Using Quantitative Software Estimation Tools and Techniques to Support Verification and Validation (V&V)

Joseph Madden
Quantitative Software Management, Inc
2000 Corporate Ridge, Suite 700
McLean, VA 22102
(703) 790-0055
Email: joseph.madden@qsm.com
Web: www.qsm.com

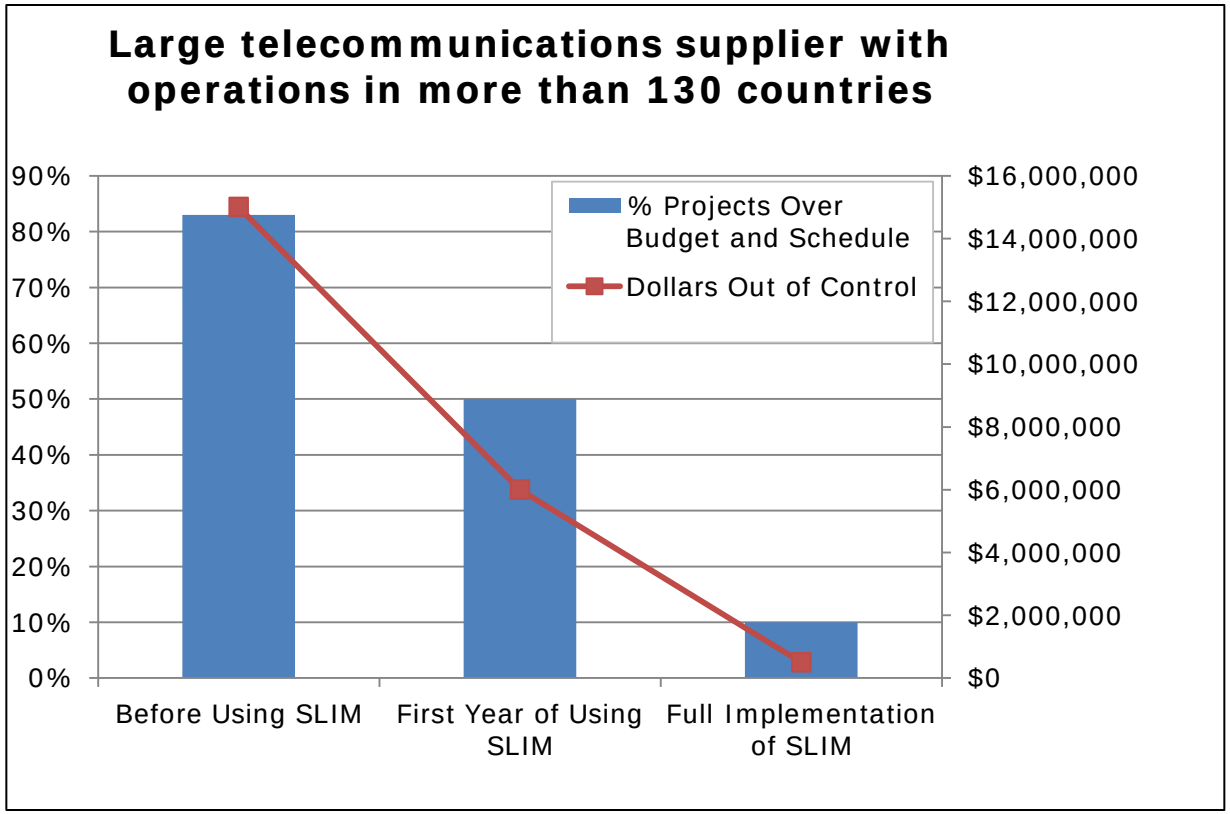
Outline

- Introduction to Quantitative Software Estimation
- Using Quantitative Software Estimation to Support V&V Planning & Execution
- Example Uses of Quantitative Estimation Techniques at the FAA
- Conclusion

About QSM

- QSM founded in 1978 by Larry Putnam, Sr., one of the original pioneers in quantitative software estimation and author of 4 books, including *Measures for Excellence* and *Five Core Metrics*.
- Putnam invented the SLIM quantitative estimation tool and began a benchmark database of historical project data. This database now includes over 10,000 validated projects covering most application domains, industry sectors, and development approaches.

Benefits of Quantitative Estimation: A Case Study



In the year before using SLIM, 10 of 12 projects (83%) exceeded budget and schedule. The cost of this excess was more than $15M. QSM was hired to implement a robust estimation process.

Within the first year of using a SLIM estimation process, the percentage of projects over schedule and budget decreased from 83% to 50%, and excess cost reduced from $15M to $6M. After full implementation of SLIM in the second year, the percentage of projects over schedule and budget dropped to 10% and the cost overruns were less than $500K[1].

[1]The above case study was published in 2003 in the book *Five Core Metrics* by Lawrence H. Putnam and Ware Myers.

