*UNIVERSITY OF SOUTHERN CALIFORNIA*
*CENTER FOR SOFTWARE ENGINEERING*

## COTS Software Integration Cost Modeling

# USC COCOTS Model

# Project Level COTS Integration Experience Survey

**Prepared by**
**Chris Abts**
**(with Betsy Clark)**

**Version 0.7**

**3 March 2000**

# Table of Contents

# I. Introduction

### Purpose and Scope

The goal of this survey is to capture from software development organizations specific COTS and NDI product integration experience at the project level;  that is, at the level at which the integration work is actually being performed. This information will be used to refine and calibrate the Constructive COTS Integration Cost (COCOTS) model currently under development at the University of Southern California. The intent of this modeling effort is to create a tool which will be able to reasonably predict the expected initial cost of integrating COTS and NDI software into a new software system development or system refresh. COCOTS is a broad-based model, currently focusing on four major sources of integration costs:  1) COTS product assessment, 2) COTS product tailoring, 3) Integration or "glue code" development, and 4) Application volatility due to use of COTS products. It currently does not account for costs associated with licensing, life cycle issues beyond initial development, or post initial development maintenance issues. While significant, these costs will be addressed in later stages of the COCOTS modeling effort.

### Disclaimer

**NO INFORMATION DIVULGED IN THIS SURVEY WILL BE USED BY ANY AGENCY OF THE FEDERAL GOVERNMENT OR ANY OTHER ORGANIZATION EITHER PUBLIC OR PRIVATE FOR THE PURPOSES OF SOURCE SELECTION, OR EVALUATIVE ASSESSMENT OF RESPONDING UNITS OR ORGANIZATIONS IN ANY FASHION.**

### Execution

This is a detailed survey.  We are looking for as much accuracy as possible, because we intend to use this data to *calibrate* our model, making consistent and accurate responses crucial.  To that end, we would prefer the return of <u>empirical</u> project effort, productivity, and sizing data over best "educated estimates, " though we recognize the latter may be all that is available in some circumstances. With that in mind, it is suggested that you quickly read through all the survey questions once before trying to answer them so you can decide beforehand whether you might need some help gathering the information requested in some of the questions.

Please record your answers to the survey directly on this form.  We ask that you make the best effort possible to provide an answer to *all* the questions.  If you are unsure of an answer, or feel a question does not apply to your project, please indicate so rather than leave a question blank.

Completed forms should be returned to the contact identified below under **Survey Point of Contact/Data Submission**.

## Time Required

Based upon pilot executions of the survey, WITH THE DATA AT HAND, it is expected that an informed individual should need on average no more than two hours to complete the questionnaire. However, gathering and preparing the information needed to complete the survey could potentially require SIGNIFICANTLY MORE TIME, perhaps even some part time effort spread over several days or weeks. Please keep this in mind if you have been asked to return this survey within a specific time frame. Again, please skim through the survey quickly one time upon first receiving it to decide how much time you will require to gather the necessary information. Also, note that some sections ask for more detailed data than others. This is a reflection of the weight placed on contribution to overall COTS integration costs. Those areas with greater overall impact are examined in greater detail.

## Survey Point of Contact/Data Submission

To return the completed survey, or if you have questions about it, or desire assistance in filling it out, please contact either:

**Mr. Chris Abts**
Center for Software Engineering
University of Southern California
Henry Salvatori Hall Room 328
Los Angeles, California 90089-0781
tel: 213.740.6470
fax: 213.740.4927
e-mail: cabts@sunset.usc.edu

or

**Dr. Elizabeth (Betsy) Bailey Clark**
Software Metrics, Inc.
4345 High Ridge Road
Haymarket, Virginia 20169
tel: 703.754.0115
fax: 703.754.0115
e-mail: BetsyClark@erols.com

## COCOTS Information on the Web:

http://sunset.usc.edu/COCOTS/cocots.html

# II. Definitions/Glossary

Application Volatility - creates difficulty in benchmarking stable system configurations, resulting from the use of COTS products which may experience multiple or frequent product releases or upgrades during system development.

Attribute - characteristic of a COTS package or associated  products and services which are evaluated and used in comparing alternative products and as input into a buy/no buy decision.

COTS Assessment - the activity of determining the appropriateness or feasibility of using specific COTS products to fulfill required system functions.

COTS software - “commercial-off-the-shelf” software commercially available as stand-alone products and which offer specific functionality needed by a larger system into which they might be incorporated.  Generally there is no access to source code for COTS products, which are treated as black boxes with application program interfaces.  (In some cases, however, some access to COTS source code is available, in which case these products have been described as “gray” or “white” box COTS.)

COTS Tailoring - the activity associated with setting or defining shell parameters or configuration options available for a COTS product, but which do not require modification of COTS source code, including defining I/O report formats, screens, etc.

NDI software - “non-developmental item” software available from some source other than the organization developing the system into which the NDI component is to be integrated. The source can be commercial, private, or public sector, just so long as the procuring organization expended no resources on the NDI component’s initial development. Source code is usually available for an NDI component, which may or may not be able to function as a stand-alone item.

Integration or “glue” code - software developed in-house and composed of 1) code needed to facilitate data or information exchange between the COTS/NDI component and the system or other COTS/NDI component into which it is being integrated, 2) coded needed to connect or “hook” the COTS/NDI component into the system or other COTS/NDI component but does not necessarily enable data exchange, and 3) code needed to provide required functionality missing in the COTS/NDI component AND which depends upon or must interact with the COTS/NDI component.

| | |
|---|---|
| AA | Percentage of reuse effort due to assessment and assimilation |
| AAF | Adaptation Adjustment Factor |
| AAM | Adaptation Adjustment Multiplier |
| ASLOC | Adapted Source Lines of Code |
| BRAK | Breakage. The amount of controlled change allowed in a software development before requirements are "frozen." |
| CASE | Computer Aided Software Engineering |
| CM | Percentage of code modified during reuse |
| COCOMO | Constructive Cost Model |
| Cost Driver | A particular characteristic of the software development that has the effect of increasing or decreasing the amount of development effort, e.g. required product reliability, execution time constraints, project team application experience. |
| COTS | Commercial Off The Shelf |
| DI | Degree of Influence |
| DM | Percentage of design modified during reuse |
| ESLOC | Equivalent Source Lines of Code |
| FP | Function Points |
| GFS | Government Furnished Software |
| IM | Percentage of integration redone during reuse |
| KASLOC | Thousands of Adapted Source Lines of Code |
| KESLOC | Thousands of Equivalent Source Lines of Code |
| KSLOC | Thousands of Source Lines of Code |
| PM | Person Months. A person month is the amount of time one person spends working on the software development project for one month. |
| SEI | Software Engineering Institute |
| SLOC | Source Lines of Code |
| SU | Percentage of reuse effort due to software understanding |
| UNFM | Programmer Unfamiliarity |

# ASSESSMENT ATTRIBUTE DEFINITIONS:

**CORRECTNESS**

**Accuracy** - The freedom of system output from error.

**Correctness** - The degree to which a COTS component is free from faults in its specification, design, and implementation.

**AVAILABILITY/ROBUSTNESS**

**Availability** - The degree to which a COTS component is operational and accessible when required for use. Often expressed as a probability.

