# COCOMO and SCORM: Cost Estimation Model for Web-Based Training


| Mike Garnsey | Lacey Edwards | Kelly Ward | Dean Marvin |
|---|---|---|---|
| Sparta, Inc. | Sparta, Inc. | General Dynamics | Joint ADL Co-Lab |
| Orlando, Florida | Orlando, Florida | Orlando, Florida | Orlando, Florida |
| Mike.Garnsey@sparta.com | Lacey.Edwards@sparta .com | Kelly.Ward@gdit.com | Dean.Marvin@jointadlcolab.org |



## ABSTRACT

The Sharable Content Object Reference Model (SCORM) provides a framework that enables standardized delivery and reuse of content for web-based training courses. In March 2002, the Office of the Secretary of Defense issued the Training Transformation Strategic Plan, which emphasized the need for Advanced Distributed Learning (ADL) and implementation standards like SCORM. In the next five years the US government will convert hundreds of classroom, correspondence, and computer-based courses to SCORM-conformant courseware. Members of the training community currently use rules of thumb to estimate the cost of creating courseware, but each company prices courseware differently. The courseware development community needs an objectively validated algorithm that can be used to estimate project costs independent of specific company processes.

The Constructive SCORM Cost Model (COSCOMO) algorithm applies the concepts behind the Constructive Cost Model (COCOMO) software cost estimation algorithm to SCORM-conformant courseware development projects. This unique new algorithm addresses cost factors such as the size of the project, experience of the team, use of development tools, product characteristics, and schedule constraints. The algorithm determines both person-months of effort and suggested duration of schedule for the project. A team of courseware developers and engineers is validating the algorithm using surveys of the courseware community and historical project data.

This paper describes the COSCOMO algorithm in detail and explains how government agencies and web-based training developers can employ the algorithm to reduce risk in developing SCORM-conformant courseware.



## ABOUT THE AUTHORS

**Mike Garnsey** is a senior principal investigator with Sparta, Inc. He is the contractor project manager and technical lead for the COCOMO-SCORM project for the Joint Advanced Distributed Learning Co-Lab. Mr. Garnsey has 15 years of experience in the DoD simulation and training field, has a B.S. in Computer Engineering from the University of South Florida and a M.S. in Simulation Systems from the University of Central Florida.

**Kelly Ward** is Director of Technology Modernization with General Dynamics Information Technology. She coordinates conversion of courseware to shareable content objects (SCOs) and application of SCORM standards for current and new customers. She has 22 years of experience in the management and production of WBT and CBT as well as analysis and design of infrastructures to support technology-based training.

**LtCol Dean Marvin, USMC (Ret)** is a Senior Military Analyst with the Joint Advanced Distributed Learning Co-Lab in Orlando, FL. He helps lead the Research and Development efforts at the Co-Lab. LtCol Marvin received a BA from Florida Southern College and an MAE from Chapman University.

**Lacey Edwards** is a Systems Engineer with Sparta, Inc. investigating new combat system infrastructures and cost management algorithms. She earned her B.S. in Computer Engineering from the University of Florida and her M.S. in Management Science and Engineering from Stanford University.

# COCOMO and SCORM: Cost Estimation Model for Web-Based Training


**Mike Garnsey**
**Sparta, Inc.**
**Orlando, Florida**
**Mike.Garnsey@sparta.com**

**Lacey Edwards**
**Sparta, Inc.**
**Orlando, Florida**
**Lacey.Edwards@sparta .com**

**Kelly Ward**
**General Dynamics**
**Orlando, Florida**
**Kelly.Ward@gdit.com**

**Dean Marvin**
**Joint ADL Co-Lab**
**Orlando, Florida**
**Dean.Marvin@jointadlcolab.org**


## INTRODUCTION

In March 2002, the Office of the Secretary of Defense issued the Training Transformation Strategic Plan, which emphasized the need for Advanced Distributed Learning (ADL) and implementation standards like the Sharable Content Object Reference Model (SCORM). SCORM provides a framework that enables standardized delivery and reuse of content for web-based training courses. Since 2002, SCORM has evolved to its present SCORM 2004 iteration and has been incorporated as an industry standard throughout the U.S. and most of the world. Although we are just beginning to realize the potential savings from SCORM, its usefulness, acceptance, and broad application will continue to be a major part of the ADL environment.

The Department of Defense Instruction (DoDI) 1322.26, entitled "Development, Management, and Delivery of Distributed Learning," was signed in 2006. The DoDI 1322.26 implements policies, assigns responsibilities, prescribes procedures, and establishes information requirements to develop manage and deliver distributed learning for DoD personnel. The Instruction mandates that content shall be SCORM-conformant (when Learning Management System (LMS) functionality is necessary). The Instruction also mandates that acquired or developed SCORM-conformant content packages include metadata, be registered in the ADL-Registry, and be maintained in DoD Components' repositories that are searchable and accessible. The DoDI 1322.26 will have a major impact on all organizations that design, develop, and/or use SCORM-conformant courseware.