The Fundamentals of Quantitative Software Estimation

The Problem

Software Development is Difficult to Estimate

Design Complexity

- "Software entities are more complex for their size than for perhaps any other human construct."[1]

Organizational Complexity

- Large software projects require a large organization. This organizational complexity "makes overview hard, thus impeding conceptual integrity. It makes it hard to find and control all loose ends. It creates a tremendous learning and understanding burden."[1]

Cost and Schedule Overruns

- The Standish Group reported that 80% of systems are delivered late and over budget and 40% of projects fail or are abandoned.[2]
- QSM benchmark database shows 70% of projects exceeded budgets and 81% did not meet schedules.[3]

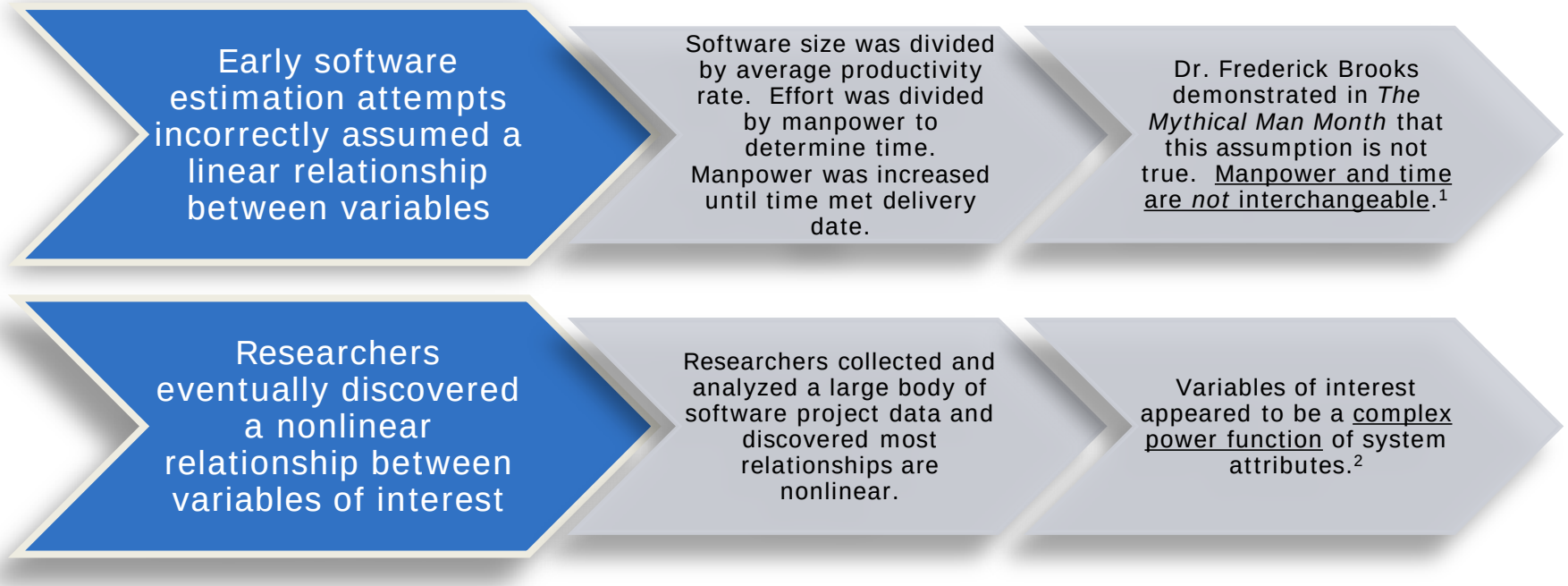
[1] *Frederick P. Brooks, Jr., "No Silver Bullet: Essence and Accidents of Software Engineering," Computer, April 1987.*

[2] *Standish Group. Computer, April 2003.*

[3] *QSM, Inc., Benchmark Database, as of April 2007.*

Variable Relationships

Research Revealed a Nonlinear Relationship Between Variables



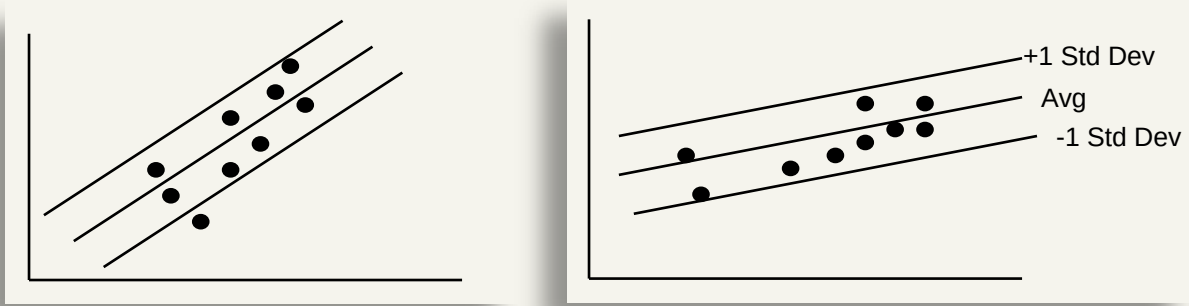
[1] *Frederick P. Brooks, Jr., "No Silver Bullet: Essence and Accidents of Software Engineering," Computer, April 1987.*

[2] *Lawrence H. Putnam. Measures for Excellence: Reliable Software On Time, Within Budget, Prentice Hall PTR, Upper Saddle River, NJ, 1992.*

Patterns in Historical Data

Curve Fitting of Historical Data Resulted in the Discovery of Estimation Equations

Scatter diagrams used to perform trend line analysis, quantifying relationships.



