Leveraging the Power of Historical Data in Software Cost Estimation

Presented at the Practical Software Measurement Users Group

June 12-16, 2017
Crystal City, VA

DASA-CE MAIS/IT Team
Presenter: Taylor Putnam-Majarian & John Staiger
Quantitative Software Management, Inc.
2010 Corporate Ridge, Suite 500
McLean, VA 22102
703.790.0055 ● 703.749.3795 (fax)
info@qsm.com ● www.qsm.com

INTRODUCTION

The Spanish philosopher, George Santayana, is credited with the observation: **“…those who do not learn from the mistakes of history are doomed to repeat them…”**

Why Collect Historical Data?

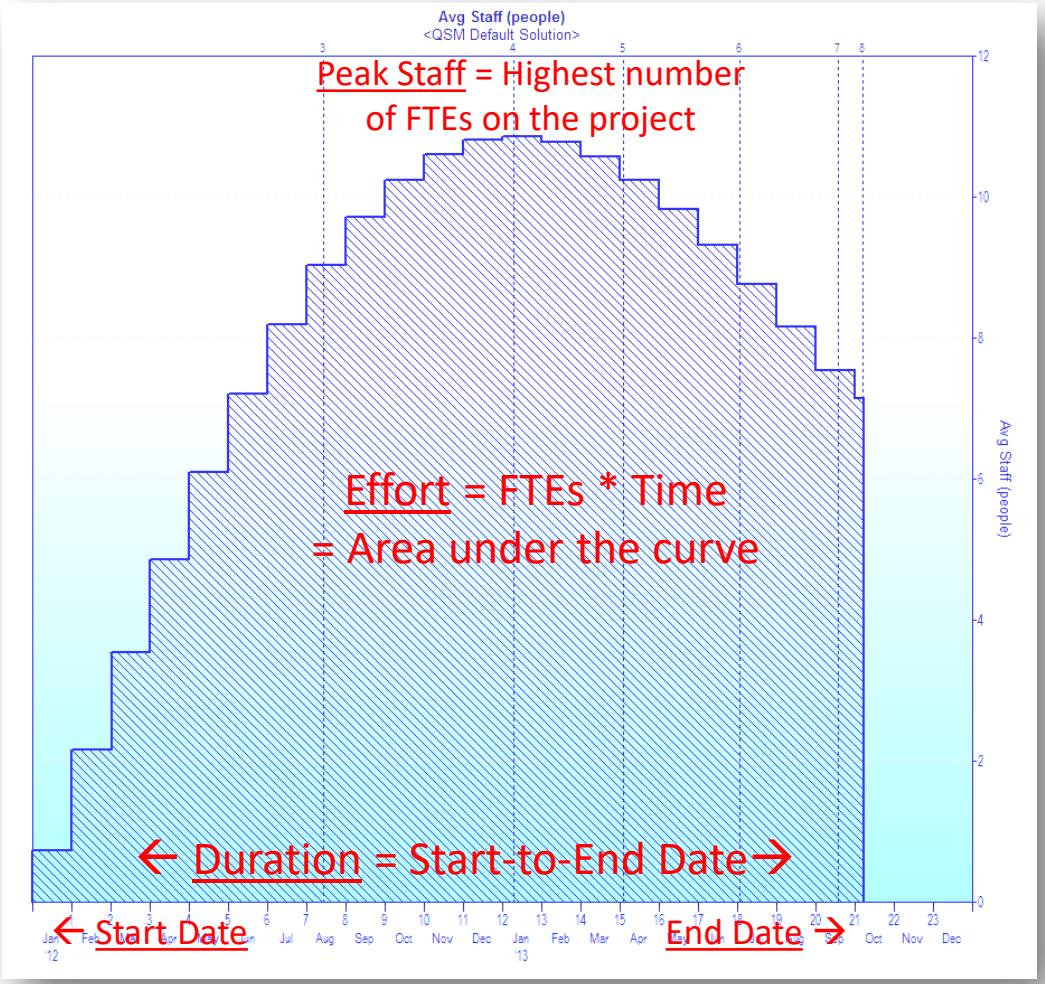
Data collection is the foundation of project estimation, tracking, and process improvement.

