QSM SLIM API usage in IBM

12 April 2012

QSM SLIM Usage Context in IBM

- IBM has nearly 1000 QSM SLIM users globally
- Full time SLIM deployment support team in place for four years with QSM guidance
- Partnered with IBM Watson research laboratory
- SLIM usage is well over 10 years
- Over 2,500 completed project in IBM SLIM DM file
- QSM has been at IBM developing estimating and performance metrics since the mid 1970's with Larry Putnam Sr.
- SLIM Suite usage
  - SLIM Estimate is the approved confirmatory estimating tool and largest usage
  - SLIM Metrics/Data Manager are global data collection and performance monitoring standard
  - SLIM Master Plan is the approved Complex System Integration (CSI)
  - SLIM Control used for troubled project tracking

SLIM API Benefits for

Data Collection

- Collect data with minimal deployment of SLIM DataManager and associated training
- Provide easily adjustable, broadly deployable data collection capability
- Provide data collection once – but for many uses that can be very easily understood and used
- Minimization of data collection errors and omissions
- Increased data collection efficiency
- Improved accuracy and capability of outlier analysis

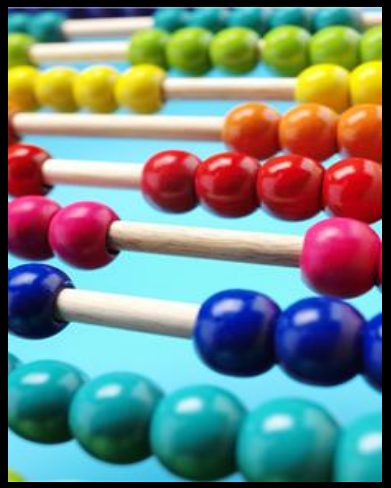
Estimation

- Decrease the number and impact of troubled projects due to poor estimating
- Conform to the estimating best practice - “two different estimating methods” - to reduce risk of estimating errors and omissions
- Increase the interest in and deployment of SLIM Estimate across GBS
- Provide basic SLIM Estimate capability for comparison purposes without extensive additional training
- Provide “single point” of documentation for two method comparisons
- Extend the use of the spreadsheet-based parametric estimating tool to additional SLIM users (UCEW, AD2.0, etc)
- Provide jointly “calibrated” estimating tools

QSM API deployment

- Estimation
  - SLIM “TAB” is a generic SLIM Estimate utility which can be added to virtually any spreadsheet to provide significant SLIM Estimate capability
  - Can be attached to bottoms up or parametric spreadsheets – or even just a SIZING spreadsheet
  - Example is SLIM integrated with the Use Case Estimating Workbook (UCEW)

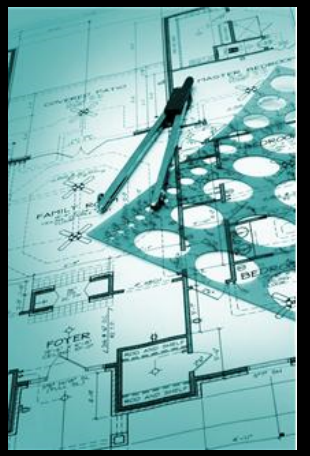
- Data
  - Data Collection for project estimates and completion actuals
  - Batch or single project data collection
  - Data Analysis/outlier identification
  - Template from basic metrics to detailed for performance benchmarking for root cause analysis
  - Easy to use and globally deployed



Estimation - SLIM TAB

SLIM TAB Integration Basics

- API and scripts are in visual basic for applications (VBA)
- All interactions are conducted on the “SLIM TAB” – no change to other estimating worksheets
- Minimal input required –more available
  - Constraints
  - Durations
  - Resources (FTE)
- Depends on initial set up of SLIM Estimate Template to provide consistent method and environmental factors
- Maintain “what if” and “mission impossible” capability within the spreadsheet
- Easy access to run SLIM Estimate directly for reporting and further analysis

