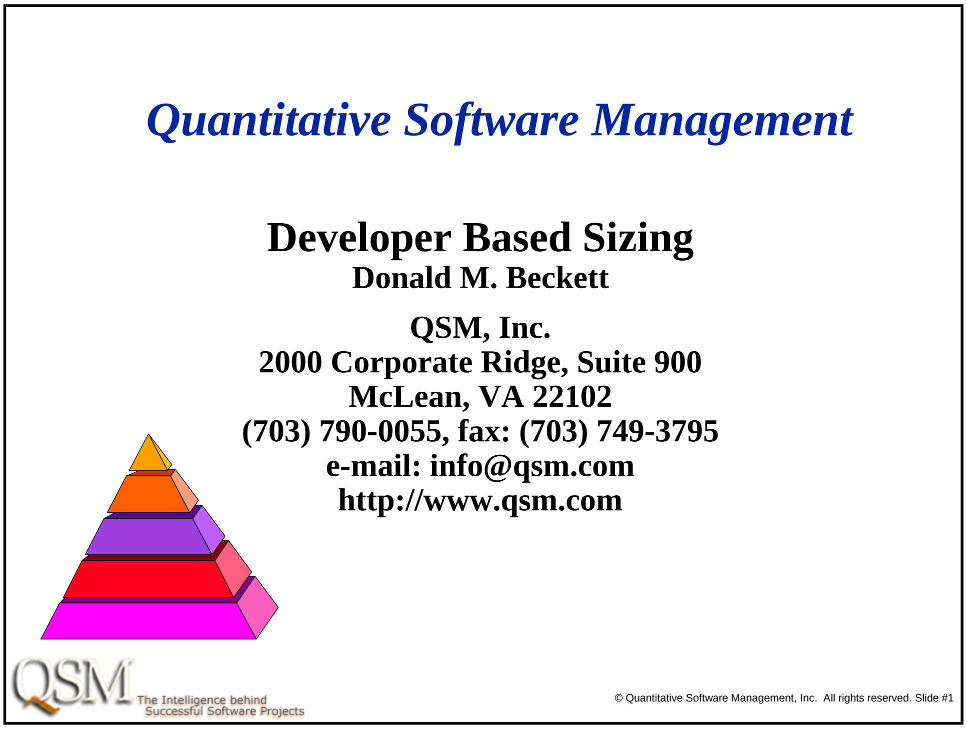


Quantitative Software Management

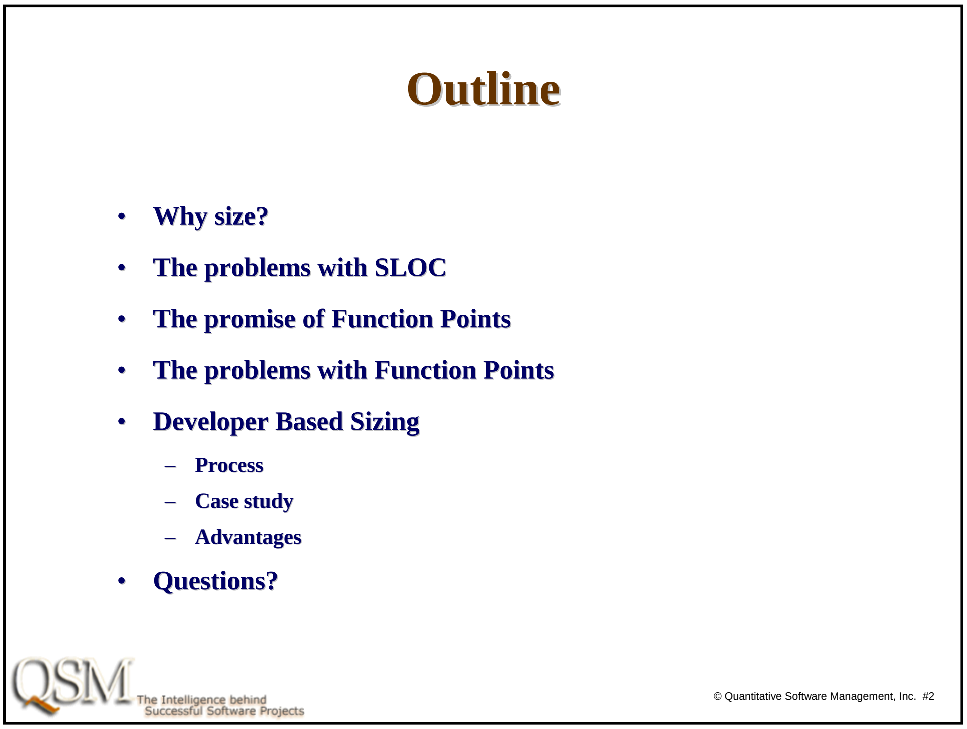
Developer Based Sizing

Donald M. Beckett

QSM, Inc.
2000 Corporate Ridge, Suite 900
McLean, VA 22102
(703) 790-0055, fax: (703) 749-3795
e-mail: info@qsm.com
http://www.qsm.com

The Intelligence behind
Successful Software Projects

© Quantitative Software Management, Inc. All rights reserved. Slide #1

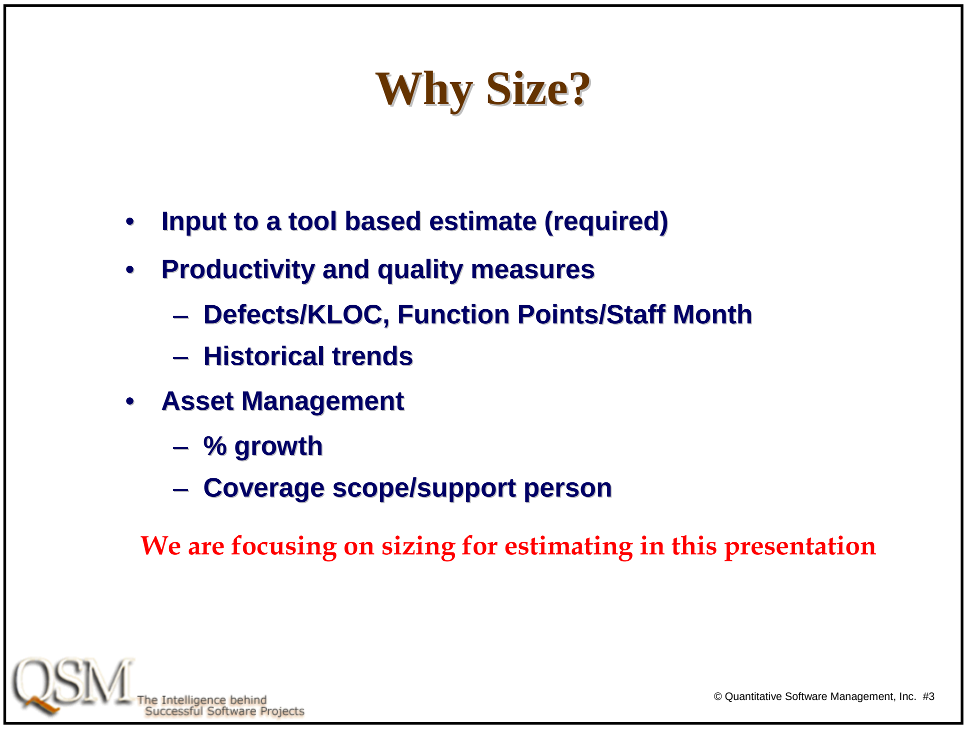


Outline

- Why size?
- The problems with SLOC
- The promise of Function Points
- The problems with Function Points
- Developer Based Sizing
  - Process
  - Case study
  - Advantages
- Questions?

The Intelligence behind
Successful Software Projects

© Quantitative Software Management, Inc. #2



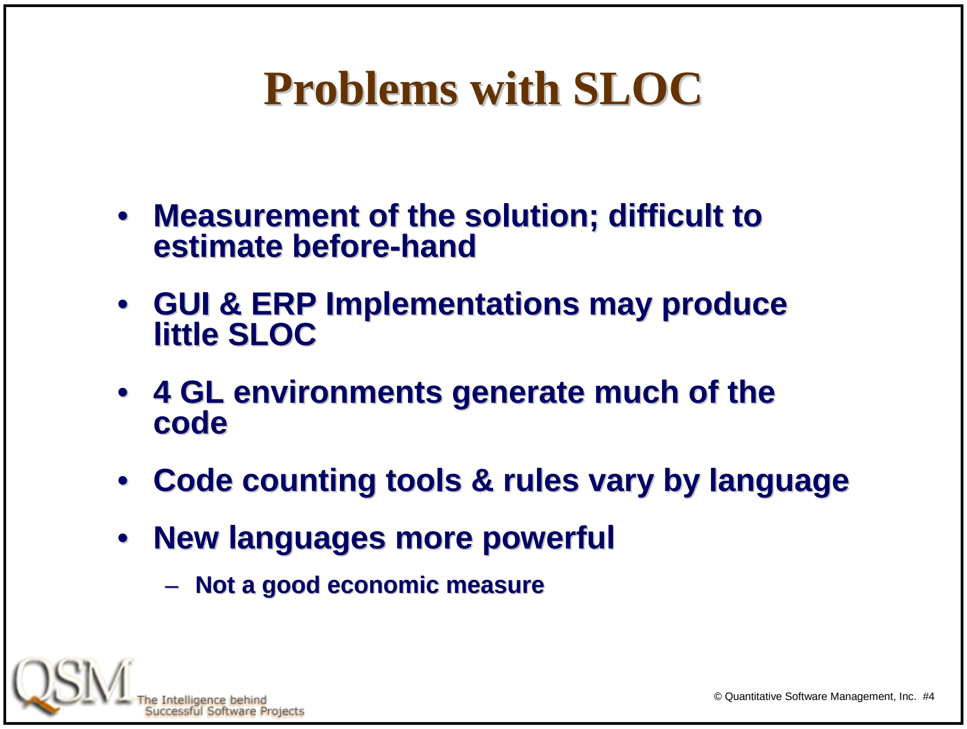
Why Size?