Establishing a repository of historical data can be useful for a variety of reasons:

- Promotes good record-keeping,
- Can be used to develop validated performance benchmarks,
- Accounts for variations in cost and schedule performance,
- Supports statistically validated trend(s) analysis,
- Helps make estimates defensible, and
- Can be used to bound productivity assumptions.

COLLECTING HISTORICAL DATA

Metrics of Interest



Phase Core Metrics:

Each phase has certain core values:

- Start Date
- End Date
- Duration
- Effort
- Peak Staff

Lifecycle Metrics:

- Project Size
- Defects Discovered in Testing

Where Do You Find Data?

When beginning the data collection process, it is important to identify potential sources of data.

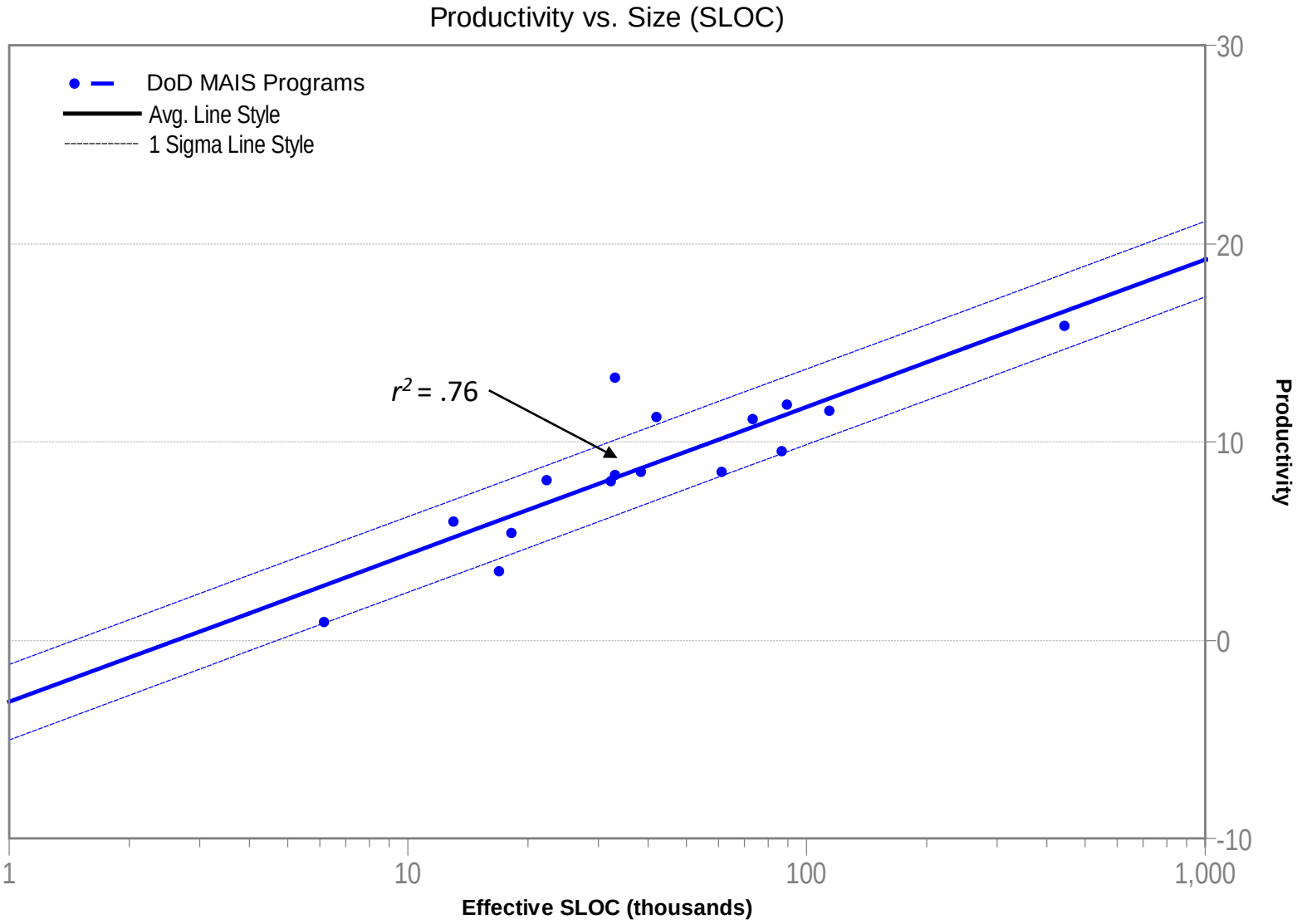
Usually this information can be found in the following artifacts:

- SRDR, CARD, and ACEIT® files
- Project schedule chart used in briefings
- Vendor Microsoft Project® file, Clarity® / PPM® export, or similar detailed planning tool
- Requirements document

Data Call

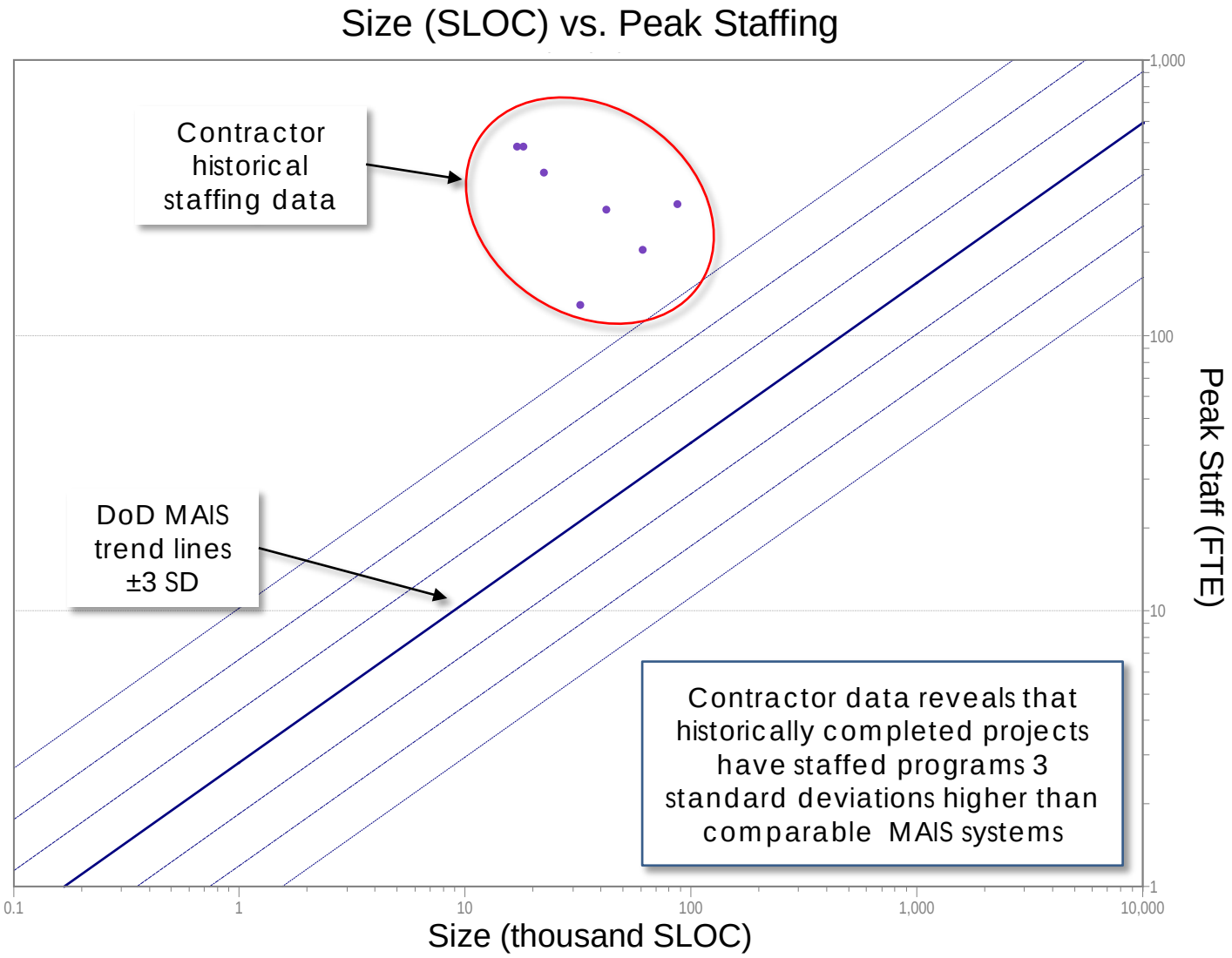
	A	B	C	D	E	F
1	**Proposed Standard SW Metrics for <system release name>**					
2	**Core Metrics (Must-Have items are highlighted)**					
3	**Category**	**Baseline Plan**	**Current Plan**	**Actual**	**Details**	**Rationale**
4	Measure	*As-of date*	*As-of date*	*As-of date*	*(See Data Dictionary tab for additional guidance)*	
5	**Size**					
6	SLOC (New)[1]	#	#	#	SW newly written for this system release.	New SW size provides a means to normalize other measures. Plan vs. Actual provides a means to conduct variance analysis.
7	SLOC (Modified)[2]	#	#	#	Changes made to baseline SW for this system release.	Modified SW size provides a means to normalize other measures. Plan vs. Actual provides a means to conduct variance analysis. Typically combined with New SLOC for analytical purposes.
8	SLOC (Reused)[3]	#	#	#	Unmodified SW brought over from a prior system release or other source (code library, COTS, etc.)	Reused SW size indicates how well existing SW (COTS, GOTS, NDI, prior system releases) is leveraged to deliver functionality for a given system release. However, reuse requires effort to integrate with newly developed SW, which tends to reduce productivities achieved through reuse.
9	Size Growth				Percentage SLOC growth from initial baseline plan to actual completion (derived)	Size growth is an indication of how well scope control is being practiced. Can be helpful to measure growth between multiple baseline plans, or other intermediate stages (e.g., system releases).
10	**Schedule/Duration (state whether calendar or fiscal year if applicable)**					