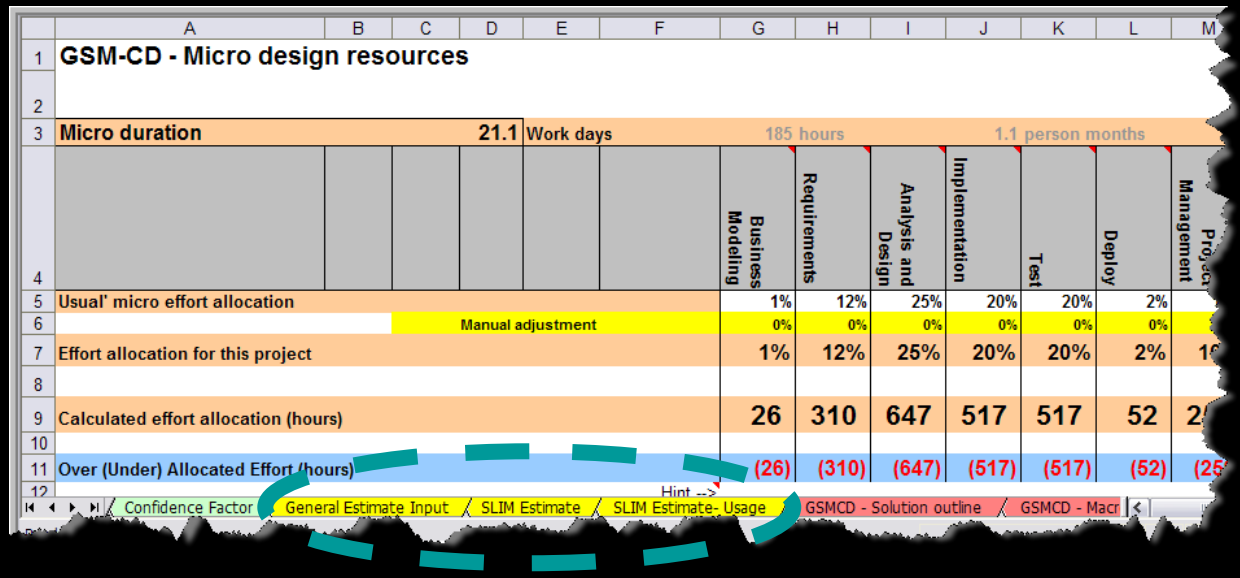


SLIM TAB Integration Basics – Beyond Process Steps

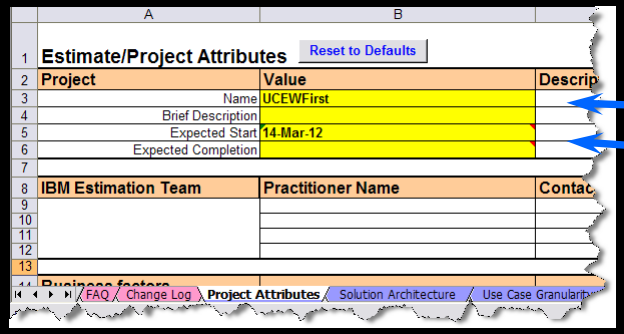
- Utilize SLIM Estimate to fully compare to a UCEW estimate
- Perform iterations (UCEW and SLIM) to close the gap between the two estimates
- Compare assumptions, assumptions, assumptions
- Be reasonable with comparison of both estimates – (linear versus non linear)
- In SLIM Estimate
  - Run multiple scenarios to find boundary conditions
  - Compare resources, effort, durations, size, etc
  - Understand probabilities (documented in BOTH tools)!!

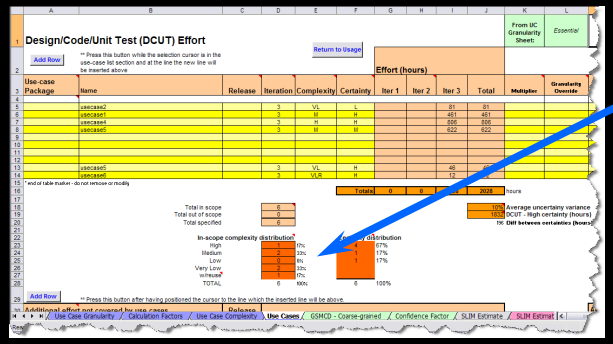
SLIM-UCEW Integration Basics – Process Steps

- Three “tabs” added to any appropriate spreadsheet
  - General Estimate Input tab is usually hidden
  - Instruction Tab
  - SLIM Estimate Tab
- Can be placed anywhere in your tab sequence to follow your process
- Once your other estimate process is done, go to the SLIM TAB



SLIM TAB first steps and usage requirements





Minimum Input

- Project Name
- Project Start Date
- Size of Project in ESLOC or native size

- Automatically provided by UCEW

Requirements:

- You must have the SLIM Suite loaded on your computer with an active license
- You must have a SLIM Estimate template with appropriate reference groups and trend lines
- Should have basic SLIM Estimate training ***(very highly recommended)***

SLIM TAB size input

- Data from parent spreadsheet
- Input cell locations into SLIM TAB
- Gearing factors are local to the SLIM TAB
- Sizing is same in SLIM Estimate Template
  - Function Points
  - RICEF
  - Integration
  - Use Cases
- Results in ESLOC
- Tab is usually “hidden” from user

