

# Assessing COTS Integration Risk Using Cost Estimation Inputs

Ye Yang  
University of Southern California  
941 w. 37<sup>th</sup> Place  
Los Angeles, CA 90089-0781  
1(213) 740 6470  
yangy@sunset.usc.edu

Barry Boehm  
University of Southern California  
941 w. 37<sup>th</sup> Place  
Los Angeles, CA 90089-0781  
1(213) 740 8163  
boehm@sunset.usc.edu

Betsy Clark  
Software Metrics, Inc.  
4345 High Ridge Rd.  
Haymarket, VA 20169  
1(703) 754 0115  
betsy@software-metrics.com

## ABSTRACT

Most risk analysis tools and techniques require the user to enter a good deal of information before they can provide useful diagnoses. In this paper, we describe an approach to enable the user to obtain a COTS glue code integration risk analysis with no inputs other than the set of glue code cost drivers the user submits to get a glue code integration effort estimate with the CONstructive COTS integration cost estimation (COCOTS) tool. The risk assessment approach is built on a knowledge base with 24 risk identification rules and a 3-level risk probability weighting scheme obtained from an expert Delphi analysis. Each risk rule is defined as one critical combination of two COCOTS cost drivers that may cause certain undesired outcome if they are both rated at their worst case ratings. The 3-level nonlinear risk weighting scheme represents the relative probability of risk occurring with respect to the individual cost driver ratings from the input. Further, to determine the relative risk impact, we use the productivity range of each cost driver in the risky combination to reflect the cost consequence of risk occurring. We also develop a prototype called COCOTS Risk Analyzer to automate our risk assessment method. The evaluation of our approach shows that it has done an effective job of estimating the relative risk levels of both small USC e-services and large industry COTS-based applications.

## Categories and Subject Descriptors

D.2.9 [Management]: Cost estimation

K.6.3 [Software Management]: Software development, software process

## General Terms

Management, measurement

## Keywords

Risk assessment, COCOTS, cost driver.

## 1. INTRODUCTION

The use of COTS products in COTS-based development (CBD) brings with it a host of unique risks quite different from those associated with software developed in-house. The “ Depending on

someone else” nature of CBD often places software processes and products under greater uncertainty. Many reported problems and lessons learned associated with CBD are consequences of poor appreciation of the risks involved and their management. There is a need to improve the risk management techniques for CBD to reap the expected benefits and avoid potential pitfalls of using COTS approach to system development.

Most risk analysis tools and techniques require the user to enter a good deal of information before they can provide useful diagnoses. For example, current CBD risk assessment techniques are largely checklist-based [5] or questionnaire-based [6], which requires large amount of user input information and also subjective judgment of domain experts. These techniques are human intensive and then often difficult due to scarcity of seasoned experts and the unique characteristics of individual projects. Risk assessment should be supported by automated processes and tools which are based on project attributes that can be quantitatively measured using project metrics.

Cost estimation models, e.g. the widely used COCOMO II model [1] for custom software development and its COTS extension -- COCOTS model [2], incorporate the use of cost drivers to adjust development effort and schedule calculations. As significant project factors, cost drivers can be used for identifying and quantifying project risk. A rule-based heuristic risk analysis method and its implementation have been developed and introduced in [3], where rules were structured around COCOMO II cost factors, reflecting an intensive analysis of the potential internal relationships between cost drivers. Another technique for integrating risk assessment with cost estimation attributes is discussed in [4].

In this paper, we describe an approach to enable the user to obtain a COTS glue code integration risk analysis with no inputs other than the set of glue code cost drivers the user submits to get a glue code integration effort estimate with the CONstructive COTS integration cost estimation (COCOTS) tool. The risk assessment approach is based on a knowledge base with 24 risk identification rules and a 3-level risk probability weighting scheme obtained from an expert Delphi analysis of the relative risks involved in the most critical combinations of COCOTS cost driver ratings.

