DATA ITEM DESCRIPTION

Title: REUSE MANAGEMENT REPORT (ReMR)

Number: DI-SESS-81771	**Approval Date:** 20090520
AMSC Number: F9071	**Limitation:** N/A
DTIC Applicable: N/A	**GIDEP Applicable:** N/A

Preparing Activity: 13 **(**ESC/AQT)
Applicable Forms:
Worksheet Questions for Reused As-is/Modified Software
Worksheet Questions for COTS/GOTS Software

Use/Relationships: The Reuse Management Report (ReMR) provides information about existing software products intended to be reused as-is or modified as part of the delivered operational software. The report also provides the acquirer insight into the current status of the activities associated with the reuse of these products as compared to the planned activities, and alternative approaches.

This Data Item Description (DID) contains the format, content and intended use information for the data product resulting from the work tasks described in the contract.

Requirements:

1. Reference documents. The applicable issue of the documents cited herein, including their approval dates and dates of any applicable amendments, notices, and revisions, shall be as cited in the contract.

2. Format. Contractor format is acceptable.

3. Content. The report shall contain the following:

3.1 Executive Summary. This section shall identify and briefly describe all software products that will be reused as-is or modified (existing software requiring change) and integrated into the delivered operational software. Both commercial off-the-shelf (COTS) and government off-the-shelf (GOTS) shall be included.

3.2. Current Status. This section shall present the current status of the software reuse activities compared to the contractor’s planned activities. Status shall include progress made and accomplishments for the engineering and management activities for each software reuse product.

3.3. Variance. This section shall identify any activities where work is not progressing in accordance with the plans and schedules, including the reasons for this lack of progress.

3.4. Milestones. This section shall describe the progress made against program milestones during the reporting period.

3.5. Alternative Approaches. This section shall describe alternative approaches for any reused as-is/modified and COTS/GOTS software products that are 1) considered high or moderate risk or 2) not available (i.e., fully documented and tested) at contract award. Alternative approaches provide options if these software reuse products cannot be implemented as planned.

3.6. Impacts of Implementing Alternative Approaches. This section shall include an assessment of the technical impacts to the program and estimates of the programmatic (i.e., effort and schedule) impacts of implementing alternative approaches.

3.7. Decision Points. This section shall include the decision points for implementing alternative approaches. These decision points shall identify when the alternative approach would need to be implemented in the event that the planned software reuse products are not available in time to preserve the program schedule.

3.8. Worksheet Questions. This section shall include the completed forms (updated, if needed) for all reused as-is/modified and COTS/GOTS software products in accordance with the instructions embedded in the forms. Forms will be included for any newly identified software reuse products.

WORKSHEET QUESTIONS FOR
REUSED AS-IS/MODIFIED SOFTWARE

The questions below should be answered in the corresponding worksheet format for each software product for which the Contractor plans to assume responsibility for the performance of the product. Software products may be reused as-is or modified. Information about the commercial off-the-shelf (COTS) or Government off-the-shelf (GOTS) software products that will be reused as-is should be provided in the worksheet format titled "Worksheet Questions for COTS/GOTS Software."

Product and Contact Information

1. What is the name of the software product to be reused as-is or modified?

2. What is the version number and date of release for the software product that is being reused as-is/modified?

3. What are the programming language(s) of this software?

4. For which system/program was the software originally developed?

5. Provide contact information, including the contact's name, the office symbol (if applicable), phone number and address for the:
    - Program manager currently responsible for the reused as-is/modified software
    - Responsible entity or source of the software, if the Government is not responsible for the software

Applicability

6. To which Computer Software Configuration Item (CSCI) (and Computer Software Component (CSC), if known) is this reused as-is/modified software product assigned?

7. What functions/requirements will the software provide? (Attach a separate sheet that shows performance requirements cross referenced to the Technical Requirements Document (TRD). Identify any mismatches in requirements between the TRD and the reused as-is/modified software product.)

8. Has the Contractor conducted an internal demonstration(s) to evaluate the applicability of this software product for this system/program?
    - If yes, provide additional information

9. Have the software product's interfaces that provide access to the functionality been evaluated?
    - If yes, provide additional information

10. Has the software product's architecture been evaluated for compatibility with the system architecture?
    - If yes, provide additional information

11. Has the software product been used on a hardware/software platform similar to the one proposed for this system/program?