**Fail safe** - Pertaining to a COTS component that automatically places itself in a safe operating mode in the event of a failure.

**Fail soft** - Pertaining to a COTS component that continues to provide partial operational capability in the event of certain failures.

**Fault tolerance** - Pertaining to a COTS component that is able to continue normal operation despite the presence of faults.

**Input Error tolerance** - The ability of a COTS component to continue normal operation despite the presence of erroneous inputs.

**Redundancy** - The presence of auxiliary components in a system to perform the same or similar functions as other elements for the purpose of preventing or recovering from failures.

**Reliability -** The ability of a COTS component to perform its required functions under stated conditions for a specified period of time; the probability that a COTS component will perform its intended functions satisfactorily for a prescribed time and under stipulated conditions.

**Robustness** - The degree to which a COTS component can function correctly in the presence of invalid inputs or stressful environmental conditions.

**Safety -** Protection against software or hardware faults that could result in harm to people, data or systems.

**SECURITY**

**Security (Access Related) -** the degree to which a system or component prevents unauthorized access to, or modification of, computer programs or data.

**Security (Sabotage Related) -** Protection against exploitable weaknesses that could result in harm to people, data, or systems.

**PRODUCT PERFORMANCE**

**Execution Performance** - The degree to which a COTS component performs its functions within given execution timing constraints.

**Information/data Capacity** - The quantity of information or logical data items that can be stored or maintained by a system or COTS component relative to the expected needs of the users.

**Precision** - The degree of exactness or discrimination with which a quantity is stated; for example, a precision of 2 decimal places versus a precision of 5 decimal places.

**Memory Performance** - The degree to which a COTS component performs its functions within given memory constraints (hard storage and/or virtual storage).

**Response time** - The elapsed time between the end of an inquiry or command to an interactive computer system and the beginning of the system's response.

**Throughput** - The amount of work that can be performed by a COTS component in a given period of time, for example, number of jobs per day.

**UNDERSTANDABILITY**

**Documentation quality** - The degree to which a COTS component contains enough information to explain its objectives, operations, properties and other attributes to be useful in understanding, tailoring, verifying, and operating it.

**Simplicity** - The degree to which a COTS component has a design and implementation that is straightforward and easy to understand.

**Testability** - The degree to which a COTS component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met.

**EASE OF USE**

**Usability/Human Factors** - The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component.

**VERSION COMPATIBILITY**

**Downward compatibility** - Pertaining to software that is compatible with an earlier or less complex version of itself, for example, a COTS component that handles files created by an earlier version of itself.

**Upward compatibility** - Pertaining to software that is compatible with a later or more complex version of itself, for example, a COTS component that handles files created by a later version of itself.

**INTERCOMPONENT COMPATIBILITY**

**Compatibility with other components** - The ability of two or more components to perform their required functions while sharing the same hardware or software environment.

**Interoperability** - The ability of two or more systems or components to exchange information and to use the information that has been exchanged.

**FLEXIBILITY**

**Extendability -** The ease with which features can be added to or around a COTS component in order to increase storage or functional capability.

**Flexibility** - The ease with which a COTS component can be tailored for use in applications or environments other than those for which it was specifically designed or is normally used.

**INSTALLATION/UPGRADE EASE**

**Installation ease** - The ease with which a COTS component can be installed within a hardware or software environment.

**Upgrade/refresh ease -** The ease with which a new version of a COTS component can be installed within a hardware or software environment.

**PORTABILITY**

**Portability -** The ease with which a COTS component can be transferred from one hardware or software environment to another.

**FUNCTIONALITY**

**Functionality** - The degree to which a COTS component has the functional capability needed by a user to solve a problem or achieve an objective; a functional capability that must be met or possessed by a COTS component to satisfy a set of requirements.

**PRICE**

**Initial purchase or lease** – The upfront cost to buy or lease a COTS component.

**Recurring costs** – The periodic (usually annual) cost for maintenance and other COTS-related support.

**MATURITY**

**Product Maturity -** The length of time that a COTS component has been commercially available and/or the size and diversity of its user base.

**Vendor Maturity** - The length of time that a vendor has been in the COTS software business and/or the size and diversity of its user base.

**VENDOR-SUPPORT**

**Response time for critical problems** - The speed with which critical problems are addressed and solutions are put in place by the vendor.

**Support** - Responsiveness in answering user questions, and in dealing with user problems in installing, testing, and using the COTS component.

**Warranty** - The vendor’s written guarantee that the product will perform as specified and that instances of non-compliance will be resolved according to a written agreement between the vendor and the buyer.

## TRAINING

**User training** - The degree to which vendor training results in users who are proficient in using the COTS component to solve problems or accomplish objectives.

## VENDOR CONCESSIONS

**Willingness to escrow source code** - Willingness of the vendor to place the source code in the hands of a third-party, thereby providing protection to the procurer in the event that the vendor goes out of business or stops supporting the COTS component.

**Willingness to Make Modifications** - Willingness to make and maintain procurer-specific modifications to the COTS product, rather than being driven solely by general market demands.

# III. Identifying Information

Date Survey Completed: ______________________________

Organization Name: ______________________________

Name of Preparer: ______________________________

Title of Preparer: ______________________________

Voice Phone Number: ______________________________

Fax Number: ______________________________

E-mail Address: ______________________________

Postal Address: ______________________________

______________________________

______________________________

# IV. Systems Data

4.1 Project Domain

Circle one:

***Core System Functionality***

Operational, Mission Critical

Operational, Non-mission Critical

Support (e.g., software development tools, logistical planning, etc.)

***Communications, Navigation, and Surveillance***

Operational, Mission Critical

Operational, Non-mission Critical

Support (e.g., software development tools, logistical planning, etc.)

***Administrative***

Operational, Business Critical

Operational, Non-business Critical

Support (e.g., software development tools, logistical planning, etc.)

***Other***

Describe:

### 4.2 Development Type

Circle one: New System System Upgrade/Refresh

### 4.3 Development Process

Circle one: Waterfall Spiral Other:__________________________
(Please describe on back of this page.)

### 4.4 Iteration

If the development process reported in 4.3 is iterative (e.g., spiral), indicate the iteration being reported.

Iteration:___________________

4.5 Current Project Phase or Activity

Report the development phase or activity the project is currently undergoing based upon one of the following three schemes:

1) It is recognized that there is always some overlap between activities even if a waterfall process is being followed but you are being asked here for the phase or activity which is the current major focus of the project. Circle that one phase.

2) It is also recognized that some modern development processes—particularly those involving COTS/NDI components—perform certain activities concurrently. For example, Requirements Definition and COTS/NDI Evaluation & Assessment may be undertaken together because system requirements will influence the suitability of given COTS/NDI components, but also the COTS/NDI components that are currently available or on the market may help determine final system requirements. If this is the case, circle the phrase "concurrent phasing" and circle all the major activities the project is currently undergoing simultaneously[=].

3) Finally, if you report that the project is currently not undergoing any development phase or activity because it is completed or in maintenance, it is assumed that the development phases for this completed project include Software Requirements through Integration/Test, including COTS/NDI Assessment. If this is not the case, please describe the correct phasing on the back of this page.

