

estimate

estimate • analyze • plan • control

Controlling Software Costs: Development is (Only) Job One

Dan Galorath

galorath@galorath.com

310 414-3222 x614

Blog: www.galorath.com/wp



Delusions of Success: How Optimism Undermines Executives' Decisions (Source: Richard Hartley, HBR)

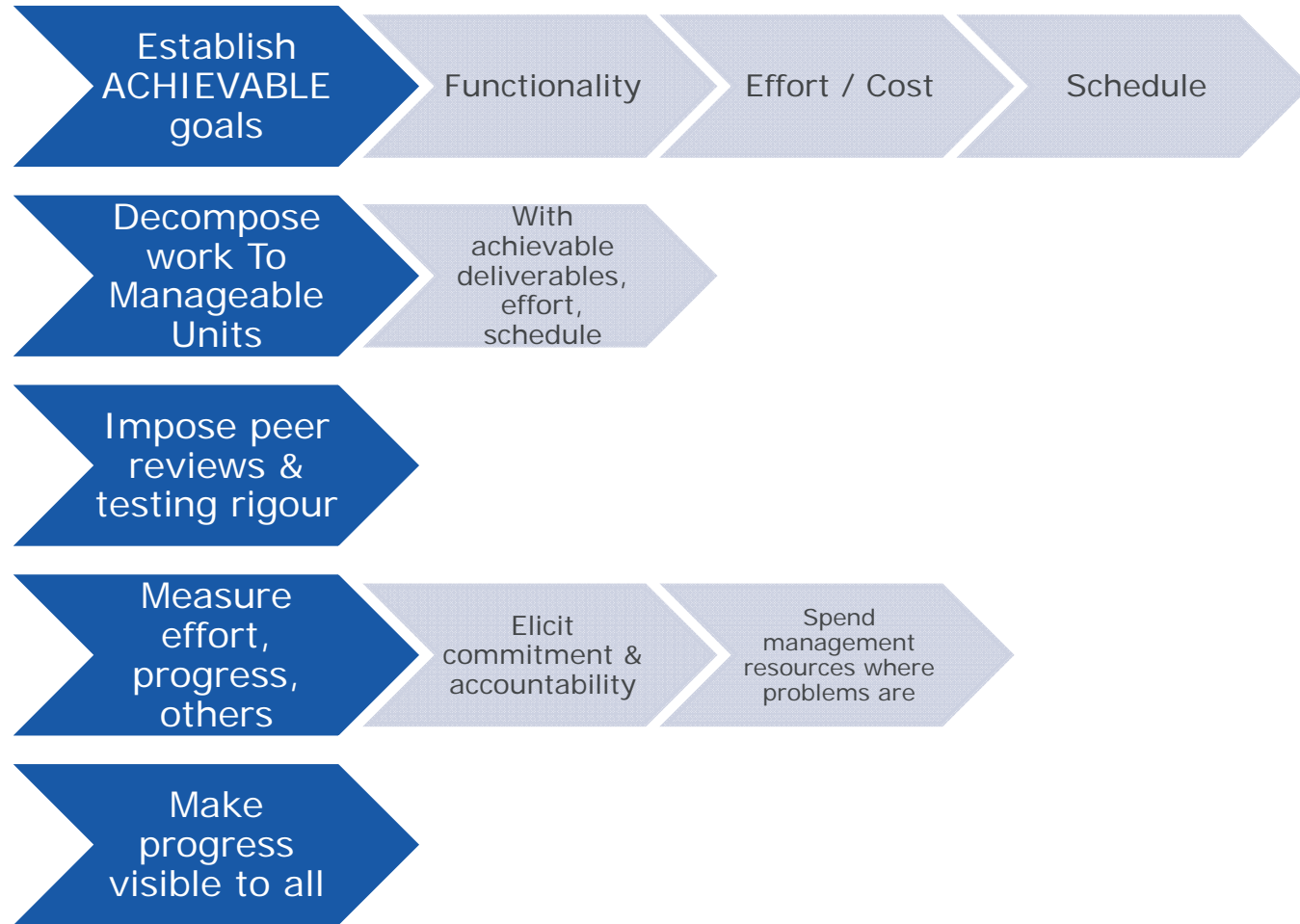


Humans seem to be hardwired to be optimists

- Routinely exaggerate benefits & discount costs
- Optimism traced to cognitive biases and organizational pressures
 - People highly optimistic most of the time
 - exaggerate talents and degree of control
 - Attribute negative consequences to external factors
 - Competition (\$, time) for new projects
- Anchoring (relying too heavily on one piece of information) magnifies optimism
 - Initial plan accentuates positive, skews subsequent analysis
 - Most pronounced for new initiatives
- Temper with “outside view”
 - Supplements traditional forecasting w/ statistical analysis of analogous efforts
 - Don't remove optimism, but temper effects
 - balance between optimism and realism
 - goals (motivate) and forecasts (decide whether to make commitment in the first

“When pessimistic opinions are suppressed, while optimistic ones are rewarded, an organizations ability to think critically is undermined”

Software Management In a Nutshell (New Or Maintenance Project)



estimate

estimate • analyze • plan • control

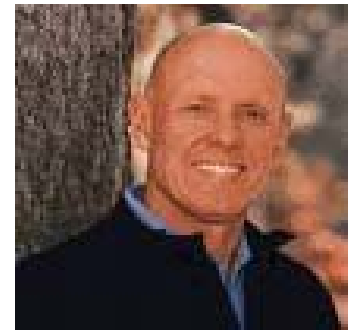
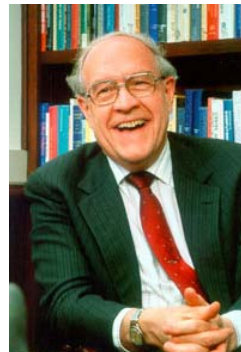
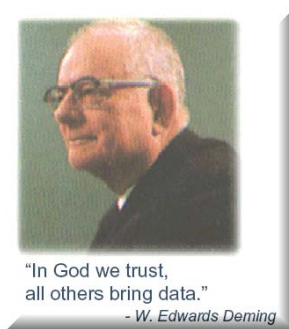
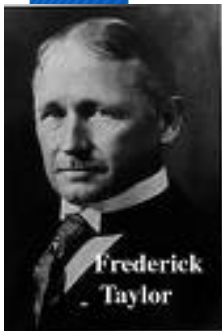
Estimation & Estimation Process



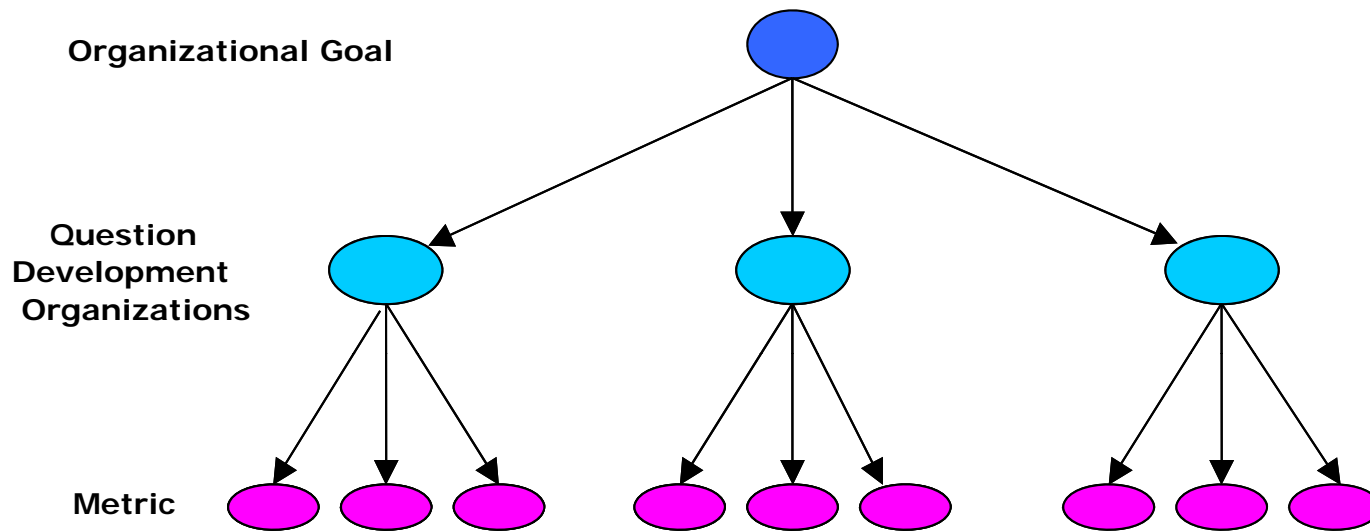
Measurement Ideas From The Pioneers



- **Frederick Taylor:** The Principals of Scientific Management
1901 "Let data and facts do the talking"
- **W. Edwards Demming:** "In God We Trust... All Others Bring Data"
- **Frederick Brooks:** "There is an incremental person when added to a software project that makes it take longer"
- **Ed Yourdon:** "Avoiding Death Marches in Software Projects"
- **Steven Covey:** "Sharpen the Saw" Focus on improvement
- **Eli Goldratt:** Improvements should increase profit Effectiveness



Choosing Appropriate Metrics with Goal Question Metric (GQM)



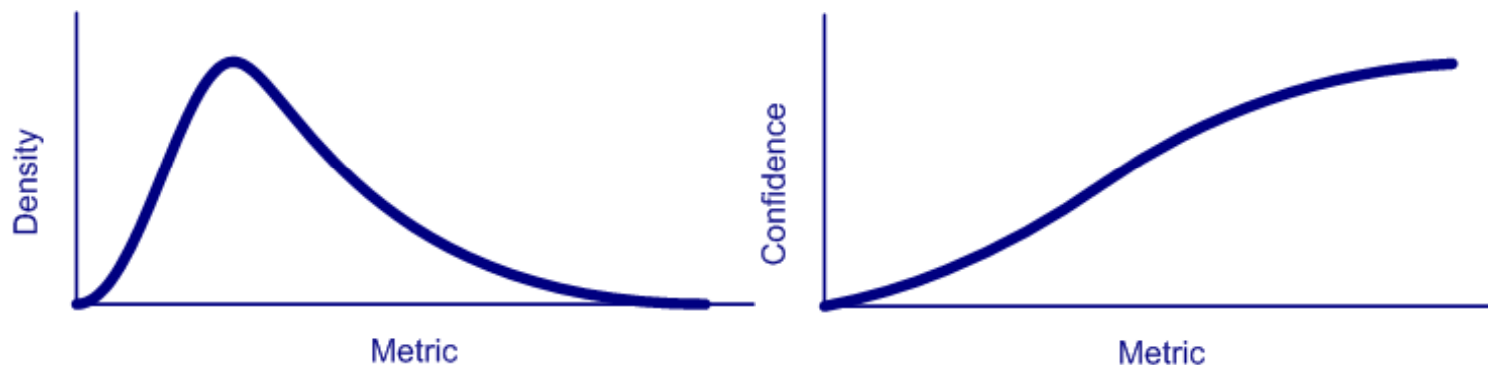
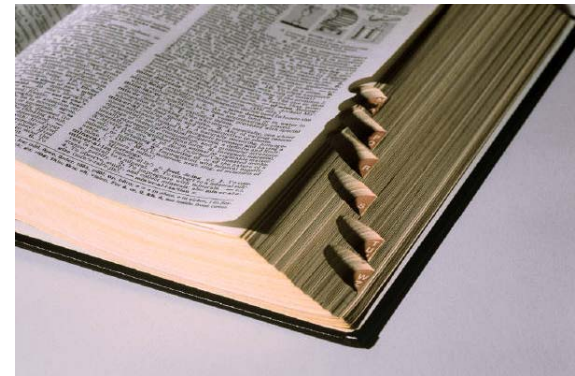
The question is not What metrics should I use so much as...

What do I want to know, motivate or learn?