Although SCORM-conformant content has application to such a wide audience, and will be mandated by this new policy, there is currently no widely accepted tool for estimating the costs of developing SCORM-conformant courseware. Members of the training community currently use industry averages to provide targets for the cost of creating courseware, but each organization prices courseware differently. The courseware development community needs an objectively validated algorithm that can be used to estimate project costs independent of specific company processes.

This paper will describe the prototype Constructive SCORM Cost Model (COSCOMO) algorithm targeted at filling this need, and will explain how government agencies and web-based training developers can accurately estimate the costs associated with developing SCORM-conformant courseware.

### COSCOMO Background

COSCOMO is a derivative of the popular software cost-estimating model COCOMO II (version 2 of the Constructive Cost Estimating Model) that provides a cost estimating capability for SCORM-conformant courseware. As Figure 1 illustrates, the main ingredients of COSCOMO are (1) the COCOMO II algorithm, (2) the ADL SCORM courseware domain, and (3) the Analysis, Design, Development, Implementation, and Evaluation (ADDIE) process.

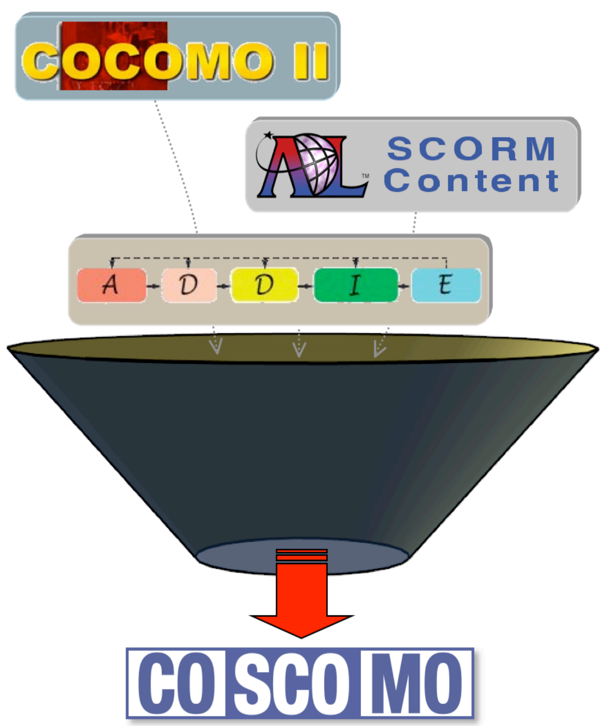


**Figure 1. COSCOMO Concept**

**COCOMO II**

The Constructive Cost Model (COCOMO) for estimating the cost of developing software was first published in 1981 by Barry Boehm. Its reliability in estimating software projects led to its wide adoption in the early 1980's in the software development community. In the 1990's, challenges such as reuse-driven approaches involving commercial-off-the-shelf (COTS) packages, applications composition, object-oriented approaches supported by distributed middleware, and software process maturity effects prompted continued research and resultant creation of COCOMO II in 1997 (Boehm).

The COCOMO II model not only continues to be widely adopted in the software development community, but has also been a popular basis for application in other domains. As of 2005, there were eleven COCOMO II model derivatives published. Note that the usual naming convention for models derived from the COCOMO II model is of the form "CO" + <derivative model identifier> + "MO." For example, the Constructive Systems Engineering Cost Model is named COSYSMO, and the Constructive Productivity-Improvement Model is name COPROMO. In keeping with this naming convention, the model name chosen for the COCOMO-SCORM content complexity measurement model is COSCOMO (Constructive SCORM Cost Model).

COCOMO II model derivatives are classified as (1) software cost models, (2) software extensions, and (3) other independent models (Boehm). New models can be created by:

- Modifying existing variables
- Adding or removing variables
- Post-processing the results of a COCOMO model
- Decomposing a problem and applying multiple models

The COSCOMO model is considered a software extension since it is a modification of the COCOMO II algorithm.

**SCORM**

SCORM is a collection of specifications adapted from multiple sources to provide a comprehensive suite of e-learning capabilities that enable interoperability, reuse, and sharable Web-based content. It is intended to provide a means of expressing instructional designs in a manner that will allow them to be executed by an LMS over the Web.

SCORM:

- Is focused on self-paced learning experiences
- Provides an intentional strategy for execution and reuse, e.g., sequence, searchability (metadata), organization/structure, etc.
- Provides a standard interface to track the student's progress, assessment/certification, completion, etc.
- Separates the design logic from the content making it easy to repurpose the content for different types of learners, different types of delivery media (books, HTML, animated slides, etc.), different contexts, and different times.

**ISD/ADDIE**

As a discipline, instructional systems design (ISD) is based upon a set of values, specialized knowledge, and intellectual skills and methodologies that pinpoint performance problems and provide a means of solving them through training and other human performance solutions. The ISD process involves a systematic approach to solving instructional or human performance problems. ISD Models are visualized representations of an instructional design process, showing the main elements or phases, and their relationships. The most basic ISD approach involves the five fundamental phases of analysis, design, development, implementation, and evaluation (ADDIE) as illustrated in Figure 2.