Circle one or more as needed:

Concurrent Phasing

Requirements Definition
COTS/NDI Assessment
Design
System Coding/COTS Integration
Unit Test
System Integration/Test
Development Completed
Maintenance

Where in the life-cycle does COTS/NDI assessment occur (For example, prior or post requirements definition)?:__________________

4.6 Delivery Scheduling

Report the nature of the project delivery scheduling.

Circle one:

| **A** | **B** | **C** | **D** |
|---|---|---|---|
| Delivery acceptance required at one location, no on-going maintenance | Delivery acceptance required at one location, maintenance on-going | Phased delivery acceptance required at more than one location, no on-going maintenance | Phased delivery acceptance required at more than one location, maintenance on-going |

4.7 Project Time Frame

1) For completed projects record the year of completion (i.e. the year of final delivery acceptance); or 2) for projects in maintenance record the year of initial delivery acceptance; or 3) for projects with a phased delivery schedule, the year of first site delivery acceptance; or 4) for projects still in development indicate the current year.

Year:______________ Year determined by criteria (circle one): 1 2 3 4

4.8 Schedule Duration

Record the number of calendar months from the time the development began (i.e. the start of system requirements definition) through either 1) the date of final delivery acceptance if the project was completed, 2) delivery to the first site if the project was/is undergoing a phased delivery of copies at multiple locations, 3) the date of initial delivery acceptance if the project is in maintenance, or 4) the current date if the project is still in development.

Months:____________ Months determined by criteria (circle one): 1 2 3 4

4.9 Development Schedule Life-cycle Phases

Circle all the life-cycle phases ***the schedule*** reported in 4.8 covers:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| System Requirements | | COTS/NDI Assessment | | Detailed Design | | System Integration and Test | |
| | Software Requirements | | Preliminary Design | | Code/COTS Integration and Unit Test | | Maintenance |

4.10 Project Total Effort

Record the total effort expended in Person-Months from the time the development began (i.e. the start of system requirements definition) through either 1) the date of final delivery acceptance if the project was completed, 2) the date of initial delivery acceptance, and then from initial delivery acceptance to the current date if the project is in maintenance, 3) the date of delivery acceptance to the first site if multiple copies of the system have a phased installation schedule to multiple locations, or 4) the current date if the project is still in development.

Development Person-Months:___________ (Maintenance Phase Person-Months:___________)

Person-months determined by criteria (circle one): 1 2 3 4

4.11 Development Effort Life-cycle Phases

Circle all the life-cycle phases ***the total effort*** reported in 4.10 covers:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| System Requirements | | COTS/NDI Assessment | | Detailed Design | | System Integration and Test | |
| | Software Requirements | | Preliminary Design | | Code/COTS Integration and Unit Test | | Maintenance |

(Note: items 4.9 and 4.11 may seem redundant, but sometimes schedule and effort data are not available over precisely identical life-cycle phases.)

4.12 Standard Person-Month

Record the average number of work hours defining a person-month in your organization.

Hours per Person-Month:_____________

4.13 Total Delivered Source Code

Record the total number of lines of source delivered at project completion (or generated to date if the project is still in development or in maintenance), including NDI code, and including COTS glue code. (This question is intended to provide us with a sense of scale of your overall system development.)

Total SLOC:_______________

4.14 SLOC Count Type

Record the unit definition for the SLOC count reported in 4.13.

Circle one:

| | |
|---|---|
| Logical SLOC | Physical SLOC (carriage returns) |
| Physical SLOC (semicolons) | Non-commented/Non-blank SLOC |
| Other: ____________________________ | |

### 4.15 Programming Languages

Record all the various programming languages used in the development and the percentage of the SLOC reported in 4.13 representing each language.

| Language | Percent SLOC |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

### 4.16 Total System Function Points

Record the total number of unadjusted function points counted for system at project completion (or to date if still in development or in maintenance), including NDI functionality, and including COTS glue functionality. (Again, This question is intended to provide us with a sense of scale of your overall system development, this time using an alternate measure.)

Total FP:_______________

### 4.17 System Architecture

Record the nature of the overall system architecture. If the architecture is essentially uniform or homogenous, circle the one style descriptor below which best describes that architecture. If the architecture is substantially a mix of multiple architectural styles, circle as many style descriptors as needed to describe the overall system architecture.

Circle as needed:

| | | | |
|---|---|---|---|
| Pipe & Filter | Distributed | Main/Subroutine | Event Based |
| Multithreaded | Blackboard/Single Layer General Repository | Closed Loop Feedback Control | Real Time |
| Rule Based | Transactional Database Centric | Layered | Other:_____________ |

### 4.18 System Architecting Process

Describe the process which was followed to arrive at the system architecture recorded in 4.17 in the space below. (For example, was there a paper analysis relative to project specifications, at least on the highest risk system elements? Was any prototyping performed or simulations on performance issues conducted? Were formal Architectural Review Boards used? Or was no formal architecting process used at all?):

# V. COTS Assessment Data

The COTS Assessment sub-model presumes COTS component assessment is done in a two-pass manner. The first pass is a "quick and dirty" activity in which minimal effort is expended, based mainly on engineering judgment of vendor supplied product specifications, and designed to rapidly remove from consideration those COTS products which on the face are not viable candidates for further consideration. The second pass is a more careful examination of the remaining COTS candidates, evaluating each product according to certain desirable attributes.

In this section, the data being requested is ***an aggregate*** of the total amount of effort expended doing COTS/NDI assessment during the system development.

**Initial Filtering Effort:**

5.1 Total number of candidate COTS products filtered:__________

5.2 Total effort spent doing initial filtering of all COTS candidates:__________(person-months)

5.3 Average filtering effort per COTS candidate:___________(person-hours *or* person-months)
*circle the correct units*

**Attribute Assessment Effort:**

5.4.Total number of COTS products assessed:__________

5.5.Total number of unique COTS products finally integrated:_________

5.6.Total effort spent on attribute assessment of all COTS candidates:_________(person-months)

5.7.Total number of *calendar months* spent assessing all candidate COTS products:__________

5.8.For each attribute listed in the table following, indicate the total amount of *effort expended* assessing COTS products during system development *in aggregate* in terms of the given attribute by checking the appropriate box according to the column definitions indicated below:

| | | |
|---|---|---|
| U | - "Unknown" | - don't know effort expended assessing this attribute. |
| EL | - "Extra Low" | - no effort expended. |
| VL | - "Very Low" | - less than or equal to one person-hour. |
| L | - "Low" | - more than one person-hour and less than or equal to one person-day. |
| N | - "Nominal" | - more than one person-day and less than or equal to one person-week. |
| H | - "High" | - more than one person-week and less than or equal to one person-month. |
| VH | - "Very High" | - more than one person-month and less than or equal to three person-months. |
| EH | - "Extra High" | - more than three person-months and less than or equal to N person-years. |

*(Please indicate your value of N in person-years: N = __________)*

Note: In the table on the following pages, the composite attributes we want you to rate are highlighted in the shaded rows. Underneath each composite attribute is at least one and sometimes several lower level attributes which are being aggregated into the composite attribute.