Linkage established between system size, development time, effort, cost, manpower, productivity, and number of defects.[1]

This analysis led to derivation of key computational equation[1]

Product = Productivity Parameter * (Effort/B)(1/3) * Time(4/3)

[1] *Lawrence H. Putnam. Measures for Excellence: Reliable Software On Time, Within Budget, Prentice Hall PTR, Upper Saddle River, NJ, 1992.*

Estimation Equations

Productivity Parameter:
Process productivity parameter for software development organization, obtained by calibration from past projects.

Constant "B":
Special skills factor which is function of size. Increases slowly with size as need for integration, testing, quality assurance, documentation, and management skill grows with increased complexity of large projects.

Effort:
Person-years of work by all job classifications for the software construction or main-build phase: design, coding, inspection, test, documentation, and supervision.

Time:
Elapsed calendar schedule in years for software construction phase.

Manpower Buildup Parameter:
Computed by calibration from past projects.

"Putnam Equations"

$$(Effort/B)^{(1/3)} * Time^{(4/3)} = (Size)/(Productivity\ Parameter)$$

$$(Total\ Effort)/Time^{3} = Manpower\ Buildup\ Parameter$$

Source: Lawrence H. Putnam. Measures for Excellence: Reliable Software On Time, Within Budget, Prentice Hall PTR, Upper Saddle River, NJ, 1992.

Key Time and Effort Rules

Small changes in duration result in large changes in effort

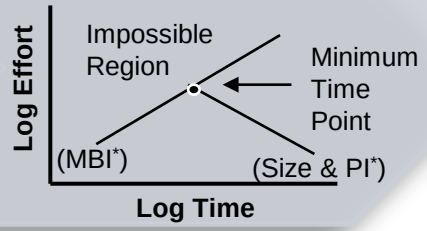
The relationship between effort and time is exponential: **Effort = Constant/Time⁴**

Extending development time from 18 to 19 months (a 5.5% increase) reduces the effort required by 19.5%.[1]

There is a minimum development time

"More software projects have gone awry for lack of calendar time than for all other causes combined."[2]

No data to the left of the MBI line has ever been seen.[1]



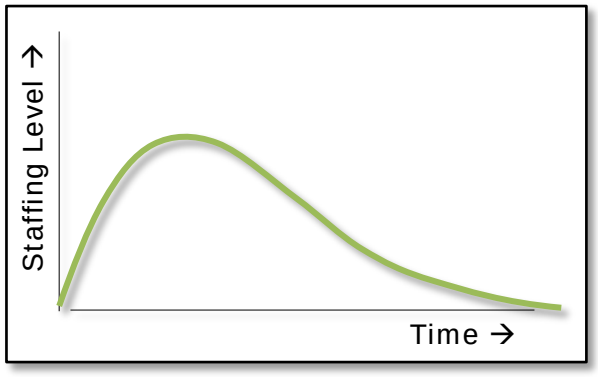
*MBI and PI are indexes based on the Manpower Buildup and Productivity parameters.

[1] *Lawrence H. Putnam. Measures for Excellence: Reliable Software On Time, Within Budget, Prentice Hall PTR, Upper Saddle River, NJ, 1992.*

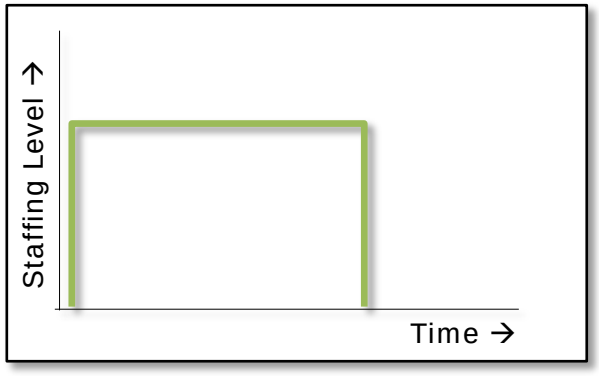
[2] *Frederick P. Brooks, Jr., "No Silver Bullet: Essence and Accidents of Software Engineering," Computer, April 1987.*

The Shape of Projects

- 1960s: Dr. Peter Norden found that engineering projects could be depicted by a Rayleigh Curve to predict staffing1
- 1970s: Larry Putnam Sr. applied the principles to software projects—this offered a mathematical way of predicting broader project behavior2