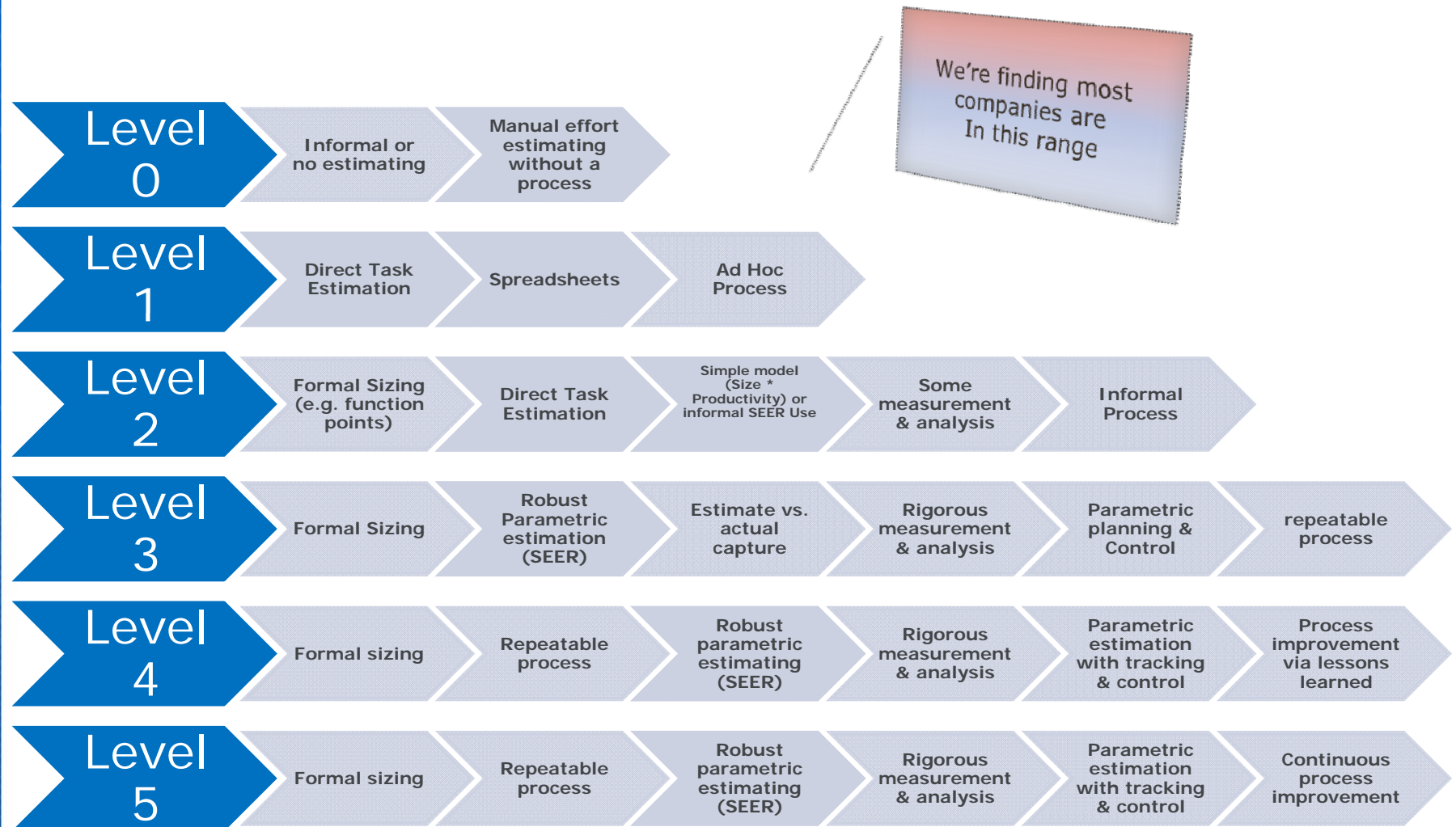
How will the data collected answer the question?

An Estimate Defined

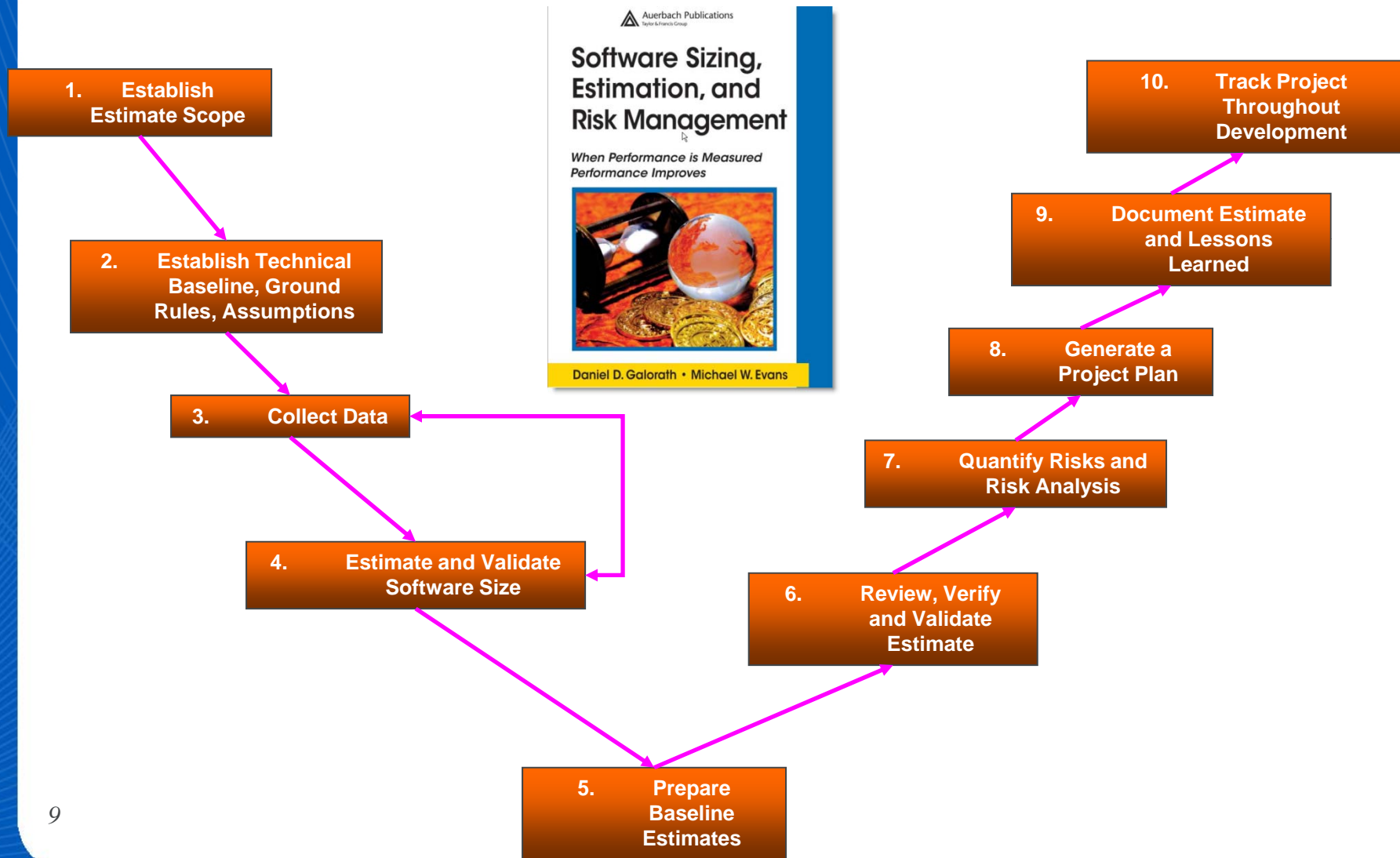
- An ***estimate*** is the most knowledgeable statement you can make ***at a particular point in time*** regarding:
 - Effort / Cost
 - Schedule
 - Staffing
 - Risk
 - Reliability
- Estimates more precise with progress
- ***A WELL FORMED ESTIMATE IS A DISTRIBUTION***



Estimation Organizational Maturity

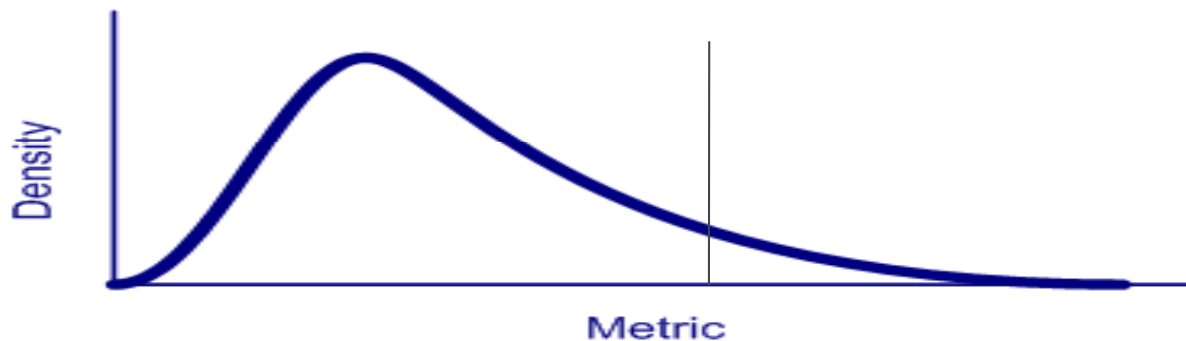


10 Step Software Estimation Process: Consistent Processes = Reliable Estimates



Manual Estimates: Human Reasons For Error (Metrics Can Help)

- 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



estimate

estimate • analyze • plan • control

Software Measurement

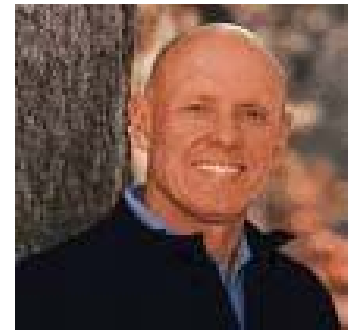
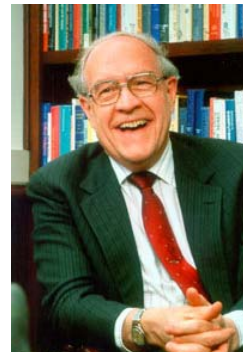
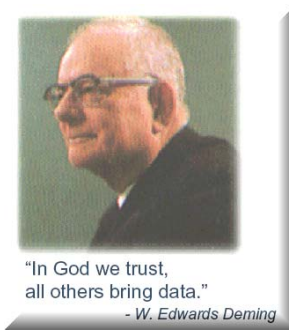
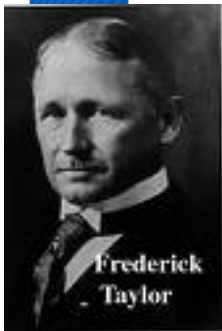


© 2008 Copyright Galorath

Some Measurement Heroes

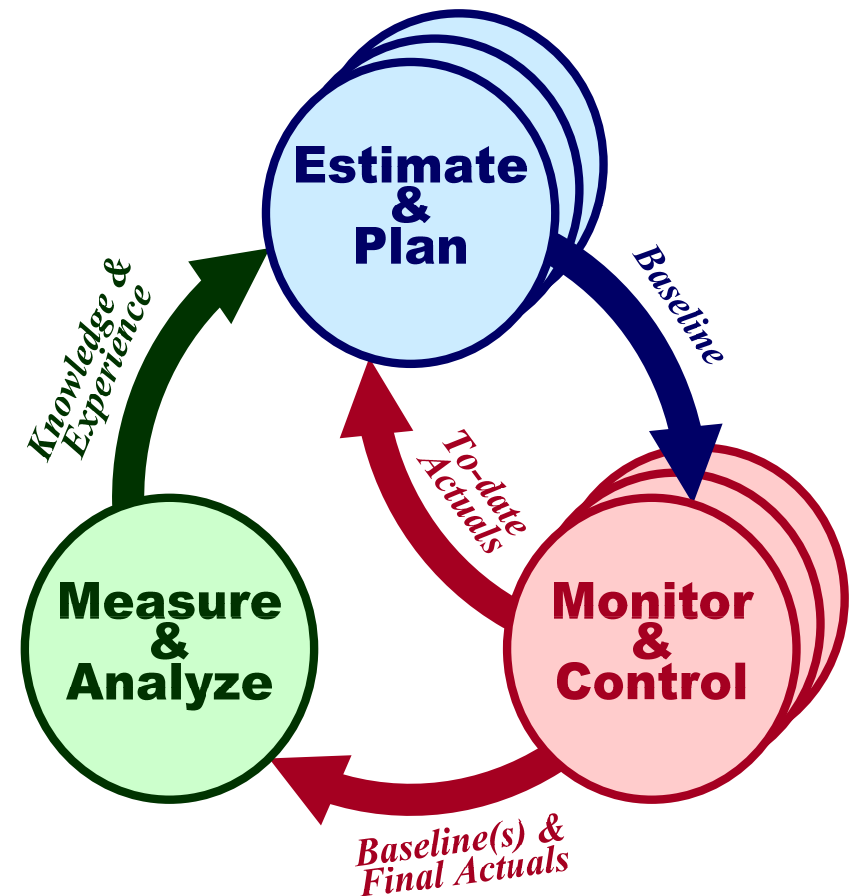


- **Frederick Taylor:** The Principals of Scientific Management
1901 "Let data and facts do the talking"
- **W. Edwards Demming:** "In God We Trust... All Others Bring Data"
- **Frederick Brooks:** "There is an incremental person when added to a software project that makes it take longer"
- **Ed Yourdon:** "Avoiding Death Marches in Software Projects"
- **Steven Covey:** "Sharpen the Saw" Focus on improvement
- **Eli Goldratt:** Improvements should increase profit Effectiveness