11	Feasibility Study Start Date[4]	date	date	date	Start Date of Feasibility: During the Feasibility phase, cost and technical feasibility are established and very high level software requirements are defined.	Phase start date bounds all effort, SLOC and duration for a given phase. Comparing baseline start date to actual start date is an indicator of the "lateness" of this phase.
12	Feasibility Study End Date[4]	date	date	date	End Date of Feasibility: During the Feasibility phase, cost and technical feasibility are established and very high level software requirements are defined.	Phase end date bounds all effort, SLOC and duration for a given phase. Comparing baseline end date to actual end date is an indicator of the "lateness" of this phase.
13	Requirements & Design Start Date[4]	date	date	date	Start of R&D: During the Requirements and Design Phase, the detailed requirements are identified and documented and the high level design is completed. Include SRDR example: "Software Requirements Analysis"	Phase start date bounds all effort, SLOC and duration for a given phase. Comparing baseline start date to actual start date is an indicator of the "lateness" of this phase.
14	Requirements & Design End Date[4]	date	date	date	End of R&D: During the Requirements and Design Phase, the detailed requirements are identified and documented and the high level design is completed. Include SRDR example: "Software Requirements Analysis"	Phase end date bounds all effort, SLOC and duration for a given phase. Comparing baseline end date to actual end date is an indicator of the "lateness" of this phase.
					Start of CUT: The Code and Unit Test Phase includes detailed design, coding integration and system level testing. This phase	