This paper is organized as follows: Section 2 gives a short background on COCOTS model on which our method is based. Section 3 discusses the method and its steps. Section 4 introduces the implementation of the method. Section 5 examines the performance of its application to two sets of COTS integration projects. Finally, we conclude the paper and discuss directions of future work.

## 2. BACKGROUND

The work in this study is based on our previous work on the COConstructive COTS integration cost estimation model (COCOTS), esp. its Glue Code sub-model, and analysis of COTS integration risk profiles. The cost factors defined in COCOTS model are used in our risk assessment method to develop and identify risk patterns, in terms of cost driver combinations, in association with particular risk profiles within COTS-based development. The following sections give quick overview on COCOTS model and COCOTS Glue Code sub-model. For more details, the reader is referred to [1] and [2].

### 2.1 COCOTS model overview

COCOTS is a member of the USC COCOMO II family of cost estimation models [1]. It is developed for estimating the expected initial cost of integrating COTS software into a new software system development or system refresh, currently focusing on three major sources of integration costs: 1) COTS product assessment, 2) COTS product tailoring, and 3) Integration or “glue code” development. COCOTS effort estimates are made by composing individual assessment, tailoring, and glue-code effort sub-models as illustrated in Figure 1.

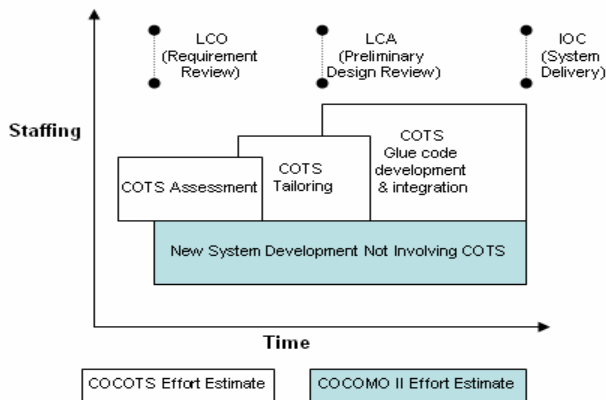


Figure 1. The COCOTS Cost Estimation Model

The COCOTS model has been largely adopted by some commercial cost estimation tools, e.g. PRICE-S [7]. It has also been found very helpful in cost estimating and resource planning in USC e-service CBA development projects [8, 9].

While COCOTS model was developed with the intention to determine the economic feasibility of COTS-based solutions, its model structure consists of most, if not all, of the important

project characteristics that should be carefully examined when assessing CBD project risk. One such example is the risk assessment method in [8] using the set of COTS assessment attributes defined in COCOTS Assessment sub-model to help determining the “sweet spot” amount of COTS assessment effort to balance project’s “too many errors” risk as well as “late delivery” risk.

For the purpose of our research, we are especially interested in using the COCOTS Glue Code sub-model as a starting point to develop a risk analyzer to assess CBD project risks. This is because among the three sub-models of COCOTS, the formulation of the Glue Code sub-model uses the same general form as does COCOMO, i.e. a well-defined and calibrated set of cost attributes which capture the critical aspects of project situation.

### 2.2 Glue Code Sub-model in COCOTS

The COCOTS Glue Code sub-model presumes that the total amount of glue code to be written for a project can be predicted and quantified in KSLOC (thousands of source line of code), and the broad project conditions in terms of personnel, product, system, and architecture issues can be characterized by standardized rating criteria. Based on these, fifteen parameters have been defined in the COCOTS Glue Code sub-model as shown in Table 1.