Supporting Successful Projects With Process

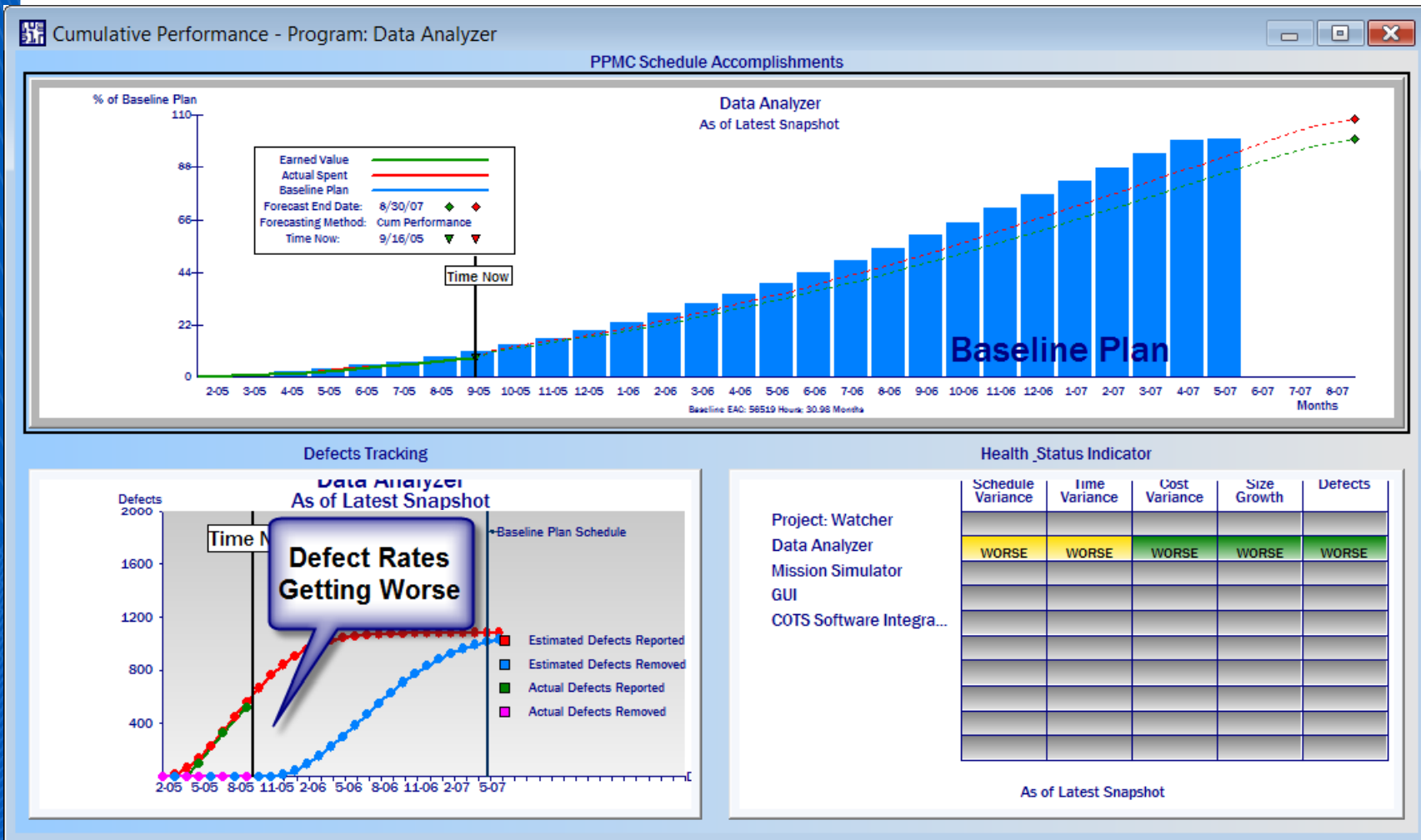
- Provide Measurement Results
- Project Planning
 - Establish Estimates
 - Develop a Project Plan
 - Obtain Commitment to the Plan
- Project Monitoring and Control
 - Monitor Project Against Plan
 - Manage Corrective Action to Closure
- Measurement and Analysis
 - Align Measurement and Analysis Activities



What To Measure: Multiplicity of Metrics

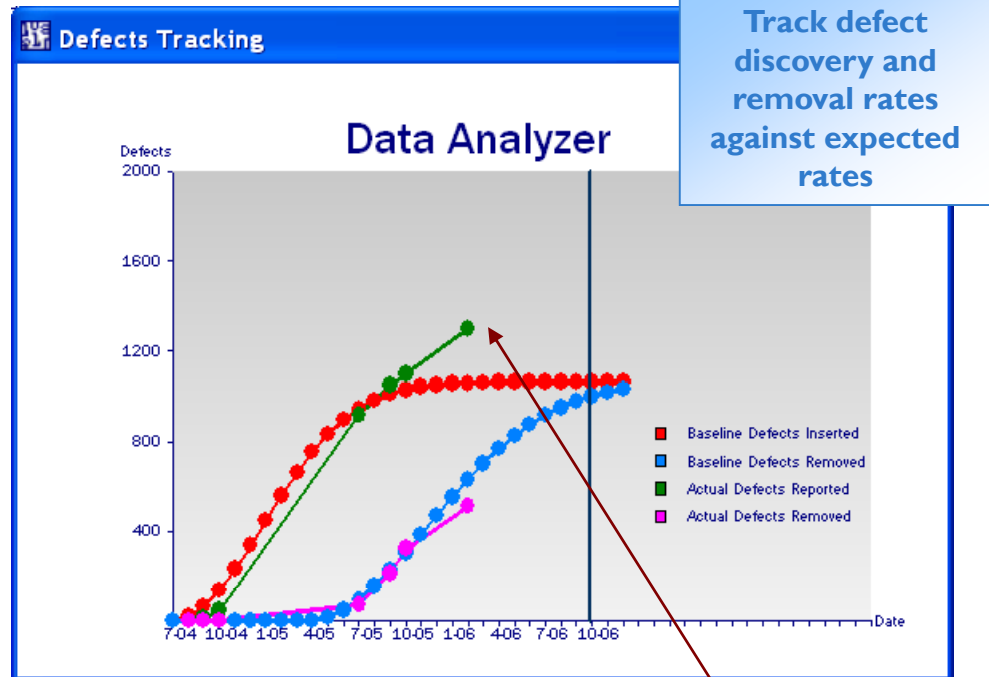
1. Obvious: Status / Trend Metrics: e.g. productivity, defects removal rate, cost, schedule
2. Most important for improvement: Effectiveness (5 max)
 - “What we are doing that we should not do” e.g. number of delivered critical defects
 - “What we are not doing that we should do” e.g. number of defects that got past inspections
 - These metrics may change over time as we improve

Defects and Size Growth Provide Early Warning

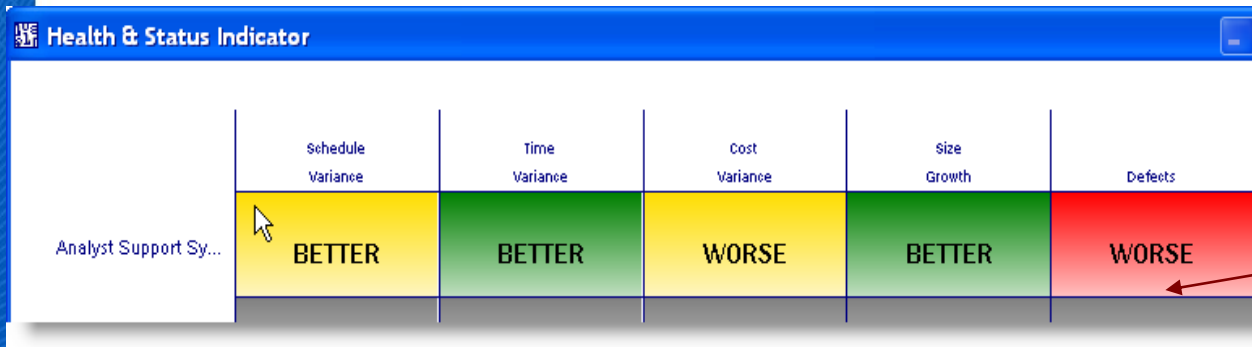


Defects and Growth Impact Software Process

Health and Status Indicator shows status and trends from the previous snapshot
Thresholds are user definable



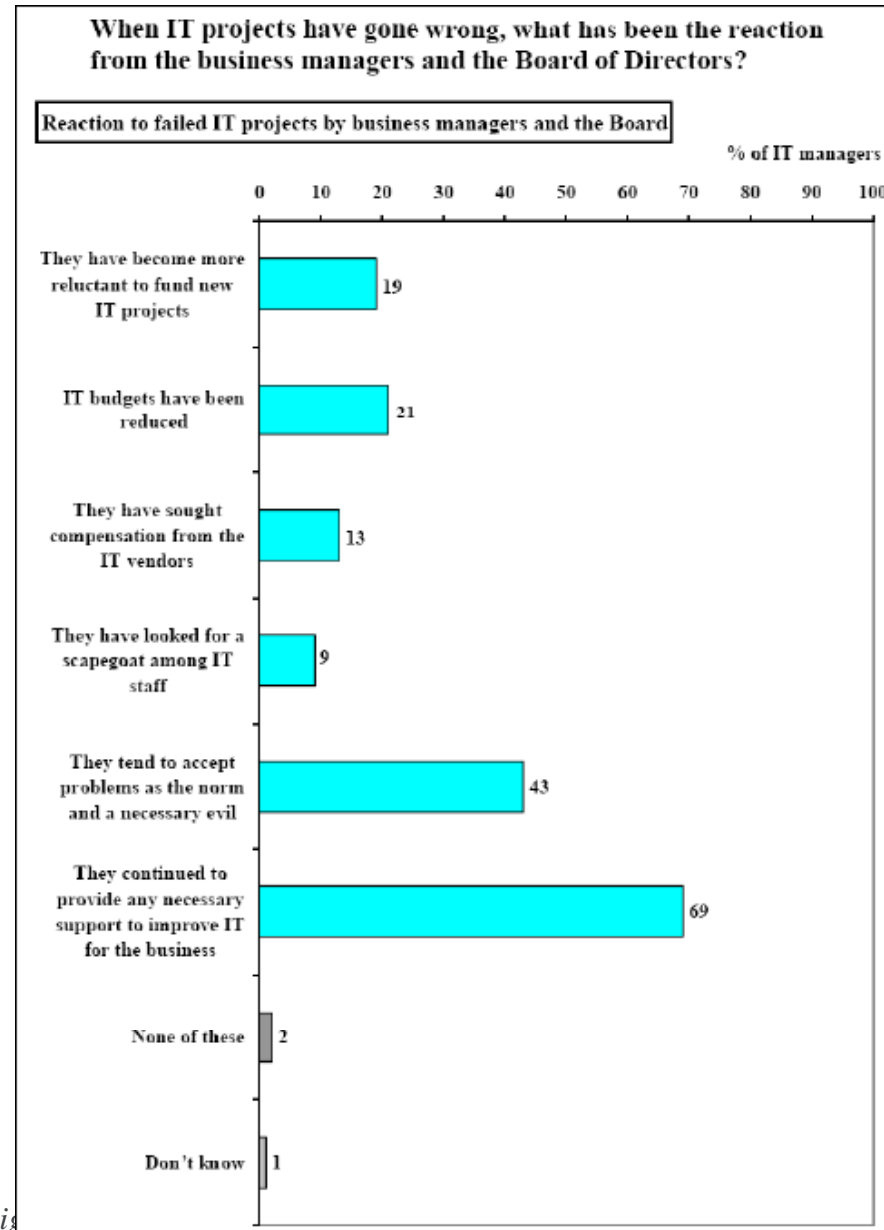
Increased defect reporting rate shows a worsening trend



Measuring Defect Insertion & Removal During Development: Better Progress Measure Than Just Earned Value