Essential Use Cases

Complexity	Number of Objects	Description/Assumption
High	16	
Medium	0	
Low	8	
Very Low	0	
w/reuse	0	

Median Use Cases

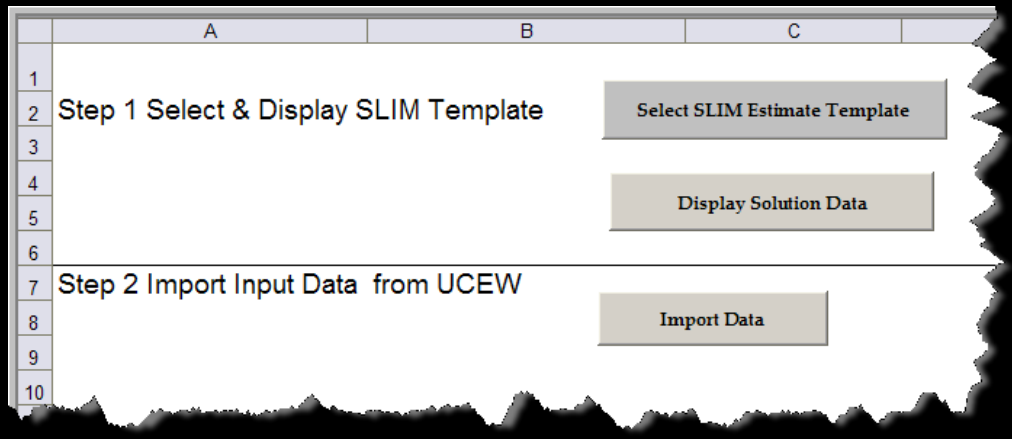
Complexity	Number of Objects	Description/Assumption
High	16	
Medium	0	
Low	8	
Very Low	0	
w/reuse	0	

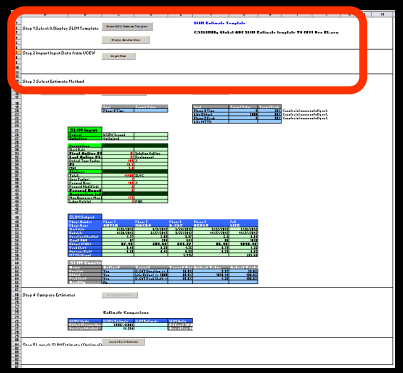
Decomposed Use Cases

Complexity	Number of Objects	Description/Assumption
High	16	
Medium	0	
Low	8	
Very Low	0	
w/reuse	0	

SLIM TAB – Once you've completed your other estimate

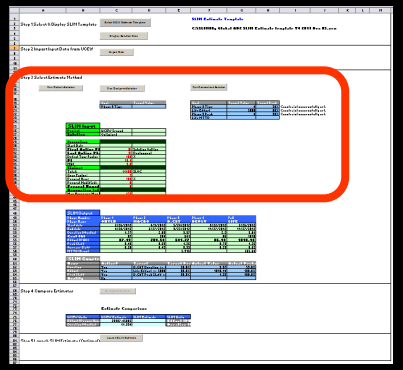
- Select approved template and import
- Display template solution data
- Then import key data from your other estimating tool





SLIM TAB – Once you've completed your other estimate

- You are provided three estimating methods
  - Default
  - Design to solution
  - Constrained solution (full)
- Error checking and validation is provide
- SLIM Input is documented



Step 3 Select Estimate Method

Run Default Solution | Run Design-to Solution | Run Constrained Solution

Goal	Target Value
Phase 3 Time	

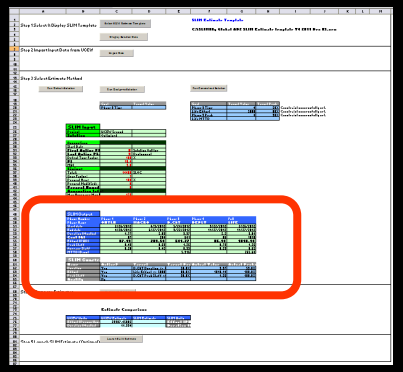
Goal	Target Value	Target Prob	
Phase 3 Time	4	50%	Constraint successfully set.
Life Effort	5000	80%	Constraint successfully set.
Phase 3 Peak	4	90%	Constraint successfully set.
Life MTTD			

SLIM Input		
Project	UCEW Second	
Solution	Optimized	

Assumptions		
Start Date		
First Active Phase	0	Solution Outline
Last Active Phase	3	Deployment

