Post Project Data Collection Form

GENERAL PROJECT DETAILS

When using this form for historic data collection in SLIM or SLIM-Control, the fields that are boxed are required. When collecting data for analysis using PADS, please try to complete the entire form.

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)	

Short description of the project:
(Maximum 4 lines please)

SYSTEM SIZE DATA

Please provide a count of the delivered system size using either Effective Source Lines of Code (ESLOC[(1)]), Function Points, or a User Metric of system size (i.e. Objects, Processes, Modules, etc.). If Function Points or a User Metric are used, a Gearing Factor should be supplied which indicates the average number of ESLOC per unit (i.e. 75 ESLOC/Function Point) in the delivered software.

	ESLOCs	Function Points	User Metric	Gearing Factor
Total				
New				
Modified or Changed				
Tested/Reused but not changed[(2)]				

(1) ESLOC are counted as the new and changed/modified logical source statements that were delivered as part of the final application. Do not include comments and blank lines, or the output of code generators (count the lines input into the code generator).

(2) If the data is being entered into PADS, the Tested/Reused category is used to identify the portion of the system that was reused but unmodifed, and may have required regression testing. In this scenario, the Total category should include the tested/reused as well as the new and modified size.

Which programming languge(s) were used and what percentage of the system was developed in each language?

Programming Language(s)	Language type(*)	ESLOCs (%)	Func. Points(%)	User Metric (%)

(*) Language type: **1** = High level language, **2** = Assembler code, **3** = 4th generation, **4** = Microcode, **5** = Other language
6 = Object oriented, **7** = DBMS lanaguage, **8** = JCL, **9** = Special language

PROJECT RESOURCES

LIFECYCLE PHASE NAME (1)	Time (Mos.)	Effort (2) (Person-Mos)	Peak Staff	Shape Of Staff Buildup (3)	Overlap (Mos.) (4)
Feasibility Study					
Functional Design					
Main Build					
Maintenance					

DATE SYSTEM WAS READY FOR DELIVERY (5) - (MM/YY) /

DEVIATION FROM PLANNING (6)

	Main Build
Effort (in person-months)	
Time (in calendar months)	

ACTUAL COSTS (7)

(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 Study** - technical/cost feasibility and system requirements; **Functional Design** - software requirements and high level design; **Main Build** -detailed design, construction, test, and delivery to customer/user (assumed that 95% of defects are found and fixed at completion of the Main Build); **Maintenance** - installation/grooming of the software in the user environment and fundamental operations and maintenance (for mission critical software, system delivery may occur in this phase rather than the Main Build).

(2) Effort is the total amount of 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) The shape of the staff buildup is an indication of how people were applied to the project in each phase. There are 2 staffing shapes for the Feasibility Study and Functional Design. These are a Level Load (equal number of people in each month of the project), or a Front Load Rayleigh (build up of staff to a peak at about 45% of the phase, then trailing down, with staff moving on to the next phase). For the Main Build, there are 3 additional staffing shapes. These are Medium Front Load (peaking at 40% of the phase), Medium Rear Load (peaking at 80% of the phase), and Rear Load (peaking at the end of the phase). For the Maintenance phase, there are also 5 choices. If staff levels were maintained at the same level as the end of the Main Build, a Level Load was used. Otherwise, some gradual reduction in staffing is typical in this phase. This reduction likely followed 1 of 4 shapes: either a Stair Step, Straight Line, Exponential, or Rayleigh.

(4) The overlap field is used to record the degree of parallel effort between the lifecycle phases. In the Feasibility Study phase report any overlap between this phase and the Functional Design in full or partial calendar months. In the Functional Design phase report any overlap between this phase and the Main Build. For example, if the Functional Design was 6 months long, and the Main Build started in month 4, there was a 2 month overlap.

(5) 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.

(6) If the total effort or time expended in the Main Build was more or less than what was initially planned for this phase, indicate the difference between the plan and the actuals. For example, if the schedule and effort were planned at 6 months and 75 person-months, and the actuals were 5 months and 80 person-months, there would then be a -1 month for schedule deviation (underrun), and a +5 person-months for effort deviation (overrun).

(7) Indicate the total actual cost for the Main Build phase. If you do not have access to the actual number, but know the fully burdened labor rate, simply multiply the labor rate by the reported number of person-months of the Main Build.

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 delivery?	
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 occurences 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 design to delivery).

	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	

Circle 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	< 1	1 - 3	> 3	Similar applications	< 1	1 - 3	> 3
Methods and techniques used	< 1	1 - 3	> 3	Programming languages	< 1	1 - 3	> 3
Development tools used	< 1	1 - 3	> 3	Hardware	< 1	1 - 3	> 3
Management team	< 1	1 - 3	> 3				

Circle the number that best describes the unexpected staff changes which occured during the Main Build:
1=**[**<10%**]**, **2**=**[**10-20%**]**, **3**=**[**21-30%**]**, **4**=**[**31-50%**]**, **5**=**[**>50%**]**

Circle the number that best describes the effectiveness of the tools and utilities used:
1=Poor **2**=Average, 3=Good, **4**=Excellent

Circle the number that best describes the average response time of the development computer for typical complies and builds during the project:
1=**[**>24 hours**]**, **2**=**[**4-24 hours**]**, **3**=**[**1-4 hours**]**, **4**=**[**5 min.-1 hour**]**, 5=**[**5 sec.-5 min.**]**, 6=**[**1-5 sec.**]**, 7=**[**<1 sec.**]**

APPLICATION TYPE AND FUNCTIONAL SPECIFICATIONS

Circle the number that best describes the application type:

1=Microcode/Firmware, **2**=Realtime, **3**=Avionic, **4**=System Software, 5=Command & Control, **6**=Telecom/Message Switching, **7**= Scientific, **8**=Process control, **9**=Business/Commercial

Circle one or more numbers that describe the characteristics of this application:

0=Online transaction, **1**=Database, **2**=Message Switching, **3**=Simulation, **4**=Communication, **5**=Network management, 6=Multiprocessor, **7**=Distributed system, **8**=Embedded system, **9**=Special system, **10**=High performance or High volumes, **11**=High reliability or availability, **13**= Significant batch processing

Circle the number that best describes the type of project:

1=New Development, **2**=Major Enhancement, **3**=Minor Enhancement, **4**=Conversion

Circle the number which best represents the percentage of change to the functional specifications during the Main Build:

1=**[**<10%**]**, **2**=**[**10-20%**]**, **3**=**[**21-30%**]**, **4**=**[**31-50%**]**, **5**=**[>**50%**]**

Circle the number that best describes the complexity of the application:

1= Completely new and complex software. Production hardware and interfaces may well be developed in parallel. Many interfaces, of which a number were not specified. Algorithms and logic newly created.
2= Completely new relatively complex software. Most interfaces were well specified. Many new algorithms and much new logic.
3= Completely new software. Interfaces well specified. Algorithms well known but much new logic required.
4= Rebuild of an existing system with substantial changes in the specifications of the software.
OR:Extension of an existing system with substantial changes in the specifications of the software
OR:Completely new software that was well specified and not complex.
5= Rebuild of an existing system with limited changes in the specifications of the software.
OR:Extension of an existing system with limited changes in the specifications of the software.
6= Conversion of an existing system to a new hardware platform with less than 15% changes in the code.

List all tools , utilities, computer based aids, and methodologies used in this project (please briefly explain what each tool is used for). Continue on another page if necessary.

List any factors that had a positive or negative effect on the project, such as sickness of the project leader, introduction of new technology, lack of readiness of externally prepared subprojects and interfaces, and so on. Continue on another page if necessary