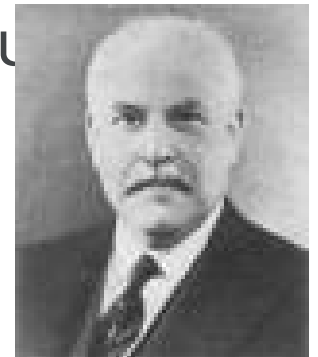
Tata Survey. When IT Projects Go Wrong



Measure What Is Real

- “The government [is] extremely fond of amassing great quantities of statistics. These are raised to the nth degree, the cube roots are extracted, and the results are arranged into elaborate and impressive displays.
- What must be kept ever in mind, however, is that **in every case, the figures are first put down by a village watchman, and he puts down anything he d..m well pleases.**
- Attributed to Sir Josiah Stamp, 1840-1941, H.M. collector of inland revenue

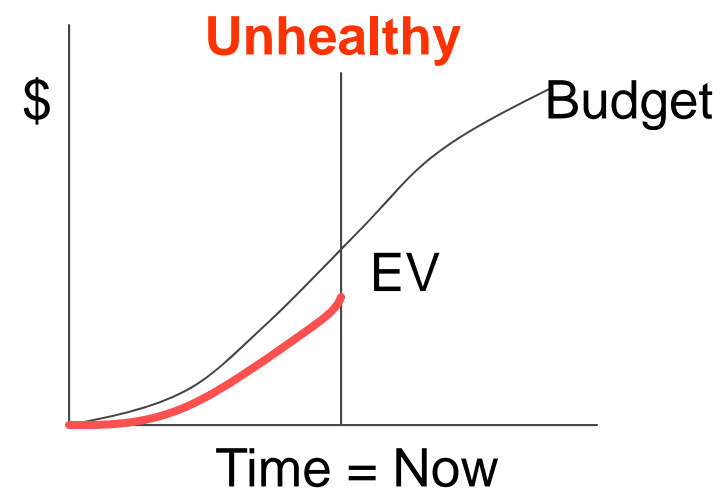
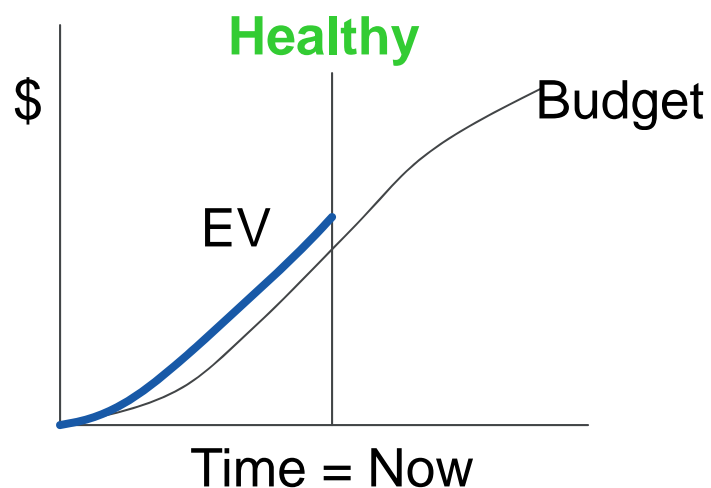
How often are our measurement programs providing less than perfect data? E.g software cost data



Use Earned Value TO Quantify Progress Versus Effort FOR DEVELOPMENT AND MAINTENANCE

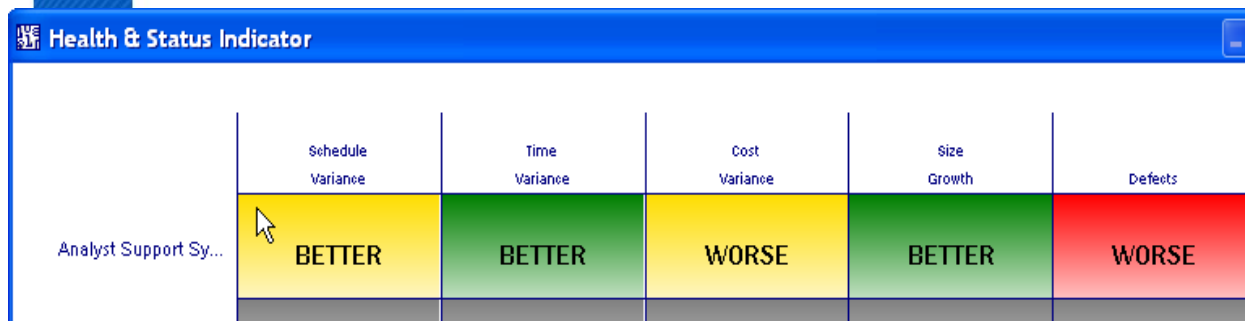
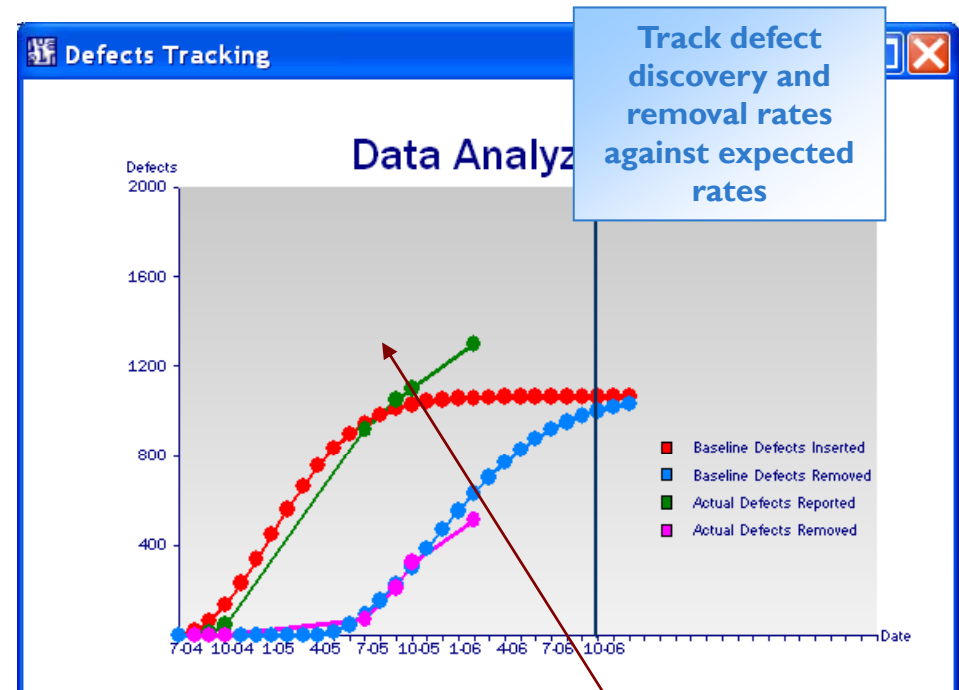


- Main concern of EVM: what has been accomplished in a given time and budget, versus what was planned for the same time and budget
 - A project is generally healthy if what has been accomplished is what was planned, or more
 - Project unhealthy if accomplishment lags expectations
- Definition: Earned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



Defects and Growth Impact Software Process

Health and Status Indicator shows status and trends from the previous snapshot
Thresholds are user definable

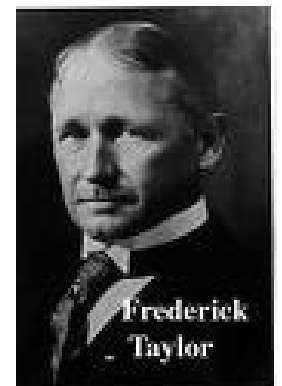


Increased defect reporting rate shows a worsening trend

The Hawthorne Effect: People Respond To Being Measured

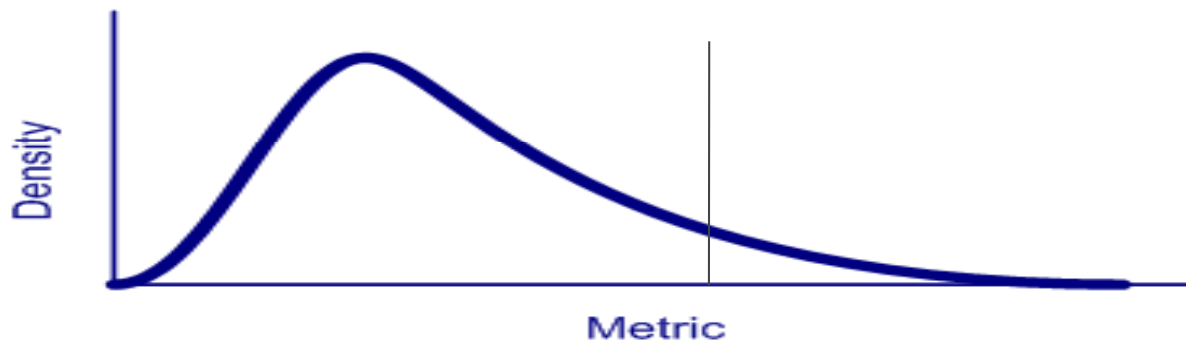


- Goal: Find optimum for productivity 1924 to 1927
- **Increase, No Control Group**; Three departments; all showed an increase of productivity, whether illumination increased or decreased.
- **Increase, Control group** = change in lighting; experimental group got sequence of increasing light. Both groups substantially increased production, no difference between groups
- **Decrease, Control group got stable lights**; other sequence of decreasing levels. Both groups steadily increased production until the light in experimental group got so low they protested and production fell off
- All back to original: Productivity went up



Manual Estimates: Human Reasons For Error (Metrics Can Help)

- 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



Core Metric: Value Provided By Software

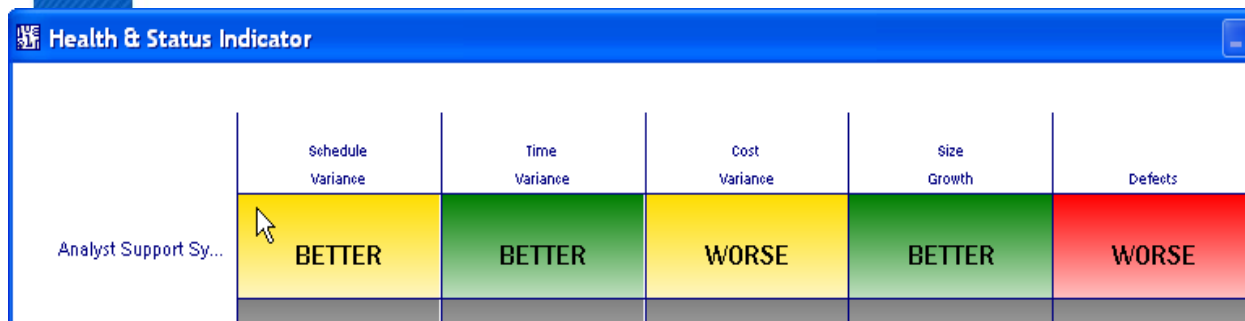
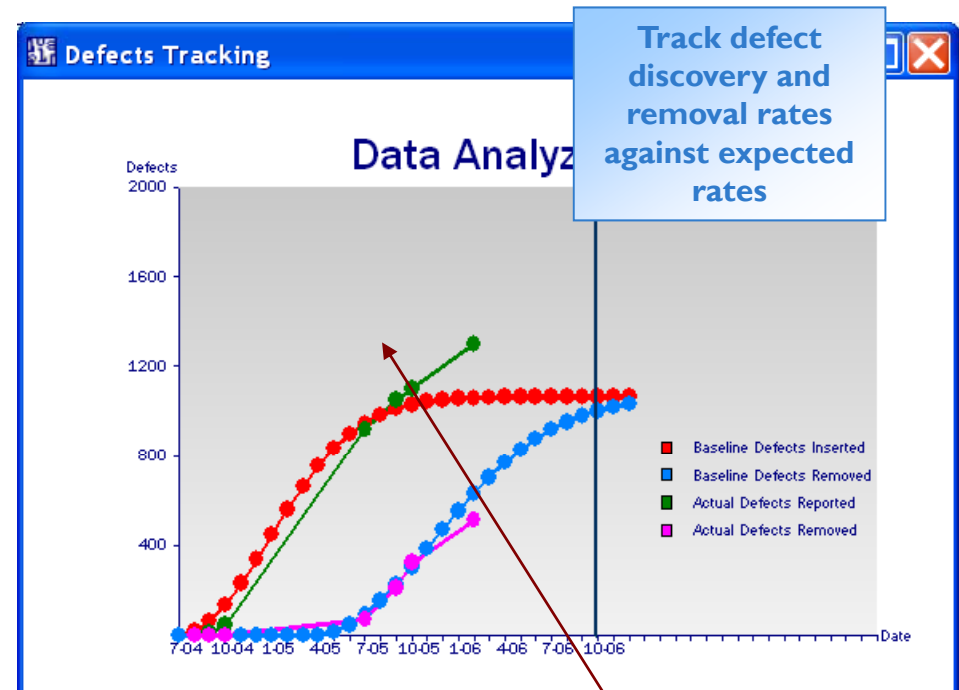