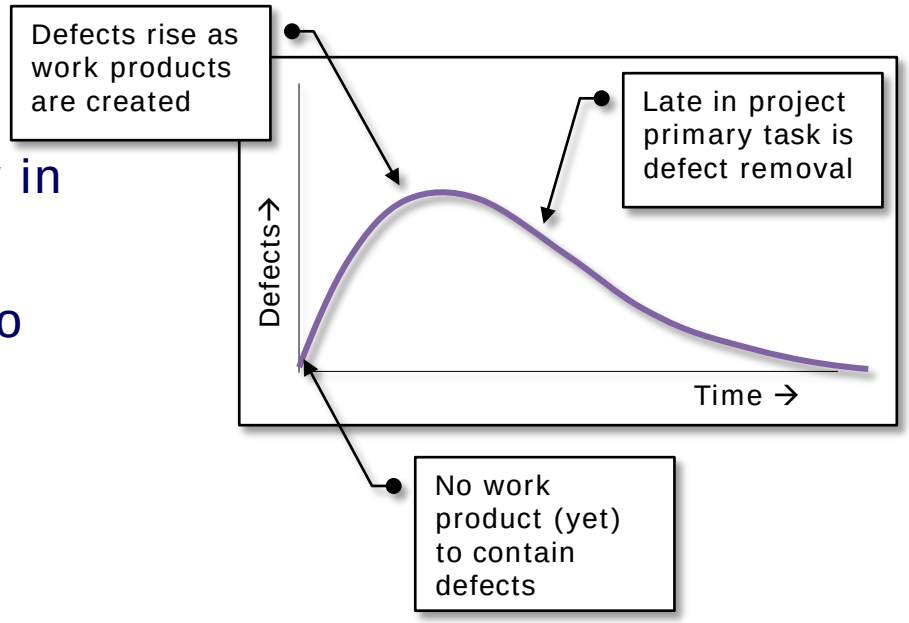
- Traditional project management often tries (often unsuccessfully) to "level-load" projects



1. *Norden, P.V.; "On the Anatomy of Development Projects," IRE Transactions on Engineering Management, 1960 Volume 7, Number 1, pp. 34-42.*
2. *Putnam, Lawrence and Myers, Ware. Measures for Excellence. Yourdon Press 1992. pp.44-45*

Reliability Model for Predicting Software Quality

- Defects in software also tend to follow a Rayleigh curve.
- We can use this to predict quality in a software product.
- SLIM-Estimate uses Mean Time To Defect (MTTD): the average time between defects appearing as a measurable indicator of quality.



Reliability Model Variable Relationships

Size: Historical data has shown that as the code size increases, the number of defects increases. The rate of defect increase is close to linear.

Peak Staffing: Historical data has shown that adding people increases the defect creation process at a rapidly accelerating rate. For example:

- A project of 350,000 SLOC using 40 people at peak staff would create approximately 2,125 total defects.
- If 60 people were used, 3 months of schedule compression would be gained but 3,010 defects would be created.

Quantitative Estimation

Inputs

Software **Size**:
- *SLOC*
- *Function Points*
- *Objects, Etc.*

Uncertainty

Process **Productivity**:
- *Methods/Tools*
- *Tech Complexity*
- *Personnel Profile*

Management **Constraints**:
- *Max People*
- *Max Budget*
- *Max Schedule*
- *Required Reliability*

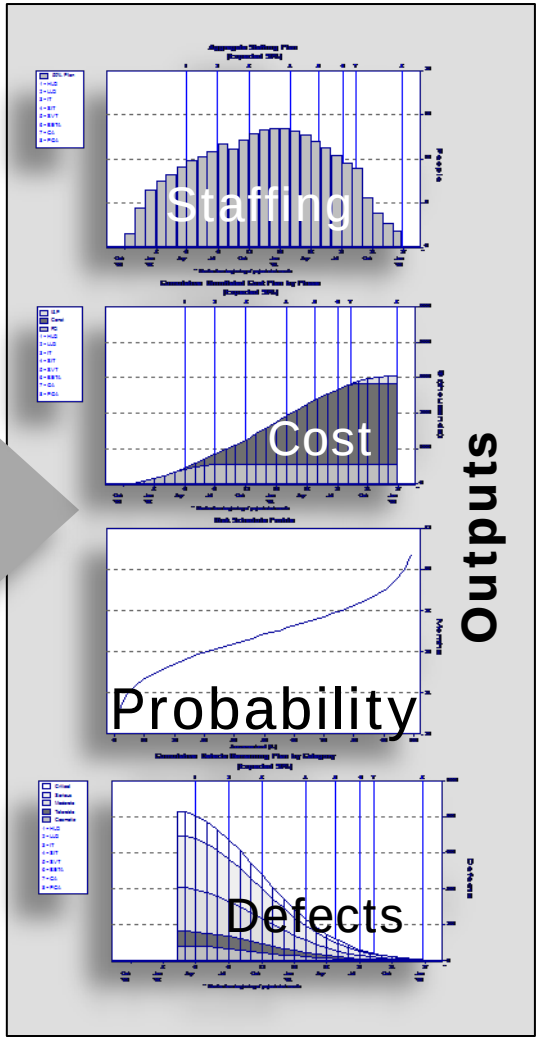
SLIM-Estimate

Optimum Estimate (Max Probability of Meeting Constraints)