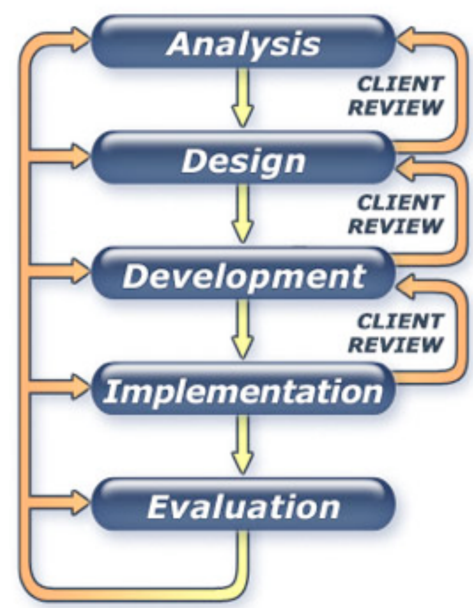


**Figure 2. ADDIE Instructional System Design Model**

This ADDIE model was developed in 1975 at Florida State University, and its five stages of the ISD process are evident in over 40 instructional design models that attempt to address the many factors involved in providing solutions to instructional problems in various contexts. ADDIE is a general purpose ISD model that is most useful for creating instructional products (Kemp), and has been adopted by the U.S. Armed Services as a valid ISD approach for training development. ADDIE has also become the framework for the DoD Instructional Systems Design and Systems Approach to Training and Education (DoD 2002).

The ADDIE model begins with the **Analysis** phase, during which designers establish a thorough understanding of all variables, impacts, needs, and facts associated with the job, task, skill, or performance requirement. After analysis, the **Design** specifications are determined; strategies, techniques, and media that will be utilized to meet performance goal(s) are identified. Once design is completed, all of the elements that were identified during the analysis phase are integrated into the solution for **Development**. During **Implementation**, the final courseware and/or support materials are delivered to the customer. Installation, integration, software and hardware testing of the courseware are performed. Finally, two types of **Evaluation** are performed – formative and summative. Formative evaluation occurs throughout the model and phases are repeated whenever conditions or variables change. As seen in Figure 2, formative evaluation during the Analysis phase includes customer and peer reviews. If discrepancies or new information are identified, the products will be modified before continuing. If new information about the job is discovered during the development phase, the phases are repeated for that new information and all products are reviewed in light of the new information. To ensure that the solution is valid and to assess training effectiveness, summative evaluation is performed. This consists of varying degrees of evaluation that may be based upon Kirkpatrick's levels of evaluation.

Government organizations and contractors often share the responsibility for completing the tasks and deliverables included in each phase of the ADDIE Model. For example, government personnel or independent specialists developing a Statement of Objectives (SOO) or Statement of Work (SOW) may conduct a preliminary training needs analysis, part of the ADDIE analysis phase. When this occurs, the contractor's entry point into the ADDIE Model and decision points within each phase change to reflect this new scope and schedule. This necessary flexibility became an added requirement for the COSCOMO model.

## METHODOLOGY

### COCOMO Methodology

COCOMO II and related models use the seven-step modeling methodology (Boehm) as shown in Figure 3.

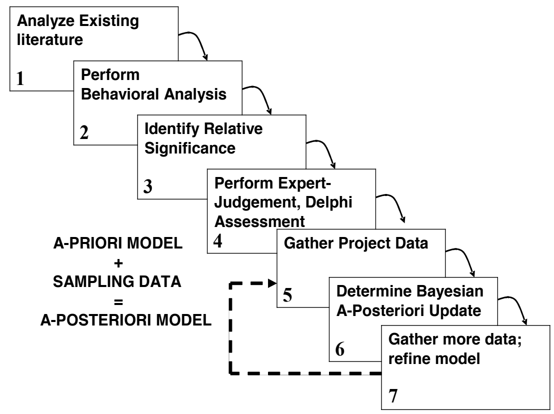


**Figure 3. COCOMO II Modeling Methodology**

### COCOMO-to-COSCOMO Methodology

The development of the COSCOMO model followed this COCOMO II modeling methodology wherever possible.

### COCOMO Variables

The COCOMO II variable set and the model's equation were analyzed to outline the process to create the COSCOMO model. A literature review and some behavioral analysis (mostly based on past experience of the team) were also conducted to support the COCOMO variable determinations. Each portion of the COCOMO equation was examined; the relationship between the variables and multipliers were identified and mapped. This analysis identified the sequence of steps that needed to be followed to correctly adjust the model for ISD/SCORM.

### ISD Cost Variables

Once the COCOMO II variables were fully analyzed, each of the variables was reviewed for relevance to ISD/SCORM projects. Most elements were directly applicable to the development of sharable content objects. A few elements required a modified definition to adapt to Interactive Multimedia Instruction (IMI) development. Finally, several elements were replaced to address remaining ISD concerns.