- Concept: Spend where you obtain the most value
 - Value = savings to company or additional revenue due to the software
- Software Fails to add value much too often
 - Users enamored with concept
 - Concept deployed
 - Little to no value contributed to company...
 - Many reasons... often no changes in business rules
- MRP is a classic example of software hyped but which did not provide value

Many Organizations May Not
Be Mature Enough To Consider Value From
the Software Team

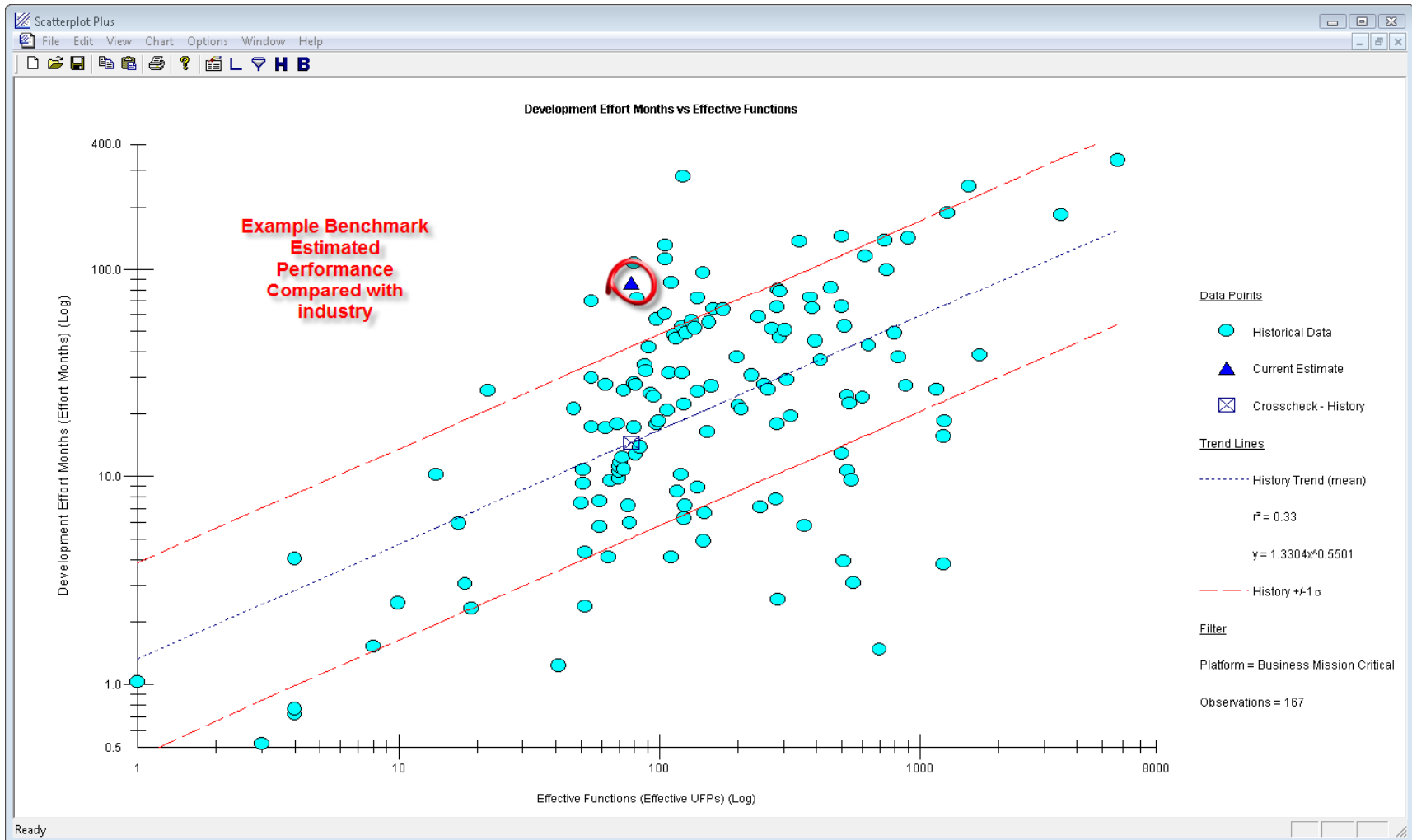
Defects and Growth Impact Software Process

Health and Status Indicator shows status and trends from the previous snapshot
Thresholds are user definable

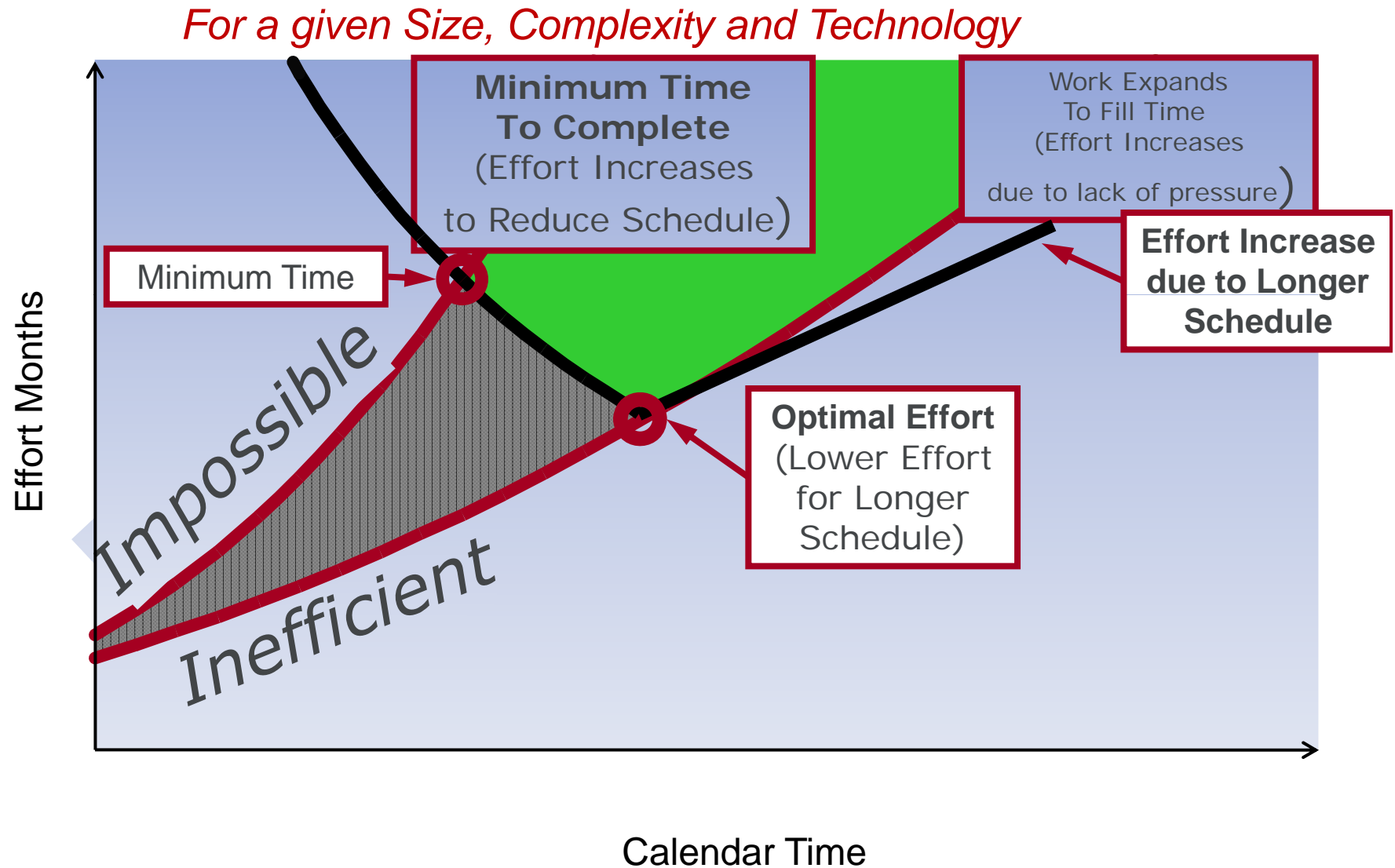


Increased defect reporting rate shows a worsening trend

Example Benchmark Versus an Estimate.. Why Are We So Expensive?



Balancing Resources & Schedule Is A Science



estimate

estimate • analyze • plan • control

Software Maintenance & Total Cost of Ownership



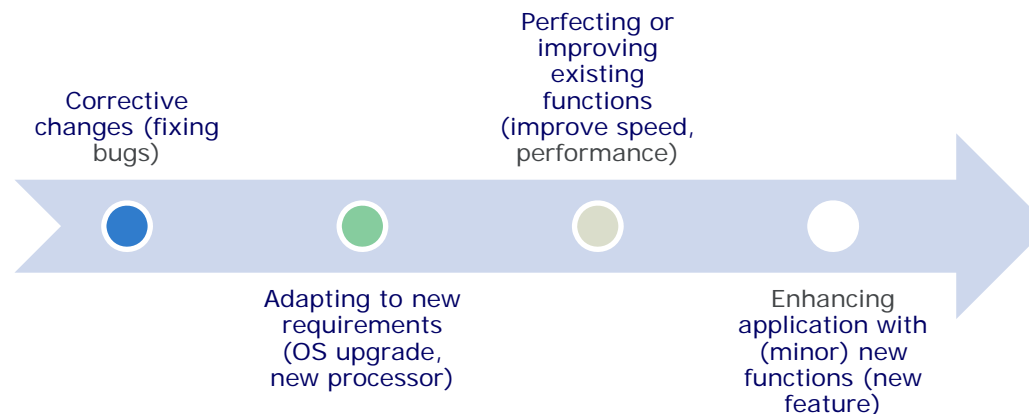
Maintenance Standards Provide Definition and Consistency



- **IEEE 1219 Standard for Software Maintenance:** Modification of software product after delivery to 1) correct faults, 2) improve performance or other attributes, 3) adapt the product to a modified environment
- **IEEE/EIA 12207 standard for software life cycle processes:** Process of a software product undergoing “modification to code and associated documentation due to a problem or the need for improvement”
 - The objective is to modify the existing software product while preserving its integrity.”
- **ISO/IEC 14764, international standard for software maintenance:** defines software maintenance in the same terms as IEEE/EIA 12207
 - Emphasizes the pre-delivery aspects of maintenance, planning

Maintenance Dissected

- Maintenance typically 75% + of the total software workload:
 - Highly dependent on rigor & operational “life expectancy”
 - Reducing maintenance costs can reduce life cycle costs significantly
- Generally includes sustaining engineering & new function development:
 - Corrective changes (fixing bugs)
 - Adapting to new requirements (OS upgrade, new processor)
 - Perfecting or improving existing functions (improve speed, performance)
 - Enhancing application with (minor) new functions (new feature)
- For every new software product we develop, we get one more to maintain -- for ?? years



Software Maintenance Critical Success Factors (Source IEEE)



Functionality: Preserve or enhance functionality



Quality: Preserve or increase quality of system



Complexity: Should not increase product complexity relative to the size



Volatility: should not lead to increase in product volatility



Costs: Relative costs per maintenance task should not increase for similarly scoped tasks



Deadlines: Agreed upon release deadlines should be kept and delays should not increase



User Satisfaction: Increase or at least not decrease



Profitability: Be profitable or at least cover its costs

Why Maintenance Is Hard



- May not have had maintenance as a goal
- System may not have been fully tested
- Documentation may be inadequate
- Maintenance staff may be inexperienced
- The tendency to produce quick & dirty fixes
- Process or language experience may have left a mess
- The "but I only changed 1 line syndrome"

