



---

# **Using SLIM to Objectively Compare COTS, New Development, and Enhancement Alternatives**

---

***September 28, 2006***

---

# Contacts

---

Questions regarding this presentation should be addressed to:

**Joseph Madden**

Senior Manager

BearingPoint

1676 International Drive

McLean, VA 22102

Tel: +1.703.747.5074

E-mail: [joseph.madden@bearingpoint.com](mailto:joseph.madden@bearingpoint.com)

# Introduction

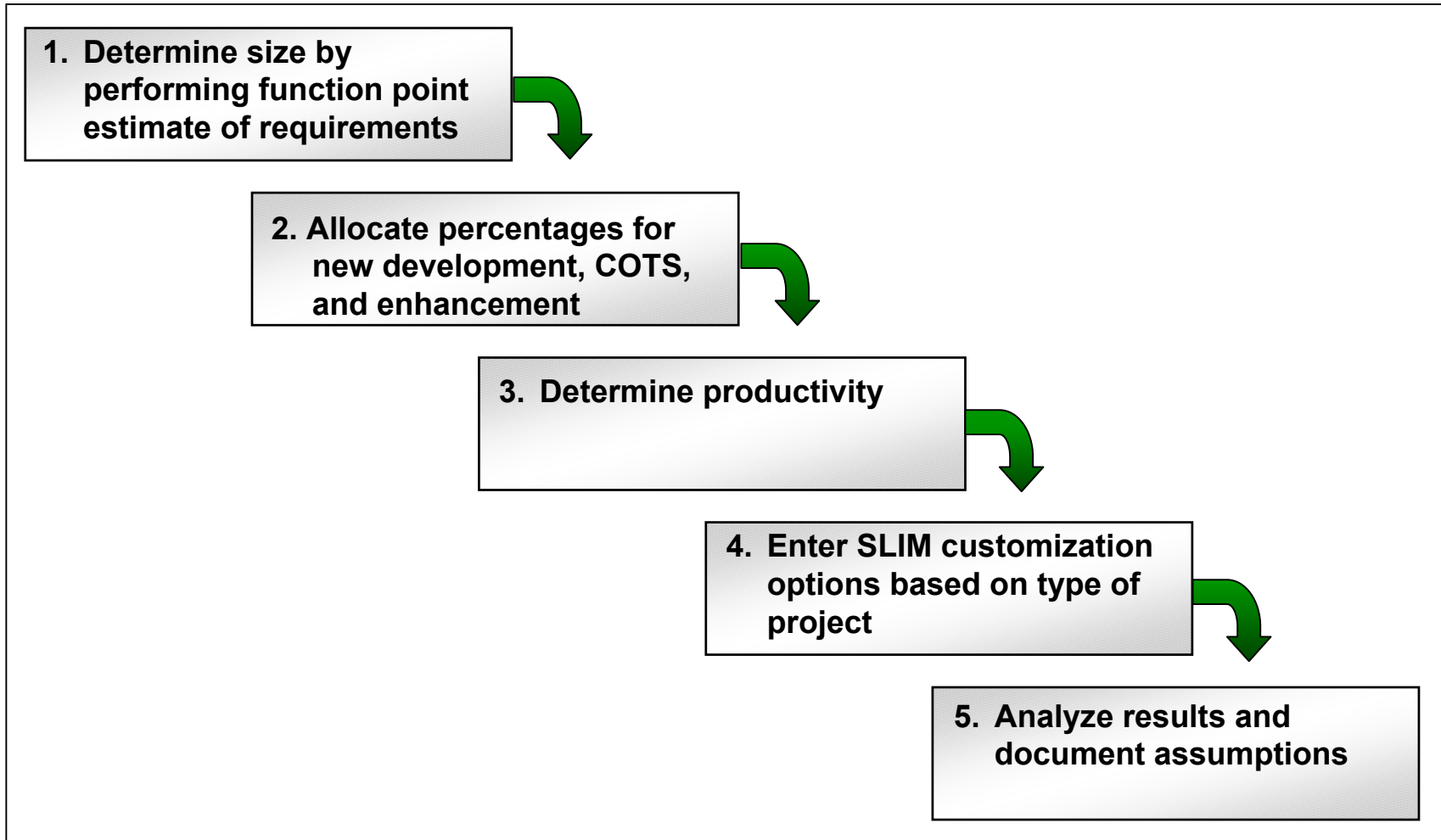
---

Unfortunately, make vs. buy vs. enhance decisions are often made subjectively and without much rigor. Examples of the impact of these decisions include:

- COTS implementation projects that require so much customization that the end result is a “Frankenstein” version of the original COTS product.
- Legacy system enhancement projects that requires a complete rewrite of the entire code base.
- New custom development projects that result in duplication of functionality that already exists in a COTS tool or legacy system.