- Input to a tool based estimate (required)
- Productivity and quality measures
  - Defects/KLOC, Function Points/Staff Month
  - Historical trends
- Asset Management
  - % growth
  - Coverage scope/support person

We are focusing on sizing for estimating in this presentation

QSM The Intelligence behind Successful Software Projects

© Quantitative Software Management, Inc. #3

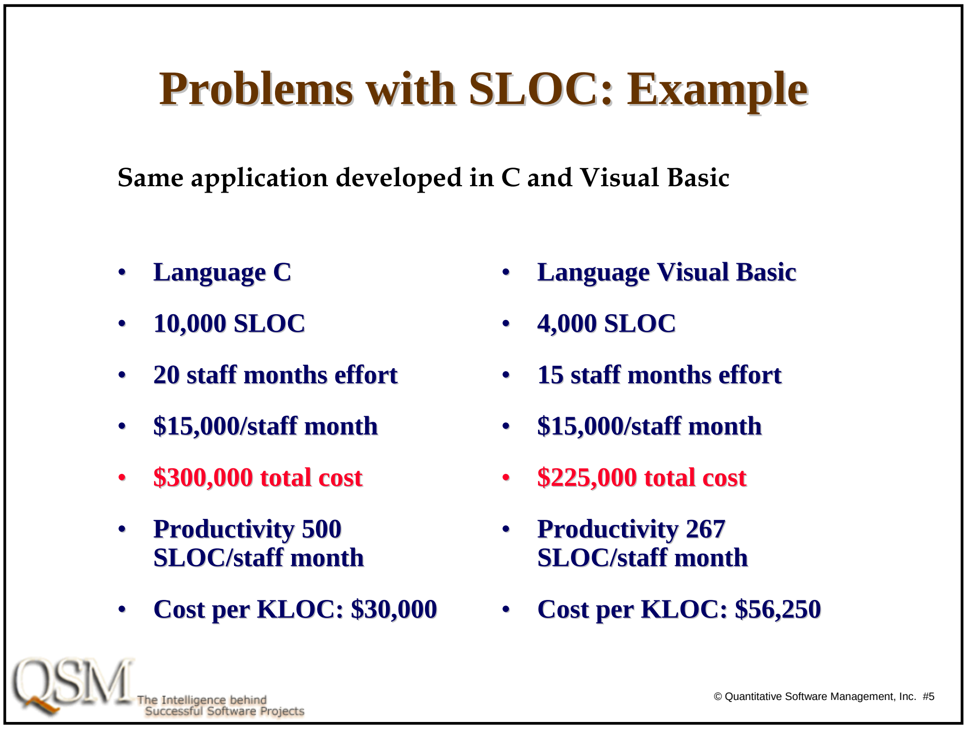


Problems with SLOC

- Measurement of the solution; difficult to estimate before-hand
- GUI & ERP Implementations may produce little SLOC
- 4 GL environments generate much of the code
- Code counting tools & rules vary by language
- New languages more powerful
  - Not a good economic measure

QSM The Intelligence behind Successful Software Projects

© Quantitative Software Management, Inc. #4



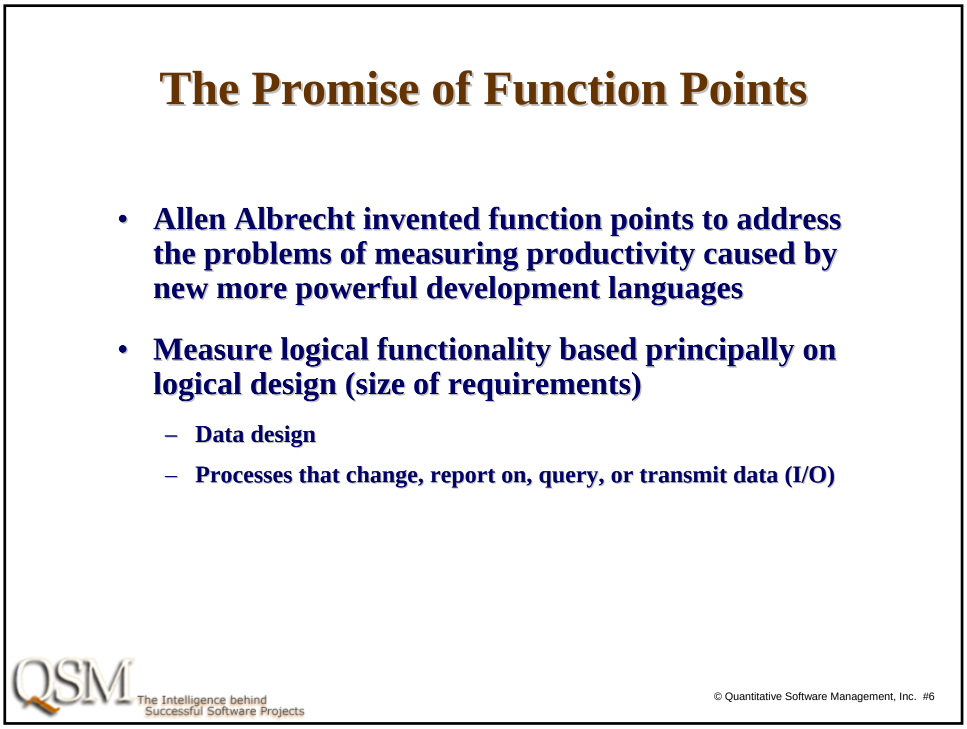
Problems with SLOC: Example

Same application developed in C and Visual Basic

• Language C	• Language Visual Basic
• 10,000 SLOC	• 4,000 SLOC
• 20 staff months effort	• 15 staff months effort
• $15,000/staff month	• $15,000/staff month
• $300,000 total cost	• $225,000 total cost
• Productivity 500 SLOC/staff month	• Productivity 267 SLOC/staff month
• Cost per KLOC: $30,000	• Cost per KLOC: $56,250

The Intelligence behind Successful Software Projects

© Quantitative Software Management, Inc. #5

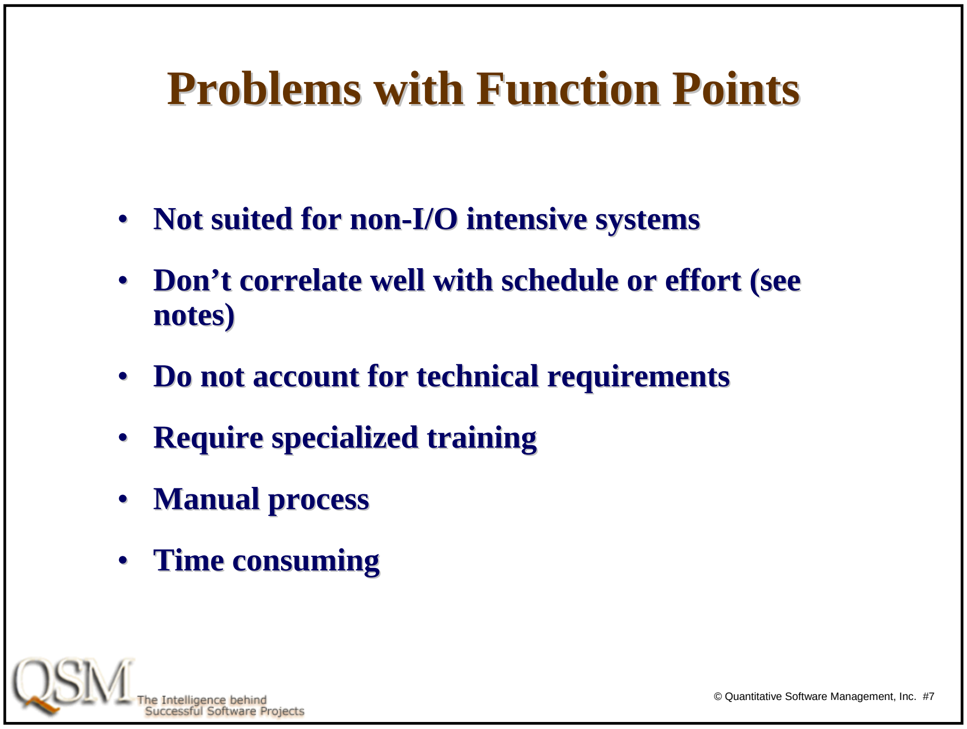


The Promise of Function Points

- Allen Albrecht invented function points to address the problems of measuring productivity caused by new more powerful development languages
- Measure logical functionality based principally on logical design (size of requirements)
  - Data design
  - Processes that change, report on, query, or transmit data (I/O)