Why Software Maintenance Metrics Are Harder

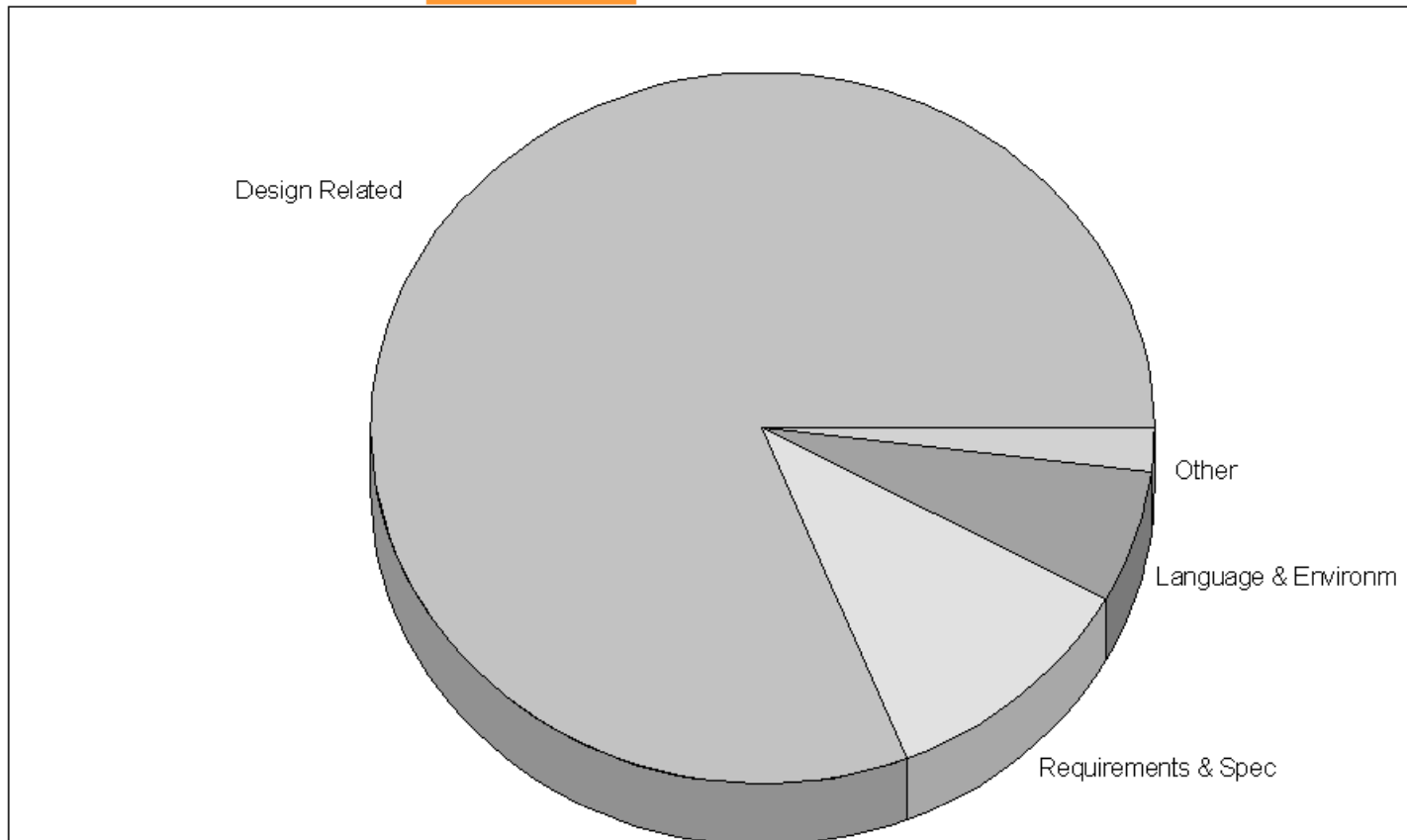


- Software Maintenance treated as A Level Of Effort Activity
- This Means You Can Maintain Software With A Larger Or Smaller Staff Depending On Your Desires / Budget

Maintaining A Car	Maintaining Software
High Maintenance: Go By The Book (Regular Oil Changes, Etc.)	<ul style="list-style-type: none">• Fix emergencies• Provide new functionality as needed• Adapt as necessary• Software may not degenerate over time
Nominal Maintenance: Go Partially By The Book (Less Frequent Oil Changes, Etc.)	<ul style="list-style-type: none">• Fix emergencies• Provide some required new functionality• Adapt when there is time
Low Maintenance: Go Slightly By The Book (Add Oil When The Low Oil Light Goes On)	<ul style="list-style-type: none">• Fix only emergencies and small adaptations• Software will degenerate over time

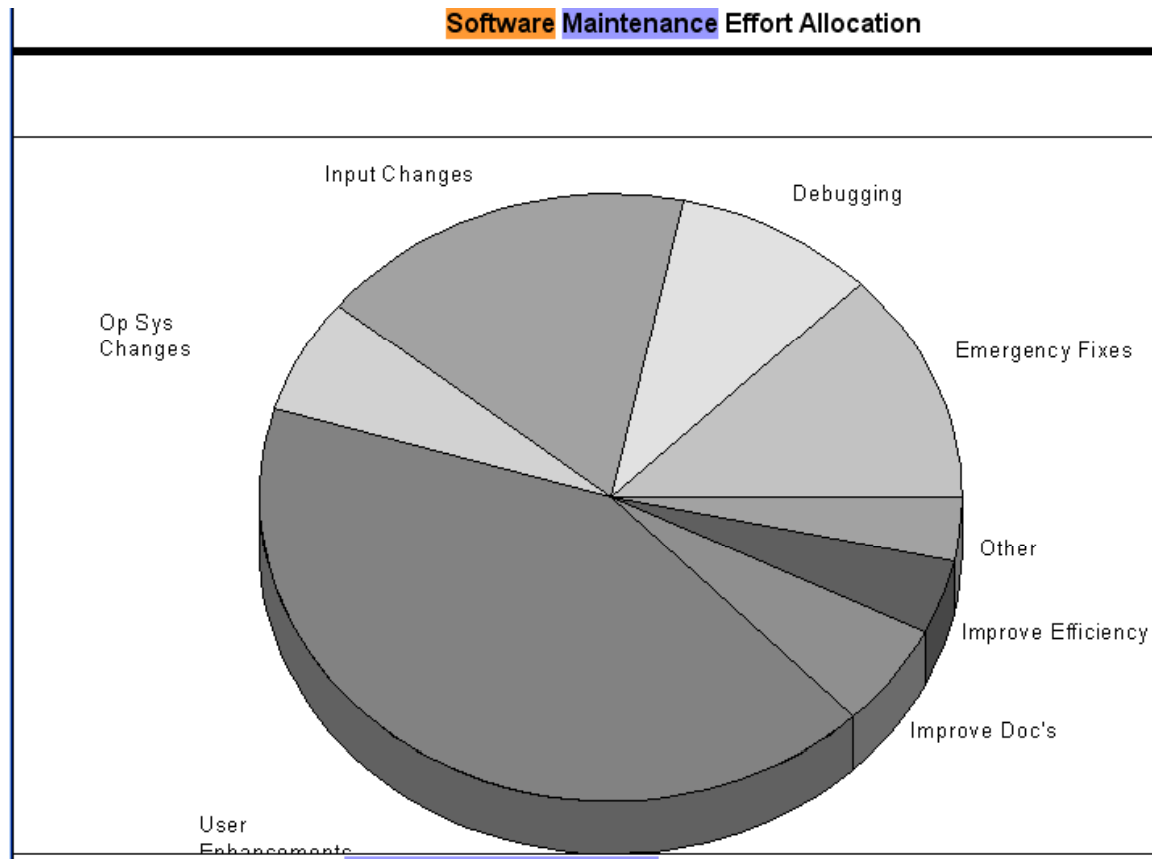
Sources of Software Errors

sources of software errors (source IEEE transactions)

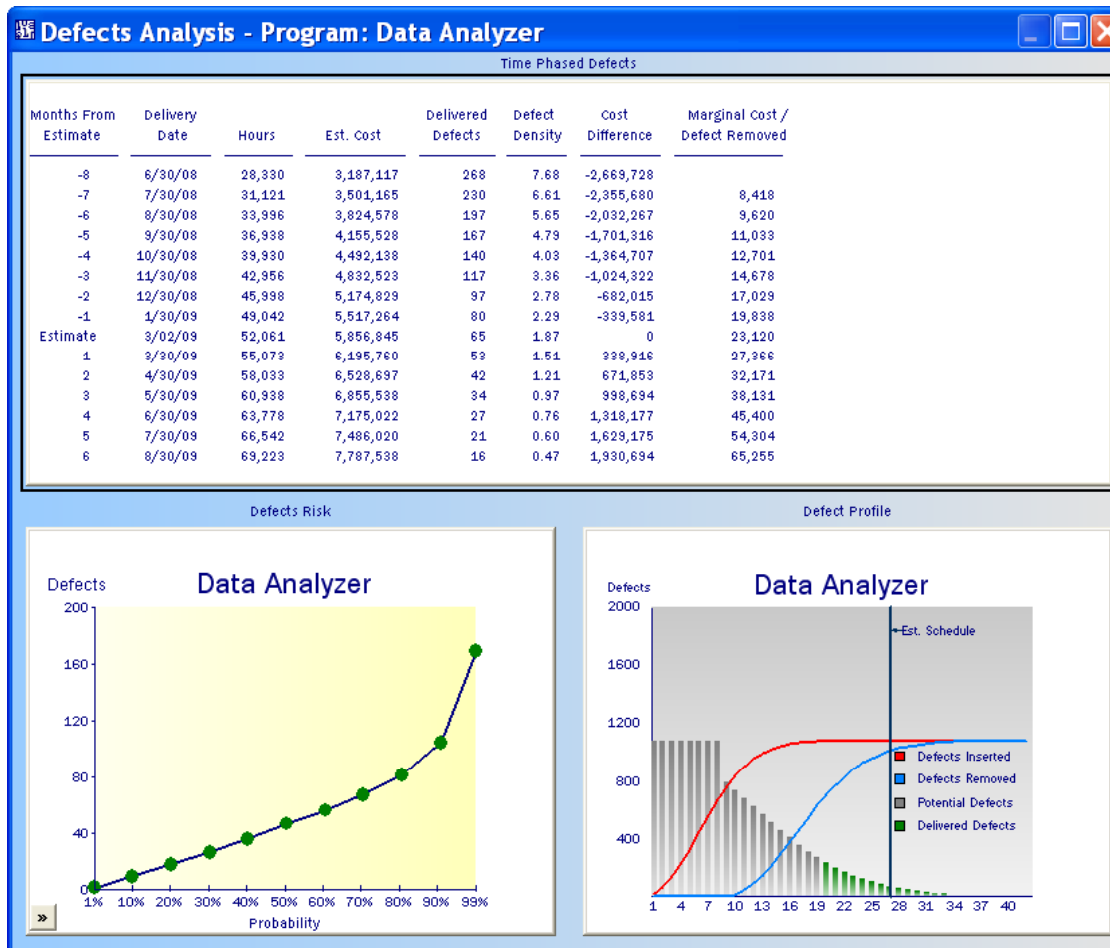


Allocation of Software Effort (Source IEEE)

Software Maintenance Effort Allocation



Development Defects Analysis Is a Clue to Maintenance Issues



Measurement Job Not Over When Development Is Complete Maintenance GQM (Adapted from Mitre)



Goal	Question	Metric(s)
Maximize Customer Satisfaction	How many problems affect the customer?	1. Current Change Backlog 2. Software Reliability
Minimize cost	How much does a software maintenance delivery cost?	
	How are costs allocated	Cost per activity
	What kinds of changes are being made?	Number of changes by type
	How much effort is expended per change	Staff hours expended by change /type
Minimize Schedule	How difficult is the delivery?	Complexity Assessment Software Maintainability Computer resource Utilization
	Are we meeting delivery schedules?	Percentage of On-Time Deliveries

Example Maintenance Metrics



- Defects Inserted per correction
- Defects removed per unit time
- Productivity for block changes
- Maintainability
- Mean time to find the next k faults
- Maintenance backlog
- Increases / decrease on maintenance backlog
- Number of trouble reports opened and closed
- Mean time until problem closed
- Defects during warranty period
- Mean time to resolution
- Defects by type and severity
- Time to respond to customer reported defects
- McCabe & Halstead complexity metrics

Example Effectiveness metrics for Maintenance



- Number of new defects created by fixes
- Number of defect corrections that were not correct
- Number of defects not repaired in promised time (Delinquent)
- **Defect Seepage.. (Customer reported defects during pre-delivery testing)**

Identify the metrics that YOUR organization needs

Software Maturity Index (Example of Metric from IEEE 982 Standard Dictionary of Measures to Produce Reliable Software)



M = number of modules in current version

A = number of added modules in current version

C = number of changed modules in current version

D = number of deleted modules in current version compared to the previous version

$$\text{SMI} = (M - (A + C + D)) / M$$

- when SMI approaches 1.0 the product is stable

Maintenance Growth Over Life



- Anticipated size growth from the point immediately after the software is turned over to maintenance to the end of the maintenance cycle
- May include additions of new functionality