Place a check mark in the appropriate box of a given composite attribute to indicate how much effort was expended assessing any or all of the lower level attributes listed under that particular composite attribute.

Also, place a check mark in the first box to the right of the lower level attributes you actually considered when determining the effort expended assessing a given composite attribute.

For example, for “Correctness,” you may indicate that the assessment effort was Very High, and you considered both of the lower level attributes *accuracy* and *correctness* as defined in the glossary. Thus you would put a check in the VH box in the shaded row for “Correctness,” and put checks in both of the boxes immediately to the right of the two lower level attributes.

But moving on to “Availability/Robustness,” you may determine only a Nominal amount of effort was expended assessing this composite attribute, and in this case the only lower level attributes considered were *availability, fail safe, reliability,* and *safety*. So you would put check marks in the N box in the shaded row for “Availability/Robustness,” and check marks in the first box to the right of *availability, fail safe, reliability,* and *safety*.

Finally, space has been left at the end of the table for you to specify your own assessment attributes if you find that you have assessment effort that is not accounted for by the existing set of pre-defined attributes.

| Attributes\Effort | | U | EL | VL | L | N | H | VH | EH |
|---|---|---|---|---|---|---|---|---|---|
| **Correctness:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Accuracy | □ | | | | | | | | |
| Correctness | □ | | | | | | | | |
| **Availability/Robustness:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Availability | □ | | | | | | | | |
| Fail Safe | □ | | | | | | | | |
| Fail Soft | □ | | | | | | | | |
| Fault Tolerance | □ | | | | | | | | |
| Input Error Tolerance | □ | | | | | | | | |
| Redundancy | □ | | | | | | | | |
| Reliability | □ | | | | | | | | |
| Robustness | □ | | | | | | | | |
| Safety | □ | | | | | | | | |
| **Security:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Security (Access Related) | □ | | | | | | | | |
| Security (Sabotage Related) | □ | | | | | | | | |
| **Product Performance:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Execution Performance | □ | | | | | | | | |
| Information/Data Capacity | □ | | | | | | | | |
| Precision | □ | | | | | | | | |
| Memory Performance | □ | | | | | | | | |
| Response Time | □ | | | | | | | | |
| Throughput | □ | | | | | | | | |
| **Understandability:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Documentation Quality | □ | | | | | | | | |
| Simplicity | □ | | | | | | | | |
| Testability | □ | | | | | | | | |
| **Ease of Use:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Usability/Human Factors | □ | | | | | | | | |
| **Version Compatibility:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Downward Compatibility | □ | | | | | | | | |
| Upward Compatibility | □ | | | | | | | | |
| **Intercomponent Compatibility:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Compatibility with Other Components | □ | | | | | | | | |
| Interoperability | □ | | | | | | | | |
| **Flexibility:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Extendibility | □ | | | | | | | | |
| Flexibility | □ | | | | | | | | |
| **Installation/Upgrade Ease:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Installation Ease | □ | | | | | | | | |
| Upgrade/Refresh Ease | □ | | | | | | | | |

**(Table continued on following page.)**

| Attributes\Effort | | U | EL | VL | L | N | H | VH | EH |
|---|---|---|---|---|---|---|---|---|---|
| **Portability:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Portability | □ | | | | | | | | |
| **Functionality:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Functionality | □ | | | | | | | | |
| **Price:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Initial Purchase or Lease | □ | | | | | | | | |
| Recurring Costs | □ | | | | | | | | |
| **Maturity:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Product Maturity | □ | | | | | | | | |
| Vendor Maturity | □ | | | | | | | | |
| **Vendor Support:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Response Time for Critical Problems | □ | | | | | | | | |
| Support | □ | | | | | | | | |
| Warranty | □ | | | | | | | | |
| **Training:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| User Training | □ | | | | | | | | |
| **Vendor Concessions:** | | □ | □ | □ | □ | □ | □ | □ | □ |
| Willingness to Escrow Source Code | □ | | | | | | | | |
| Willingness to Make Modifications | □ | | | | | | | | |
| **Other (Please define)** | | | | | | | | | |
| 1) ______________________ | | □ | □ | □ | □ | □ | □ | □ | □ |
| 2) ______________________ | | □ | □ | □ | □ | □ | □ | □ | □ |
| 3) ______________________ | | □ | □ | □ | □ | □ | □ | □ | □ |
| 4) ______________________ | | □ | □ | □ | □ | □ | □ | □ | □ |
| 5) ______________________ | | □ | □ | □ | □ | □ | □ | □ | □ |
| 6) ______________________ | | □ | □ | □ | □ | □ | □ | □ | □ |

# VI. COTS Tailoring Data

COTS tailoring involves those normal activities required to prepare or initialize a COTS component for use in a specific system WITHOUT adapting or modifying the normal or available functionality of the COTS component.

(Adapting or modifying functionality, e.g., to account for mismatches between your business process and a COTS product's functionality, is typically done in the glue code and is handled by the COCOTS glue code sub-model through proper sizing of the glue code effort and rating of the glue code parameters. Large scale database population or conversion is also not considered "tailoring." This effort is captured in the COCOMO II model with the DATA parameter. However, specification of data definition templates and formats *is* considered tailoring if this is part of the normal activity needed to initialize a given COTS component.)

Major tailoring activities include parameter specification, script writing, I/O report and GUI screen specification, and set-up of security/user access protocols.

The COTS Tailoring sub-model presumes that aggregate project COTS tailoring activity can be characterized by an overall level of complexity, which in turn has implications for the overall effort expended on tailoring all the COTS components in a system.

**Tailoring Effort:**

6.1.Total number of COTS components in system *tailored*:__________

6.2.Total effort spent tailoring all COTS components in system:_________(person-months)

6.3.Total number of *calendar months* spent tailoring all COTS components:__________

**Tailoring Activity Complexity:**

Complexity of aggregate COTS tailoring activities is determined in the model using a subjective average of the individual complexity of five equally weighted factors (four tailoring activities plus one tailoring aid) presented in table VI.A on the following page. To determine aggregate tailoring complexity, first rate the five factors in table VI.A individually according to the criteria given in the table. ***(Again, keep in mind that you are doing a mental averaging of each factor as it was performed or applied across all COTS components in the system.)*** Next, sum the total point score as described in the table for the combination of ratings you selected. Then determine which gross category that score corresponds to on the rating scale provided on the page following the table. Finally, using your best engineering judgment, adjust your final rating for aggregate complexity above or below the center mark of the gross category as needed.