Evaluate Practical Alternatives

Generate Plans

Outputs



Quantitative Control

Tracking of Plan Against Approved Baseline

Inputs

Monthly Metrics

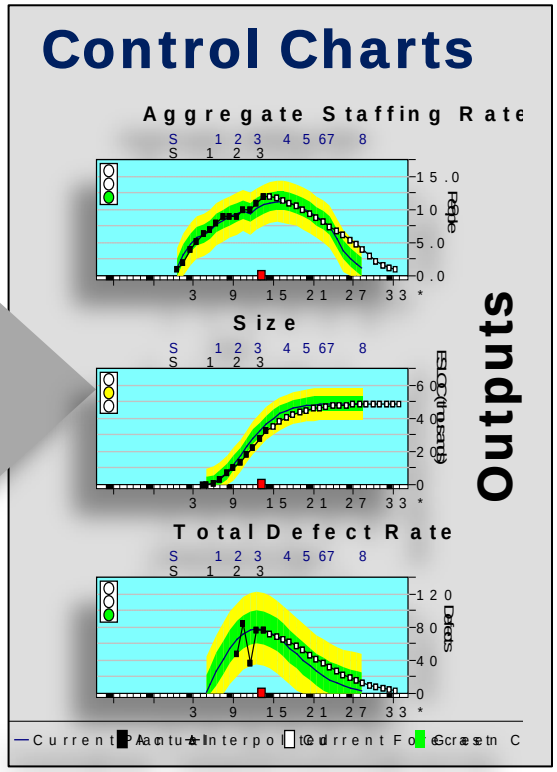
- Software Development **Size** Actuals
- Cost Accounting **Effort**/Cost Actuals
- Software Test **Defect** Actuals
- Project Management **Milestones**

SLIM-Control

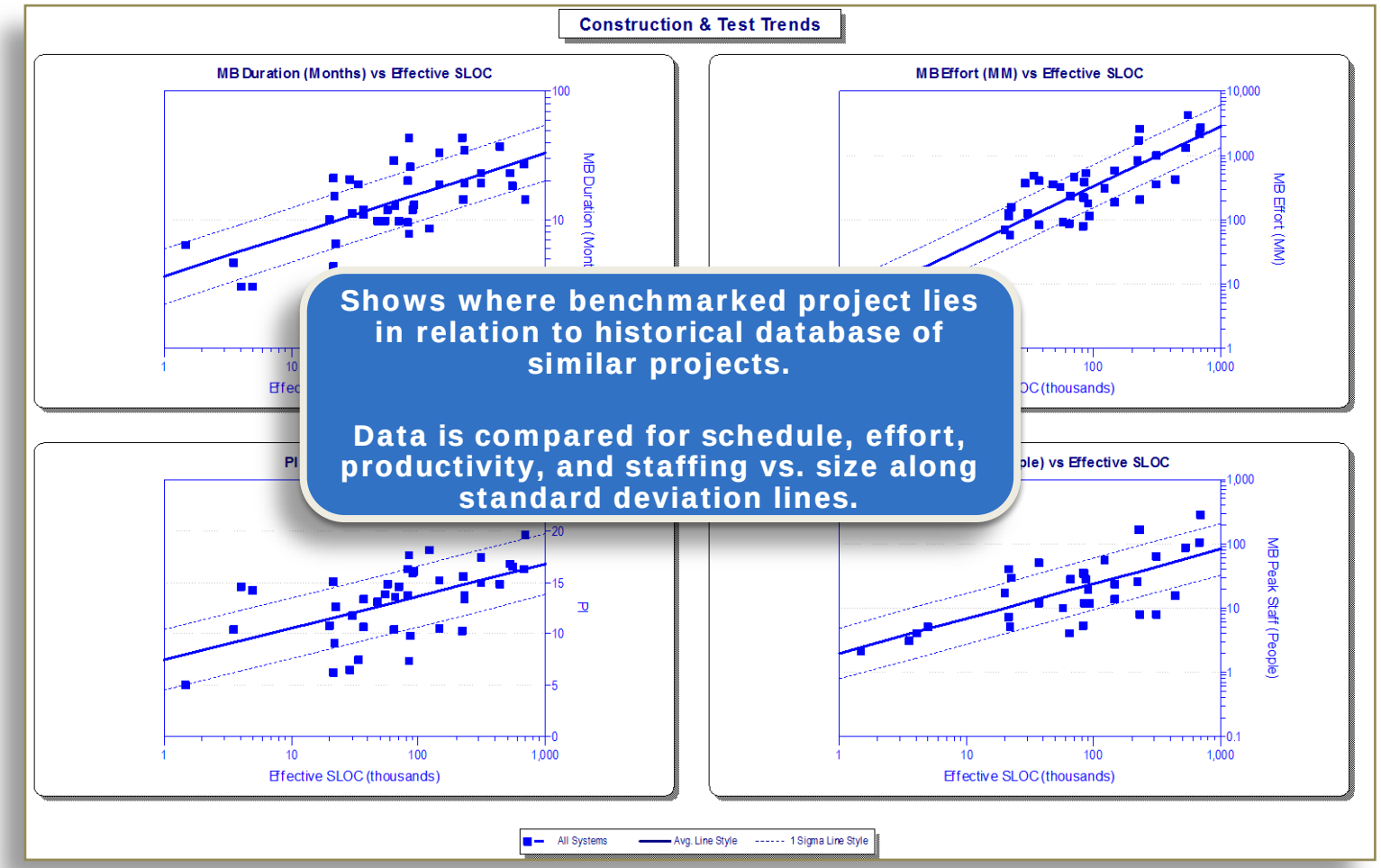
- Variance from Plan
- Forecast to Complete
- Evaluation of Alternative Management Strategies