The Intelligence behind Successful Software Projects

© Quantitative Software Management, Inc. #6

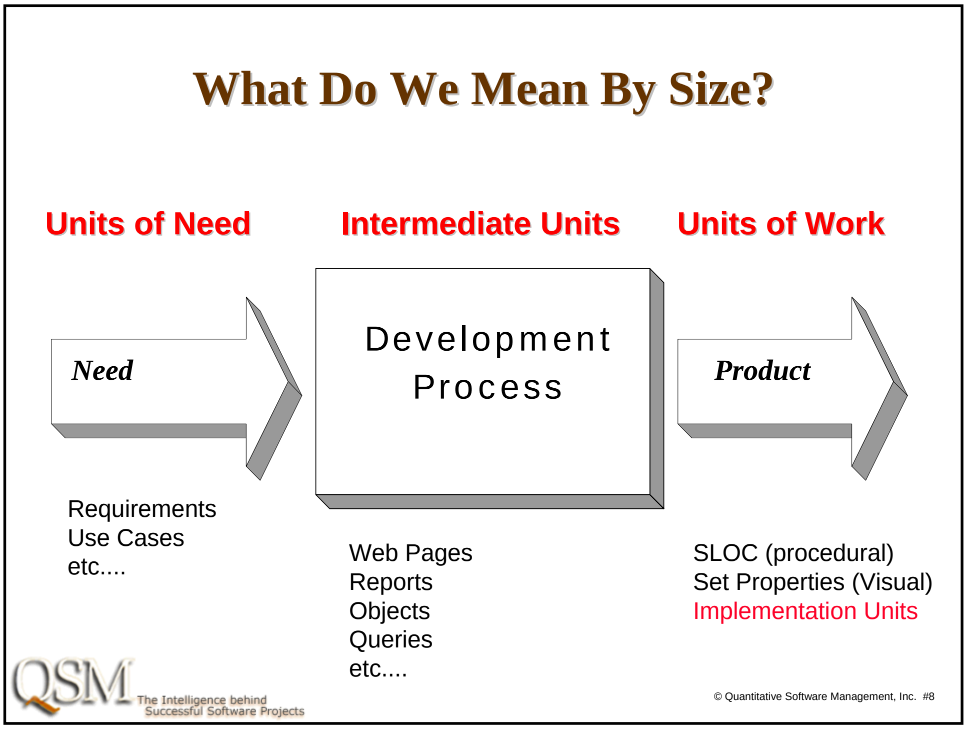


Problems with Function Points

- Not suited for non-I/O intensive systems
- Don’t correlate well with schedule or effort (see notes)
- Do not account for technical requirements
- Require specialized training
- Manual process
- Time consuming

The Intelligence behind Successful Software Projects

© Quantitative Software Management, Inc. #7



What Do We Mean By Size?

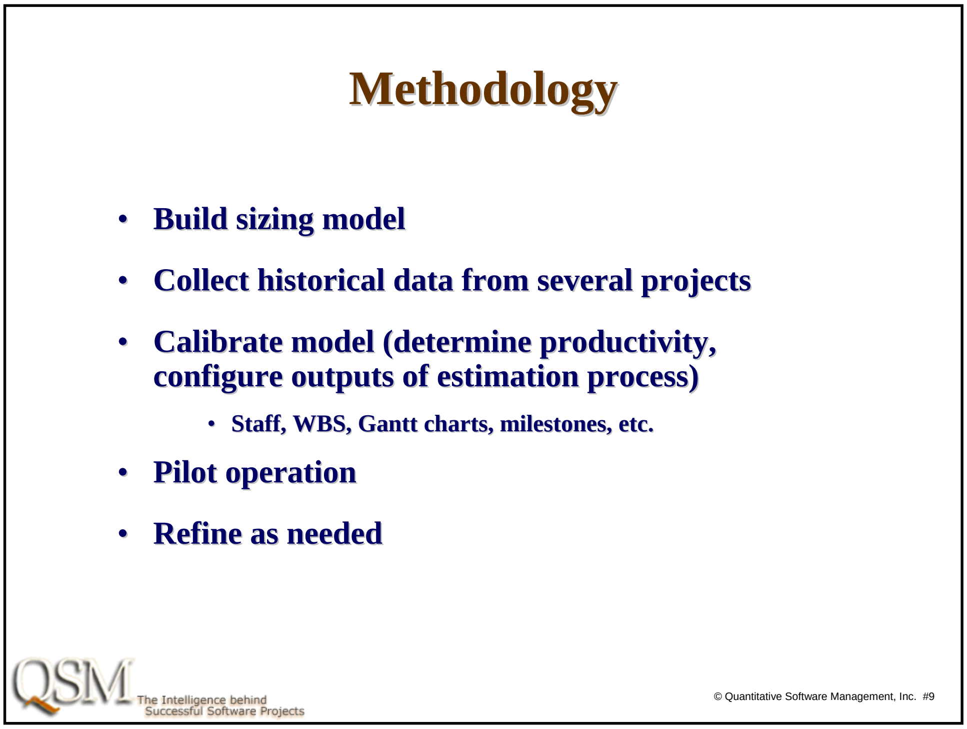
Requirements
Use Cases
etc....

Web Pages
Reports
Objects
Queries
etc....

SLOC (procedural)
Set Properties (Visual)
Implementation Units

The Intelligence behind Successful Software Projects

© Quantitative Software Management, Inc. #8

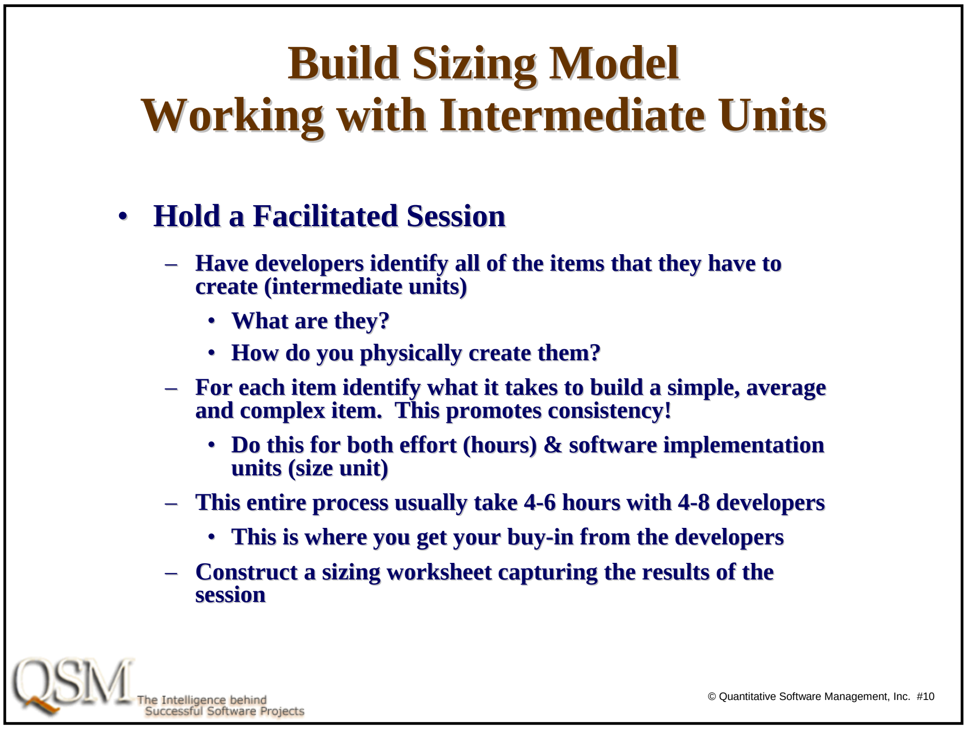


Methodology

- Build sizing model
- Collect historical data from several projects
- Calibrate model (determine productivity, configure outputs of estimation process)
  - Staff, WBS, Gantt charts, milestones, etc.
- Pilot operation
- Refine as needed

QSM The Intelligence behind Successful Software Projects

© Quantitative Software Management, Inc. #9



Build Sizing Model Working with Intermediate Units

- Hold a Facilitated Session
  - Have developers identify all of the items that they have to create (intermediate units)
    - What are they?
    - How do you physically create them?
  - For each item identify what it takes to build a simple, average and complex item. This promotes consistency!
    - Do this for both effort (hours) & software implementation units (size unit)
  - This entire process usually take 4-6 hours with 4-8 developers
    - This is where you get your buy-in from the developers
  - Construct a sizing worksheet capturing the results of the session

QSM The Intelligence behind Successful Software Projects

© Quantitative Software Management, Inc. #10

Facilitated Session Hints

- **Developer productivity varies significantly**
  - **Make certain that both optimistic and pessimistic views are represented**
  - **Let the developers arrive at a consensus**
  - **Don’t allow personalities to dominate (be a facilitator!)**
- **Remember:**
  - **Developers focus on what they have to do**
  - **Coding & unit testing are typically around 30% of total project effort. (Effort is a relative measure of size)**
  - **Remember to factor other activities into the estimation model**