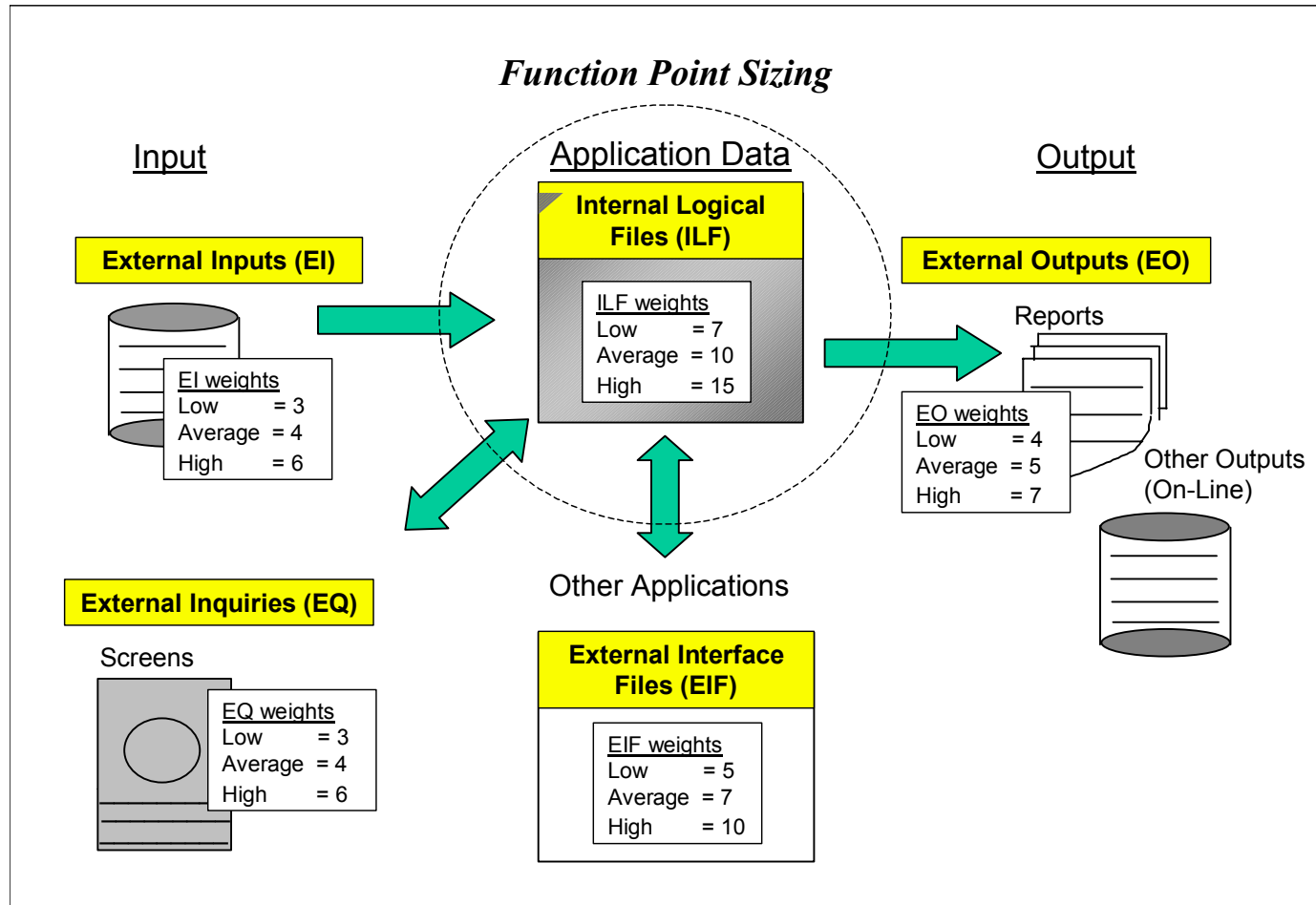
# Approach

Our five step approach provides a method for comparing COTS, new development and enhancement alternatives more objectively.



# Step 1: Determine Size Using Function Point Analysis

The most important first step is to size the functional requirements using a method that is technology independent and based on a standard. Function Point Analysis is one such method.



# Function Points vs. Lines of Code

---

Function points were invented by Allan Albrecht of IBM in 1979 to address problems with using lines of code for software size:

- There is no standard for a line of code.