- If yes, provide additional information

Extent of Modification

12. Briefly describe the tasks (e.g., modification, integration, test) required to make the reused as-is/modified software functional within this system.

13. What organization will perform the modifications to this software product?

14. What organization will integrate the reused as-is/modified software with the system's software?

15. What is the effective size of the reused as-is/modified software product and extent of the modification, if applicable? Complete this table, in an Excel workbook, according to the definitions and instructions attached.

A	B	C	D	E	F	G	H	I	J
Total Delivery or Delivered Build	ID Number of CSCI Contained In	Name of CSCI Contained In	CSC Name	Module or Class Level	Development Contractor/ Subcontractor	Sizing Method	New Software	Total Pre-existing Software	Deleted Software

K	L	M	N	O	P	Q	R
Modified Software	Redesign Required (%)	Reimple-mentation Required (%)	Retest Required (%)	Weight for Design Phase (%)	Weight for Implementation Phase (%)	Weight for Test Phase (%)	Effective Size for Modified Software

S	T	U	V	W	X
Reused As-is/ Lifted Software	Reuse As-is/ Lift Factor Required (%)	Effective Size for Reused As-is/Lifted Software	Total Effective Size	Effective Size Representing Software Growth	Total Size

Maturity

16. What is the extent of testing of the software that is to be reused as-is/modified (e.g., completed unit tests, completed CSC tests, completed CSCI tests)?

17. Has formal qualification has been conducted?
    - If yes, provide additional information

18. Has the software been certified and accredited?
    - If yes, provide additional information (e.g., specific certifications and accreditations)

19. Has the software been fielded in an operational environment?
    - If yes, provide additional information

20. Has the software been fielded in an operational environment?
    - If yes, provide additional information (e.g., which systems/programs, whether these systems/programs have fielded the software)

21. Is the software in long-term maintenance?
    - If yes, provide additional information (e.g., organization maintaining the software)

Availability

22. How does the Contractor have access to the software to be reused as-is/modified (e.g., developed the software in-house, has or will acquire the software from another contractor/vendor, or requesting the software be provided by the Government)?

23. Is the software currently available?

- If not, describe the software delivery schedule, including all critical dates that could affect program success.

24. Is the Contractor's solution dependent on another Government program for this software?

- If yes, briefly discuss if cross program (and contractor to contractor) relationships have been established, how they will be managed, and how the Contractor plans to stay informed about the evolving software functionality.

Other Attributes

Designed for Reuse

25. Identify any attributes (e.g., standards, design patterns, architecture paradigms) of the reused as-is/modified software that support reuse.

Contractor's Experience with Software

26. Will the Contractor have any access to the software developers, who were part of the original software development team?

- If yes, provide additional information

27. Has the organization (that will be performing the modifications to this software product for this program) reused as-is or modified (e.g., altered the design, made changes to the code) the software previously?

- If yes, provide additional information (e.g., for what systems/programs, how many of the software developers have modified this software product before)

28. Has the organization (that will be integrating this software product for this program) integrated the software previously?

- If yes, provide additional information (e.g., for what systems/programs, how many of the software integration engineers have integrated this software product before)

Documentation

29. What supporting engineering and management documentation for the reused as-is/modified software is available for the software developers?

30. What supporting documentation for the reused as-is/modified software is available for the end users?

31. What documentation (both development and end user) will be delivered to the Government?

32. Describe the test procedures that will support the conduct of the comprehensive

regression testing for the reused as-is/modified software?

- Do these procedures exist or do they need to be created?

Standards

33. What development standards (e.g., IEEE/EIA Std 12207.0-2008) were followed during the development of the software intended to be reused as-is/modified?

Data and Software Rights

34. What rights will the Government have to the data and software? Identify what data and software rights are being provided to the Government using the relevant Defense Federal Acquisition Regulation Supplement (DFARS) clause definitions (DFARS 227.7103-3 and 227.7203-3.1. What is the name of the COTS/GOTS software product to be reused as-is?

35. Does the reused as-is/modified software require the Government to purchase any COTS software licenses? If yes, provide the commercial software licenses for review.

36. Do you intend to transfer any COTS software licenses to the Government? If yes, provide the commercial software licenses for review.

Defect Reports

37. How many Defect Reports (DRs) are currently open for the software?

38. Provide a listing of all (open and closed) DRs by category/priority, date when opened, description of problem and planned/actual date of closure.