The Intelligence behind
Successful Software Projects

© Quantitative Software Management, Inc. #11

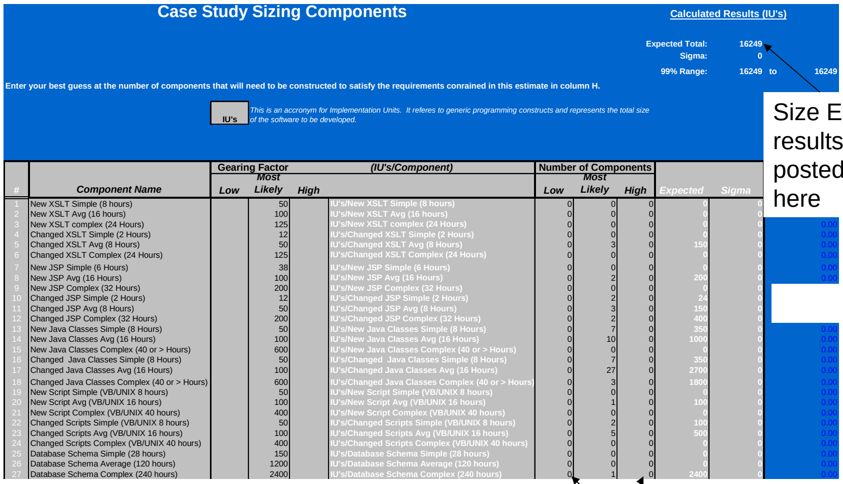
Sizing Guidelines: Repeatable Process

Case Study Sizing Components	Guidelines for bounding deliverable and work effort
New XSLT Simple (8 hours)	One transform/no paging/output only
New XSLT Avg (16 hours)	1-2 transforms/paging/no computed links
New XSLT complex (24 Hours)	Greater than 3 transforms/ paging/computed links
Changed XSLT Simple (2 Hours)	One transform/no paging/output only
Changed XSLT Avg (8 Hours)	1-2 transforms/paging/no computed links
Changed XSLT Complex (24 Hours)	Greater than 3 transforms/ paging/computed links
New JSP Simple (6 Hours)	Simple redirect/ display error message/ simple results set/ 1 call
New JSP Avg (16 Hours)	Simple form (15 controls)/ data validations/pop ups/ user inputs
New JSP Complex (32 Hours)	Prefilled forms/ User preferences/ data validation/ pop ups/ user input/ database calls
Changed JSP Simple (2 Hours)	changing a format
Changed JSP Avg (8 Hours)	adding more forms/ more controls/ more calls
Changed JSP Complex (32 Hours)	Rewrite of JSP/ adding more complex capabilities
New Java Classes Simple (8 Hours)	basic logic/ less than 1 page of non trivial code
New Java Classes Avg (16 Hours)	less than 2 pages of non trivial code
New Java Classes Complex (40 or > Hours)	5 or more pages of non trivial code
Changed Java Classes Simple (8 Hours)	basic logic/ less than 1 page of non trivial code
Changed Java Classes Avg (16 Hours)	less than 2 pages of non trivial code
Changed Java Classes Complex (40 or > Hours)	5 or more pages of non trivial code
New Script Simple (VB/UNIX 8 hours)	basic logic/ less than 1 page of non trivial code
New Script Avg (VB/UNIX 16 hours)	less than 2 pages of non trivial code
New Script Complex (VB/UNIX 40 hours)	5 or more pages of non trivial code
Changed Scripts Simple (VB/UNIX 8 hours)	basic logic/ less than 1 page of non trivial code
Changed Scripts Avg (VB/UNIX 16 hours)	less than 2 pages of non trivial code
Changed Scripts Complex (VB/UNIX 40 hours)	5 or more pages of non trivial code
Database Schema Simple (28 hours)	less than 5 tables/ apprximately 20 fields
Database Schema Average (120 hours)	5-15 tables/ 75 fields
Database Schema Complex (240 hours)	Greater than 30 tables/ 150 fields
Database Proceedures Simple (8 hours)	basic logic/ less than 1 page of non trivial code
Database Proceedures Average (16 hours)	less than 2 pages of non trivial code
Database Proceedures Complex (40 hours)	5 or more pages of non trivial code
SQL Query Simple (1 hour)	1-2 table touched
SQL Query Avg (2 hour)	3-4 tables touched
SQL Query Complex (4 hour)	greater than 5 tables touched
Data Migration Simple (40 hours)	1-20 elements/ attibute matching high/key same/ scrubbing low
Data Migration Avg (60 hours)	20-50 elements/ attribute matching moderate/ key mixed/ scrubbing moderate
Data Migration Complex (120 hours)	50 plus elements/ attibute matching low/ key different/ scrubbing high
Data Sync Simple (1 hours)	shareplex
Data Sync Avg/Complex (80 hours)	multiple 3rd party databases
Configuration Parameter File (4 hours)	set up configuration

The Intelligence behind
Successful Software Projects

© Quantitative Software Management, Inc. #12

Sizing Spreadsheet Example



Case Study Sizing Components

Calculated Results (IU's)

Expected Total: 16249
Sigma: 0
99% Range: 16249 to 16249

Enter your best guess at the number of components that will need to be constructed to satisfy the requirements conrained in this estimate in column H.

IU's — *This is an accronym for Implementation Units. It referes to generic programming constructs and represents the total size of the software to be developed.*

#	Component Name	Gearing Factor			(IU's/Component)	Number of Components					
		Low	Most Likely	High		Low	Most Likely	High	Expected	Sigma	
1	New XSLT Simple (8 hours)		50		IU's/New XSLT Simple (8 hours)	0	0	0	0	0	
2	New XSLT Avg (16 hours)		100		IU's/New XSLT Avg (16 hours)	0	0	0	0	0	
3	New XSLT complex (24 Hours)		125		IU's/New XSLT complex (24 Hours)	0	0	0	0	0	0.00
4	Changed XSLT Simple (2 Hours)		12		IU's/Changed XSLT Simple (2 Hours)	0	0	0	0	0	0.00
5	Changed XSLT Avg (8 Hours)		50		IU's/Changed XSLT Avg (8 Hours)	0	3	0	150	0	0.00
6	Changed XSLT Complex (24 Hours)		125		IU's/Changed XSLT Complex (24 Hours)	0	0	0	0	0	0.00
7	New JSP Simple (6 Hours)		38		IU's/New JSP Simple (6 Hours)	0	0	0	0	0	0.00
8	New JSP Avg (16 Hours)		100		IU's/New JSP Avg (16 Hours)	0	2	0	200	0	0.00
9	New JSP Complex (32 Hours)		200		IU's/New JSP Complex (32 Hours)	0	0	0	0	0	
10	Changed JSP Simple (2 Hours)		12		IU's/Changed JSP Simple (2 Hours)	0	2	0	24	0	
11	Changed JSP Avg (8 Hours)		50		IU's/Changed JSP Avg (8 Hours)	0	3	0	150	0	
12	Changed JSP Complex (32 Hours)		200		IU's/Changed JSP Complex (32 Hours)	0	2	0	400	0	
13	New Java Classes Simple (8 Hours)		50		IU's/New Java Classes Simple (8 Hours)	0	7	0	350	0	0.00
14	New Java Classes Avg (16 Hours)		100		IU's/New Java Classes Avg (16 Hours)	0	10	0	1000	0	0.00
15	New Java Classes Complex (40 or > Hours)		600		IU's/New Java Classes Complex (40 or > Hours)	0	0	0	0	0	0.00
16	Changed Java Classes Simple (8 Hours)		50		IU's/Changed Java Classes Simple (8 Hours)	0	7	0	350	0	0.00
17	Changed Java Classes Avg (16 Hours)		100		IU's/Changed Java Classes Avg (16 Hours)	0	27	0	2700	0	0.00
18	Changed Java Classes Complex (40 or > Hours)		600		IU's/Changed Java Classes Complex (40 or > Hours)	0	3	0	1800	0	0.00
19	New Script Simple (VB/UNIX 8 hours)		50		IU's/New Script Simple (VB/UNIX 8 hours)	0	0	0	0	0	0.00
20	New Script Avg (VB/UNIX 16 hours)		100		IU's/New Script Avg (VB/UNIX 16 hours)	0	1	0	100	0	0.00
21	New Script Complex (VB/UNIX 40 hours)		400		IU's/New Script Complex (VB/UNIX 40 hours)	0	0	0	0	0	0.00
22	Changed Scripts Simple (VB/UNIX 8 hours)		50		IU's/Changed Scripts Simple (VB/UNIX 8 hours)	0	2	0	100	0	0.00
23	Changed Scripts Avg (VB/UNIX 16 hours)		100		IU's/Changed Scripts Avg (VB/UNIX 16 hours)	0	5	0	500	0	0.00
24	Changed Scripts Complex (VB/UNIX 40 hours)		400		IU's/Changed Scripts Complex (VB/UNIX 40 hours)	0	0	0	0	0	0.00
25	Database Schema Simple (28 hours)		150		IU's/Database Schema Simple (28 hours)	0	0	0	0	0	0.00
26	Database Schema Average (120 hours)		1200		IU's/Database Schema Average (120 hours)	0	0	0	0	0	0.00
27	Database Schema Complex (240 hours)		2400		IU's/Database Schema Complex (240 hours)	0	1	0	2400	0	0.00