Table 1. COCOTS Glue Code sub-model cost drivers

Cost Factors	Name	Definition
Size Driver	Glue Code Size	The total amount of COTS glue code developed for the system.
Scale Factor	AAREN	Application Architectural Engineering
Cost Drivers	ACIEP	COTS Integrator Experience with Product
	ACIPC	COTS Integrator Personnel Capability
	AXCIP	Integrator Experience with COTS Integration Processes
	APCON	Integrator Personnel Continuity
	ACPM T	COTS Product Maturity
	ACSEW	COTS Supplier Product Extension Willingness
	APCPX	COTS Product Interface Complexity
	ACPPS	COTS Supplier Product Support
	ACPTD	COTS Supplier Provided Training and Documentation
	ACREL	Constraints on Application System/Subsystem Reliability

	AACPX	Application Interface Complexity
	ACPER	Constraints on COTS Technical Performance
	ASPRT	Application System Portability

Except the size driver, Glue Code Size, which is measured by thousand source line of code (KSLOC), all other cost factors are defined using a 5-level rating scale including Very Low, Low, Nominal, High, and Very High ratings. Therefore, a COCOTS estimation input is generally composed by the size input in KSLOC, and a specific symbolic rating ranging from Very Low to Very High for each of the rest 14 cost factors.

### 3. RISK ASSESSMENT USING COCOTS COST FACTORS

To construct a CBD risk analyzer from the cost factor ratings in a COCOTS estimation input, we start with knowledge engineering work by formulating risk rules and quantification scheme from COCOTS cost drivers and iteratively eliciting expert knowledge. The workflow steps in our risk assessment method are illustrated in Figure 2 and discussed in detail next starting with the central knowledge base.

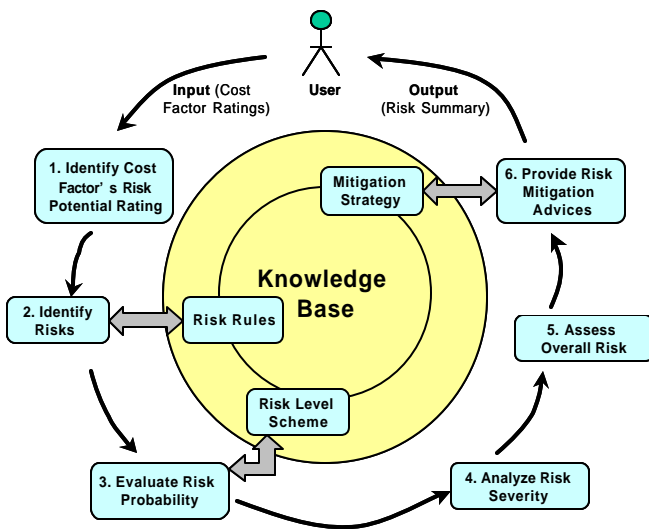


Figure 2. Risk Assessment Workflow

#### 3.1 Constructing the knowledge base

As shown in Figure 2, the knowledge base consists in the following three elements, risk rules, risk level scheme, and mitigation strategy. To construct the knowledge base, we developed a Delphi survey to collect experts' judgment on risky combinations of COCOTS cost factors and their quantification weighting scheme. Up to date, we have received response from 5

domain experts and established the risk rule table and risk level weighting scheme based on these Delphi feedbacks. The risk mitigation strategy is drafted based upon our previous risk analysis work in [9], aiming to provide specific mitigation plan advices with respect to the particular risk items identified by our method, esp. in the early phase of their project.

##### 3.1.1 Risk rules

Following the term definition in [3], we describe a CBD risk situation as a combination of two cost attributes at their extreme ratings, and formulate such combination into a risk rule. One example is a project condition whereby COTS products complexity (APCPX) is very high and the staff's experience on COTS products (ACIEP) is low. In such case, cost and/or schedule goals may not be met, since time will have to be spent understanding COTS, and this extra time may not have been planned for. Hence, a corresponding risk rule is formulated as:

*IF ((COTS Product Complexity > Nominal)  
AND (Integrator's Experience on COTS Product < Nominal))  
THEN there is a project risk.*

Table 2 lists results of risk situation identification from the 5 Delphi survey responses. The shaded table entries represent the risky combinations of COCOTS cost factors identified in the Delphi responses. Different cell shading patterns correspond to different percentage of responses over the total responses. For example, there are 24 risk combinations identified by more than half of the responses (i.e. identified by at least 3 experts in our case, shading pattern of "≥50%" in the table). Another 25 more risk combinations are identified in two of the five responses (shading pattern of "40%" in the table). The 20% shading pattern refers those risk combinations only identified in one of the five responses.

