SIZING AND ESTIMATION - Key Information for SLIM Forecasting

GENERAL INFORMATION

1. Project Name
2. Date form Completed
3. Completed by
4. Telephone and Fax Numbers
5. Role in the project
6. Group/Division within Company

LIFECYCLE PHASES

7. Please identify which of the following 4 phases should be included in the SLIM estimate, and what type of staffing profile you think is appropriate.

	INCLUDE (1) (YES/NO)	STAFFING SHAPE (2)	PEAK STAFFING POINT (3)
FEASIBILITY STUDY			
FUNCTIONAL DESIGN			
MAIN BUILD			
MAINTENANCE			

1) The SLIM estimate can include up to 4 development phases. All estimates will include the Main Build phase. A brief description of each phase is as follows (contact the Project Office for details): **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 end of the Main Build).

2) There are 5 fundamental staffing shapes which can be applied in SLIM. For initial setup, it is adequate to identify the staffing shape by selecting either Level Load or Rayleigh. The Level Load staffing profile should be selected when a constant number of people will be applied during the entire phase (typical of the early phases of the lifecycle and small team projects); The Rayleigh staffing profile should be selected in all other cases when staffing will increase to a certain peak level and then either level off or tail off.

3) The peak staffing point is applicable if you selected the Rayleigh staffing profile. This will help to identify when you expect the peak staffing to occur. A Selection of Front will have a peak at approximately 45% of the phase. Selecting Medium Front will place the peak at approximately 60% of the phase. Medium Rear will position the peak staff at approximately 80% of the phase, and Rear will postion the peak staff at 100% of the phase. Alternatively, you may select Default and SLIM will determine the optimum peak staffing point.

PROJECT CONSTRAINTS

8. Please identify any management constraints that may be imposed on the project development team. Constraints apply to the entire lifecycle including all phases identified in item 7 on this form. Indicate N/A if a constraint does not exist (i.e.it is common to have a dollar budget constraint, but not know the equivalent number of person-months that the project is limited to).

	Constraint	Desired Probability *
Time (# of Months or Date)		
Effort (Person-Months)		
Cost (Dollar Limit)		
Maximum number of staff (1)		
Minimum number of staff (1)		
Reliability of the system (2)		

(1) Include both internal and external/contractor staff (if any), using "full-time-equivalents." For example, two half people would be one full-time-equivalent. If you estimate in actual hours to be worked (net hours) then divide by the appropriate number of hours per month to determine the full-time-equivalent staff.

(2) The reliability constraint is quantified using the Mean Time to Defect (MTTD). MTTD is the average number of operational hours/days between faults in the system. It is possible to have a reliability constraint for ALL faults, or a specific category such as "Show Stoppers". For example, your Reliability Constraint could be 2 Days MTTD at delivery for ALL faults or 15 days for "Show Stoppers".

* The Desired Probability should represent how critical it is to meet the project constraint. For example, a 90% desired probability for a 12 month schedule constraint would imply that it is business critical to hit the target schedule with adequate "risk protection" from changes in the plan.

APPLICATION TYPE

9. What is the approximate percentage mix of application types in the system? (e.g. 100% Business or 90% Business, 5% System Software, and 5% Telecom)

Type (s)	%

Type can be:

- Business/Financial
- Process control
- Real time
- Telecom/Message Switching
- Command & Control
- Microcode / Firmware
- System Software
- Scientific

SYSTEM SIZE AND LANGUAGES

10. Please provide an estimate of the system size using one or more metrics as indicated in the table below. If the system can be decomposed into functional components or modules, the statistical reliability of the size esimate will be much higher. The worksheet at the back of this form can be used to decompose the size estimate in support of the total numbers provided here. The Project Office can provide additional support using the 5 independent sizing techniques available in SLIM.

	Minimum (99%)*	Expected (50%)*	Maximum (99%)*
Size of the system in ESLOC (1)			
Size of the system in Function Points (1)			
Size of the system in User Metric (1)(2)			
Gearing Factor (2)			

(1): Either Effective Source Lines of Code (ESLOC), Function Points, or a User Metric may be used to estimate size of the system. ESLOC include all new and modifed lines of code excluding comments and blanks. Function Points include Inputs, Outputs, Files, Interfaces, and Inquiries. User Metrics may be any unit of size including Objects, Modules, Programs, Screens, etc,. The Metrics Office document set contains guidelines for each of these size metrics.