Size Estimate results get posted up here

Number of Implementation units per component

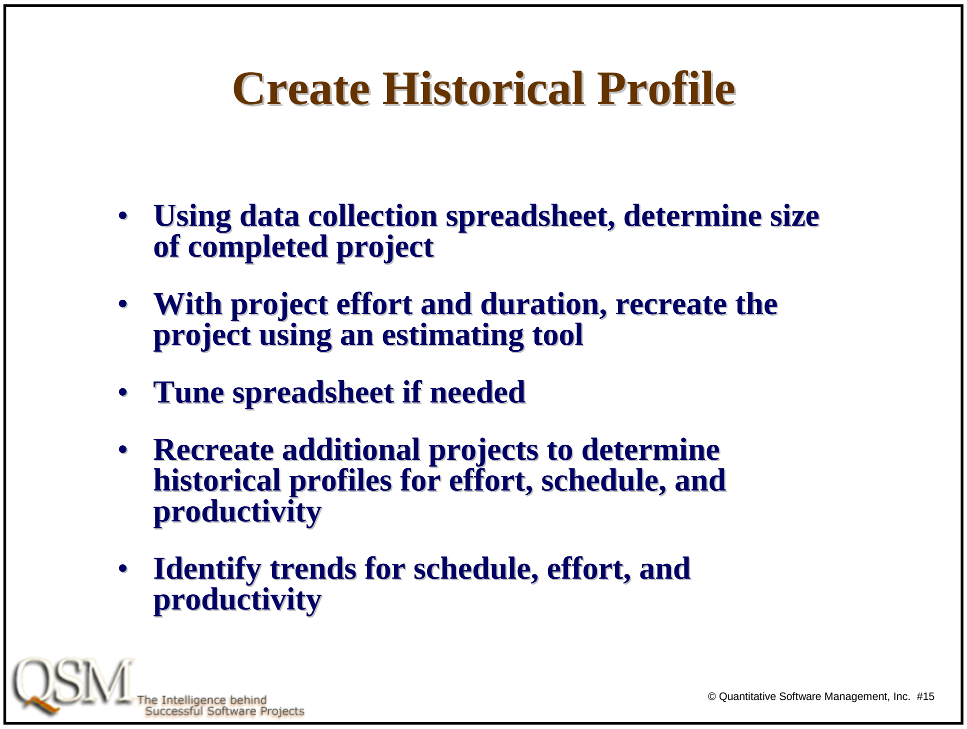
Estimate of the number of components goes here

The Intelligence behind Successful Software Projects

Methodology

- **Build sizing model**
- **Collect historical data from several projects**
- **Calibrate model (determine productivity, configure outputs of estimation process)**
  - **Staff, WBS, Gantt charts, milestones, etc.**
- **Pilot operation**
- **Refine as needed**

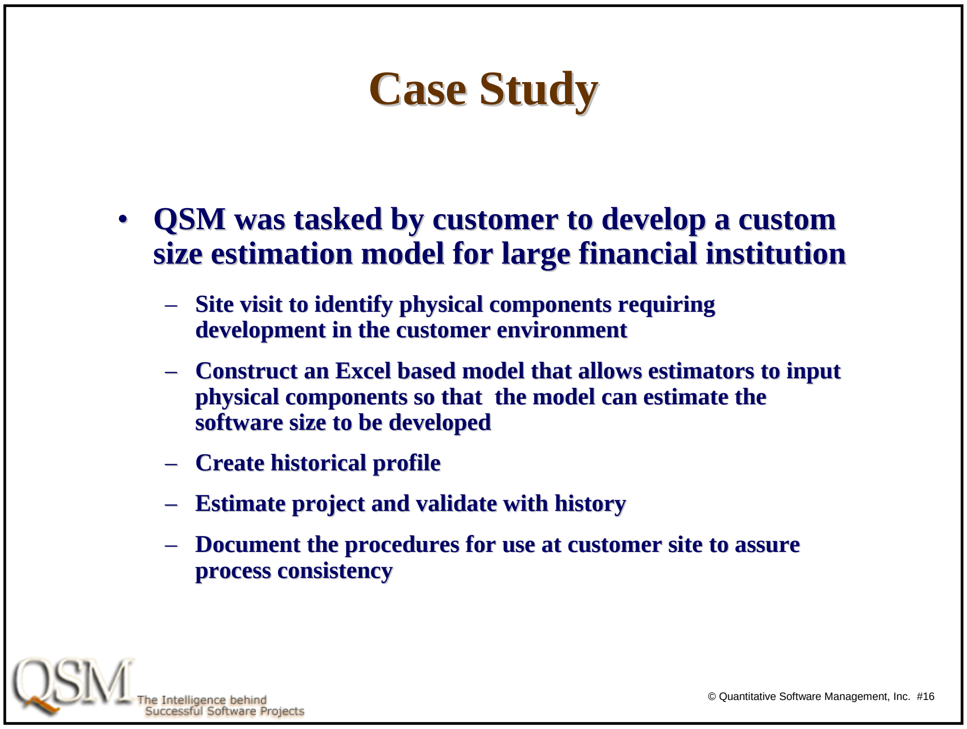
The Intelligence behind Successful Software Projects



Create Historical Profile

- Using data collection spreadsheet, determine size of completed project
- With project effort and duration, recreate the project using an estimating tool
- Tune spreadsheet if needed
- Recreate additional projects to determine historical profiles for effort, schedule, and productivity
- Identify trends for schedule, effort, and productivity

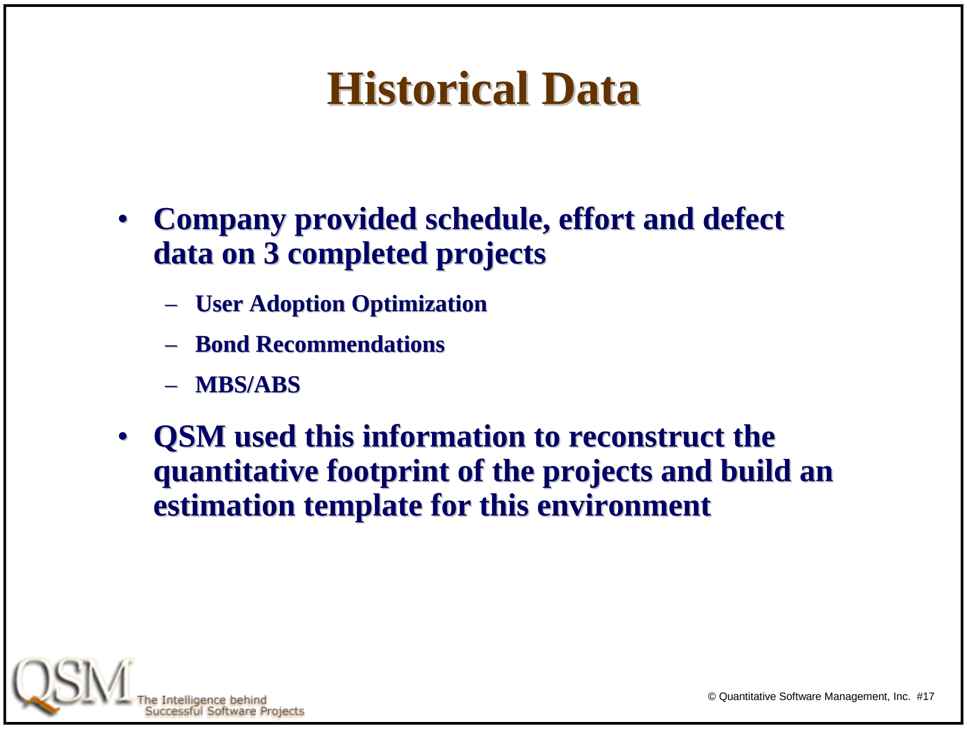
© Quantitative Software Management, Inc. #15



Case Study

- QSM was tasked by customer to develop a custom size estimation model for large financial institution
  - Site visit to identify physical components requiring development in the customer environment
  - Construct an Excel based model that allows estimators to input physical components so that the model can estimate the software size to be developed
  - Create historical profile
  - Estimate project and validate with history
  - Document the procedures for use at customer site to assure process consistency

© Quantitative Software Management, Inc. #16

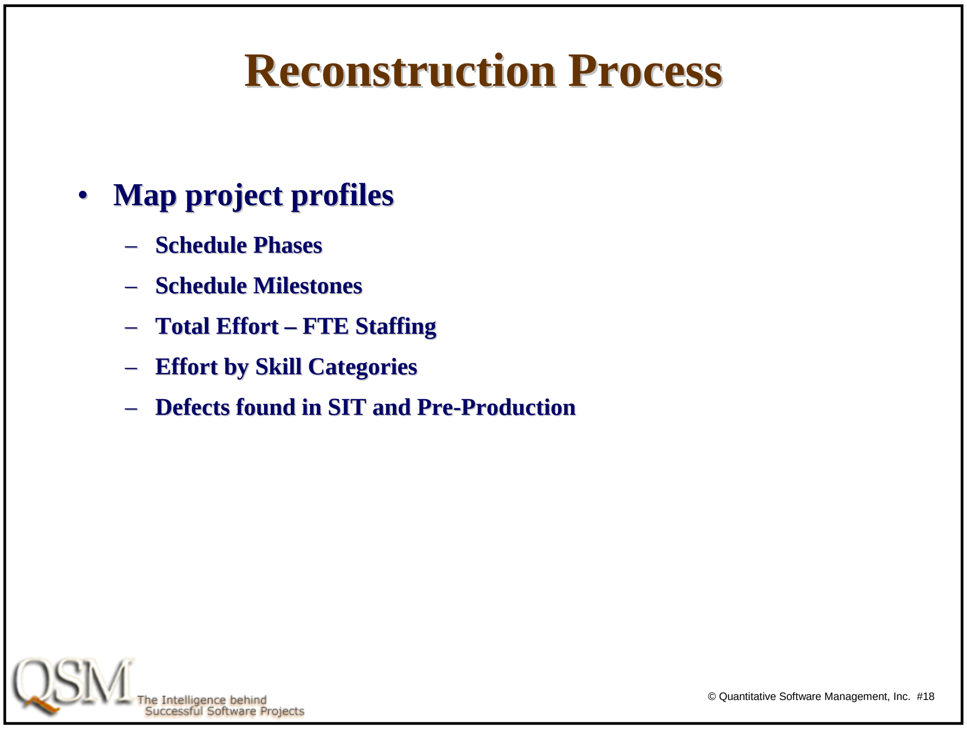


Historical Data

- Company provided schedule, effort and defect data on 3 completed projects
  - User Adoption Optimization
  - Bond Recommendations
  - MBS/ABS
- QSM used this information to reconstruct the quantitative footprint of the projects and build an estimation template for this environment