Each COSCOMO element is defined from the perspective of an IMI application. These definitions have been included in the prototype COSCOMO tool to enable the ISD to better utilize and understand each element. For example, Size determination was redefined in the COSCOMO model to reflect hours of courseware versus lines of code in the COCOMO II model. To accurately build this variable, users must break down the hours of courseware into the four levels of courseware interactivity as identified in Table 1 (MIL-HDBK-29612-3A).

**Table 1. Levels of Courseware Interactivity**

| Level | Description |
|---|---|
| Level 1 - Passive | The student acts solely as a receiver of information |
| Level 2 - Limited participation | The student makes simple responses to instructional cues |
| Level 3 - Complex participation | The student makes a variety of responses using varied techniques in response to instructional cues |
| Level 4 - Real-time participation | The student is directly involved in a life-like set of complex cues and responses |

This calculates the initial size factor. The user then provides an estimate of reused courseware based upon the level of rework required to integrate existing content into SCORM-conformant IMI.

**Delphi**
The COCOMO II modeling methodology often involves several iterations of the Delphi assessment process to attempt to capture what the experts believe has an influence on development effort. For the COSCOMO model, our group of domain experts and courseware project managers reached consensus after three iterations of the Delphi assessment process.

**Validation**
Next, historical project data was collected to validate the cost estimating relationships in the COSCOMO model. Finally, the data collected from both methods was analyzed to set the COSCOMO model option values. This was done using Bayesian statistical techniques that provide the ability to balance expert data and historical data.

The Bayesian approach makes use of prior information that is not part of the historical data by providing an optimal combination of the expert data and historical data (Boehm). Essentially, if the precision of the expert data is bigger than the precision of the historical data, a stronger weight is assigned to the expert data resulting in the combined mean to be closer to the expert data mean than the historical data mean.

## THE COSCOMO ALGORITHM

Figure 4 illustrates the operational concept of the COSCOMO model. The current form of the model is given in Equation 1. Note that this model retains the fundamental form of the original COCOMO II model.

$$PM = A * (Size)^{E} * \prod_{i=1}^{15} EM_i$$

$$\text{where} \qquad (1)$$

$$E = B + 0.01 * \sum_{j=1}^{5} SF_j$$

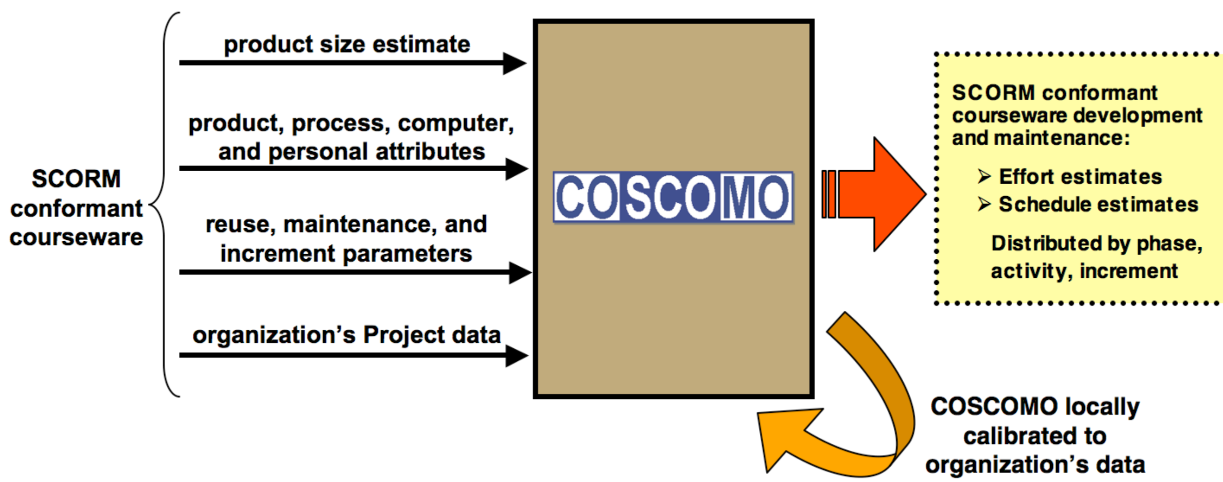


**Figure 4. COSCOMO Operational Concept**

Table 2 describes each of the variables for this equation.

**Table 2. COSCOMO Variables**

| Variable | Description |
|---|---|
| $SF_j$ | scale factor for the $j^{th}$ scale driver. Provides project-specific adjustments to the size of the project |
| E | diseconomies of scale driven by five scale factors |
| $EM_i$ | effort multiplier for the $i^{th}$ cost driver in the equations |
| PM | effort in Person-Months |
| A | calibration constant derived from historical project data |
| Size | adjusted number of courseware hours |
| B | calibration constant derived from historical project data. Can be customized based on a specific organization's historical project data |

The five scale factors are personnel and project related factors to account for the relative economies or diseconomies of scale encountered for projects of different sizes. They are as follows:

Development Flexibility (FLEX): How flexible is this project with respect to changes in requirements, interfaces, schedules, etc?