Outputs

Control Charts



Quantitative Benchmarking



Shows where benchmarked project lies in relation to historical database of similar projects.

Data is compared for schedule, effort, productivity, and staffing vs. size along standard deviation lines.

Using Quantitative Software Estimation to Support V&V Planning and Execution

Making software schedules and associated V&V activities more predictable

Estimating V&V Effort and Schedule

Within the control of the V&V team:

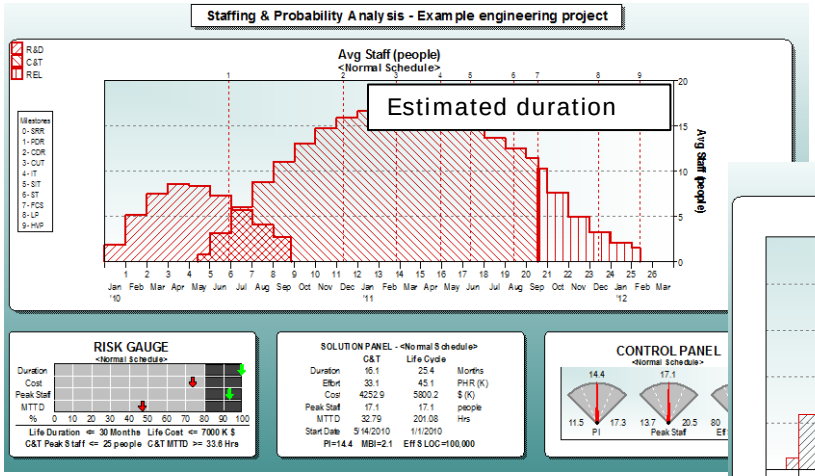
- Duration and effort for complete inspection or testing cycle

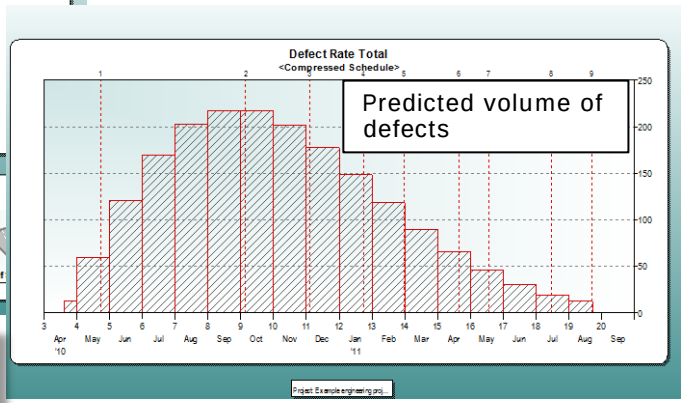
+

Outside the control of the V&V team:

- Delivery date for the software
- Volume of defects and rework
- Number of test cycles required to reach target quality

Quantitative estimation techniques can help predict these items.