SLIM TAB – Once you've completed your other estimate

- SLIM estimate is provided by phase
- Constraints and basis are displayed



SLIM Output

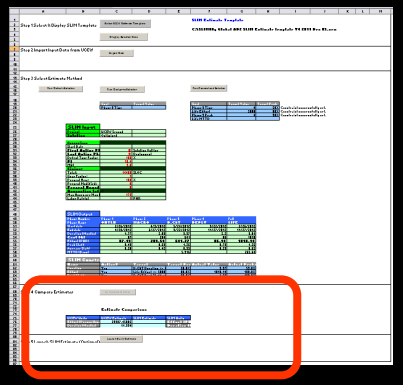
Phase Number Phase Name	Phase 1 OUTLN	Phase 2 MACRO	Phase 3 D,C&T	Phase 4 DEPLY	Full LIFE
Start date	2/26/2012	4/3/2012	5/24/2012	9/22/2012	2/26/2012
End date	4/20/2012	6/27/2012	9/22/2012	11/27/2012	11/27/2012
Duration (Months)	1.77	2.83	3.97	2.2	9.01
Cost ($)	87	204	641	86	1018
Effort (PHR)	**87.19**	**203.54**	**641.27**	**86.14**	**1018.14**
Peak Staff	0.43	0.63	1.26	0.73	1.26
Average Staff	0.28	0.42	0.93	0.23	0.65
MTTD (Days)			5.916		265.64

SLIM Constraints

Name	Active?	Target	Target Probability	Actual Value	Actual Probability
Duration	Yes	D,C&T Duration <= 4 Mo	50.0%	3.97	53.8%
Effort	Yes	Life Effort <= 5000 PHR	80.0%	1018.14	100.0%
Peak Staff	Yes	D,C&T Peak Staff <= 4 p	90.0%	1.26	100.0%
Quality	No				

SLIM TAB – Once you've completed your other estimate

- Compare both estimates
- Run iterations to minimize the delta and understand the differences
- Run SLIM Estimate for full documentation and additional "what ifs"



Step 4 Compare Estimates

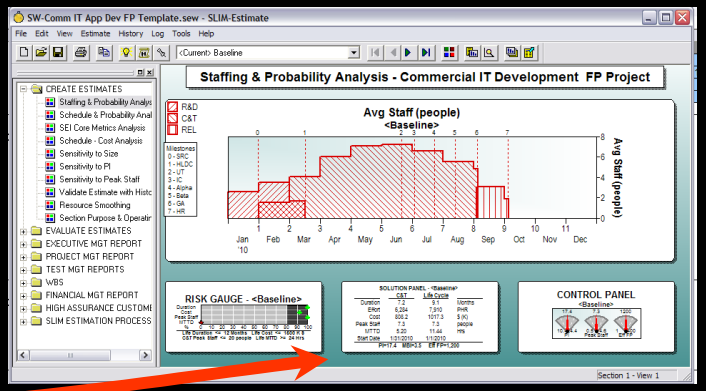
Compare Estimates

Estimate Comparison

UCEW Units	UCEW Estimate	SLIM Estimate	SLIM Units
Effort (Person Hours)	4909.5	1495.3	Effort (PHR)
Duration (Months)	5.883	6.87	Duration (Months)

Step 5 Launch SLIM Estimate (Optional)