- Lines of code do not measure a completed product.
- The lines of code required for a given function vary widely between different programming languages and tools (e.g. on average, COBOL requires almost twice as many lines of code as Visual Basic).
- The lines of code written for a given function vary widely between programmers of various skill levels.
- Lines of code reward poor design and penalize tight design.
- Lines of code penalize reuse.

# Function Points vs. Lines of Code (Cont.)

---

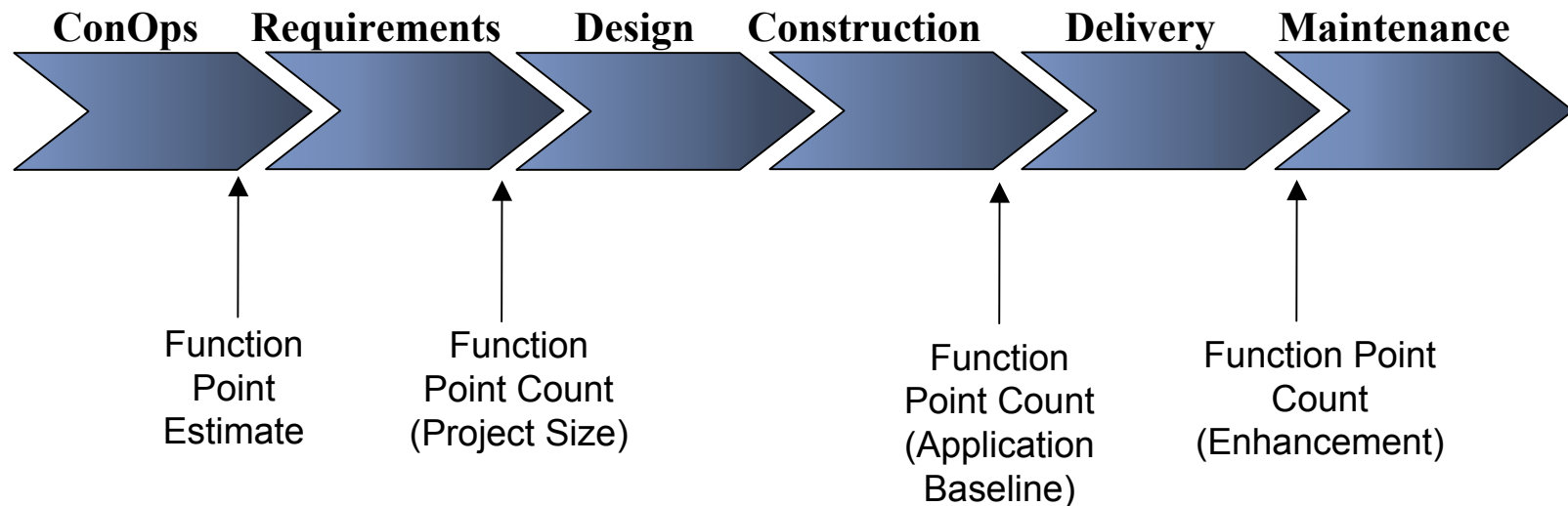
Function points offer many advantages over lines of code and other sizing techniques:

- **Technology and Platform Independent** – Function point analysis sizes software based on user requirements, allowing for apples-to-apples comparisons between projects using different technology and/or platforms. It can be used to objectively compare various alternatives such as COTS, enhancement, and new development.
- **Industry-wide Acceptance** - Function point analysis is based on an International Organization for Standardization (ISO) standard. It has been used for over 25 years and is the most widely used method for measuring the functional size of software. Over 400 government and industry organizations belong to the International Function Point Users Group (IFPUG).
- **Consistent Measurement** – IFPUG ([www.ifpug.org](http://www.ifpug.org)) maintains a function point counting standard as well as a formal certification program. Counts performed by Certified Function Point Specialists (CFPS) do not vary by more than 10 percent.
- **Can Be Used Early in the Software Life Cycle** – Reliable software estimates can be performed early in the requirements phase, allowing for more predictable schedules.

# Function Point Analysis During the Life Cycle of an Application

Function point analysis can be used to estimate project size long before any code is written. It can also be used later in the life cycle to validate the as-built functionality and to estimate the size of proposed enhancements.

## *System Development Life Cycle*





# Step 2: Allocate Percentages of Functionality to Each Alternative

The example below identifies functions that can be satisfied by COTS vs. custom code.

| Function  | Type | Count             | Custom Code        | Custom Code | COTS Box        | COTS Box Subtotal | COTS Config        | COTS Config |
|---|------|-------------------|--------------------|-------------|-----------------|-------------------|--------------------|-------------|
| Create Home Study Allowance Approval Package        | EI   | 4                 | 4                  |             |                 |                   |                    |             |
| View list of Home Study Allowance Approval Packages | EQ   | 3                 |                    |             |                 |                   | 3                  |             |
| Review Proposed Home Study Ed Allow Pkg             | EQ   | 4                 |                    |             |                 |                   | 4                  |             |
| Delete Proposed Home Study Ed Allow Pkg             | EI   | 3                 |                    |             |                 |                   | 3                  |             |
| Approve Proposed Home Study Ed Allow Pkg            | EI   | 4                 |                    |             |                 |                   | 4                  |             |
| Print Proposed Home Study Ed Allow Pkg              | EQ   | 4                 |                    |             | 4               |                   |                    |             |
| Special Needs Education Allowance Approval Package  | ILF  | 10                |                    |             |                 |                   | 10                 |             |
| Special Needs Education Survey                      |      |                   |                    |             |                 |                   |                    |             |
| Special Needs Recommendation Memo                   |      |                   |                    |             |                 |                   |                    |             |
| Create Special Needs Allowance Approval Package     | EI   | 4                 | 4                  |             |                 |                   |                    |             |
| View list of Special Needs Ed Allow Pkg             | EQ   | 3                 |                    |             |                 |                   | 3                  |             |
| Review Proposed Special Needs Ed Allow Pkg          | EQ   | 4                 |                    |             |                 |                   | 4                  |             |
| Delete Proposed Special Needs Ed Allow Pkg          | EI   | 3                 |                    |             |                 |                   | 3                  |             |
| Approve Proposed Special Needs Ed Allow Pkg         | EI   | 4                 |                    |             |                 |                   | 4                  |             |
| Print Proposed Special Needs Ed Allow Pkg           | EQ   | 4                 |                    |             | 4               |                   |                    |             |
| Course Categories                                   | ILF  | 7                 | 7                  |             |                 |                   |                    |             |
| View Course Categories                              | EQ   | 3                 | 3                  |             |                 |                   |                    |             |
| Update Course Categories                            | EI   | 3                 | 3                  |             |                 |                   |                    |             |
| Counties for Special Ed Input Calculations          | ILF  | 7                 | 7                  |             |                 |                   |                    |             |
| View Counties for Special Ed Input Calculations     | EQ   | 3                 | 3                  |             |                 |                   |                    |             |
| Update Counties for Special Ed Input Calculations   | EI   | 3                 | 3                  |             |                 |                   |                    |             |
| Disability Types                                    | ILF  | 7                 | 7                  |             |                 |                   |                    |             |
| View Disability Types                               | EQ   | 3                 | 3                  |             |                 |                   |                    |             |
| Update Disability Types                             | EI   | 3                 | 3                  |             |                 |                   |                    |             |
| Base School for Each Post                           | ILF  | 7                 | 7                  |             |                 |                   |                    |             |
| View Base School at Each Post                       | EQ   | 3                 | 3                  |             |                 |                   |                    |             |
| Update Base School at Each Post                     | EI   | 3                 | 3                  |             |                 |                   |                    |             |
| <b>Subtotal Phase 2</b>                             |      | 806               |                    | 397         |                 | 50                |                    | 359         |
|   |      | <b>Total Size</b> | <b>Custom Code</b> |             | <b>COTS Box</b> |                   | <b>COTS Config</b> |             |
|   |      | 1123              | 557                |             | 99              |                   | 467                |             |