The Intelligence behind Successful Software Projects

© Quantitative Software Management, Inc. #17



Reconstruction Process

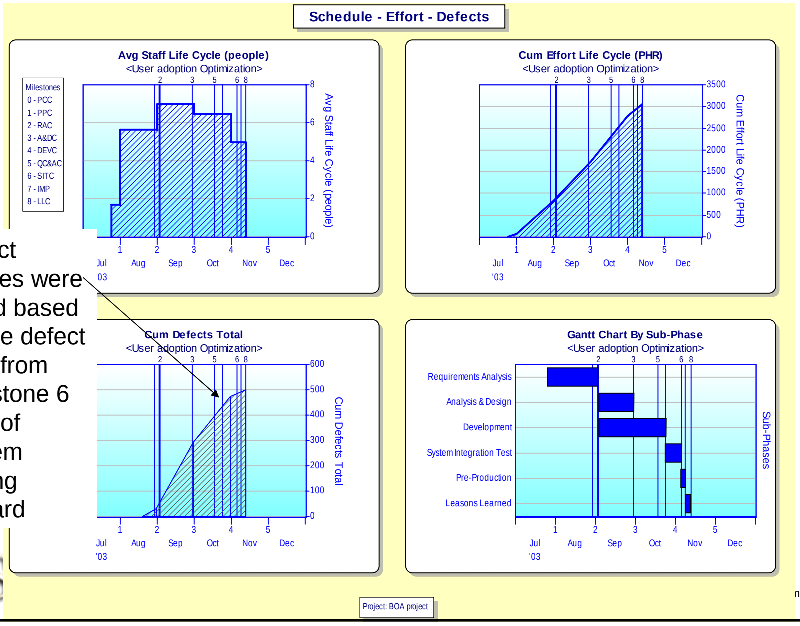
- Map project profiles
  - Schedule Phases
  - Schedule Milestones
  - Total Effort – FTE Staffing
  - Effort by Skill Categories
  - Defects found in SIT and Pre-Production

The Intelligence behind Successful Software Projects

© Quantitative Software Management, Inc. #18

User Adoption Optimization Project Reconstruction

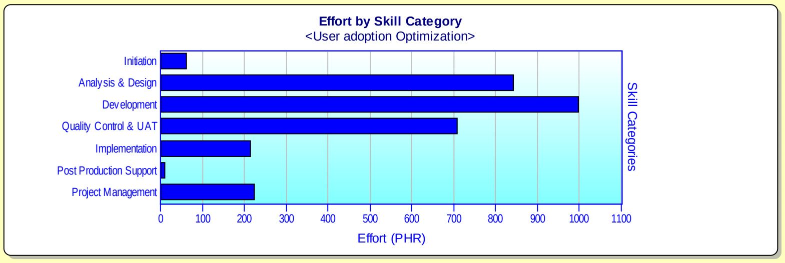
User Adoption Optimization Reconstruction



Defect profiles were tuned based on the defect data from milestone 6 start of system testing onward

User Adoption Optimization Skill Categories

Effort By Skill Category



Effort by Skill Category - BOA project
<User adoption Optimization>

Skill Categories	Effort (PHR)	%
Initiation	62.00	2.03
Analysis & Design	841.39	27.53
Development	996.39	32.60
Quality Control & UAT	708.54	23.18
Implementation	213.93	7.00
Post Production Support	10.03	0.33
Project Management	223.78	7.32

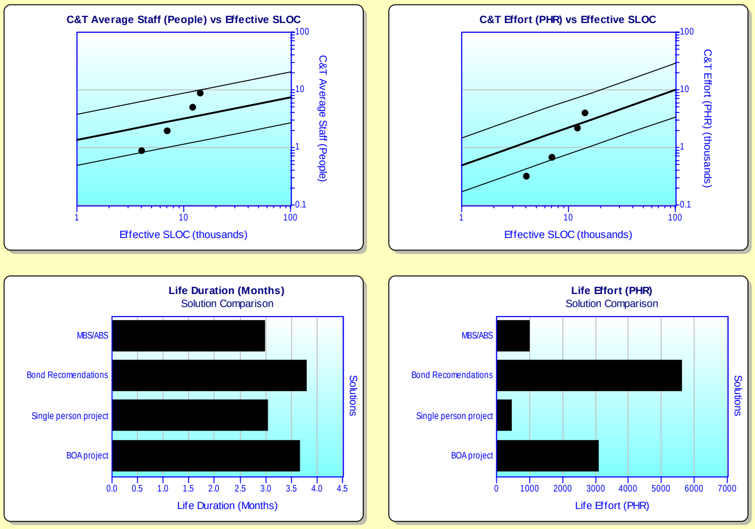
Based on the percentages

Project: BOA project

Successful Software Projects

Summary of Projects

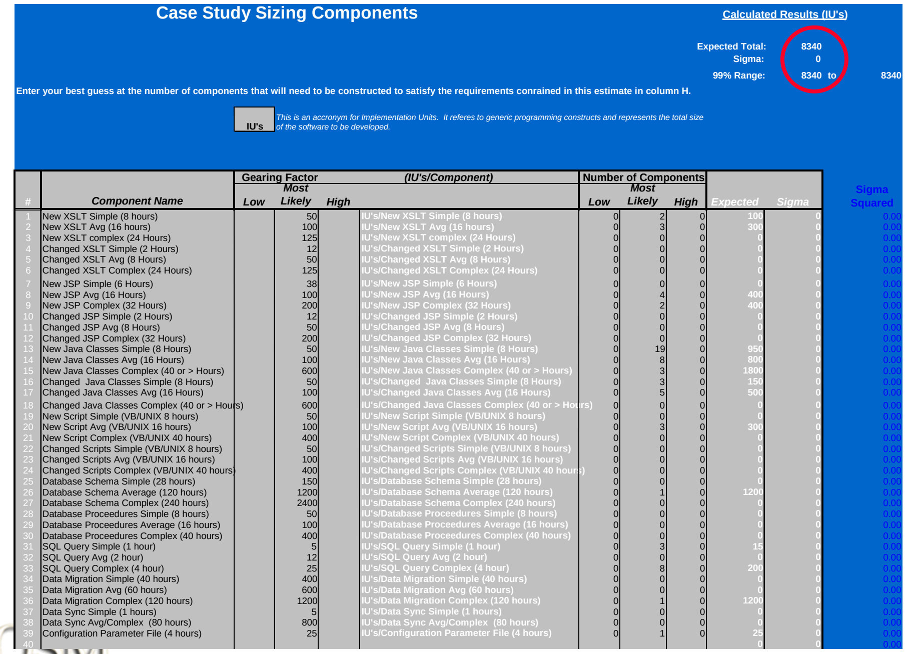
Compare Estimates to Historical Data



The Intelligence behind
Successful Software Projects

Example of Size Estimate

- New release to be developed
  - 2 simple and 3 average New XSL Transformations
  - 4 Average and 2 Complex New JSPs
  - 19 Simple/ 8 Average/ 3 Complex New Java Classes
  - 3 Simple/ 5 Average Changed Java Classes
  - 3 Average New Scripts
  - 1 Average Database Schema
  - 3 Simple/ 8 Complex SQL Queries
  - 1 Complex Data Migration
  - 1 configuration parameter file
- These are the developer inputs to the model



Case Study Sizing Components

Calculated Results (IU's)

Expected Total: 8340
Sigma: 0
99% Range: 8340 to 8340

Enter your best guess at the number of components that will need to be constructed to satisfy the requirements conrained in this estimate in column H.

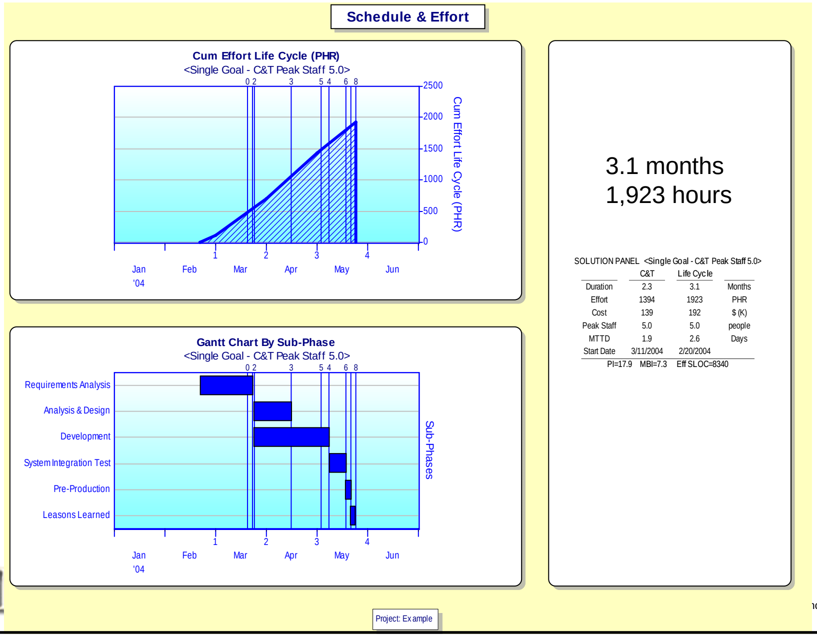
IU's *This is an accronym for Implementation Units. It referes to generic programming constructs and represents the total size of the software to be developed.*