Maintenance and Support Strategy

39. What organization is expected to maintain the modified software?

Releases/Updates

40. Will the Contractor incorporate future releases of the reused as-is/modified product into the system's software baseline?

- If yes, how will these releases be incorporated

Dead and Unused Code

41. Identify any dead code (i.e., unreachable, unnecessary, or inoperative code that is not required for any purpose) and unused code (i.e., code used in applications other than this program) from the reused as-is and modified software products. Discuss how dead and unused code will be handled, how it will be tested, and whether it presents any risks to the program.

WORKSHEET QUESTIONS FOR REUSED AS-IS/MODIFIED SOFTWARE
Instructions for Completing Question 15

Definitions: Total Delivery or Delivered Build (Col. A): If there are multiple delivered builds (blocks, increments, etc.), enter the build identifier for the sizing information provided. Enter "Total" if the sizing information represents the total delivery. A separate table should be completed for each delivered build as well as the total delivery.

ID Number of CSCI Contained In (Col. B): Enter the identification numbers for the Computer Software Configuration Item (CSCI), in which the reused as-is/modified software product is contained.

Name of CSCI Contained In (Col. C): Enter the name of the CSCI, in which the reused as-is/modified software product is contained.

CSC (Col. D): Enter the names of the Computer Software Components (CSCs), if known. A separate row should be completed for each CSC.

Module or Class Level (Col. E): Enter the software module or class level, if known. A separate row should be completed for each software module or class level.

Development Contractor/Subcontractor (Col. F): Enter the name of the contractor or subcontractor responsible for the development of each CSCI.

Sizing Method (Col. G): Enter either Source Lines of Code (SLOC) or Function Points (FP). Standard definitions for SLOC and FPs are provided below. Fully explain any non-standard definition on a separate sheet. If an alternative sizing measure is used, the counting method should be described in detail. This table can be adapted to accommodate an alternative measure, but the type of information requested in these instructions must be included.

> Lines of Code: Non-Comment lines of source code for the computer program. Source lines to include are: All executable source lines such as (1) Control, (2) Mathematical, (3) Conditional, (4) Deliverable Job Control, (5) Data Declaration Statements, and (6) Data Typing and Equivalence; and input/output/format. Source lines to exclude are: debug statements, continuation of single statement to multiple lines, machine/library generated statement, and non-deliverable test statements.
>
> Function Points: Unadjusted function points, IFPUG compatible. Use this only if your size methods are function based rather than line based.

New Software (Col. H): Enter the new non-comment lines of source code or the new number of unadjusted function points, IFPUG compatible for the computer program.

New code is software developed from scratch and is not modified or reused as-is in any way from any pre-existing design or code.

Total Pre-existing Software (Col. I): Enter the number of lines of code or functions in a pre-existing software package (before reuse as-is/modification/deletion), including lines of code or functions that may not be pertinent to this program/system.

Deleted Software (Col. J): Enter the number of lines of code or functions which will be deleted from the pre-existing software package (Col. I). The deletion will be accomplished by physical omission or commenting out.

Modified Software (Col. K): Enter the total number of lines of code or functions that will be modified from a pre-existing software package through re-design or re-implementation, and then integrated and tested in the new software product baseline. If there are multiple delivered builds, this number should represent the code developed in previous builds that may need to be modified or re-tested with the code being developed for the current build.

Redesign Required (Col. L): Enter the percentage of the pre-existing software to be modified (Col. K) that requires redesign to make this software functional within the new environment.

Reimplementation Required (Col. M): Enter the percentage of the pre-existing software to be modified (Col. K) that requires reimplementation (i.e., code and unit test) to make this software functional within the new environment.

Retest Required (Col. N): Enter the percentage of the pre-existing software to be modified (Col. K) that requires retesting (i.e., CSC integration/test and CSCI integration/test, but excluding CSCI-to-CSCI integration/test) to ensure this software functions within performance, reliability, and other criteria after the modifications.

Weight for Design Phase (Col. O): Enter the percentage of the software development effort (i.e., design, implementation and test) attributed to the design phase. The weights (Col. O-Q) represent the phase distribution of effort, i.e., the distribution typically observed for each phase. Note that the sum of the weights for the design, implementation and test phases must equal 100 percent.