| | Individual Activity & Aid Complexity Ratings | | | | | |
|---|---|---|---|---|---|---|
| **Tailoring Activities & Aids** | **Very Low** (point value = 1) | **Low** (point value = 2) | **Nominal** (point value = 3) | **High** (point value = 4) | **Very High** (point value = 5) | **Corre-sponding Points** |
| **Parameter Specification** | Zero to 50 parms to be initialized. | 51 to 100 parms to be initialized. | 101 to 500 parms to be initialized. | 501 to 1000 parms to be initialized. | 1001 or more parms to be initialized. | ------- |
| **Script Writing** | Menu driven;<br>1 to 5 line scripts;<br>1 to 5 scripts needed. | Menu driven;<br>6 to 10 line scripts;<br>6 to 15 scripts needed. | Hand written;<br>11 to 25 line scripts;<br>16 to 30 scripts needed. | Hand written;<br>26 to 50 line scripts;<br>31 to 50 scripts needed. | Hand written;<br>51 or more line scripts;<br>51 or more scripts needed. | ------- |
| **I/O Report & GUI Screen Specification & Layout** | Automated or standard templates used;<br>1 to 5 reports/screens needed. | Automated or standard templates used;<br>6 to 15 reports/screens needed. | Automated or standard templates used;<br>16 to 25 reports/screens needed. | Hand written or custom designed;<br>26 to 50 reports/screens needed. | Hand written or custom designed;<br>51 or more reports/screens needed. | ------- |
| **Security/Access Protocol Initialization & Set-up** | 1 security level;<br>1 to 20 user profiles;<br>1 input screen/user. | 2 security levels<br>21 to 50 user profiles;<br>2 input screens/user. | 3 security levels<br>51 to 75 user profiles;<br>3 input screens/user. | 4 security levels<br>76 to 100 user profiles;<br>4 input screens/user. | 5 or more security levels<br>101 or more user profiles;<br>5 or more input screens/user. | ------- |
| **Availability of COTS Tailoring Tools** | Tools were highly useful. | Tools were very useful. | Tools were moderately useful. | Tools were somewhat useful. | No tools available. | ------- |

**Total Point Score = ___________**

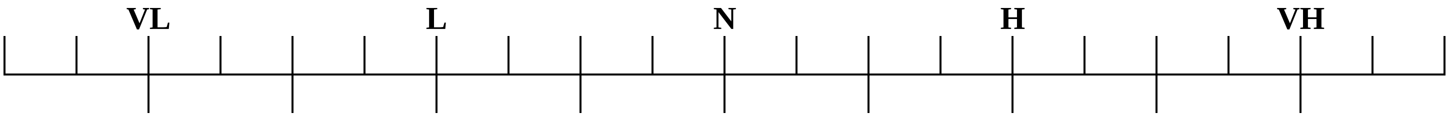
**Table VI.A - Aggregate Tailoring Activity Complexity**

***Example:*** *individual ratings of Low for Parameter Spec, Very Low for Scripts ,Very High for I/O Reports, Nominal for Security, and Very High for Tailoring Tools would result in a point total of 16, indicating a gross combined rating of Nominal. To recognize the existence of at least one individual rating of Very Low, however, it might be reasonable to circle the tic mark exactly halfway between the Low and Nominal categories on the scale below when assigning a final complexity rating for aggregate COTS tailoring activity. Note that the minimum point total possible is 5 and the maximum is 30.*

6.4 - Aggregate Tailoring Complexity.

Circle the appropriate tic mark based upon the criteria in the preceding table:

| **Very Low** | **Low** | **Nominal** | **High** | **Very High** |
|---|---|---|---|---|
| Point total is between 5 and 10. | Point total is between 11 and 15. | Point total is between 16 and 20. | Point total is between 21 and 25. | Point total is between 26 and 30. |



Explain rationale for your rating:

# VII. COTS Glue Code Data

The COTS Glue Code model refers to the total amount of COTS glue code developed for the system in aggregate, including glue code created for COTS products layered beneath other COTS products, as opposed to being integrated directly into the main system.

7.1 Total number of COTS components represented by glue code described in this section.

No. Components:___________

7.2 All functions provided by the COTS components counted in question 7.1.

Circle as needed:

Spreadsheet
Word Processing
Scheduling
Mathematical Utilities

Communications
User Display
Database
Signal Processing

Message Handling
CASE Environment
Diagnostics
Compiler

Other: ______________________________

7.3 Component Integration Nature

Indicate the percentage of COTS components counted in question 7.1 for which the integration activity is a:

New Component Integration:__________%

Component Upgrade/Refresh:__________%

7.4 Specific Glue Code Development Activity Duration

Record the number of calendar months needed to complete all the glue code from the time the first COTS component counted in question 7.1 was integrated to the last COTS component.

Months:___________

7.5 Specific Glue Code Development Activity Effort

Record the total effort expended in Person-Months needed to complete all the glue code from the time the first COTS component counted in question 7.1 was integrated to the last COTS component.

Person-Months:___________

* * * *

*We would like to collect integration or "glue" code sizing and breakage data in physical and logical lines of code as well as unadjusted function points. Please submit all size measures that are available.*

***Note: Glue code as defined for the purposes of this survey is composed of 1) code needed to facilitate data or information exchange between the COTS/NDI component and the system into which it is being integrated, 2) coded needed to connect or "hook" the COTS/NDI component into the system but does not necessarily enable data exchange, and 3) code needed to provide required functionality missing in the COTS/NDI component AND which depends upon or must interact with the COTS/NDI component.***

### 7.6 Total Delivered Lines of Component Glue Code

Record the total number of lines of glue code delivered for all COTS components.

Glue SLOC:_______________

### 7.7 Glue SLOC Count Type

Record the unit definition for the SLOC count reported in question 7.6.

Circle one:

Logical SLOC
Physical SLOC (carriage returns)
Physical SLOC (semicolons)
Non-commented/Non-blank SLOC
Other: _____________________________

### 7.8 Glue Code Programming Languages

Record all the various programming languages used in the development of the glue code and the percentage of the SLOC reported in question 7.6 representing each language.

| Language | Percent SLOC |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

7.9 Glue Code by Unadjusted Function Points

Record the total number of unadjusted function points attributable to data function types (*internal logical files*, *external interface files*) and transaction function types (*external inputs, external outputs, external inquiries*) counted for the glue code delivered with all COTS components.

Glue UFP:_______________

7.18 Glue Code Breakage

Record the percentage breakage in the glue code that occurred by project completion. "Breakage" is defined to be code that had to be discarded or reworked as a result of a change in system requirements OR the need to integrate a newer or upgraded version of a COTS product. It does NOT include code that had to be reworked as result of bugs found during testing or improper implementation of requirements.

Percentage Breakage (Glue SLOC):________

Percentage Breakage (Glue UFP):________

# VIII. Glue Code Development Cost Drivers

These drivers should be assessed while considering the total amount of COTS glue code developed for the system, as described in Section VII. That is, where the criteria given below refer to “a COTS component,” think in terms of how a given driver would most accurately be rated when considering all COTS components taken together. You’re doing a kind of mental averaging here across all COTS components integrated. ***The key here is to remember that you are trying to qualify the average or overall conditions that obtained when all the glue code was being developed***, whether or not that glue code was written to accommodate only one or many different COTS components.

Fourteen Cost Drivers have been defined for the COTS integration cost estimation model. You are asked to rate each driver according to a specific metric defined for each driver on a scale ranging from Very Low to Very High as the given metric applies to the circumstances of the component integration effort being reported. Descriptions of the concepts being captured by each driver have been provided to help you make your assessment. (Note that these descriptions are usually more encompassing than the specific metric by which you are asked to make your rating.) Also, a graduated scale has been provided to allow you to make incremental ratings between the five gross ratings. Record your answers by circling the tic marks on the scales, one mark per cost driver. Note that some of the cost drivers do not allow ratings at all levels (i.e., Very Low, Low, etc.). Finally, for each question in this section, the word “**UNKNOWN**” has been placed just above each rating scale. If for any driver you do not know with any reasonable confidence the appropriate rating and cannot determine that information, please circle “**UNKNOWN**” rather than indicating either nominal or some other rating on the scale.