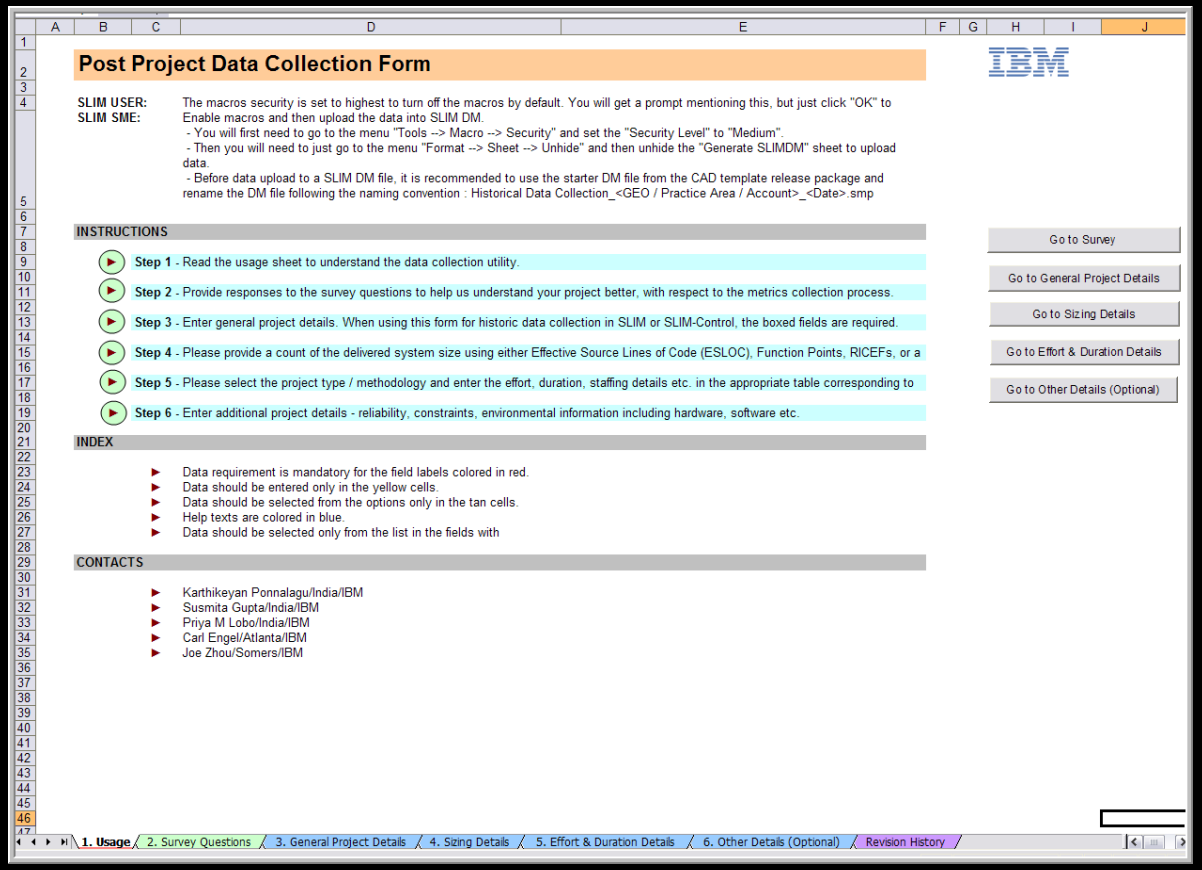
Launch SLIM Estimate



Data Collection

Data Collection – Single Project, Tab 1 Usage Instructions

- Example is relatively complete
- One button update to DM file
- Versions easily tailored
  - Minimal data
  - Project Type
- Six tabs
  1. Instructions and links on Usage tab
  2. Survey Questions for data quality
  3. General Project Details
  4. Sizing Details
  5. Effort and Duration Details
  6. Optional Data



Data Collection Survey – Tab 2

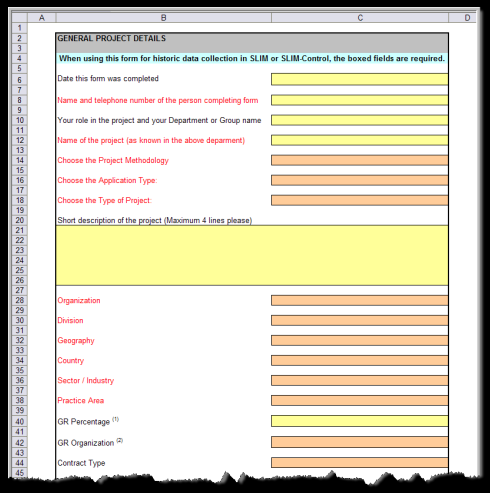
- Data quality has a huge impact on validity of reference groups and trend lines.
- Our process includes collecting “metrics” on the person filling out the form.
- Low confidence data is not thrown away, but may not be used for trend line development