Weight for Implementation Phase (Col. P): Enter the percentage of the software development effort (i.e., design, implementation and test) attributed to the implementation phase.

Weight for Test Phase (Col. Q): Enter the percentage of the software development effort (i.e., design, implementation and test) attributed to the test phase.

Effective Size for Modified Software (Col. R): Enter the number of lines of code or functions that represent the pre-existing lines or functions that will be modified (Col. K) and are adjusted based on the applicable percentages (Col. L-N) and weights (Col. O-Q). Effective size represents the software size equivalent to developing the code from scratch. The formula, in this spreadsheet, used for calculating Effective Size for the Modified Software is:

Col. R = Col. K * ((Col. L * Col. O) + (Col. M * Col. P) + (Col. N * Col. Q))

Explain the method used for calculating effective size if it differs from the formula in this table.

Reused As-is/Lifted Software (Col. S): Enter the number of lines of code or functions that will be reused as-is or lifted, with no modification of design or code, from a pre-existing software package.

Reuse As-is/Lift Factor Required (Col. T): Enter the percentage that is applied to the pre-existing software to be reused as-is/lifted (Col. S) to estimate effective size. This percentage is similar in concept to the percentages for redesign required, reimplementation required and retest required, but is a composite factor applied to the software that will be reused as-is/lifted. Reused as-is/lifted code, by definition, will not require modification.

Reused as-is software products may require new code, such as glue code, wrappers, or plug-ins, or parameterization, but the software product itself will remain unchanged. The new lines of source code or functions associated with the software (e.g., glue code, integration code) should be included as New Software (Col H). The effort required to understand the product and its interfaces, integrate the product as part of a CSC and CSCI, and perform testing should be reflected in the percentage in order to estimate effective size.

Effective Size for Reused As-is/Lifted Software (Col. U): Enter the number of lines of code or functions that represent the pre-existing lines or functions that will be reused as-is/lifted (Col. 18) and are adjusted based on the applicable percentage (Col. 19). Effective size represents the software size equivalent to developing the code from scratch. The formula, in this spreadsheet, used for calculating Effective Size for the Reused As-is/Lifted Software is:

Col. U = Col. S * Col. T

Explain the method used for calculating effective size if it differs from the formula in this table.

Total Effective Size (Col. V): Enter the number of lines of code or functions that represent all new lines or functions (Col. H) as well as the pre-existing lines or functions that are modified (Col. K) or reused as-is/lifted (Col. S) and are adjusted based on the applicable percentages (Col. N-N, T) and weights (Col. O-Q). Effective size represents the software size equivalent to developing the code from scratch. The formula, in this spreadsheet, used for calculating Total Effective Size is:

Col. V = Col. H + Col. R + Col. U

Effective Size Representing Software Growth (Col. W): Enter the number of effective lines or functions that are included in the effective size estimate (Col. V) to capture software growth. Provide the definition of software growth used for the sizing estimate and the method used to estimate software growth.

Total Size (Col. X): Enter the number of new (Col. H) and pre-existing (Col. K and S) lines of code or functions. Total size represents the total amount of new software that would need to be developed for the new software baseline, if no code were to be reused as-is or modified. Note that Total Size does not include the pre-existing software that will be deleted.

WORKSHEET QUESTIONS FOR COTS/GOTS SOFTWARE

The questions below should be answered in the corresponding worksheet format for each commercial off-the-shelf (COTS) or Government off-the-shelf (GOTS) software product that will be reused as-is. COTS/GOTS software include the products, for which the software provider, either a commercial vendor or the Government, assumes responsibility for the performance of the software product. The source code is not necessarily provided to the Contractor. The COTS/GOTS software products may require new code, such as glue code, wrappers, or plug-ins, or parameterization, but the COTS/GOTS product itself will remain unchanged.

Product and Contact Information

1. What is the name of the COTS/GOTS software product to be reused as-is?

2. What is the version number and date of release for the COTS/GOTS software product that is being reused as-is?

3. Provide contact information, including the contact's name, the office symbol (if applicable), phone number and address for the:
- COTS/GOTS software provider
- Government program manager currently responsible for a program, if any, that uses the COTS/GOTS software product

Applicability

4. To which Computer Software Configuration Item (CSCI) (and Computer Software Component (CSC), if known) is this software product assigned?