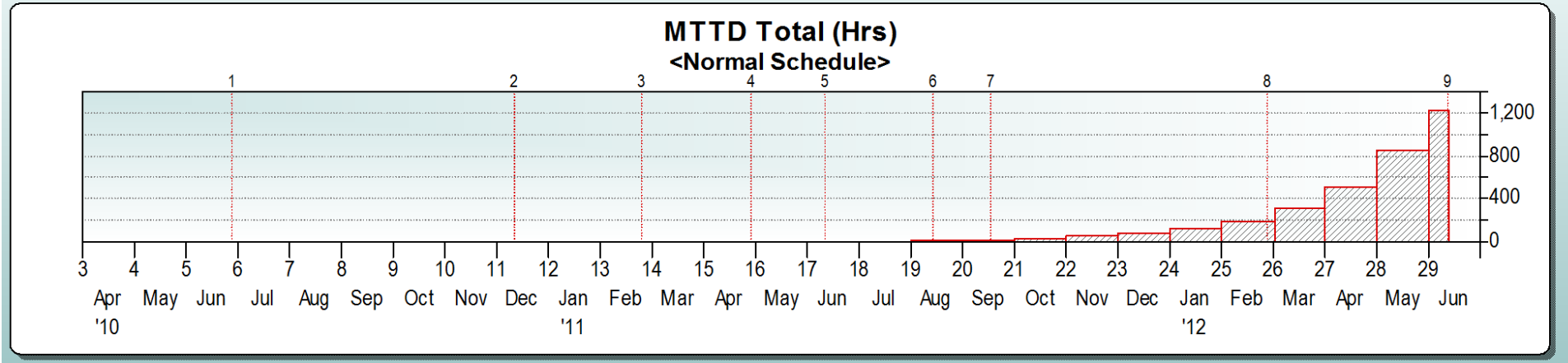


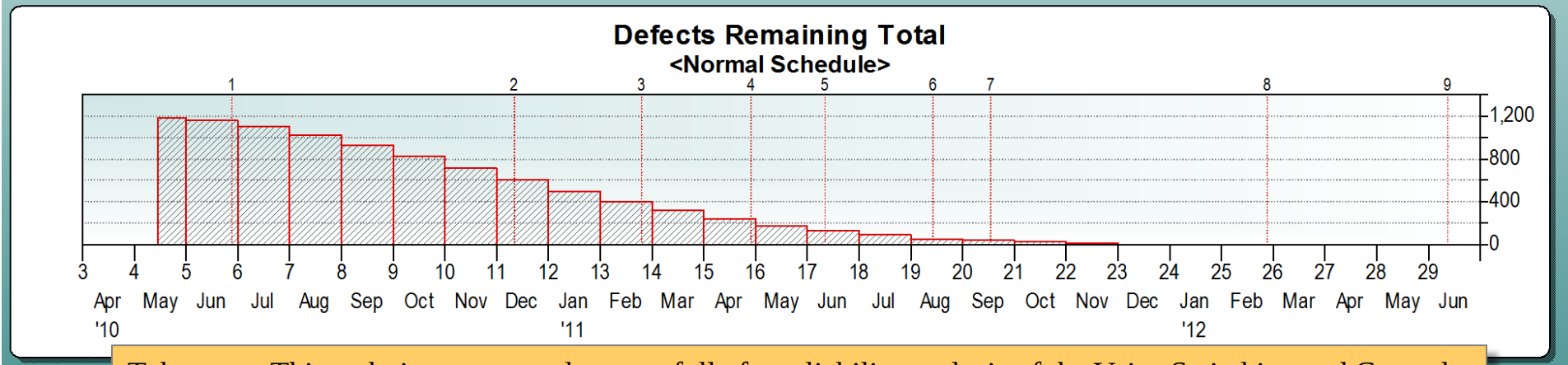


Takeaway: Quantitative estimation can help make V&V effort and duration more predictable