#	Component Name	Gearing Factor Low	Most Likely	High	(IU's/Component)	Number of Components Low	Most Likely	High	Expected	Sigma	Sigma Squared
1	New XSLT Simple (8 hours)		50		IU's/New XSLT Simple (8 hours)	0	2	0	100	0	0.00
2	New XSLT Avg (16 hours)		100		IU's/New XSLT Avg (16 hours)	0	3	0	300	0	0.00
3	New XSLT complex (24 Hours)		125		IU's/New XSLT complex (24 Hours)	0	0	0	0	0	0.00
4	Changed XSLT Simple (2 Hours)		12		IU's/Changed XSLT Simple (2 Hours)	0	0	0	0	0	0.00
5	Changed XSLT Avg (8 Hours)		50		IU's/Changed XSLT Avg (8 Hours)	0	0	0	0	0	0.00
6	Changed XSLT Complex (24 Hours)		125		IU's/Changed XSLT Complex (24 Hours)	0	0	0	0	0	0.00
7	New JSP Simple (6 Hours)		38		IU's/New JSP Simple (6 Hours)	0	0	0	0	0	0.00
8	New JSP Avg (16 Hours)		100		IU's/New JSP Avg (16 Hours)	0	4	0	400	0	0.00
9	New JSP Complex (32 Hours)		200		IU's/New JSP Complex (32 Hours)	0	2	0	400	0	0.00
10	Changed JSP Simple (2 Hours)		12		IU's/Changed JSP Simple (2 Hours)	0	0	0	0	0	0.00
11	Changed JSP Avg (8 Hours)		50		IU's/Changed JSP Avg (8 Hours)	0	0	0	0	0	0.00
12	Changed JSP Complex (32 Hours)		200		IU's/Changed JSP Complex (32 Hours)	0	0	0	0	0	0.00
13	New Java Classes Simple (8 Hours)		50		IU's/New Java Classes Simple (8 Hours)	0	19	0	950	0	0.00
14	New Java Classes Avg (16 Hours)		100		IU's/New Java Classes Avg (16 Hours)	0	8	0	800	0	0.00
15	New Java Classes Complex (40 or > Hours)		600		IU's/New Java Classes Complex (40 or > Hours)	0	3	0	1800	0	0.00
16	Changed Java Classes Simple (8 Hours)		50		IU's/Changed Java Classes Simple (8 Hours)	0	3	0	150	0	0.00
17	Changed Java Classes Avg (16 Hours)		100		IU's/Changed Java Classes Avg (16 Hours)	0	5	0	500	0	0.00
18	Changed Java Classes Complex (40 or > Hours)		600		IU's/Changed Java Classes Complex (40 or > Hours)	0	0	0	0	0	0.00
19	New Script Simple (VB/UNIX 8 hours)		50		IU's/New Script Simple (VB/UNIX 8 hours)	0	0	0	0	0	0.00
20	New Script Avg (VB/UNIX 16 hours)		100		IU's/New Script Avg (VB/UNIX 16 hours)	0	3	0	300	0	0.00
21	New Script Complex (VB/UNIX 40 hours)		400		IU's/New Script Complex (VB/UNIX 40 hours)	0	0	0	0	0	0.00
22	Changed Scripts Simple (VB/UNIX 8 hours)		50		IU's/Changed Scripts Simple (VB/UNIX 8 hours)	0	0	0	0	0	0.00
23	Changed Scripts Avg (VB/UNIX 16 hours)		100		IU's/Changed Scripts Avg (VB/UNIX 16 hours)	0	0	0	0	0	0.00
24	Changed Scripts Complex (VB/UNIX 40 hours)		400		IU's/Changed Scripts Complex (VB/UNIX 40 hours)	0	0	0	0	0	0.00
25	Database Schema Simple (28 hours)		150		IU's/Database Schema Simple (28 hours)	0	0	0	0	0	0.00
26	Database Schema Average (120 hours)		1200		IU's/Database Schema Average (120 hours)	0	1	0	1200	0	0.00
27	Database Schema Complex (240 hours)		2400		IU's/Database Schema Complex (240 hours)	0	0	0	0	0	0.00
28	Database Proceedures Simple (8 hours)		50		IU's/Database Proceedures Simple (8 hours)	0	0	0	0	0	0.00
29	Database Proceedures Average (16 hours)		100		IU's/Database Proceedures Average (16 hours)	0	0	0	0	0	0.00
30	Database Proceedures Complex (40 hours)		400		IU's/Database Proceedures Complex (40 hours)	0	0	0	0	0	0.00
31	SQL Query Simple (1 hour)		5		IU's/SQL Query Simple (1 hour)	0	3	0	15	0	0.00
32	SQL Query Avg (2 hour)		12		IU's/SQL Query Avg (2 hour)	0	0	0	0	0	0.00
33	SQL Query Complex (4 hour)		25		IU's/SQL Query Complex (4 hour)	0	8	0	200	0	0.00
34	Data Migration Simple (40 hours)		400		IU's/Data Migration Simple (40 hours)	0	0	0	0	0	0.00
35	Data Migration Avg (60 hours)		600		IU's/Data Migration Avg (60 hours)	0	0	0	0	0	0.00
36	Data Migration Complex (120 hours)		1200		IU's/Data Migration Complex (120 hours)	0	1	0	1200	0	0.00
37	Data Sync Simple (1 hours)		5		IU's/Data Sync Simple (1 hours)	0	0	0	0	0	0.00
38	Data Sync Avg/Complex (80 hours)		800		IU's/Data Sync Avg/Complex (80 hours)	0	0	0	0	0	0.00
39	Configuration Parameter File (4 hours)		25		IU's/Configuration Parameter File (4 hours)	0	1	0	25	0	0.00
40									0	0	0.00

Example Estimate

- Estimation inputs
  - Size estimated to be 8,340 IU’s from sizing spreadsheet
  - Productivity Index = 17.9 Calibrated based on Historical Projects
  - Peak staff 5 people (FTE) – Project constraint
  - Start Date 2/20/04

Schedule & Effort



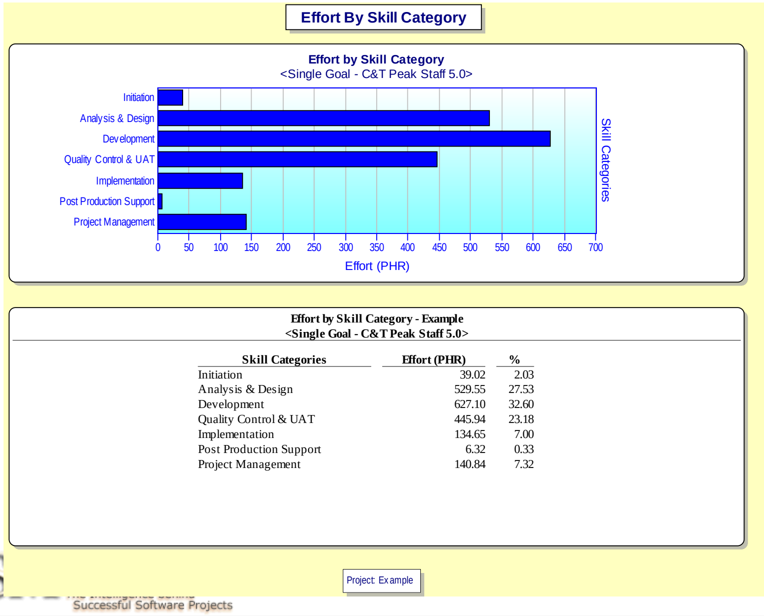
3.1 months
1,923 hours

SOLUTION PANEL <Single Goal - C&T Peak Staff 5.0>

	C&T	Life Cycle	
Duration	2.3	3.1	Months
Effort	1394	1923	PHR
Cost	139	192	$ (K)
Peak Staff	5.0	5.0	people
MTTD	1.9	2.6	Days
Start Date	3/11/2004	2/20/2004	

PI=17.9 MBI=7.3 Eff SLOC=8340

Effort by Labor Category

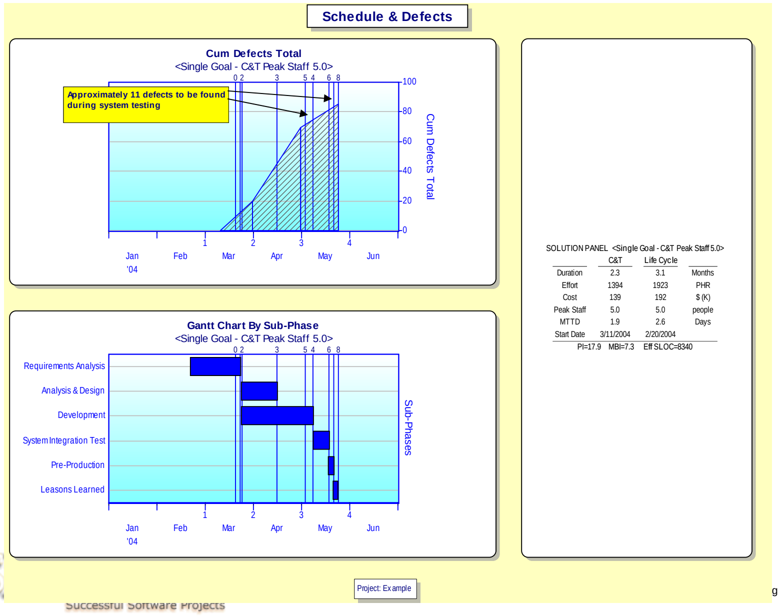


Effort by Skill Category - Example
<Single Goal - C&T Peak Staff 5.0>

Skill Categories	Effort (PHR)	%
Initiation	39.02	2.03
Analysis & Design	529.55	27.53
Development	627.10	32.60
Quality Control & UAT	445.94	23.18
Implementation	134.65	7.00
Post Production Support	6.32	0.33
Project Management	140.84	7.32