Rating

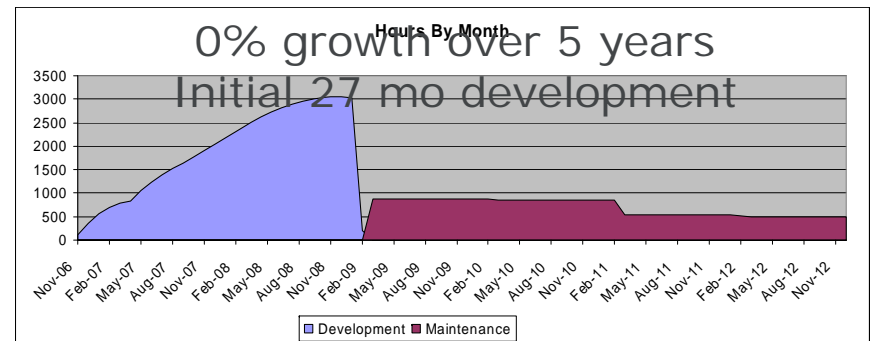
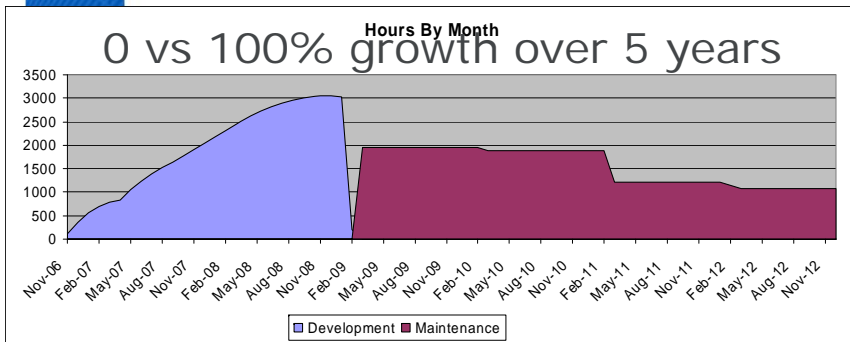
Description

- 100% Major updates adding many new functions
- 35% Moderate updates adding some new functions
- 20% Minor updates & enhancements to existing functions
- 5% No updates expected, some minor enhancements
- 0% Sustaining engineering only

100% growth over 5 years
Initial 27 mo development

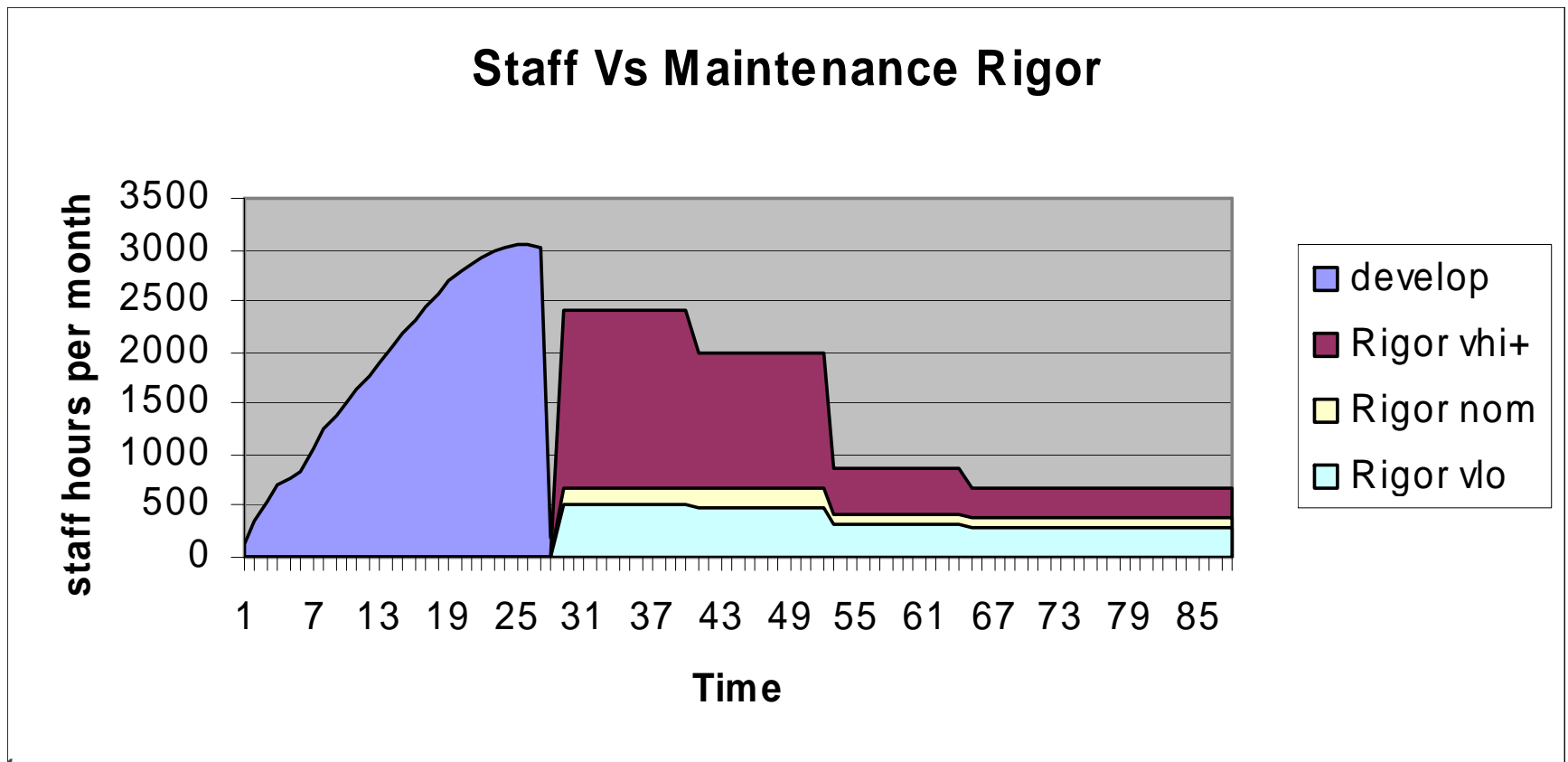
Quick Estimate

	Program: Data Analyzer Estimate	Program: Data Analyzer Reference	Diff.
Development Schedule Months	27.07	27.07	0%
Development Effort Months	342.51	342.51	0%
Development Effort Hours	52,061	52,061	0%
Development Base Year Cost	5,856,845	5,856,845	0%
Maintenance Effort Months	584.23	260.59	124%
Defect Prediction	65	65	0%
Constraints	MIN TIME	MIN TIME	



Key Driver: Maintenance Level (Rigor)

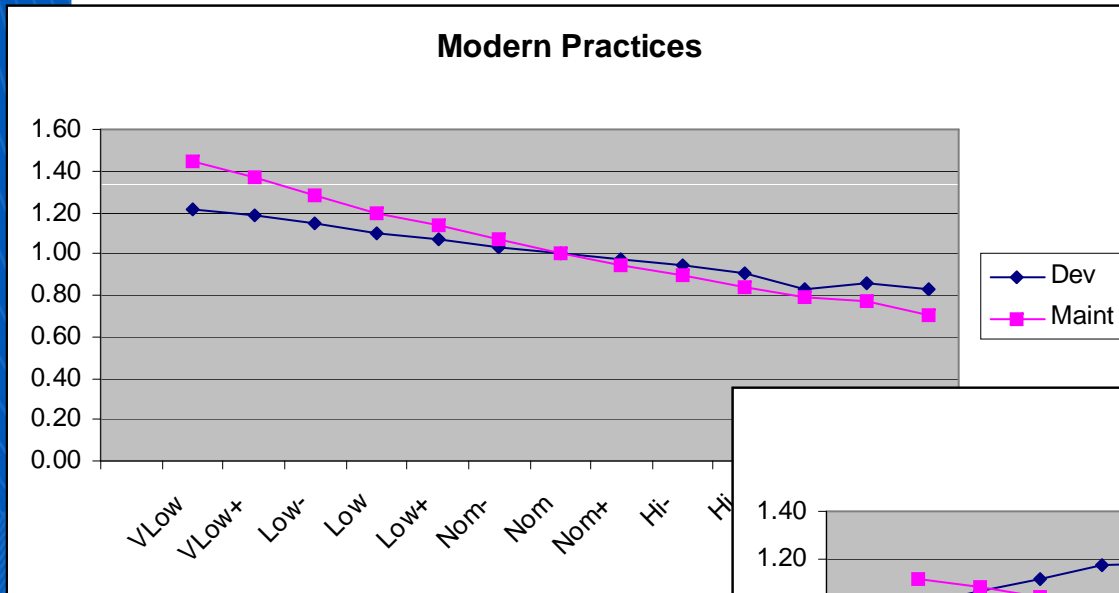
Most Projects Spend Low During Maintenance



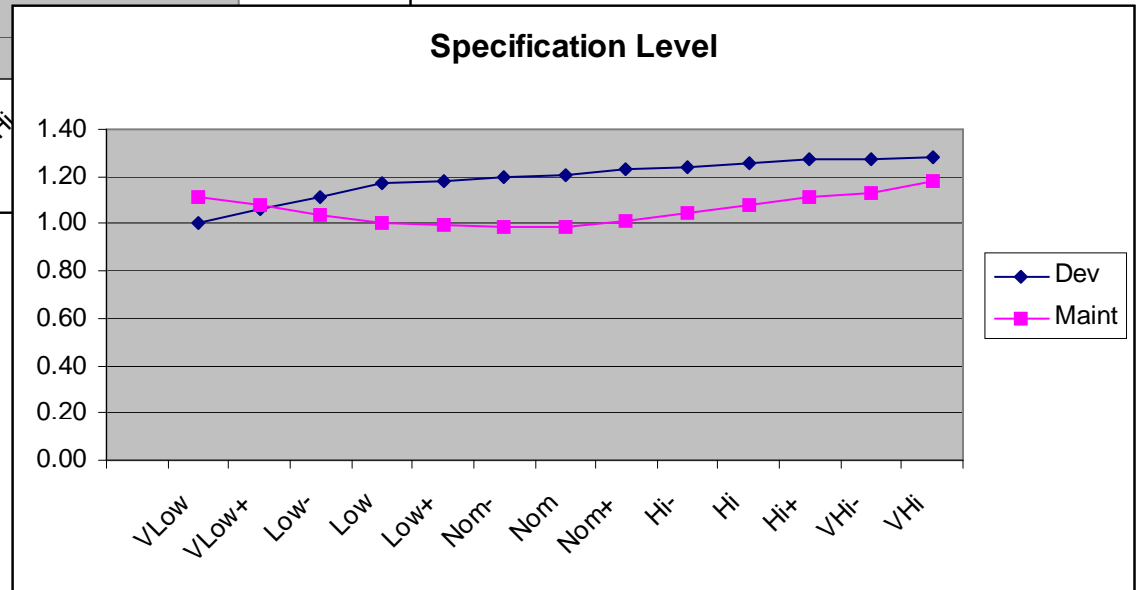
People, Process, Technology Sensitivity Development Vs Maintenance – 1