Process Maturity (PMAT): The organizations' maturity rating organized around the Software Engineering Institute's (SEI) Capability Maturity Model (CMM). There are two ways of rating Process Maturity. The first captures the result of an organized evaluation based on the CMM. The second is organized around the 18 Key Process Areas in the SEI CMM.

Precedentedness (PREC): How unique or common is this course to your organization? (If a project is unprecedented or not similar to other previously developed projects within an organization, then its precedentedness is low.)

Architecture / Risk Resolution (RESL): The degree of design thoroughness and risk elimination.

Team Cohesion (TEAM): This scale factor accounts for the sources of project turbulence and entropy due to difficulties in synchronizing the project's stakeholders: users, customers, developers, maintainers, others.

If $E < 1.0$, the project exhibits economies of scale, meaning that if the product's size is doubled, the project effort is less than doubled. Conversely, if $E > 1.0$, the project exhibits diseconomies of scale, meaning that the project's productivity decreases as the product size increases. The later is generally attributed to two main factors: growth of interpersonal communications overhead and growth of large-system integration overhead (Boehm).

Figure 5 provides two illustrations of how these five scale factors contribute to COSCOMO cost estimates.

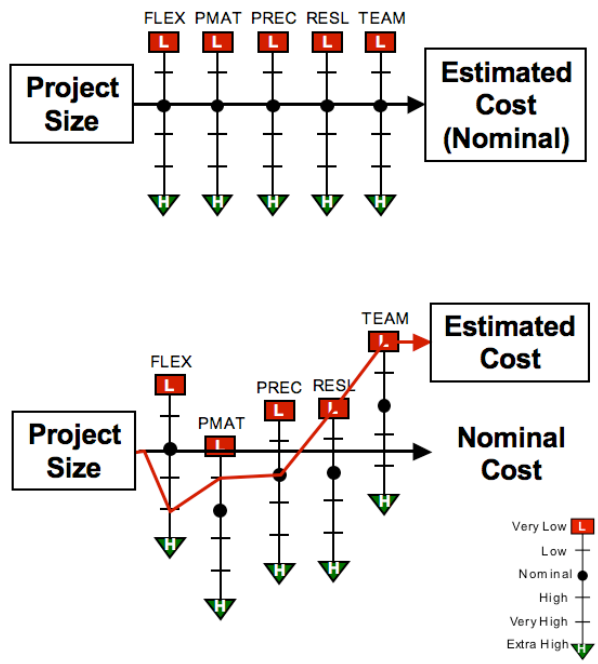


**Figure 5. Impacts of Scale Factors on COSCOMO Cost Estimates**

The upper graph of Figure 5 illustrates that when the input values for the five scale factors are at their default 'Nominal' levels, the scale factors have no impact on the estimated cost value that the COSCOMO model generates.

Of more interest is when the input values of the five scale factors are at levels other than 'Nominal', as depicted in the lower graph of Figure 5. This lower graph reflects the scale factor values for a hypothetical project with 'Very High' development flexibility, 'Low' Process Maturity, 'Nominal' precedentedness, 'Very Low' architecture/ risk resolution, and 'Very Low' team cohesion. As illustrated in this lower graph, higher scale factor value levels reduce COSCOMO cost estimates, and conversely, lower scale factor value levels increase COSCOMO cost estimates.

In COSCOMO, the effort multipliers perform the same function as in the original COCOMO II model. They are qualitative measures that estimate variations in level of effort that are associated with the product, platform, personnel, and project. Table 3 organizes by category and describes the resulting fifteen COSCOMO effort multipliers.

Of the seventeen original COCOMO II multipliers, five (APEX, PLEX, DTEX, SITE, SCED) are reused as-is, three have been removed (Execution Time Constraints (TIME), Main Storage Constraints (STORE), and Database Size (DATA)), one has been added (BAND), and the rest have been tailored in the COSCOMO model.

The COCOMO II model estimates the amount of effort required to design, develop, test, and integrate a software product. According to Boehm, et. al. (Boehm), the effort estimate from COCOMO II applies to the elaboration and construction phases of the rational unified process (RUP).