Project: Example

Estimate of Defects Found in System Testing



Approximately 11 defects to be found during system testing

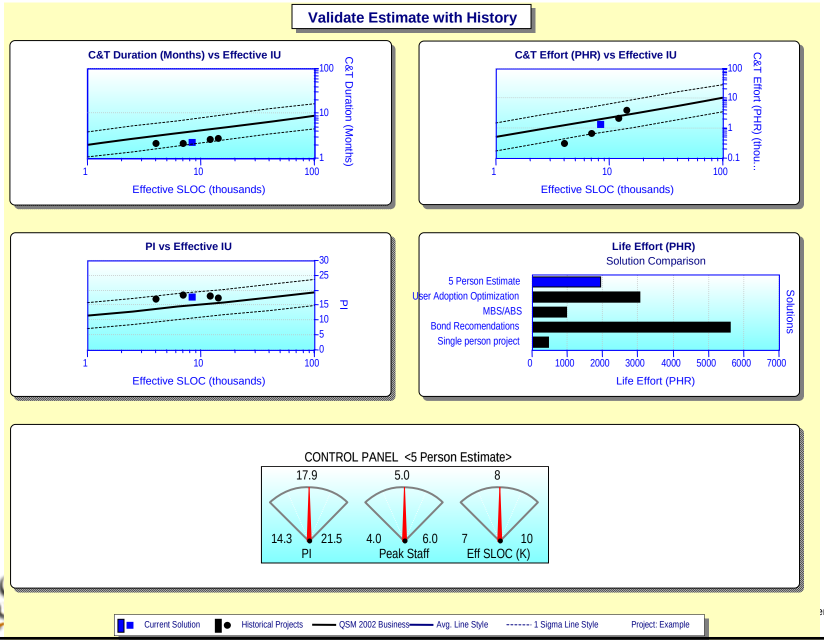
SOLUTION PANEL <Single Goal - C&T Peak Staff 5.0>

	C&T	Life Cycle	
Duration	2.3	3.1	Months
Effort	1394	1923	PHR
Cost	139	192	$ (K)
Peak Staff	5.0	5.0	people
MTTD	1.9	2.6	Days
Start Date	3/11/2004	2/20/2004	

PI=17.9 MBI=7.3 Eff SLOC=8340

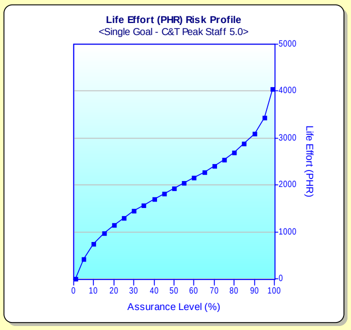
Project: Example

Estimate Compared to Historical Data



Effort Probability Analysis

Effort Probability Profile



Life Effort (PHR) Risk Profile - Example
<Single Goal - C&T Peak Staff 5.0>

Assurance Level (%)	Life Effort (PHR)
5	416.96
10	749.79
15	974.40
20	1,152.93
25	1,306.09
30	1,443.61
35	1,571.00
40	1,691.81
45	1,808.61
50	1,923.41
55	2,038.22
60	2,155.01
65	2,275.83
70	2,403.22
75	2,540.74

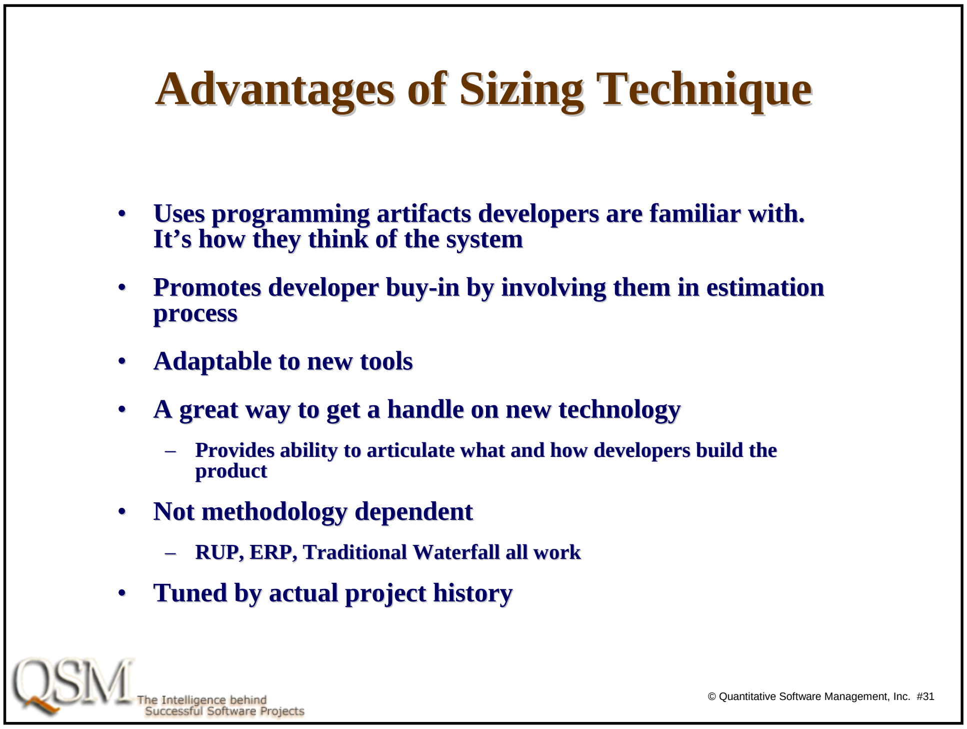
plan

high assurance

Effort Probability Profile

This report shows the effort probability for the overall project and the key milestones. The expected effort estimate is positioned at the 50% probability level. High assurance effort estimate would be in the 75% to 99% range.

Project: Example

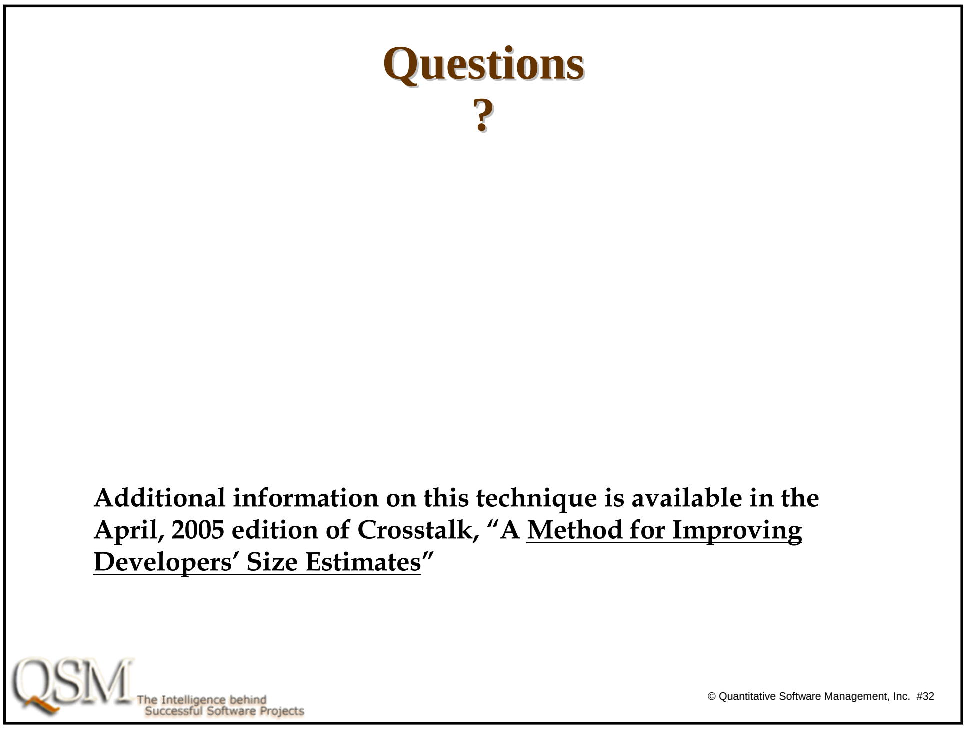


Advantages of Sizing Technique

- Uses programming artifacts developers are familiar with. It's how they think of the system
- Promotes developer buy-in by involving them in estimation process
- Adaptable to new tools
- A great way to get a handle on new technology
  - Provides ability to articulate what and how developers build the product
- Not methodology dependent
  - RUP, ERP, Traditional Waterfall all work
- Tuned by actual project history

The Intelligence behind
Successful Software Projects

© Quantitative Software Management, Inc. #31



Questions
?

Additional information on this technique is available in the April, 2005 edition of Crosstalk, "A Method for Improving Developers' Size Estimates"

The Intelligence behind
Successful Software Projects

© Quantitative Software Management, Inc. #32