	A	B	C	D
1				
2		DATA QUALITY		
3			Overall Confidence in the Data	
4				
5		#	SURVEY QUESTION & RESPONSE OPTIONS	Put X to answer
6		1	**How is the final metrics data captured verified?**	
7		a	The metrics data captured is verified with the responsible stakeholders of the project	
8		b	The metrics data captured is verified with the project data source & the measurement procedure is verified against project quality plan document / data harvesting procedure document	
9		c	The metrics data captured is not verified	
10		2	**What is the experience level of capturing and preparing project metrics?**	
11		a	Data capturing is done by experienced project member like project manager	
12		b	The project member responsible to collect the data has limited experience of project manager, but has many years of experience in software engineering (including programming and/or architecture design)	
13		c	The project member responsible to collect the data is relative new to data capturing process	
14		3	**How is staffing for each phase in the project measured?**	
15		a	By asking others (like architects)	
16		b	Taken from the project planning document that has been well updated	
17		c	I am the project manager, I know this from my memory	
18		d	Taken from the time sheet of the employees working on this project for each phase	
19		4	**How is effort for each phase in the project calculated?**	
20		a	By asking the team members	
21		b	Taken from the timesheets that is entered regularly in the project	
22		c	I am the project manager, I know this from my memory	
23		5	**How the effective SLOC counted?**	
24		a	SLOC is counted based on the guidelines provided in data harvesting procedure document for counting tool and definitions of new/modified/reuse code.	
25		b	The SLOC count is asked to others (like architects)	
26		c	SLOC is counted using code counter tool that were used in the past (If this is selected, provide name of code counting tool)	
27		6	**In case delivered size is specified in units other than SLOC, how was the function point/use cases/ user metric count arrived at?**	
28		a	By visiting the design & requirements documents and arriving at the size	
29		b	There is another source that has such information (If this is selected, provide source name)	
30		c	Based on individual memory	
31		d	By asking others	
32				

Data Collection – Details Tabs 3 and 4

- General Project Details Tab 3
  - Fundamentals
  - Key to analysis granularity
  - GR data

- Sizing Detail Tab 4
  - Flexible sizing
  - Gearing factors in SLIM Estimate
  - Languages



GENERAL PROJECT DETAILS

When using this form for historic data collection in SLIM or SLIM-Control, the boxed fields are required.

Date this form was completed

Name and telephone number of the person completing form

Your role in the project and your Department or Group name

Name of the project (as known in the above deparment)

Choose the Project Methodology

Choose the Application Type:

Choose the Type of Project:

Short description of the project (Maximum 4 lines please)

Organization

Division

Geography

Country

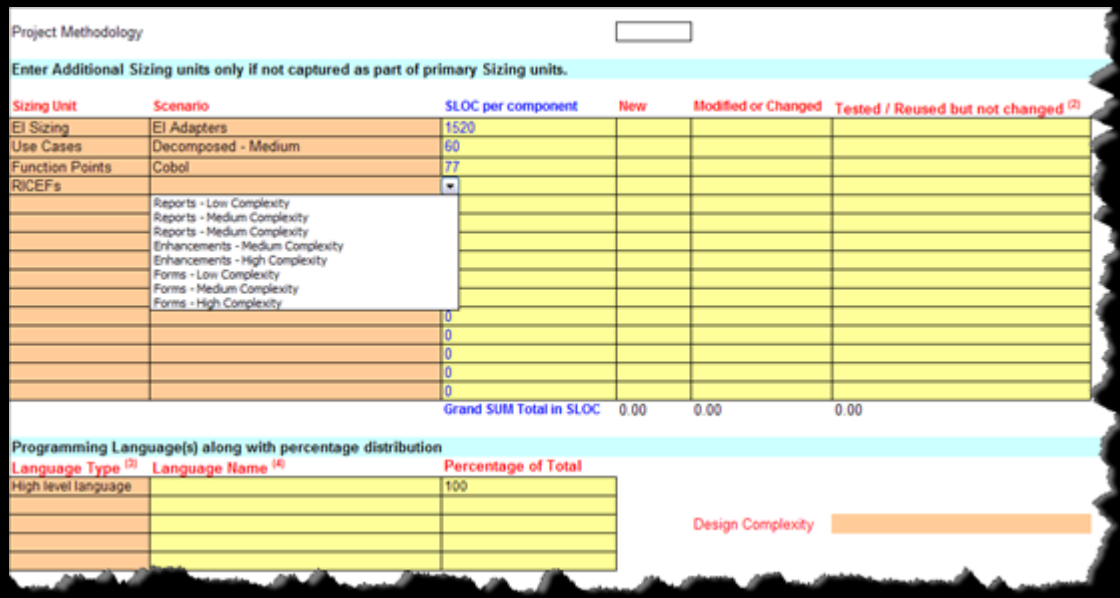
Sector / Industry

Practice Area

GR Percentage (1)

GR Organization (2)

Contract Type



Project Methodology

Enter Additional Sizing units only if not captured as part of primary Sizing units.