Defense MAIS Trends



USING TREND LINES

Assessing Past Performance

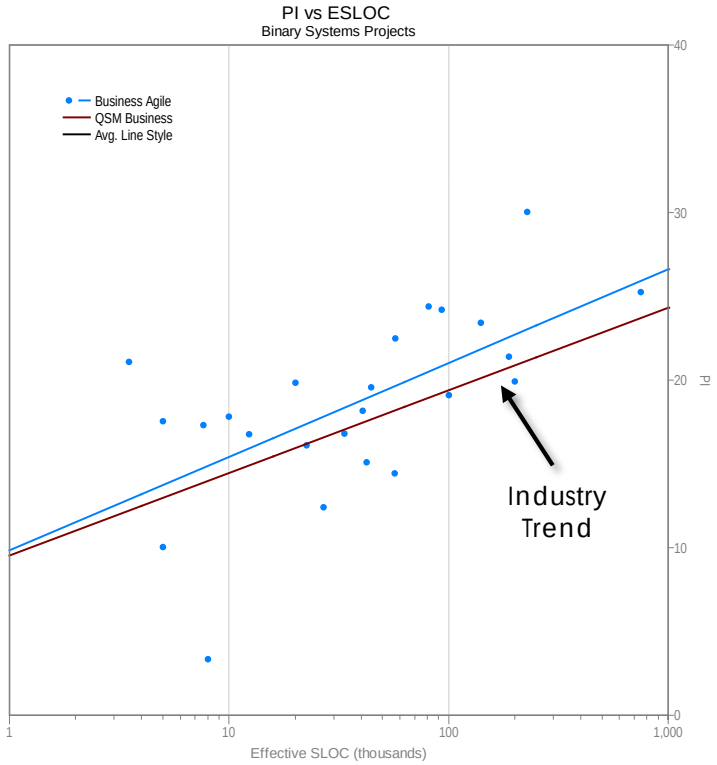


Core Metrics to Evaluate

As a first step, it's a good idea to assess the following metrics. Compare them against industry trend lines and/or internal trend lines customized to your organization's environment.

- Productivity
- Duration (Time to Market)
- Effort Expended
- Staff
- Reliability/ Defects (if available)

Quantify differences between trend lines.



Production Equation

$$Size = PI\ (time)^{4/3}\ (effort)^{1/3}$$




Delivered System Size proportional to Effort over Time at some Productivity

A measure of Value Delivered	**A measure of** Resources Expended	**A measure of** Duration Required	**A measure of** Capability and Difficulty of the task

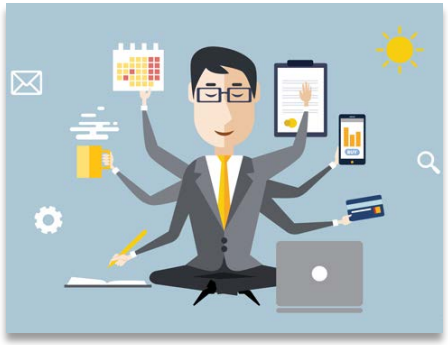
The Production Equation can be rearranged algebraically to solve for any of the above variables