**Table 3. COSCOMO Effort Multipliers**

| | |
|---|---|
| **Product Multipliers** | |
| Required Reliability (RELY) | How reliable the product needs to be, and how much extra effort goes into the development process to ensure that level of reliability |
| Product Complexity (CPLX) | How complex the product is |
| Development for Reusability (RUSE) | How widely the product and its components will be shared in this or other projects, possibly throughout several services |
| Required Documentation (DOCU) | How much documentation is required compared to the amount of documentation that is necessary to support the product during its lifetime |
| **Platform Multipliers** | |
| Platform Volatility (PVOL) | How often the deployment platform changes |
| Bandwidth Restrictions (BAND) | The speed of the network across which the web-based training will be delivered |
| **Personnel Multipliers** | |
| Senior ISD, Human Performance Team Capability (SCAP) | The relative capability of the senior analysts, senior instructional system designers, and human performance factors team as compared to others in the industry |
| ISD, Development Team Capability (DCAP) | The relative capability of the instructional system designers, programmers, and other development team members as compared to others in the industry |
| Personnel Continuity (PCON) | How often the composition of the project team changes, measured in annual turnover |
| Courseware Applications Experience (APEX) | Weighted average of the experience of the team in developing courseware applications |
| Platform Experience (PLEX) | Weighted average of the experience of the team in developing applications for the deployment platform |
| Development Tools Experience (DTEX) | Weighted average of the experience of the team in using the development tools chosen for the project |
| **Project Multipliers** | |
| Availability of Lifecycle Tools (LIFE) | Level of development and lifecycle support tools available to the project team |
| Multi-site Development (SITE) | How distributed the project team is, and how they communicate |
| Schedule Expansion (SCED) | The relative length of the schedule for the project as compared to the schedule for typical projects of the same effort |

COSCOMO estimates the effort involved in developing a courseware product, therefore its effort estimate applies to only those phases of courseware development that correspond to the Elaboration and Construction phases of the RUP. Based on our comparison of the ADDIE process to the RUP, we determined that the Elaboration and Construction phases collectively correspond to the Design, Development, Implementation, and Evaluation phases of the ADDIE process.

Since COSCOMO produces an estimate of effort for only four of the five ADDIE phases, the ADDIE Phase Section of the COSCOMO tool provides a mechanism for extrapolating the effort needed for the remaining Analysis phase based on an estimate of the relative effort required for each of the five phases. In addition, this portion of the tool displays the estimated effort required to complete each of the other four phases. The COSCOMO tool provides a dynamic graphical display of the relative and absolute distributions of effort across the five ADDIE phases. This visualization allows the user to view the effects of various phase distributions, and to quickly determine where cost-cutting efforts might have the most significant effects, for example.

## PRACTICAL APPLICATION OF COSCOMO

While there may be many unknowns during the estimation phase, such as delivery medium, level of interactivity, or level of instruction, the COSCOMO user will be able to explore "what-if" scenarios for cost comparisons. Input data used during the estimation process may also be captured for future reference and used in the government's request for proposal (RFP) development process.

### Government

Government agencies responsible for either managing or developing SCORM-conformant courseware, will find the potential of a SCORM cost-estimation tool beneficial. A large training organization, such as the Army's Training & Doctrine Command (TRADOC), would use this tool for estimating development costs for new courseware, updating legacy courseware, or re-purposing current content. This would prove invaluable whether they are focusing on the development of a single course or preparing a large contract for developing multiple courses. The COSCOMO tool could be used for providing a basis of estimate for contracting courseware development efforts, developing annual budgets, or for prioritizing courses to be developed.

As an example of how this tool can be utilized to establish contractor requirements, Figure 6 identifies those elements that are key cost drivers. By modifying or amplifying requirements to explicitly request the correct amount of documentation necessary to support life-cycle maintenance, acquisition experts should receive more refined estimates.