Table 2. Delphi results of risk situations identified from COCOTS model

	ASPRT	ACPER	AACPX	ACREL	ACPTD	ACPPS	APCPX	ACSEW	ACPMT	APCON	AXCIP	ACIPC	ACIEP	AAREN	SIZE
SIZE															
AAREN															
ACIEP															
ACIPC															
AXCIP															
APCON															
ACPMT															
ACSEW															
APCPX															
ACPPS															
ACPTD															
ACREL															
AACPX															
ACPER															
ASPRT															

■ ≥50%    ■ 40%    ■ 20%

To be more conservative, we select the 24 risk combinations that recognized in more than half of the Delphi responses as the foundation to formulate the risk rules for the knowledge base.

Please note that for the sake of simplicity in our study, we assume a symmetric risk matrix existing between all COCOTS cost factors. Therefore the region under the diagonal line remains blank.

Another observation from Table 2 is that some cost factors such as Size, AAREN, ACREL, APCPX, ACIPC, and ACPMT, are the “most sensitive factors” that can easily have risky interactions with others (see the shaded area in the table). Also note that interactions of cost attributes that are essentially orthogonal to each other are not identified as risk situations.

The other two elements of the knowledge base, risk level scheme and mitigation strategy will be introduced later when the workflow continues on to bring the appropriate context for them.

### 3.2 Identifying cost factor’s risk potential rating

As discussed in section 2.2, a COCOTS estimation input is a vector of 15 values, one numeric value carrying the size estimate in KSLOC, and 14 symbolic values representing the specific ratings ranging from Very Low to Very High for each of the other 14 cost factors.

In order to capture and analyze the underlying relation between cost attributes and the impact of their specific ratings on project risk, some transformation work on the inputted cost driver ratings is necessary. These include the resolving the difference between continuous representation of size driver and discrete representation of the other 14 cost factors, as well as establishing a mapping mechanism between cost driver ratings and the probability of risk caused by these ratings. Through these treatments, the risk assessment method can be developed and implemented on an automatable means that can be evaluated systematically with little involvement of human subjective measures. Next we introduce these treatments on the inputted COCOTS cost factors ratings.

#### 3.2.1 Derive risk potential rating for inputted size

To determine risk potential ratings for the inputted glue code size driver, we ask each domain expert to discretize the “Glue Code Size” driver into a 4-level risk potential rating schema and the responses are listed in Table 3. The 4 levels of risk potential rating are defined as OK, Moderate, Risk Prone, and Worst Case, with an increasing probability of causing problems when conflicting with other cost factor ratings. The median values for each rating level are used in our knowledge base to derive risk potential rating from an inputted size number.

**Table 3. Delphi Responses for Size Rating**

Rating	OK	Moderate	Risk Prone	Worse Case
Response 1	1	2	10	50
Response 2	2	5	10	25
Response 3	1	3	10	10
Response 4	1	2	10	50
Response 5	1	2	10	50
Median	1	2	10	50
Stdev	0.447214	1.30384	0	18.5741756

Therefore, the continuous representations in KSLOC are transformed into symbolic ones for simplicity to model in our study, as well as for generality to compare and analyze together with the rest 14 cost attributes.

#### 3.2.2 Derive risk potential ratings for the other cost factors

This transformation is more straightforward compared with the handling on size driver, because the other 14 cost factors are inputted based on the 5-level rating scheme, from Very Low to Very High. The same 4-level risk potential rating scheme is used to distinguish the high or low risk potential of each driver from its specific cost driver rating. Table 4 shows the detailed mapping schema for the other 14 cost factors with respect to the influences of their cost driver ratings on project risk.

