Evans, et al.)

- Failure to Apply Essential Project Management Practices 57%
- Unwarranted Optimism and Unrealistic Management Expectations 41%
- Failure to Implement Effective Software Processes 30%
- Premature Victory Declarations 20%
- Lack of Program Management Leadership 13%
- Untimely Decision-Making 8%
- Lack of Proactive Risk Management 3%

Estimation Lessons Learned

- **The longer the development schedule the higher the risk of key personnel turnover**
- **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
- **Measure early**
  - Before trouble
  - Early milestones that are under-budget are not necessarily a good sign
  - Programs that skimp on the early, upfront planning; requirements and design work, will most likely be in trouble later.
- **Follow known, proven development processes**
- **Estimation, planning, tracking, controlling – then using the information to do better next time**
- **Recognize that you have a problem**