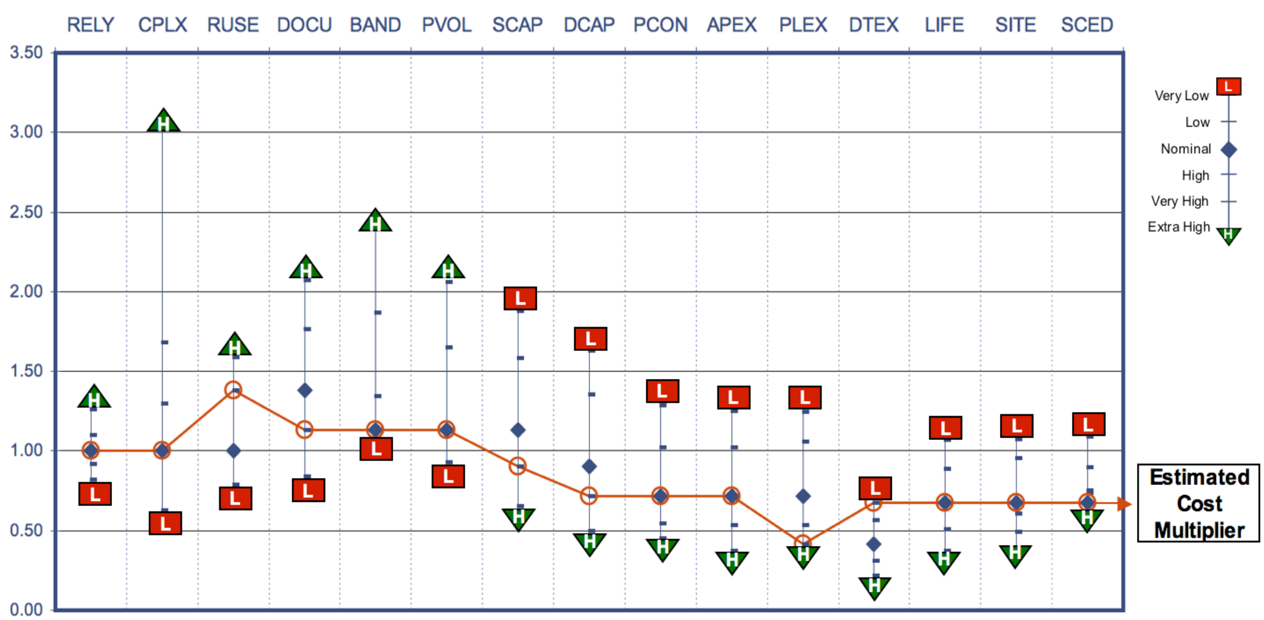


**Figure 6. Impacts of Effort Multipliers on Cost Estimation**

**Content Development Vendors**

In the training industry, custom content developers are frequently responding to requests for proposals (RFP) or rough orders of magnitude (ROM). The COSCOMO tool can be tailored to reflect an organization's unique scale factors such as team cohesion and process maturity. Through constant use of COSCOMO, the organization will learn which variables are lower or higher than industry norms. Once tailored to reflect the organization's differences, the tool becomes an even more reliable predictor for project estimation. The constant user can also preset effort multipliers and scale factors that remain common across all projects.

As an example for content developers, a project manager initially interacts with the tool to generate the size factor, rank the effort multipliers, and rank the scale factors based upon the organization, the proposed team, and the requirements in the statement of work. The tool provides the initial number of person-months for the project. The project manager may then apply labor rates to the number of hours to arrive at an initial estimate. Using the chart depicted in Figure 6, the manager can identify which variables are driving the estimate. In Figure 6, the ranking for Development for Reusability (RUSE) increases the person-month estimate. The graph helps the manager identify risks like platform volatility that must be tracked and managed. Also evident from the graph, high senior team capability (SCAP) and high development team capability (DCAP) are contributing significantly to reducing the project cost. If key drivers such as SCAP or DCAP change during the execution of a project, modifying the entries in the tool may help identify the amount of schedule and effort impact on the project based upon those changes.

**What if**

A final use case available to both audiences is in exploring how changes in project requirements and effort multiplier elements affect the person-month estimate. In the example above, increasing the experience and capability of SCAP and DCAP by one level each will decrease the estimate by 37%. Using the best team possible will reduce the person-month estimate up to 68% below the original estimate! If a fixed budget restricts the amount and level of courseware that can be developed, a manager can manipulate the size factor to explore various combinations of hours and levels to maximize the funding available.

Figure 7 shows how COSCOMO can be used to help address issues of change at the project definition level. You can enter your organization's customary values via the COSCOMO parameters, and indicate which ones will undergo change. COSCOMO will estimate how these changes will affect the project's expected cost and schedule, and will provide you and your stakeholders with a framework for re-scoping the project if estimated cost and schedule are unsatisfactory.

Frequently, changes in project objectives, priorities, available components, or personnel occur during project execution. If these are anticipated, COSCOMO can support a variant of the project definition process above to converge on a stakeholder-satisfactory re-scoping of the project.

A more serious case occurs when the changes are unanticipated and largely unnoticed: via personnel changes; COTS product, reusable component, or tool shortfalls; requirements creep; or platform discontinuities.