5. What functions/requirements will the software provide? (For submissions with the proposal, attach a separate sheet that shows functions cross referenced to the Technical Requirements Document (TRD). Identify any mismatches in functionality between the TRD and the COTS/GOTS product. After contract award, attach a separate sheet that shows performance requirements cross referenced to the TRD. Identify any mismatches in requirements between the TRD and the COTS/GOTS product.)

6. Has the Contractor conducted an internal demonstration(s) to evaluate the applicability and usability of this product for this system/program?
- If yes, provide additional information

7. Have the COTS/GOTS product's interfaces that provide access to the functionality been evaluated?
- If yes, provide additional information

8. Has the software product's architecture been evaluated for compatibility with the system architecture?
- If yes, provide additional information

9. Are the COTS/GOTS product's development and target hardware/software platforms similar to the ones proposed for this system/program?

- If yes, provide additional information

Approach to Integration/Test

10. Briefly describe the tasks (e.g., development of glue code, integration, test) required to make the COTS/GOTS software functional within this system.

11. What organization will be responsible for developing any new code (e.g., glue code, integration code) needed?

12. What organization will integrate this COTS/GOTS product with the system's software?

13. Does the Contractor need access to the source code?

- If yes, does the Contractor have access to the source code?

14. What is the effective software size associated with the COTS/GOTS software product? Complete this table, in an Excel workbook, according to the definitions and instructions attached.

A	B	C	D	E	F	G
Total Delivery or Delivered Build	**ID Number of CSCI Contained In**	**Name of CSCI Contained In**	**CSC Name**	**Development Contractor/ Subcontractor**	**Sizing Method**	**New Software**

H	I	J	K	L	M
Reused As-is/ Lifted Software	**Reuse As-is /Lift Factor Required (%)**	**Effective Size for Reused As-is/Lifted Software**	**Total Effective Size**	**Effective Size Representing Software Growth**	**Total Size**

Maturity

15. When was the COTS/GOTS product first released?

16. How many versions (both major and minor updates) have subsequently been released?

17. Has this COTS/GOTS product been successfully used on any Government program?

- Has it been system-level tested?
- Has it been certified and accredited?
- Has it been fielded?
- If yes, provide additional information (e.g., for what Government program, when, Government agency witnessing test, specific accreditations and certifications)

Availability

18. Is the COTS/GOTS software currently available?

- If not, describe the software delivery schedule, including all critical dates that could affect program success

19. Is the Contractor's solution dependent on another Government program for this software?
    - If yes, discuss if cross program (and contractor to contractor) relationships have been established, how they will be managed, and how the Contractor plans to stay informed about the evolving functionality of the COTS/GOTS software

Other Attributes
Contractor's Experience with COTS/GOTS Product

20. Does the Contractor have experience using the proposed COTS/GOTS software?
    - If yes, for what systems/programs?
21. Does the Contractor have experience integrating the proposed COTS/GOTS software?
    - If yes, for what systems/programs?
22. Has the Contractor worked with the vendor of the COTS software product before?
    - If yes, on which program or in what capacity?

Documentation

23. If yes, on which program or in what capacity?

Data and Software Rights

24. What rights will the Government have to the data and COTS/GOTS software? Identify what data and software rights are being provided to the Government using the relevant Defense Federal Acquisition Regulation Supplement (DFARS) clause definitions (DFARS 227.7103-3 and 227.7203-3.)
25. Does the reused COTS/GOTS software require the Government to purchase any COTS software licenses? If yes, provide the commercial software licenses for review.
26. Do you intend to transfer any COTS software licenses to the Government? If yes, provide the commercial software licenses for review.

Licensing

27. How will the COTS/GOTS software be licensed (e.g., per seat, per site, per host) for both development and run-time for this program?
28. Will the Government be expected to keep track of run-time licenses?
29. Will there be any automated enforcement mechanisms (e.g., license managers, activation)?

Maintenance and Support Strategy

30. What organization is expected to maintain the COTS/GOTS software?

31. For what time frame, will the COTS/GOTS software be maintained?

Releases/Updates

32. Will the Contractor incorporate future releases of the COTS/GOTS product into the system’s software baseline?

- If yes, how will these releases be incorporated?