(2): If you are using a User Metric for system size (e.g. Objects, Procedures, Screens) also enter a Gearing Factor. The Gearing Factor represents the average ESLOC per User Metric (e.g. 225 ESLOC per Object). If you do not have this information, QSM industry average gearing factors will be applied.

* The Minimum, Maximum, and Expected size values are used to identify the degree of uncertainty associated with an estimate. The Minimum value should represent the samllest possible total system size, with a 99% confidence that the system can not be any smaller. The Expected size should be the current best assessment of size. The Maximum size should be the largest possible total system size, with a 99% confidence that the system can not be any larger. It's okay to have a wide spread between Minimum and Maximum values. This is typical early in a project's lifecycle, and will narrow as requirements are baselined.

11. Which programming language(s) will be used and for what percentage of the whole system?

Name of the language	Type (*)	%

(*) Type (1) = High order (2) = Low level (3) = 4-th Generation (4) = Microcode (5) = Other (6) = Object Oriented (7) = DBMS (8) = JCL (9) = Special Language

PROCESS PRODUCTIVITY

12. What Process Productivity Index (PI) do you think is reasonable to expect on this project? What is your degree of confidence in the PI selected?

Process Productivity Index (PI) [1]	Uncertainty [2]

(1) The QSM Process Productivity Index (PI) is a macro measure of project complexity and efficiency of the development processes and evnironment. Details of this measure can be found in Measures for Excellence, Putnam and Myers, Prentice Hall 1992 (see your Metrics Office Manager for a copy of this book). The PI can be obtained from historic project data, or may be determined by SLIM using its knowledge base in accordance with your input on the complexity and environment (see questions below).

(2) There are four (4) levels of uncertainty which can be applied in SLIM; Certain, Slightly Uncertain, Rather Uncertain, Very Uncertain. A selection of Certain would imply that significant historic data exists to support the PI being used. If historic data does not exist, or the project is significantly different than the historic data, then additional uncertainty levels should be applied. Slightly uncertain means that the PI selected is good to +/- .5. Rather uncertain will allow the PI to vary by +/- 1 value, and Very uncertain will vary the PI by +/- 2 values. This selection will have a direct result on the risk analysis that SLIM provides. Review the Metrics Office document set, or contact your Metrics Office Manager for further guidance.

TOOLS & METHODS - PROJECT COMPLEXITY - TEAM SKILLS

Each of the questions below is useful in adjusting the SLIM estimate to account for variations in environments and project complexity from one application to the next (iIf your selection of the PI in question 12 was based on extensive historic data, these questions may be ignored). For questions 13 - 15, answers should be provided using a relative scale from 0 to 10, with 5 being average. Indicate N/A if the information is not applicable or unkown. (Additional detailed questions may be examined in each of these areas - contact the Project Office for more information).

13. **Tools and Methods** - Are the development tools and lifecycle methodologies available to this project well integrated and very familiar to the team?(8 to 10); Are the tools/methods an average set, which are familiar to some of the team members but new to others?(4 to 7); or Are the tools and methods outdated and little used, and minimally integrated with little use within the team?(0 to 3).

Examples of tools and methods included in this category are diagramming, testing, configuration management, and screen painting tools. Also included in this category are automated methods or manual procedures which define the software development lifeycycle.

14. **Complexity** - Is the technical complexity of this application extensive, with many external interfaces, complex integration requirements, and extensive data, algorithm, and logic intensity?(8 to 10); Is this an average complexity application with well understood interfaces, stable requirements, and less than 75% brand new algorithm and logic design?(4 to 7); or Is this a straightforward application with little or no external interfaces, minimal integration requirements, and a small percentage of new algorithms, data, and logic design?(0 to 3).

15. **Team Skills** - Is the mix of skills across the team well rounded with strong experience in the funcitonal area and the application specific complexities? Does the group “gel” together nicely and have strong motivation supported by an experienced management team which provides the needed tools and training? (8 to 10); Is the mix of skills more average, with some senior staff and some new to the team?. Has the team not yet had time to become cohesive and is motivation varied accross team members?(4 to 7); or Is the team mostly comprised of new hires or staff new to the functional area, with minimal training and less experienced project management?(0 to 3).