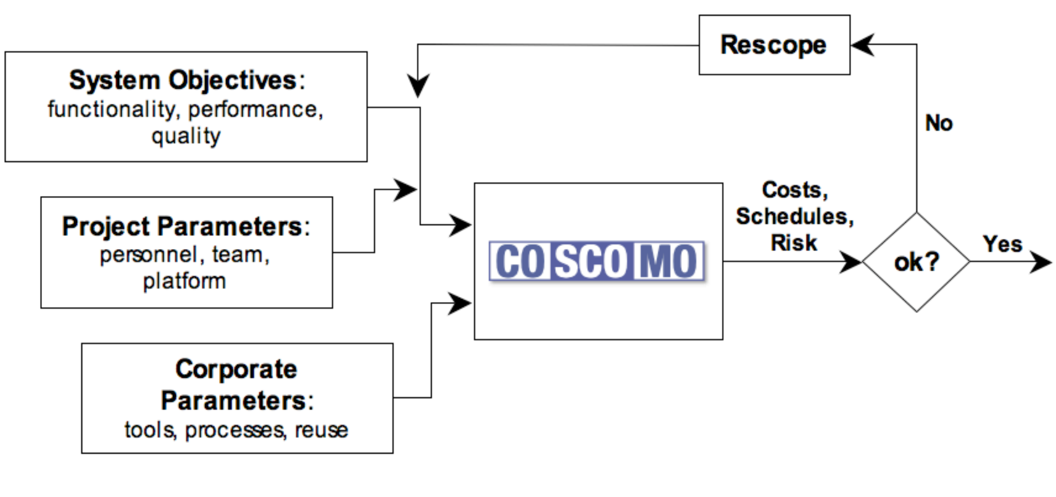


**Figure 7. Using COSCOMO to Cope with Change during Project Definition**

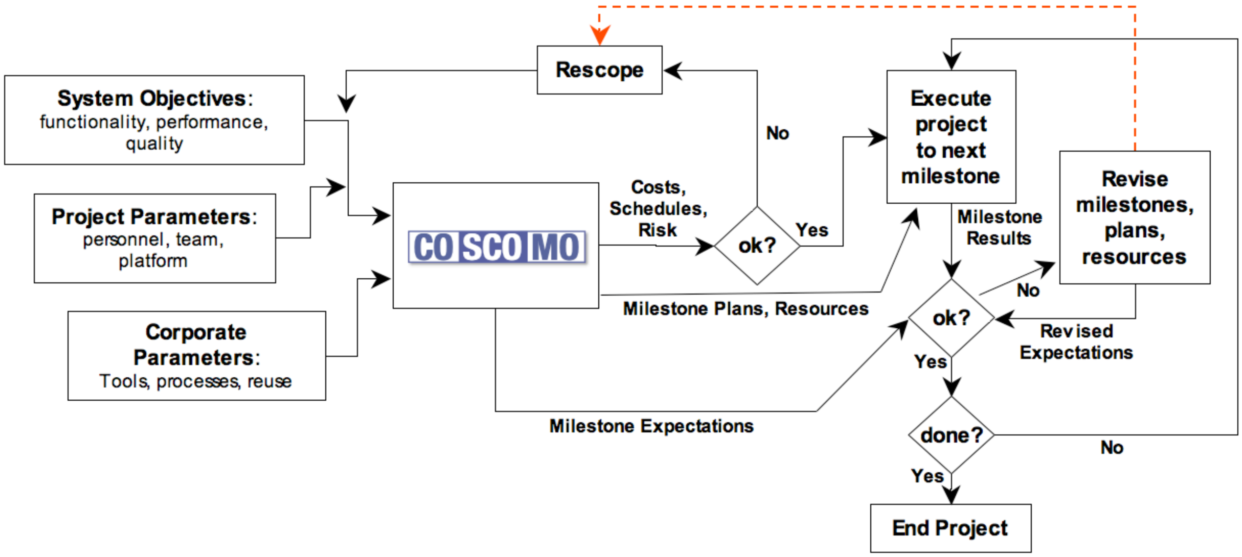


**Figure 8. Using COSCOMO to Cope with Change during Project Execution**

In such cases, COSCOMO phase and activity distributions can be used to develop a quantitative milestone plan or an earned-value system for the project, which enable plan deviations to be detected, and appropriate corrective actions to be taken involving COSCOMO in project re-scoping. Figure 8 illustrates this concept of how COSCOMO can be used to help address issues of change during the execution of a project. For example, when the project executes to the next milestone and the results do not meet stakeholder expectations, then that phase may be lengthened to allow the phase products to be modified and resubmitted. The COSCOMO element responses are reviewed to identify which responses may account for the missed milestone (i.e. Personnel Continuity or Precedentedness). Additionally, the distribution of effort across the ADDIE model phases would be modified to reflect the lengthened phase.

## SUMMARY

The prototype COSCOMO algorithm discussed in this paper is a first step in providing an open and standardized cost estimation tool for the interactive courseware community. The reliability of the COCOMO family of models is often measured by the percentage of test cases that the model will estimate within 30% of the actual project costs. With a very small initial set of historical project data, the COSCOMO model estimates 50% of its test cases within this 30% range. As with other such cost estimation models, it is expected that COSCOMO will continue to improve in its ability to accurately predict project costs accurately as it is calibrated against more data sets. As a reference, the COCOMO II model, with its large base of data sets that have been collected over two decades, estimates 64% of its test cases within 30%. Government agencies and web-based training developers are encouraged to start employing COSCOMO as a risk reduction tool in developing SCORM-conformant courseware. The tool is available for download from the Joint ADL Co-lab website (www.jointadlcolab.org).

## ACKNOWLEDGEMENTS

The authors wish to recognize Dr. Roger Smith for his contributions in conceptualizing and initiating the research and development project for COSCOMO.

## REFERENCES

Boehm, B., et al. (2000). *Software Cost Estimation with COCOMO II*. Prentice Hall.

Department of Defense Handbook MIL-HDBK-29612-3A, (2001). “Development of Interactive Multimedia Instruction (IMI) (Part 3 of 5 Parts)”

Department of Defense Instruction (DoDI) 1322.26, (2006). “Development, Management, and Delivery of Distributed Learning”

Kemp, J.E., Morrison, G.R., & Ross, S.M. (1996). *Designing Effective Instruction, 2nd Edition*. Prentice-Hall.

Kirkpatrick, Donald (1998). *Evaluating Training Programs: The Four Levels*. Berrett-Koehler Publishers, San Francisco, CA.