## Circle the appropriate tic mark on the scales below:

## <u>Integration Personnel Drivers</u>

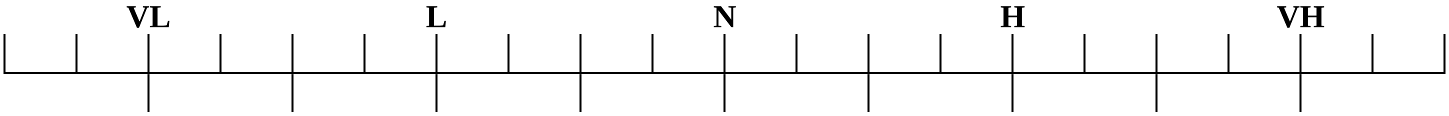
8.1 <u>ACIEP - COTS/NDI Integrator Experience with Product</u>

How much experience did/does the development staff have with running, integrating, and maintaining the COTS/NDI products?

*Metric: months/years of experience with product.*

**UNKNOWN**

| Very Low | Low | Nominal | High | Very High |
|---|---|---|---|---|
| Staff on average has no experience with the products. | Staff on average has less than 6 month's experience with the products. | Staff on average has between 6 month's and 1 year's experience with the products. | Staff on average has between 1 and 2 years' experience with the products. | Staff on average has more than 2 years' experience with the products. |



Explain rationale for your rating:

8.2 ACIPC - COTS/NDI Integrator Personnel Capability

What were/are the overall software development skills and abilities which your team as a whole on average brought/bring to the product integration tasks AS WELL AS experience with the specific tools, languages, platforms, and operating systems used/being used in the integration tasks?

*Metric: months/years of experience.*

**UNKNOWN**

| **Very Low** | **Low** | **Nominal** | **High** | **Very High** |
|---|---|---|---|---|
| Staff on average has no development experience or with the specific environmental items listed. | Staff on average has less than 6 month's development experience or with the specific environmental items listed. | Staff on average has between 6 month's and 1 year's development experience or with the specific environmental items listed. | Staff on average has between 1 and 2 years' development experience or with the specific environmental items listed.. | Staff on average has more than 2 years' development experience or with the specific environmental items listed.. |

**VL** **L** **N** **H** **VH**

Explain rationale for your rating:

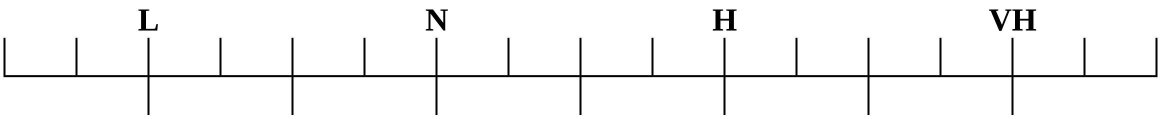
### 8.3 AXCIP - Integrator Experience with COTS/NDI Integration Processes

Does a formal and validated COTS/NDI integration process exist within your organization and how experienced was/is the development staff in that formal process?

*Metric: a mix of conditions including SEI CMM level, ISO 9001 certification, and number of times integration team as a whole on average has used the defined COTS/NDI integration process.*

**UNKNOWN**

| **Low** | **Nominal** | **High** | **Very High** |
|---|---|---|---|
| CMM level =1<br>OR<br>there is no formally defined COTS/NDI integration process. | [CMM level =2<br>OR<br>ISO 9001 certified]<br>AND<br>there is a formally defined COTS/NDI integration process<br>AND<br>the integration team has never used the process before. | [CMM level =3<br>OR<br>ISO 9001 certified]<br>AND<br>there is a formally defined COTS/NDI integration process<br>AND<br>the integration team has used the process 1 or 2 times before. | [CMM level > 3<br>OR<br>ISO 9001 certified]<br>AND<br>there is a formally defined COTS/NDI integration process<br>AND<br>the integration team has used the process 3 or more times before. |



Explain rationale for your rating:

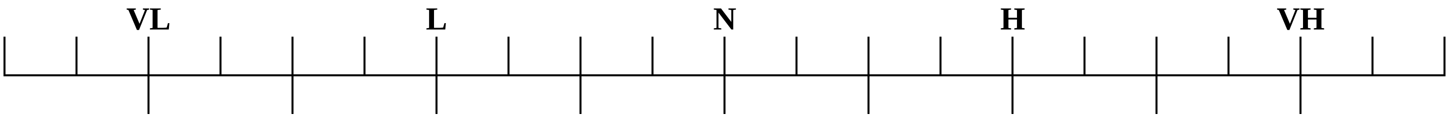
### 8.4 APCON - Integrator Personnel Continuity

How stable was/is your integration team? Are the same people staying around for the duration of the tasks, or must you keep bringing in new people and familiarizing them with the particulars of the project because experienced personnel leave?

*Metric: annual integration personnel turnover rate (a high personnel turnover rate implies a low personnel continuity).*

**UNKNOWN**

| Very Low | Low | Nominal | High | Very High |
|---|---|---|---|---|
| 48% or more per year. | Between 24% and 47% per year. | Between 12% and 23% per year. | Between 6% and 11% per year. | 5% or less per year. |



Explain rationale for your rating:

## COTS/NDI Component Drivers

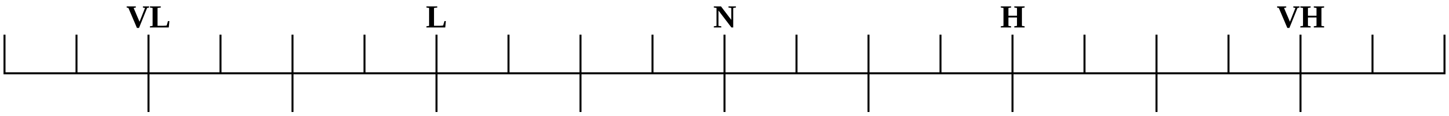
8.5 ACPMT - COTS/NDI Product Maturity

How many copies have been sold or used previously of the major versions (as opposed to release of those versions) of the COTS/NDI components you integrated or intend to integrate? How long have the versions been on the market or available for use? How large are the versions' market shares or installed user bases? How thoroughly have the versions been used by others in the manner you used or intend to use them?