# Gearing Factor

---

The SLIM tool uses what is called a gearing factor, Source Lines of Code (SLOC) per function point, to account for differences in implementation tools. QSM, Inc. maintains a table of gearing factors on its website. Based on this table, and advice from QSM, Inc., we made the following assumptions in the examples that follow:

- New custom code will be in a modern, object-oriented, programming language similar to Java with a gearing factor of 80.
- Enhancements to the legacy system will be made in PowerBuilder with a gearing factor of 31 .
- COTS configurations will be made with a gearing factor of 10.

# Example: New Development Alternative

The example below is an alternative that assumes new development for most functions.

| Phase 2 New Development  |           |         |           |                        |    |
|--------------------------|-----------|---------|-----------|------------------------|----|
| Reused, unmodified code  | Size (FP) | % Total |           |                        |    |
| COTS out-of-the-box      | 3         | 0%      |           |                        |    |
| Phase 1 functionality    | 317       | 28%     |           |                        |    |
|                          | 320       | 28%     |           |                        |    |
| System integration work  |           |         | % SI work | SLIM gearing factor    |    |
| COTS configure           | 19        | 2%      | 2%        | 10                     | 0  |
| Custom code              | 784       | 70%     | 98%       | 80                     | 78 |
|                          | 803       | 72%     |           | Blended gearing factor | 78 |
| Total size input to SLIM | 1123      | 100%    |           |                        |    |

A blended gearing factor is derived for any functions that are not satisfied with reused, unmodified code. The gearing factor is then entered into SLIM along with the total size and percentage of new, modified and reused code.

Expected Total Size

Total FP

1123

Gearing Factor

78

New %

72

Mod %

0

Reused %

28

Size Calculator...