SOFTWARE REUSE

Each of the questions below is geared to applications which include reused software. This may be internally developed reusable objects, subroutines or other class libraries, as well as commercial off-the-shelf software packages (if your selection of the PI in question 12 was based on extensive historic data, these questions may be ignored).
For questions 16 - 27, answers should be provided using a relative scale from 0 to 10, with 5 being average. Indicate N/A for those questions which are not applicable or unkown. Additional guidance is avaiable in the Guidelines section at the back of this form.

16. What is the approximate percentage that the reused functionality represents relative to the entire system? (e.g. reused functionality/total functionality *100, using the sizing metric from question 10).

17. What is the complexity of integrating the reused code with any new code development?

18. What is the teams experience in working with the reused software?

19. What is the realtive number of functional interfaces present in the reused software?

20. What is the realtive percentage of functional interfaces that will be used?

21. What is the complexity of using the functional interfaces?

22. What is the relative time required to select a commercial product?

23. What is the relative analysis effort required to assess the impact on existing code?

24. How useful is the software documentation (if it is available and/or required)?

25. How effective is external customer support (if it is needed)?

26. How extensive are user documentation requirements after integrating the reused software?

27. How extensive are regression testing requirements after incorporating the reused software?

WORKSHEET FOR SIZE DECOMPOSITION

MODULE or FUNCTION	Current size	New or Added			Changed or Modified		
		Min. (99%)	Exp (50%).	Max. (99%)	Min. (99%)	Exp. (50%)	Max. (99%)

GUIDELINES FOR ANSWERING SOFTWARE REUSE QUESTIONS (16-27)

Following are some general guidlines to help with interpretation of the questions related to software reuse.

Question 16: Percent functionality of the system the reused software represents ?
To estimate the reused portion of the system, use a size measure that can be applied to the entire system (i.e., SLOC, Function Points, Objects, Requirements). Percent Reused Functionality = (Reused Functionality/Total System Functionality) * 100. Often an intelligent approximation works best, especially when purchased software is included in the reused portion of the system.

- Unknown. The average will be used. 50% of the system represents reused functionality.
- Low end of the scale, 1 to 20% of the system represents reused functionality.
- Middle of the scale, 40 to 60% of the system represents reused functionality.
- High end of the scale, 80 to 99% of the system represents reused functionality.

Question 17: Complexity of integrating products with new code (0 to 10)?
In answering this question please consider both the number and complexity of the interfaces required.

- Unknown. The average (5) will be used.
- Low end of the scale, 0-2 range. There is minimal complexity integrating the reused software products with the new code. There are not many interfaces necessary. The interfaces that are needed are well understood, simple, well isolated and straightforward.
- Middle of the scale, 4-6 range. The job of integrating the existing software is moderately difficult. There are either many simple interfaces or a limited number of complex interfaces required.
- High end of the scale, 8-10 range. The job of integrating the existing software is extremely complex. Often this software is very old, there are many interfaces and a significant portion of the existing software needs to be modified in many scattered places.

Question 18. Experience using the specific products (0 to 10)?
Please consider the average experience of the entire development team, their level of experience and the length of time that they used the commercial packages and/or existing code.

- Unknown. The average (5) will be used.
- Low end of the scale, 0-2 range. Team has minimal or no development experience using the commercial packages and/or existing code.
- Middle of the scale, 4-6 range. Team has previous development experience using the commercial packagesand/or existing code on one project.
- High end of the scale, 8-10 range. Team has previous development experience using the commercial packages and/or existing code on two or more projects or has had intensive experience on one project.

Question 19: Number of functional interfaces present (0 to 10)?
Functional interfaces refer to unique entry points into the software package. Examples are calls to subroutines, function calls, object properties, object methods/events. In answering this question please include the total number of functional interfaces that are present in the commercial packages and/or existing code. More time and effort is required as the number of functional interfaces increase. It is necessary to assess all the functional interfaces to locate the ones that are most appropriate to use.

- Unknown. The average (5) will be used.
- Low end of the scale, 0-2 range. There are fewer than 20 functional interfaces
- Middle of the scale, 4-6 range. There are 100 - 150 functional interfaces.
- High end of the scale, 8-10 range. There are greater than 500 functional interfaces.