*Metric: time on market (if COTS)/time available for use (if NDI).*

**UNKNOWN**

| Very Low | Low | Nominal | High | Very High |
|---|---|---|---|---|
| Versions in pre-release beta test. | Versions on market/available less than 6 months. | Versions on market/available between 6 months and 1 year. | Versions on market/available between 1 and 2 years. | Versions on market/available more than 2 years. |



Explain rationale for your rating:

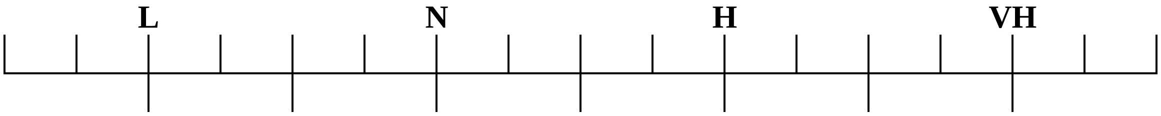
### 8.6 ACSEW - COTS/NDI Supplier Product Extension Willingness

How willing were/are the suppliers of the COTS/NDI products to modify the design of their software to meet your specific needs, either by adding or removing functionality or by changing the way it operates? In the case of COTS components, this refers to changes that would appear in market releases of the product. In the case of NDI components, this refers to changes that would appear in copies being distributed to all users of the component. This does NOT include specialty changes in the COTS/NDI component that would appear in your copy only.

*Metric: number and nature of changes supplier will make.*

**UNKNOWN**

| Low | Nominal | High | Very High |
|---|---|---|---|
| Suppliers will not make any changes. | Suppliers will make a few minor changes. | Suppliers will make one major change and several minor ones. | Suppliers will make two or more major changes along with any minor changes needed. |



Explain rationale for your rating:

## 8.7 APCPX - COTS/NDI Product Interface Complexity

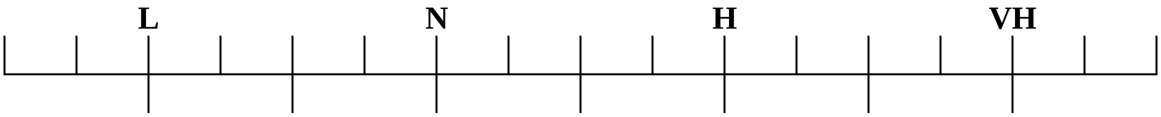
What are the nature of the interfaces between the COTS/NDI components and the glue code connecting them to the main application? Are there difficult synchronization issues? Must the interfaces balance conflicting criteria (e.g., security, safety, accuracy, ease of use, speed)?

*Metric: the scale for this driver uses a subjective average of the three equally weighted facets of interface complexity described in table VIII.A on the following page. To rate this driver, first rate the three items (interface conventions, control aspects, data) in table VIII.A individually according to the criteria given in the table. Next, sum the total point score as described in the table for the combination of ratings you selected, and determine which gross category that score corresponds to on the scale below. Finally, using your best engineering judgment, adjust your final rating for this driver above or below the center mark of the gross category as needed.*

*Example: individual ratings of Low for Interface , Low for Control , and Very High for Data would result in a point total of 9, indicating a gross combined rating of Nominal. To recognize the existence of at least one individual rating of Very High, however, it might be reasonable to circle the tic mark immediately to the right of the center tic mark in the Nominal category on the scale below when assigning a final rating for this driver.*

**UNKNOWN**

| Low | Nominal | High | Very High |
|---|---|---|---|
| Point total is between 5 and 7. | Point total is between 8 and 10. | Point total is between 11 and 13. | Point total is between 14 and 15. |



Explain rationale for your rating:

| **Complexity Elements** | **Very Low** (point value = 1) | **Low** (point value = 2) | **Nominal** (point value = 3) | **High** (point value = 4) | **Very High** (point value = 5) | **Corre-sponding Points** |
|---|---|---|---|---|---|---|
| **Interface Conventions** (e.g., naming, relevant usage scenarios, service signature, service order) | **N/A** | Nearly all API conventions are clear and consistent. | Most API conventions are clear and consistent. | Few API conventions are clear and consistent. | API conventions are non-existent. | ------- |
| **Control Aspects** (e.g., consistent and clear error handling/recovery) | **N/A** | Nearly all control aspects are well defined and consistently applied. | Most control aspects are well defined and consistently applied. | Few control aspects are well defined and consistently applied. | No control aspects are well defined and consistently applied. | ------- |
| **Data** (e.g., conversion, number/range typing) | No data conversion required. | Little data conversion required and standard data types used. | Some data conversion required and standard data types used. | Significant data conversion required and/or use of non-standard data types. | Extensive data conversion required and/or use of non-standard data types. | ------- |

**Total Point Score =** __________

**Table VIII.A - Facets of Complexity**

Use this table in evaluating complexity drivers 8.7 (APCPX) and 8.11 (AACPX). Use it once for APCPX, then repeat its use for AACPX. Rate each complexity element described in the table individually, recording the point value associated with your rating in the far right column. Then sum all three point values to arrive at a total point score (minimum score possible is 5, maximum score is 15). Then apply that total point score to the scales provided for each of the two cost drivers as indicated under their descriptions.

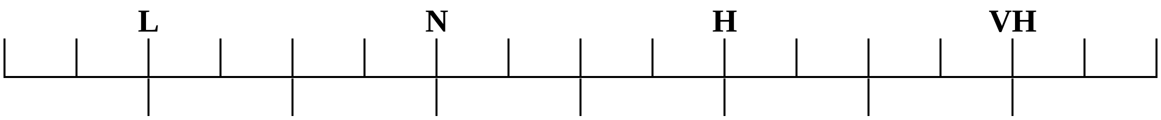
8.8 ACPPS - COTS/NDI Supplier Product Support

What is the nature of the technical support for the COTS/NDI components that was/is available AND PROCURED for the integration team during the development, either directly from the component suppliers or through third parties?

*Metric: the level of support available and procured.*

**UNKNOWN**

| Low | Nominal | High | Very High |
|---|---|---|---|
| Products are unsupported. | Help desk support. | Trained technical support. | Formal consulting help. |



Explain rationale for your rating:

## 8.9 ACPTD - COTS/NDI Supplier Provided Training and Documentation

How much training and/or documentation for the COTS/NDI components was/is available AND PROCURED for the integration team during the development, either directly from the component suppliers or through third parties?

*Metric: the amount of training and/or documentation available and procured.*

**UNKNOWN**

| Very Low | Low | Nominal | High | Very High |
|---|---|---|---|---|
| No training and very little documentation procured. | Roughly ¼ of the needed training and/or documentation procured. | Roughly ½ of the needed training and documentation procured. | Roughly ¾ of the needed training and/or documentation procured. | As much training and/or documentation procured as needed. |

**VL L N H VH**

Explain rationale for your rating:

## APPLICATION/SYSTEM Drivers

8.10 ACREL - Constraints on System/Subsystem Reliability

How severe are the overall reliability constraints on the system or subsystem into which the COTS/NDI components was/is being integrated? What are the potential consequences if the components fail to perform as required in any given time frame? (Note that *availability* is considered an issue different than reliability and is NOT addressed in this cost driver.)