Accurately predicting when software is likely to reach target reliability

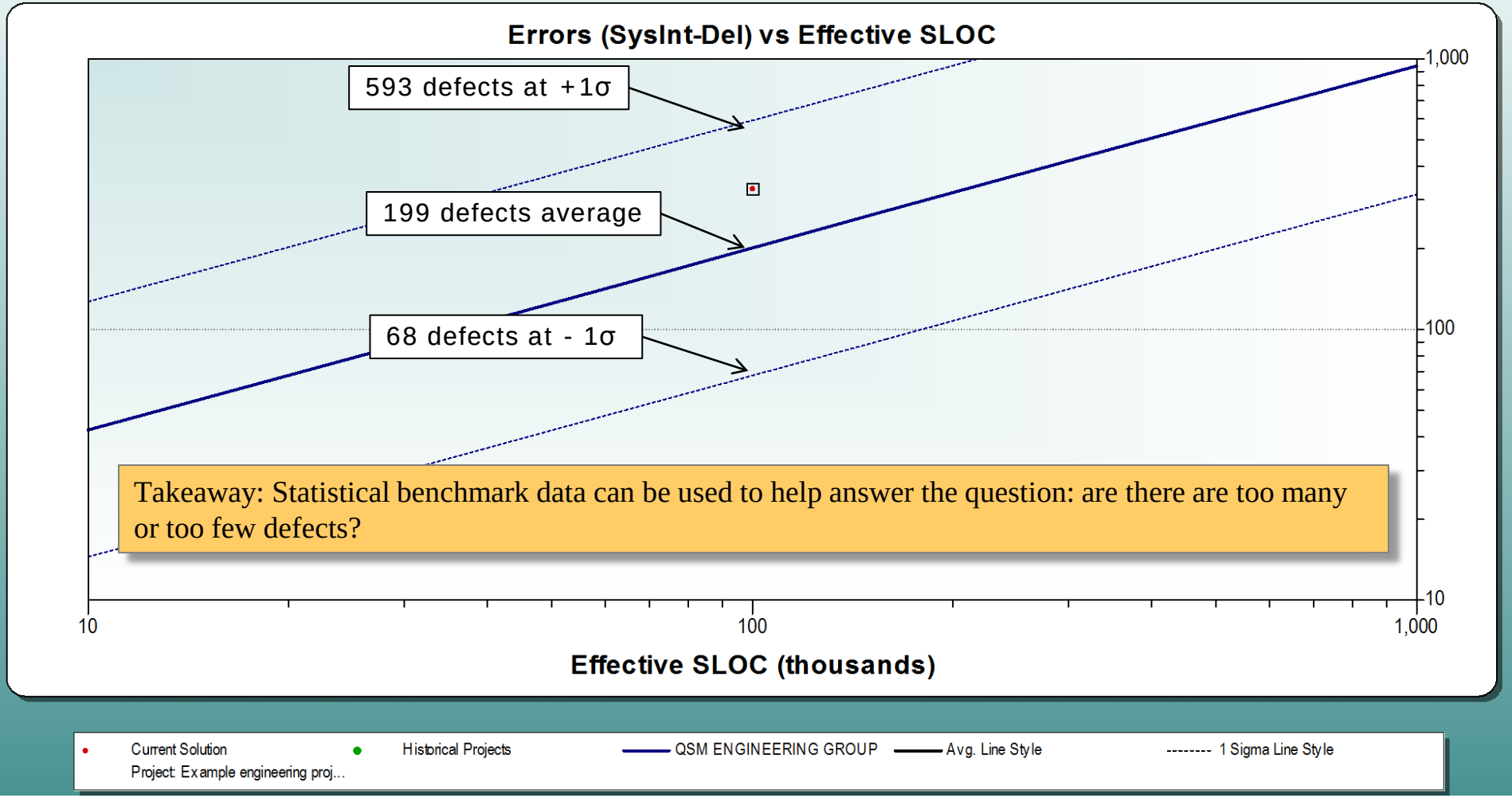
Total Defect Estimate - Example engineering project





Takeaway: This technique was used successfully for reliability analysis of the Voice Switching and Control System (VSCS) software at FAA prior to delivery to ensure it was stable enough to deliver and deploy

Understanding whether the number of actual defects discovered compares favorably with other projects



Examples of Quantitative Software Estimation Techniques at the FAA

Example Use of Quantitative Software Estimation Techniques at the FAA

- **RRI Program**. Independent Estimate and Risk Assessment for the FAA for a new generation air-ground router and communication system being built by five separate contractors (three US, one French, one Irish) building different pieces of the system.
- **Terrorist Screening Software**. Estimates of schedule cost and risk for each of 9 airlines to build terrorist screening software to use in airport screening departure lounges. Recommended to FAA reasonable costs for each system for negotiation between airline and FAA. Validated estimates using SLIM Metrics and QSM database.
- **OASIC**. Bid Evaluation of competitors to do an upgrade to a flight planning system. Calibrated historic data, did an analysis of each vendor's bid and provided some alternatives for FAA to consider.
- **URET**. Independent estimate of an FAA system prior to award. Calibration data and defect analysis of a historic project by the same vendor was done for tuning purposes.
- **STARS**. Independent estimate of an airfield radar system for the FAA. Calibration and analysis of a vendor bid prior to award. Emphasis on whether they could do the project in the specified time frame.
- **ITWS**. Competitive Bid Evaluation of 3 vendors competing for a new weather system for the FAA. Calibration of each vendor's historic data, a baseline government should cost scenario for comparison, and evaluation of each vendor's bid compared with the baseline, QSM tend lines, and specially tailored data sample. Determine whether bids were consistent with past performance.
- **VSCS**. Reliability analysis of the software prior to delivery to ensure it was stable enough to deliver and deploy.
- **Expert Witness Case**. Served as expert witness in a protest of termination of a contract by FAA against the prime contractor in the building of Radio Control Equipment for air traffic control. FAA asserted that Prime Contractor did not perform. QSM examined the plans and data concerning the software development to compared it with industry norms for real time control software to see if the progress, effort, and defect performance was consistent with others building software of a similar nature. Case was settled out of court about two weeks prior to trial.
- **Airfield Radar System**. Bid evaluation of contractor's ability to perform work required on airfield based radar system. Once the project was under way, performed monitoring and control and re-planning.

Conclusion

- Quantitative estimation supports more objective management decisions.
- Significant insight can be gained from a few core software metrics: size, duration, staffing and defects.
- Improved predictability makes everyone's job becomes easier, including V&V staff.

Contact

Joseph Madden
Vice President

Quantitative Software Management, Inc.
2000 Corporate Ridge; Suite 700
McLean, VA 22102
O 703.790.0055
Email: **joseph.madden@qsm.com**

www.qsm.com