COTS Vendor Viability

33. How long has the COTS vendor been in business?

34. How many of the vendor’s employees are dedicated to the implementation of the product and to the product’s support?

35. What was the funding source for the development of this product?

36. What is the vendor’s customer base (e.g., commercial, Government defense, or Government nondefense)?

WORKSHEET QUESTIONS FOR COTS/GOTS SOFTWARE
Instructions for Completing Question 14

Total Delivery or Delivered Build (Col. A): If there are multiple delivered builds (blocks, increments, etc.), enter the build identifier for the sizing information provided. Enter "Total" if the sizing information represents the total delivery. A separate table should be completed for each delivered build as well as the total delivery.

ID Number of CSCI Contained in (Col. B): Enter the identification numbers for the Computer Software Configuration Item (CSCI), in which the COTS/GOTS software product is contained.

Name of CSCI Contained in (Col. C): Enter the name of the CSCI, in which the COTS/GOTS software product is contained.

CSC (Col. D): Enter the names of the Computer Software Components (CSCs), if known. A separate row should be completed for each CSC.

Development Contractor/Subcontractor (Col. E): Enter the name of the contractor or subcontractor responsible for the development of each CSCI.

Sizing Method (Col. F): Enter either Source Lines of Code (SLOC) or Function Points (FP). Standard definitions for SLOC and FPs are provided below. Fully explain any non-standard definition on a separate sheet. If an alternative sizing measure is used, the counting method should be described in detail. This table can be adapted to accommodate an alternative measure, but the type of information requested in these instructions must be included.

> Definitions:
> Lines of Code: Non-Comment lines of source code for the computer program. Source lines to include are: All executable source lines such as (1) Control, (2) Mathematical, (3) Conditional, (4) Deliverable Job Control, (5) Data Declaration Statements, and (6) Data Typing and Equivalence; and input/output/format. Source lines to exclude are: debug statements, continuation of single statement to multiple lines, machine/library generated statement, and non-deliverable test statements.
>
> Function Points: Unadjusted function points, IFPUG compatible. Use this only if your size methods are function based rather than line based.

New Software (Col. G): Enter the new non-comment lines of source code or the new number of unadjusted function points, IFPUG compatible for the computer program. New code is software developed from scratch and is not modified or reused as-is in any way from any pre-existing design or code. The new lines of source code or functions associated with reusing as-is any COTS/GOTS software (e.g., glue code, integration code) should be included in this column.

Reused As-is/Lifted Software (Col. H): Enter the number of lines of code or functions that will be reused as-is or lifted, with no modification of design or code, from a pre-existing COTS/GOTS software package. If COTS/GOTS software size is not encompassed in the Contractor's overall sizing methodology, attach a separate sheet to explain the estimating methodology.

Reuse As-is/Lift Factor Required (Col. I): Enter the percentage that is applied to the pre-existing COTS/GOTS software to be reused as-is/lifted (Col. H) to estimate effective size.

Reused as-is software products may require new code, such as glue code, wrappers, or plug-ins, or parameterization, but the software product itself will remain unchanged. The new lines of source code or functions associated with the software (e.g., glue code, integration code) should be included as New Software (Col G). The effort required to understand the product and its interfaces, integrate the product as part of a CSC and CSCI, and perform testing should be reflected in the percentage in order to estimate effective size.

Effective Size for Reused As-is/Lifted Software (Col. J): Enter the number of lines of code or functions that represent the pre-existing lines or functions that will be reused as-is/lifted (Col. H) and are adjusted based on the applicable percentage (Col. I). Effective size represents the software size equivalent to developing the code from scratch. The formula, in this spreadsheet, used for calculating Effective Size for the COTS/GOTS software is:

Col. J = Col. H * Col. I

Explain the method used for calculating effective size if it differs from the formula in this table.

Total Effective Size (Col. K): Enter the number of lines of code or functions that represent all new lines or functions (Col. G) as well as the pre-existing lines or functions that are reused as-is/lifted (Col. J). Effective size represents the software size equivalent to developing the code from scratch. The formula, in this spreadsheet, used for calculating Total Effective Size is:

Col. K = Col. G + Col. J

Effective Size Representing Software Growth (Col. L): Enter the number of effective lines or functions that are included in the effective size estimate (Col. K) to capture software growth. Provide the definition of software growth used for the sizing estimate and the method used to estimate software growth.

Total Size (Col. M): Enter the number of new (Col. G) and pre-existing (Col. H) lines of code or functions. Total size represents the total amount of new software that would need to be developed for the new software baseline, if no code were to be reused as-is.

4. End of DI-SESS-81771