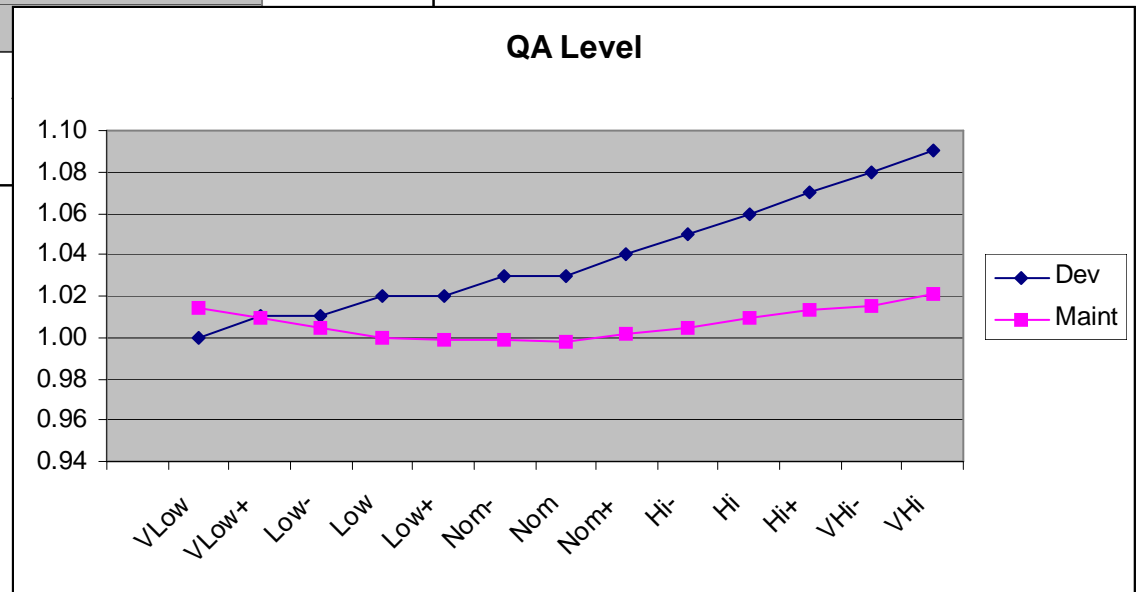
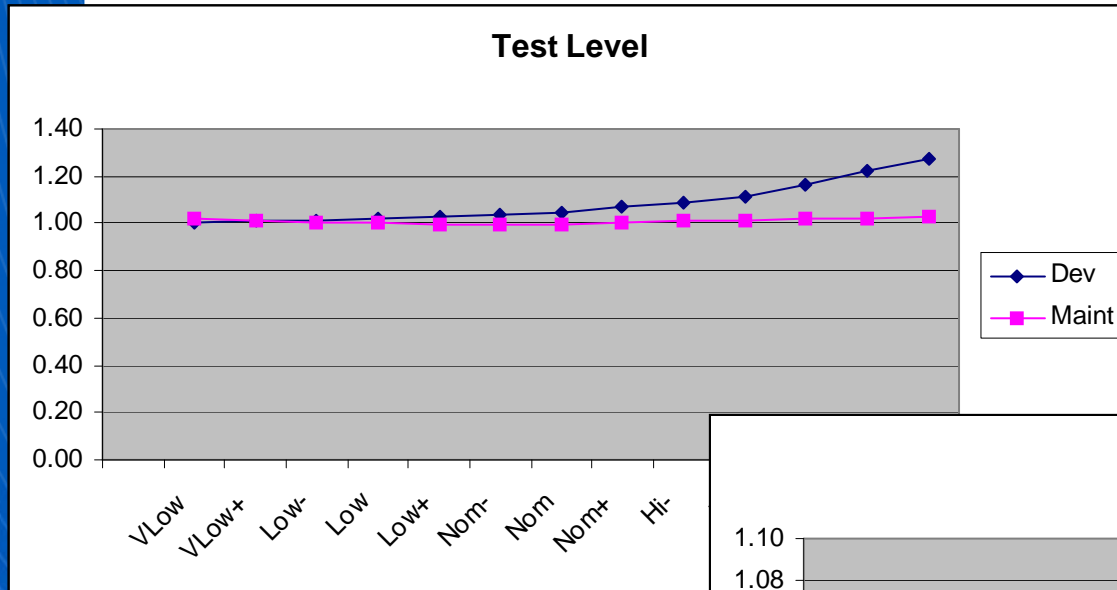
Modern Practices



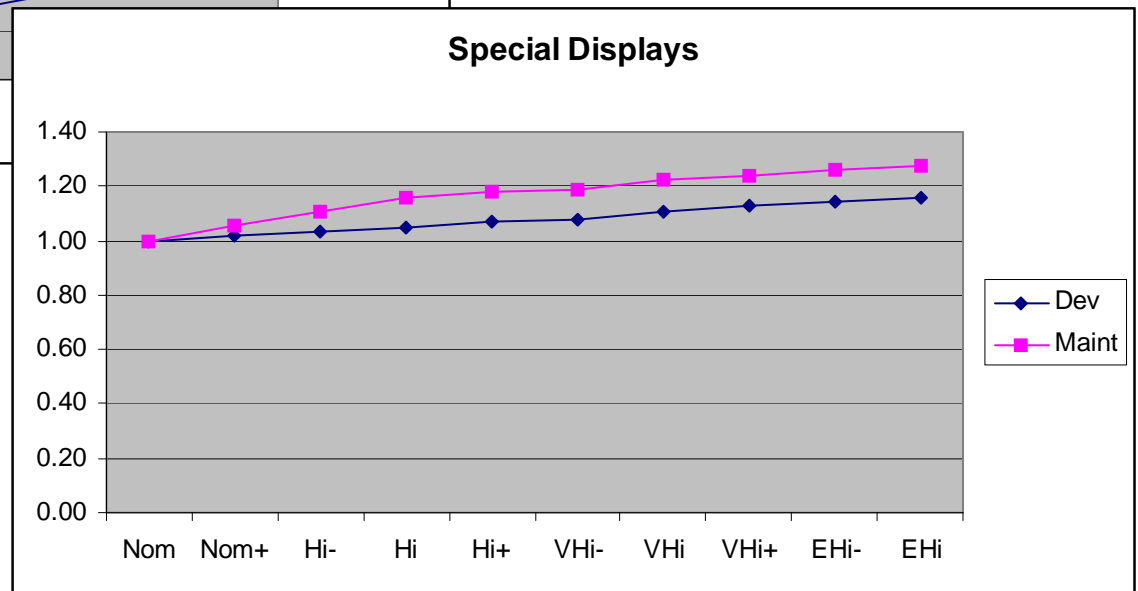
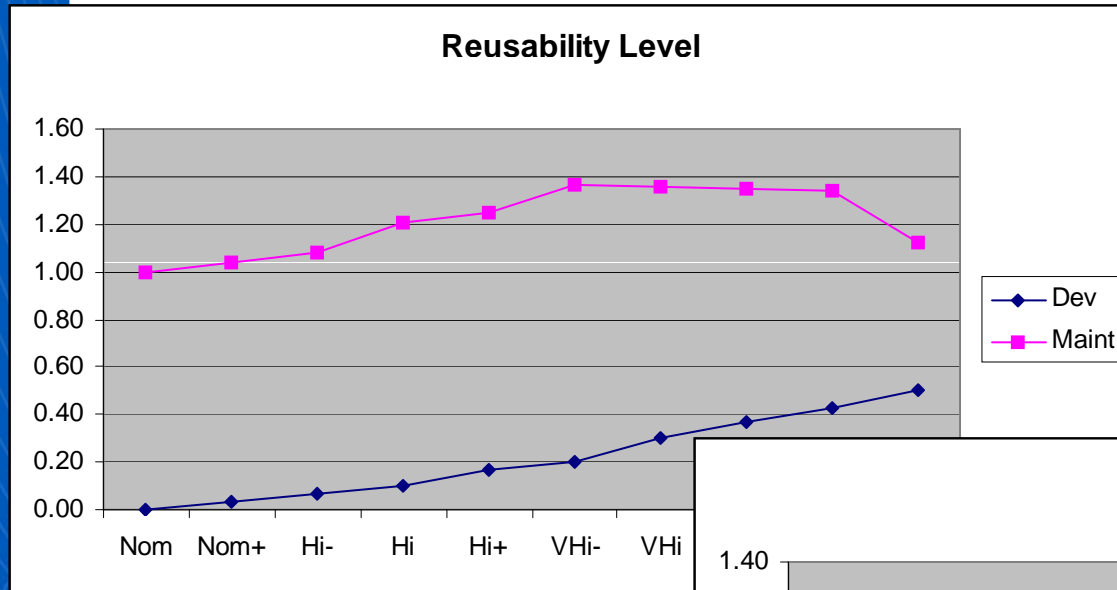
Specification Level



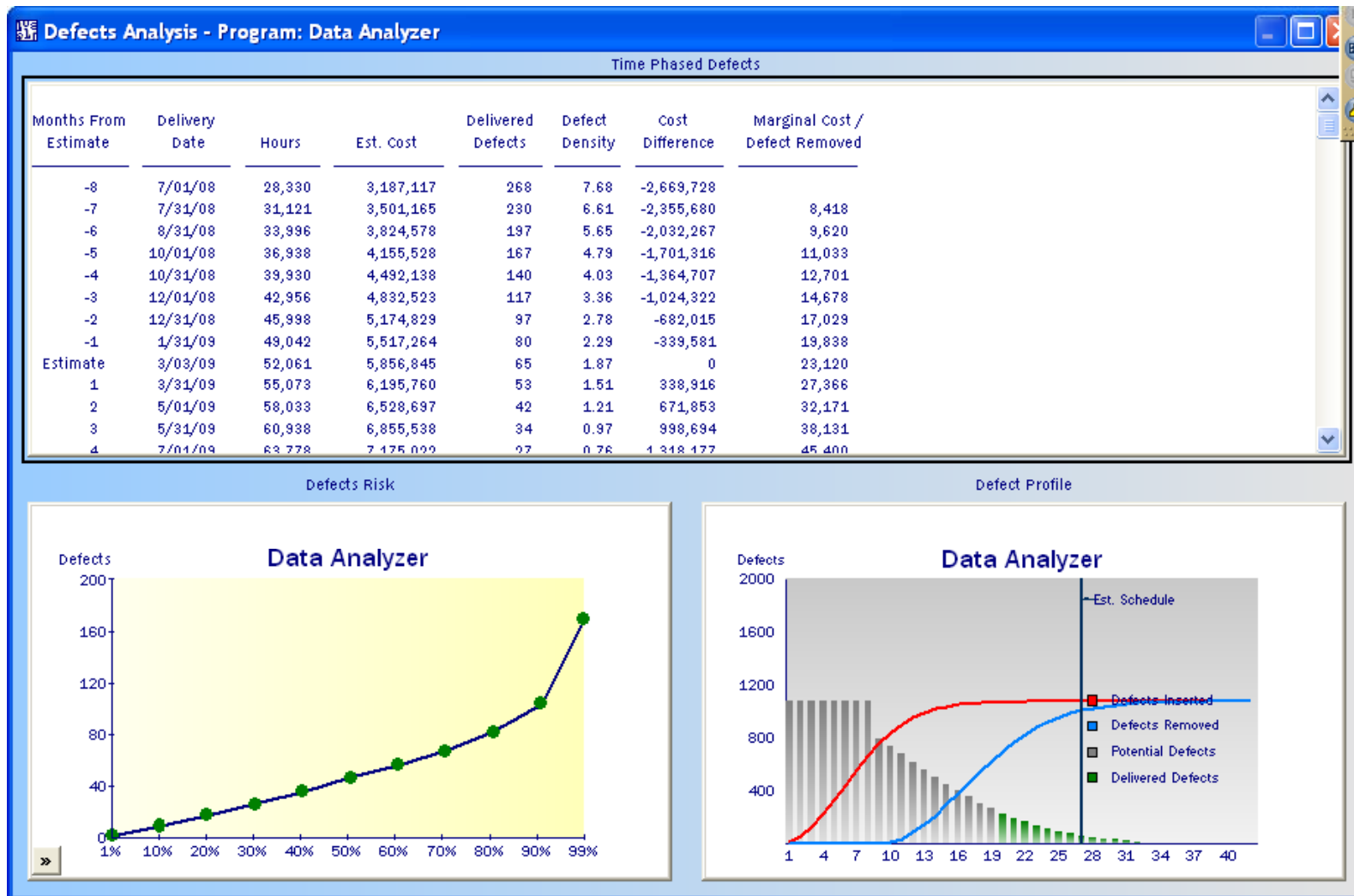
Development Vs Maintenance - 2



Development Vs Maintenance - 3



Defects Can Be Reduced By Further Development Testing but Not Eliminated



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



- Based on 350 projects:
- Failure to Apply Essential Project Management Practices
- Unwarranted Optimism and Unrealistic Management Expectations
- Failure to Implement Effective Software Processes
- Premature Victory Declarations
- Lack of Program Management Leadership
- Untimely Decision-Making
- Lack of Proactive Risk Management

Conclusions



- Maintenance can be 75% of total ownership costs
- Development decisions, processes and tools impact maintenance costs
- While software maintenance is often treated as a level of effort activity there are consequences:
 - Quality, functionality and reliability
- Consider total ownership costs and risks
- Applied measurement is a critical component of software and systems management
- Measure what you want people to focus on
- Continue emphasis on standards and definition

SEER by Galorath Offerings



Projects Delivered...

On Time. On Budget.
As Specified.



• Products:



**SEER for
Software**



SEER for IT



**SEER for
Hardware,
Electronics
& Systems**



**SEER for
Manufacturing**



**SEER Services
& Support**

Dan Galorath

310 414-3222 x614

galorath@galorath.com

Blog: www.galorath.com/wp