Estimating Future Releases

If you know…

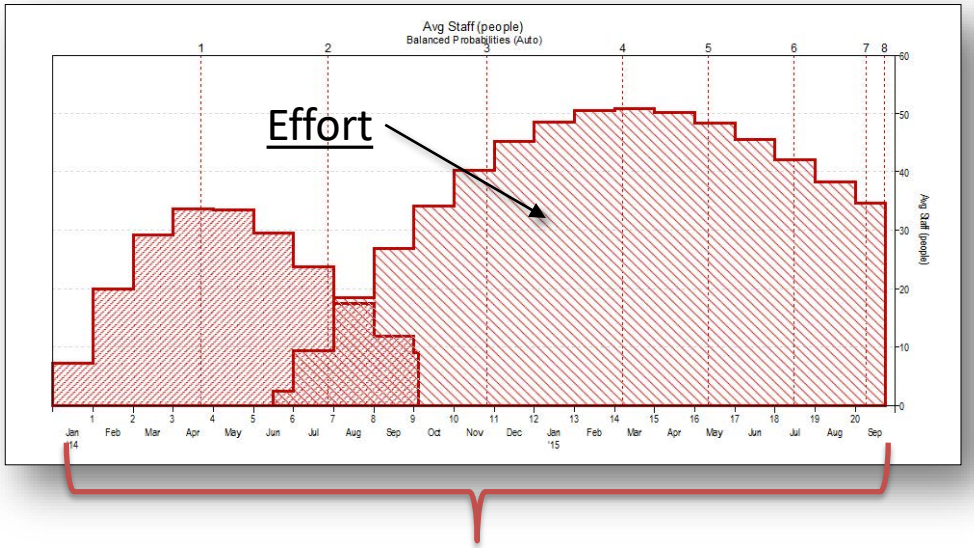
Delivered System Size

&



Productivity

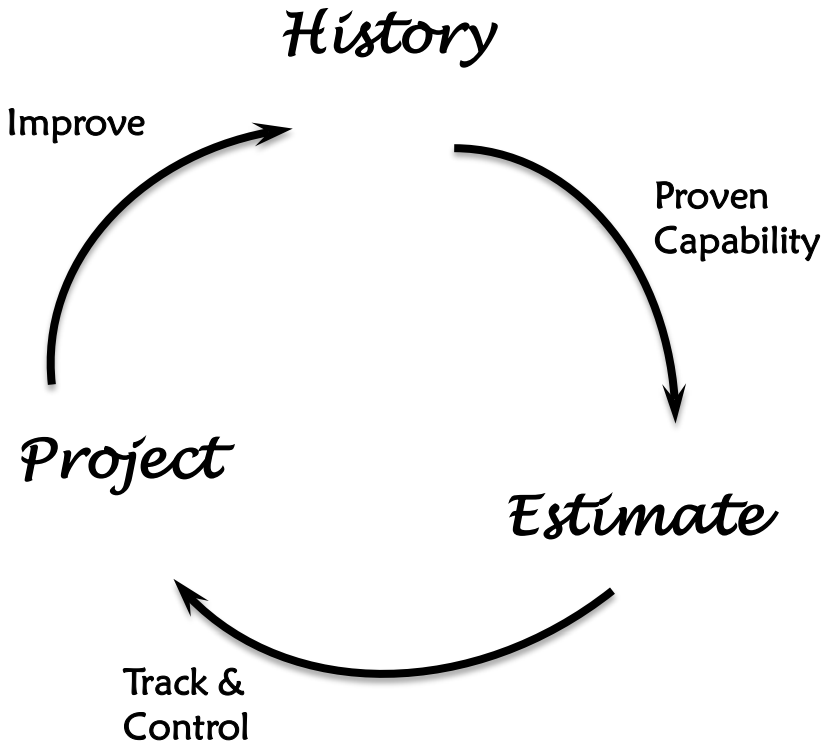
You can calculate…



REFINING THE CYCLE

Using Historical Data Throughout the Lifecycle

1. Use data to estimate early in the planning stage
2. Track actual performance against the estimate
3. Collect data on the actual performance
4. Update the trend lines
5. Repeat



QUESTIONS?

DASA-CE MAIS/IT Team
Presenter: Taylor Putnam-Majarian & John Staiger
Quantitative Software Management, Inc.
2010 Corporate Ridge, Suite 500
McLean, VA 22102
703.790.0055 ● 703.749.3795 (fax)
info@qsm.com ● www.qsm.com