Cost Factors	Cost Factor Rating	Risk Probability Rating
AAREN, ACIEP, ACIPC, AXCIP, APCON, ACPMT, ACSEW, ACPPS, ACPTD	Very Low	Worst Case
	Low	Risk Prone
	Nominal	Moderate
	High	OK
	Very High	OK
APCPX, ACREL, AACPX, ACPER, ASPRT	Very Low	OK
	Low	OK
	Nominal	Moderate
	High	Risk Prone
	Very High	Worst Case

**Table 4. Mapping between cost factor’s rating to its risk potential rating**

### 3.3 Identifying project risk situations

With the 24 risk rules formulated and stored in the knowledge base and the transformed cost driver’s risk potential ratings, our method will automatically check, match, and generate a list of risk situations for the specific project.

### 3.4 Evaluating the probability of risk

Risk impact is a term defined to measure the potential impact of certain risk, and the equation to calculate risk impact is the probability of loss multiplying the cost of loss due to the risk occurring. This section and the next two sections will discuss how the individual risk is quantified in terms of evaluating its probability and severity, and how the overall project is aggregated from individual risks.

		Attribute 1			
		Worst Case	Risk Prone	Moderate	OK
Attribute 2	Worst Case	Severe	Significant	General	
	Risk Prone	Significant	General		
	Moderate	General			
	OK				

### 3.4.1 Risk probability weighting scheme

For an identified risk situation, different rating combinations are evaluated to determine the relative probability level of risk. For instance, it is perceivable that a project with a Worst\_Case APCPX and a Worst\_Case ACIEP is having a greater probability to run into problems than one just with a Risk\_Prone APCPX and a Moderate ACIEP.

Corresponding to the 4 level risk potential rating scheme, we define 4 levels of risk probability as visualized in Table 5 where each axis is the risk potential ratings of a cost attribute. A risk situation corresponds to an individual cell containing an identified risk probability. In this step, risk rules use cost driver's risk potential ratings to index directly into these tables of risk probability levels.

**Table 5. Assignment of Risk Probability Levels**

To quantify risk probability levels, a quantitative nonlinear weighting scheme obtained from the expert Delphi survey is used to assign a risk probability level of

- 0.1 for general risk,
- 0.2 for significant risk, and
- 0.4 for severe risk.

Additionally, 0 is used as the probability of the blank cells in Table 5, situations where the risk is unlikely to occur.

### 3.5 Analyzing the severity of risk

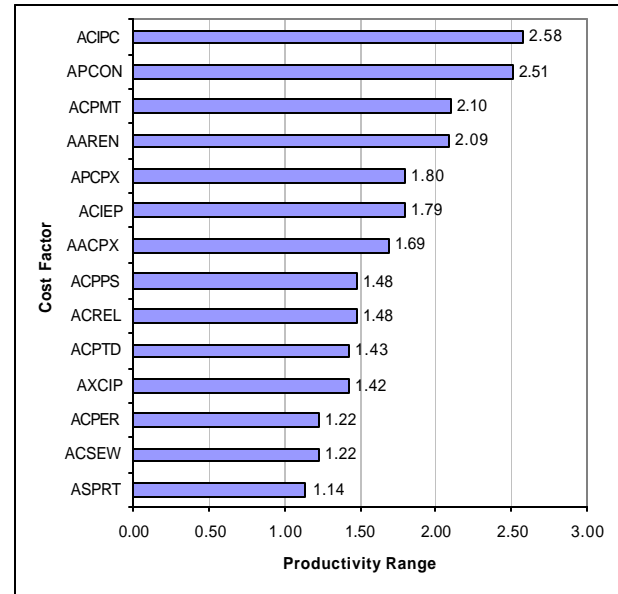
Besides the probability of risk occurring, we need to develop a means to represent and evaluate the relative severity of risk occurring. One key term used here is the productivity range of cost factors.

In COCOMO II family, the productivity range (PR) for a cost driver is defined as the ratio between its highest and lowest values (when treated as an effort multiplier). For a scale factor, the productivity range is calculated using a different formula, with raising a power of the difference between its highest and lowest values to a default project size of 100KSLOC.