Eff FP 809

Uncertainty Range

Low

High

99% Eff Range

786 to 1,460

# Example: COTS Alternative

The example below is an alternative that assumes COTS will be used to satisfy functionality wherever possible. Those functions not satisfied by the COTS product will have to be developed using custom code.

|                          |             |            |           |                        |           |
|--------------------------|-------------|------------|-----------|------------------------|-----------|
| <b>Phase 2 COTS</b>      |             |            |           |                        |           |
|                          |             |            |           |                        |           |
| Reused, unmodified code  | Size (FP)   | % Total    |           |                        |           |
| COTS out-of-the-box      | 50          | 4%         |           |                        |           |
| Phase 1 functionality    | 317         | 28%        |           |                        |           |
|                          | 367         | <b>33%</b> |           |                        |           |
|                          |             |            |           |                        |           |
| System integration work  |             |            | % SI work | SLIM gearing factor    |           |
| COTS configure           | 359         | 32%        | 47%       | 15                     | 7         |
| Custom code              | 397         | 35%        | 53%       | 80                     | 42        |
|                          | 756         | <b>67%</b> |           | Blended gearing factor | <b>49</b> |
|                          |             |            |           |                        |           |
| Total size input to SLIM | <b>1123</b> | 100%       |           |                        |           |

# Example: Enhancement Alternative

The example below is an alternative that assumes enhancement of a legacy system to satisfy as many functions as possible.

|                                |             |             |           |                        |           |
|--------------------------------|-------------|-------------|-----------|------------------------|-----------|
| <b><i>Phase 2 Overhaul</i></b> |             |             |           |                        |           |
|                                |             |             |           |                        |           |
| Reused, unmodified code        | Size (FP)   | % Total     |           |                        |           |
| COTS out-of-the-box            | 3           | 0%          |           |                        |           |
| Existing code                  | 172         | 15%         |           |                        |           |
| Phase 1 functionality          | 317         | 28%         |           |                        |           |
|                                | 492         | <b>44 %</b> |           |                        |           |
|                                |             |             |           |                        |           |
| System integration work        |             |             | % SI work | SLIM gearing factor    |           |
| COTS configure                 | 19          | 2%          | 3%        | 15                     | 0         |
| Changed code                   | 354         | 32%         | 56%       | 32                     | 18        |
| Custom code                    | 258         | 23%         | 41%       | 80                     | 33        |
|                                | 631         | <b>56 %</b> |           | Blended gearing factor | <b>51</b> |
|                                |             |             |           |                        |           |
| Total size input to SLIM       | <b>1123</b> | 100%        |           |                        |           |

## Step 3: Determine Productivity

---

For the example alternatives analysis, we started with a baseline Productivity Index (PI) representative of the SLIM database average productivity for applications of similar size and type. We then made several adjustments to the PI. For each of the alternatives, we made PI adjustments based on the percentage of reused, unmodified code that will have to be re-tested. This includes:

- COTS out-of-the-box
- As-is code from the current system
- Code from a previous phase

For the enhancement alternative, we made PI adjustments to tooling/methods to account for the fact that the existing legacy system is written in tool versions that are no longer supported.

# Step 4: Enter SLIM Customization Options

---

The SLIM tool provides a number of customization options. Some of the customization options that should be considered include:

- Phases to be included in the estimate - often COTS and enhancement alternatives require a front end feasibility study phase.
- Labor rate assumptions for cost – are they different for each alternative?
- Application type

# Step 5: Analyze Results and Document Assumptions

---

Once the SLIM assumptions and constraints are determined, the final step is to analyze tradeoffs between cost, schedule, and peak staff to come up with the final SLIM cost and schedule estimates for each alternative. Some additional items to consider outside of the SLIM tool include:

- Data migration costs.
- COTS license fees
- Operations and maintenance costs



# Summary

---

The SLIM tool can be used successfully to compare COTS, new development, and enhancement alternatives. The key takeaways are:

- Size the requirements using a technology independent method such as function point analysis.
- Identify the tools that will be used for each alternative and determine the gearing factor.
- For each alternative, determine how each function will be implemented (e.g. COTS out of the box, COTS configuration, custom code, enhancement to legacy code). Calculate a blended gearing factor and percentage of reused, unmodified code.
- Calibrate the PI for each alternative.
- If appropriate for a given alternative, add an extra front end feasibility study phase.