*Metric: the potential threat if the component fails to perform as expected.*

**UNKNOWN**

| Low | Nominal | High | Very High |
|---|---|---|---|
| Threat is low; if a failure occurs losses are easily recoverable (e.g., document publishing). | Threat is moderate; if a failure occurs losses are fairly easily recoverable (e.g., support systems). | Threat is high; if a failure occurs the risk is to mission critical requirements. | Threat is very high; if a failure occurs the risk is to safety critical requirements. |

**L** **N** **H** **VH**

Explain rationale for your rating:

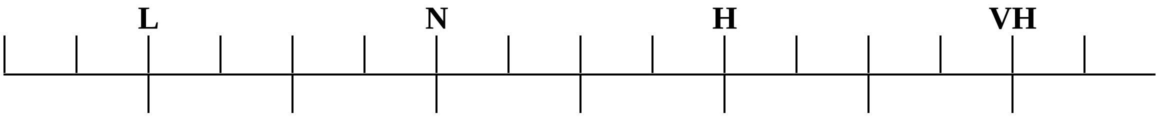
## 8.11 AACPX - Application Interface Complexity

What are the nature of the interfaces between the main application system or subsystem and the glue code used to connect the system to the COTS/NDI components? Are there difficult synchronization issues? Must the interface balance conflicting criteria (e.g., security, safety, accuracy, ease of use, speed)?

*Metric: the same subjective averaging of the items in table VIII.A as used for the driver APCPX. See the explanation provided for rating that driver under item 8.7, and then repeat the use of table VIII.A to evaluate this current driver.*

**UNKNOWN**

| **Low** | **Nominal** | **High** | **Very High** |
|---|---|---|---|
| Point total is between 5 and 7. | Point total is between 8 and 10. | Point total is between 11 and 13. | Point total is between 14 and 15. |



Explain rationale for your rating:

| **Complexity Elements** | **Very Low** (point value = 1) | **Low** (point value = 2) | **Nominal** (point value = 3) | **High** (point value = 4) | **Very High** (point value = 5) | **Corresponding Points** |
|---|---|---|---|---|---|---|
| **Interface Conventions** (e.g., naming, relevant usage scenarios, service signature, service order) | **N/A** | Nearly all API conventions are clear and consistent. | Most API conventions are clear and consistent. | Few API conventions are clear and consistent. | API conventions are non-existent. | ------- |
| **Control Aspects** (e.g., consistent and clear error handling/recovery) | **N/A** | Nearly all control aspects are well defined and consistently applied. | Most control aspects are well defined and consistently applied. | Few control aspects are well defined and consistently applied. | No control aspects are well defined and consistently applied. | ------- |
| **Data** (e.g., conversion, number/range typing) | No data conversion required. | Little data conversion required and standard data types used. | Some data conversion required and standard data types used. | Significant data conversion required and/or use of non-standard data types. | Extensive data conversion required and/or use of non-standard data types. | ------- |

**Total Point Score =** __________

**Repeat of Table VIII.A for 8.11 calculation**

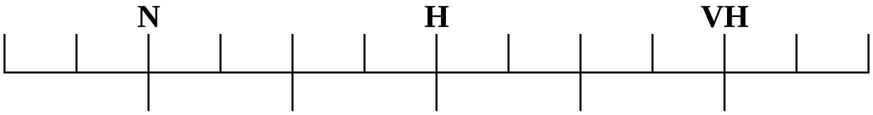
8.12 ACPER - Constraints on System/subsystem Technical Performance

How severe were/are the technical performance constraints (e.g., storage, memory, reserve, flow through capacity, etc.) on the application system or subsystem that the COTS/NDI components needed to/must meet?

*Metric: the presence or absence of constraints.*

**UNKNOWN**

| Nominal | High | Very High |
|---|---|---|
| There are no technical constraints or real time processing needs. | Real time processing must be performed<br>OR<br>other technical constraints exist. | Real time processing must be performed<br>AND<br>other technical constraints exist. |



Explain rationale for your rating:

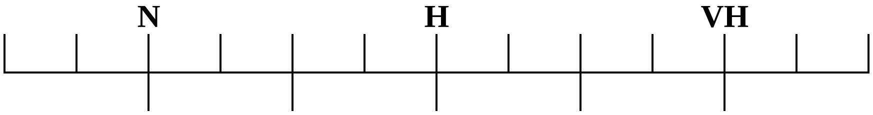
8.13 ASPRT - System Portability

What were/are the overall system or subsystem portability requirements that the COTS/NDI component needed to/must meet?

*Metric: the nature of portability requirements.*

**UNKNOWN**

| Nominal | High | Very High |
|---|---|---|
| There are no portability requirements at the system/subsystem level. | System must be portable across platforms within the same family (e.g., across different versions of UNIX). | System must be portable across divergent platforms (e.g., from UNIX to VMS). |



Explain rationale for your rating:

## 8.14 AAREN - Application Architectural Engineering

How adequate/sophisticated were the techniques used to define and validate the overall systems architecture?

*Metric: architecture validation techniques.*

**UNKNOWN**

| Very Low | Low | Nominal | High | Very High |
|---|---|---|---|---|
| No architecture validation done. | Paper analysis performed. | Peer reviews of architectural design (including interface definitions). | Prototyping/demos of the architecture performed. | Simulations of the architecture created. |

**VL** **L** **N** **H** **VH**

Explain rationale for your rating:

# IX. Application Effort Due to COTS Volatility Data

The Application Effort Due to COTS Volatility Model is defined in both approximate and detailed versions. The Approximate model determines added effort as simply a percentage of overall new application effort based on the percentage of code breakage due to COTS volatility as determined in question 7.18. The Detailed model adds the refinement of including the COCOMO II non-linear scale factors. *(When used to perform an estimate, rather than ask for the actual new application effort, both models will assume an original Application development effort has been estimated using COCOMO II.)*

9.1 New application coding effort ***excluding*** effort due to COTS integration, i.e., exclusive of glue code and tailoring:

New application code effort:__________________(person-months)

9.2 Percentage of new application coding effort due to breakage ***excluding*** breakage related to COTS integration, i.e., as a result of requirements change, but *not* due to design or programming error *(this is the same breakage term as defined for COCOMO II):*

Application code breakage due to requirements change:_____________%

9.3 Percentage of new application coding effort due to breakage ***as a result of COTS product volatility,*** i.e., as a result of releases by the vendor of updated versions of COTS components.

Application code breakage due to COTS volatility:_____________%

9.4 COCOMO II Scale Factor Ratings:

Circle the appropriate box for each factor, one box per row:

| Scale Factor | Very Low | Low | Nominal | High | Very High | Extra High |
|---|---|---|---|---|---|---|
| **Precedentedness** | thoroughly unprecedented | Largely unprecedented | somewhat unprecedented | generally familiar | largely familiar | thoroughly familiar |
| **Development Flexibility** | rigorous | Occasional Relaxation | some relaxation | general conformity | some conformity | general goals |
| **Architecture/Risk Resolution** | little (20%) | some (40%) | often (60%) | generally (75%) | mostly (90%) | full (100%) |
| **Team Cohesion** | some difficult interactions | Basically cooperative interactions | largely cooperative | highly cooperative | seamless interactions | N/A |
| **Process Maturity** | Chaos | CMM Level 1 | CMM Level 2 | CMM Level 3 | CMM Level 4 | CMM Level 5 |

* percentage of module interfaces specified, percentage of significant risks eliminated.

# X. Free Form Comments

Please write down any comments or descriptions of items you feel are important to COTS integration but that were missing from this survey:

**This concludes this survey. Thank you for your efforts. Your cooperation is greatly appreciated and goes a long way to ensuring the usefulness of the COCOTS software integration cost estimation tool.**