Sizing Unit	Scenario	SLOC per component	New	Modified or Changed	Tested / Reused but not changed (2)
EI Sizing	EI Adapters	1520			
Use Cases	Decomposed - Medium	60			
Function Points	Cobol	77			
RICEFs					
	Reports - Low Complexity				
	Reports - Medium Complexity				
	Reports - Medium Complexity				
	Enhancements - Medium Complexity				
	Enhancements - High Complexity				
	Forms - Low Complexity				
	Forms - Medium Complexity				
	Forms - High Complexity				
		0			
		0			
		0			
		0			
		0			
		Grand SUM Total in SLOC	0.00	0.00	0.00

Programming Language(s) along with percentage distribution

Language Type (3)	Language Name (4)	Percentage of Total
High level language		100

Design Complexity

Data Collection – Tab 5 Effort and Duration

- Project Method - AD, Oracle, Package, etc
- Basic phase/effort data
- Accuracy is important
- Phase/template confirmation

PROJECT EFFORT AND DURATION

Please select the project type / methodology and enter the effort, duration, staffing details etc. in the appropriate table below, corresponding to each project lifecycle phase.

Project Methodology

Enter Project Lifecycle Data

Lifecycle Phase Name (1)	Start Date	End Date	Effort (Person-Hours) (2)	Peak Staff
Solution Outline / Feasibility Study				
Macro Design / Functional Design				
Micro Design, Build and Test / Main Build (Unit, System, Integration & User Acceptance Testing)				
Deploy & Close / Maintenance				

Date System was Ready for Delivery (4) - (MM/YY)

(1) The lifecycle is divided into 4 phases in SLIM. Enter the resource information for all phases which are applicable to the completed project. A brief description of each phase is as follows: **Feasibility**

(2) Effort is the total person months expended by all development staff during a given phase. The number of person months should be reported in Full Time Equivalents (FTEs), including both internal and external/contractor staff (if any). For example, two people working half-time in project management would be included as one FTE. If you track effort in actual hours worked (net hours), then divide by the appropriate number of hours per month to determine the FTE effort.

(3) Indicate the month and year that the system was ready for delivery. This should coincide with the end of the Main Build phase. In the case of mission critical software, if the system was not delivered until sometime after the end of the Main Build, please make a note of when in phase 4 this occured.

Data Collection – Tab 6 Other Details

- Rarely Used
- Does provide increased analysis capability
- Primarily PM data

RELIABILITY OF THE DELIVERED SYSTEM

How many errors [1] were found during the period from the start of integration testing to the time the system was ready for
How many errors were found during the first month that the system was in commercial use/production?
What was the Mean Time to Defect (MTTD [2]) in hours during the first month the system was in production?

(1) An error is defined as each unique discrepancy between the program test results and the specification or code. Don't include queries or feature requests, only actual program errors. These counts should include all error severity categories, from cosmetic to critical. If the breakout by severity category is known, please note the distribution.

(2) The Mean Time to Defect (MTTD) is the average time between occurrences of a unique error (a defect once the product is shipped). This can be calculated by dividing the number of hours the system was operational by the number of defects discovered by the user(s).

PROJECT CONSTRAINTS

Please indicate any initial project constraints or limiting factors that were imposed on the project's Main Build phase (detailed	**Main Build**
Cost ($ x 1000)	
Maximum available staff	
Duration (in calendar months)	
Computer Resource Limits (**0** = none, **1** = some, **2** = significant)	

ENVIRONMENTAL DETAILS

List the hardware and software used during the development:

Hardware brand and type	
Operating System	

Choose the number of years experience (averaged over the whole team, where appropriate) at the start of the project:

Aspect	**Years Experience**	**Aspect**	**Years Experience**
Similar projects		Similar applications	
Methods and techniques used		Programming languages	
Development tools used		Hardware	
Management team			

Choose the number that best describes the unexpected staff changes which occured during the Main Build:

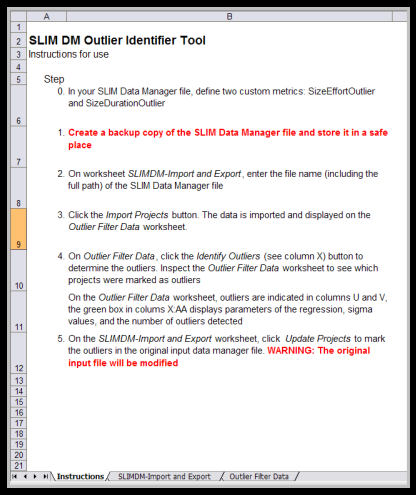
Choose the number that best describes the effectiveness of the tools and utilities used:

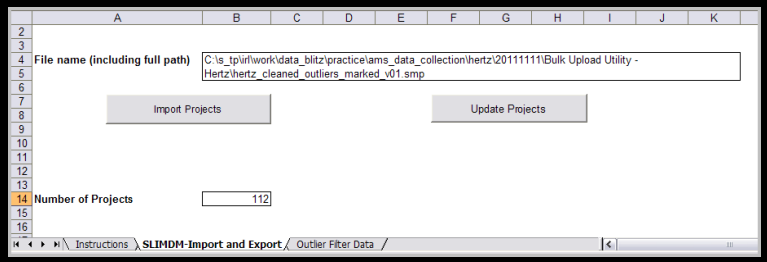
Choose the number that best describes the average response time of the development computer for typical complies and

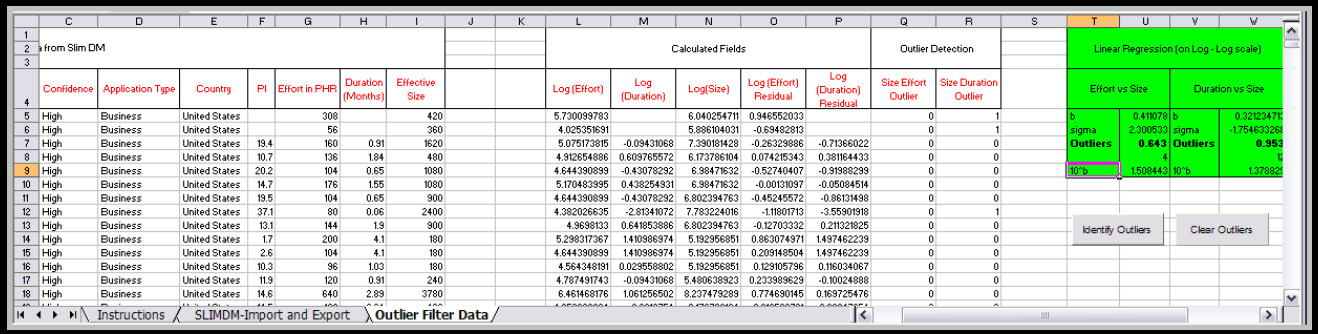
Data Analysis

Data Analysis – Outlier Determination

- Previously performed in SLIM Metrics visually – tended to be subjective
- Identify and investigate anomalies
- Data collection is expensive – don't waste data
- X-Y comparison flexibility
- Productivity and quality improvement tool
- 3 Tabs
- Allows independent data analysis







Summary

- IBM uses the API to
  - Reduce the cost/time to value realization of SLIM Estimate deployment
  - Improve the rate of adoption of improved estimating tools
  - Improve capability for performance benchmarking and trendline/reference group generation
  - Reduce data collection errors thus improving trend line accuracy

- General comments
  - QSM has a great starter set of examples and documentation on line
  - Fundamental script programming knowledge required only
  - Don't try to solve all of your estimating or data collection problems at one time with giant applications built on the API
  - Do start small to sell the concepts and get implementation buy in
  - Be careful of API usage without understanding SLIM Estimate

Developers/Authors

- Carl Engel - Carl Engel is the Estimating Program Manager for IBM's Global Business Services responsible for the development and deployment of performance benchmarking and estimating process, methods and tools including the support for nearly 1,000 SLIM Suite users. Carl has been with IBM for 12 years as an Associate Partner and has had previous roles as the program manager for IBM's worldwide project management methodology and tools. He is an IBM certified Executive Project Manager, PMP with over 30 years of program and project management experience primarily in very large scale efforts in the nuclear industry and U.S. National Laboratories: engecarl@us.ibm.com
- Karthikeyan Ponnalagu - Karthikeyan Ponnalagu joined IBM in 2000 and is working with IBM Research - India Bangalore for the past 6 years. Recently , he has been specifically involved with Estimation Data Harvesting, SLIM API based tooling development for leveraging SLIM Data manager and SLIM Estimate capabilities and challenges in planning and estimation for globalization projects. He has developed a Multi-Step Data Filtering (Completeness, Correctness and Outlier) and Clustering Technique that can be repeatedly deployed with different data sets and for multiple account centric needs. He also built the Automation Utilities for SLIM DM file generation and 2 sigma based Outlier deduction,. He also developed SLIM API based integration Worksheets for BAO and UCEW Estimation tools that are Excel based. He also developed a "Task Allocation Network" concept for addressing Globalization in Estimating projects
- Joe Zhou - Dr. Nianjun Zhou (Joe) is a research staff member at IBM T.J. Watson Research Center. He is serving as PIC (Professional Interest Community) Chair of Services Computing at IBM research. His current research areas mainly focus on services sciences and service computing to achieve IT and services solution optimization. He is leading research for resource estimation and optimization of IT solution and maintenance under global integration environment; and leading a solution framework development using standardized toolset, method, and assets to simplify service engagement.