Question 20: Percent of functional interfaces that will be used (0 to 10)?
Functional interfaces refer to unique entry points into the software package. In answering this question, consider the percent of the available interfaces that will actually be used.

- Unknown. The average (5) will be used.
- Low end of the scale, 0-2 range. 1 to 20% of the functional interfaces present will be used.
- Middle of the scale, 4-6 range. 40 to 60% of the functional interfaces present will be used.
- High end of the scale, 8-10 range. 80 to 100% of the functional interfaces present will be used.

Question 21: Complexity of using the functional interfaces (0 to 10)?
In answering this question please consider complexity of the interfaces that will be used.

- Unknown. The average (5) will be used.
- Low end of the scale, 0-2 range. The functional interfaces are simple, easy to learn and use.
- Middle of the scale, 4-6 range. There is a mix of simple and complex functional interfaces.
- High end of the scale, 8-10 range. The functional interfaces are extremely complex, difficult to learn and use.

Question 22: Research time required to select products (0 to 10)?
Answer this question as a percentage of the total project schedule. Examples for 6, 12 and 24 month projects are shown below.

Total project schedule	Low5%	Average10%	High20%
6 Months	1 week	2 weeks	1 month
12 Months	2 weeks	5 weeks	2.5 months
24 Months	5 weeks	2.5 months	5 months

- Unknown. The average (5) will be used.
- Low end of the scale, 0-2 range. 0 to 5% of the total project schedule will be dedicated to researching and selecting reusable software products.
- Middle of the scale, 4-6 range. 8 to 14% of the total project schedule will be dedicated to researching and selecting reusable software products.
- High end of the scale, 8 to 10 range. 18 to 20% or greater of the total project schedule will be dedicated to researching and selecting reusable software products.

Question 23: Analysis effort required to assess impact to existing code (0 to 10)?

- Unknown. The average (5) will be used.
- Low end of the scale, 0-2 range. 0 to 5% of the total project schedule will be dedicated to assessing the impact to existing code.
- Middle of the scale, 4-6 range. 8 to 14% of the total project schedule will be dedicated to assessing the impact to existing code.
- High end of the scale, 8 to 10 range. 18 to 20% or greater of the total project schedule will be dedicated to assessing the impact to existing code.

Question 24: Usefulness of system documentation if required (0 to 10)?

- Unknown. The average (5) will be used.
- Low end of the scale, 0-2 range. Little formal documentation is available. What is available is of limited use.
- Middle of the scale, 4-6 range. There is documentation available on using and interfacing with the reused software, but it is only moderately effective.
- High end of the scale, 8 to 10 range. There is extensive documentation on using and interfacing with the reused software and it is highly effective.

Question 25: Effectiveness of external customer support if required (0 to 10)?

- Unknown. The average (5) will be used.
- Low end of the scale, 0-2 range. It is very difficult to communicate with the developers or supporters of the reused software. When available, the support is of questionable benefit.
- Middle of the scale, 4-6 range. Customer support personnel are moderately helpful when problems or questions about using the software come up. There may be a 2 to 3 day turnaround in getting a response.
- High end of the scale, 8 to 10 range. Customer support personnel are extremely knowledgable and helpful when problems or questions about the reused software occur.

Question 26: Additional user documentation required for reused products (0 to 10)?

- Unknown. The average (5) will be used.
- Low end of the scale, 0-2 range. Less than 20% of the total documentation effort will be dedicated to documenting the reused software.
- Middle of the scale, 4-6 range. 25 to 50% of the total documentation effort will be dedicated to documenting the reused software.
- High end of the scale, 8 to 10 range. More than 75% of the total documentation effort will be dedicated to documenting the reused software.

Question 27: Additional testing required for reused products (0 to 10)?

- Unknown. The average (5) will be used.
- Low end of the scale, 0-2 range. 0 to 5% of the total testing will be devoted to the reused portion of the software.
- Middle of the scale, 4-6 range. 40 to 60% of the total testing will be devoted to the reused portion of the software.
- High end of the scale, 8 to 10 range. More than 75% of the total testing will be devoted to the reused portion of the software.