Following these definitions, the productivity ranges of 14 COCOTS cost factors except size are calculated and shown in Figure 3. We believe the productivity range is a sound measurement to reflect the cost consequence of risk occurring, since it combines both expert judgment and industry data calibration during the development of COCOTS model.

**Figure 3. Productivity range of COCOTS cost factors**

For the glue code size driver, a number of 2 is also obtained from the expert Delphi analysis as its productivity range for evaluating risks involving size.



### 3.6 Assessing overall project risk

Section 3.4 and 3.5 discuss the evaluation of risk probability and risk severity with respect to every individual risk. This section talks about how to assess overall project risk by aggregating individually evaluated risks.

The overall project risk can be quantified following equations:

$$\text{Project risk} = \sum_{i=1}^{15} \sum_{j=1}^{15} \text{riskprob}_{ij} \times PR_i \times PR_j$$

**Figure 4. Risk quantification equation**

In the above formula,  $\text{riskprob}_{ij}$  to correspond to the nonlinear relative probability of the risk occurring as discussed in Section 3.4, and the product of  $PR_i$  and  $PR_j$  represents the cost consequence of the risk occurring as discussed in Section 3.5. Project risk is then computed as the sum risk impact of individual risks.

To interpret the overall project risk number, we use a normalized scale from 0 to 100 with benchmarks for low, moderate, high, and very high overall project risk. The scale is designed as follows: 0-5 low risk, 5-15 moderate risk, 15-50 high risk, and 50-100 very high risk. The value of 100 represents the situation where each cost driver is rated at its most expensive extremity, i.e. its worst case rating.

### 3.7 Providing risk mitigation advices

Finally, risk mitigation advices as discussed in our previous work in [9, 10] can be provided to the user with respect to each risk combinations identified by the risk analyzer. This will be further illustrated in the case study in Section 4.1 next.

Proj	ACII	ACII	AXC	APC	ACP	ACS	APC	ACF	ACF	ACR	AAC	ACF	ASF	AAF
A	WC	WC	RP	M	RP	RP	M	RP	WC	RP	M	RP	RP	M
B	M	OK	RP	WC	WC	RP	OK	RP	WC	M	OK	M	M	OK

## 4. CASE STUDY AND PROTOTYPE

### 4.1 Application case study

In this section, we illustrate the application of the risk assessment approach on two USC e-service CBA development projects. The risk analysis process of the case study is explained step by step. The discussion about the analysis results comparing with project data is described later. Table 6 shows COCOTS cost driver ratings. Project A has 400 lines of glue code and Project B has 1350 lines of glue code.

**Table 6.** Project glue code cost factor rating

Project	ACIEP	ACIPC	AXCIP	APCON	ACPMT	ACSEW	APCPX	ACPPS	ACPTD	ACREL	AACPX	ACPER	ASPRT	AAREN
A	VL	VL	L	N	L	L	N	L	VL	H	N	H	H	N
B	N	VH	L	VL	VL	L	L	L	VL	N	L	N	N	VH

Following the risk analysis steps discussed in section 3, we will show how to get the risk analysis result for project A and B step by step:

Step 1: Identify cost factor’s risk potential rating. Original size and cost driver ratings need to be mapped to risk potential ratings. According to Table 3 and 4 in Section 3, the size for project A and B are mapped into OK and Moderate ratings respectively. The transformation of all other cost drivers is shown in Table 7.

**Table 7.** Transformed cost factor risk potential rating

Step 2: Identify risks. We can find out risk situations for the project by matching the transformed glue code and cost driver risk potential ratings with risk rules in knowledge base. For instance, ACREL is High and APCPX is Nominal. Based on the risk rules in KB, this rating combination will not produce a risk. Another example, ACREL is High and ACIPC is Very Low. Based on the risk rules in KB, this rating combination will produce a risk. After exploring all the pairs, we get the specific risk situations for project A and B as showed in Figure 5:

ACREL & ACIPC	ACREL & ACPMT
ACREL & ACPMT	ACREL & AAREN
APCPX & ACIEP	ACPMT & AXCIP
APCPX & ACIPC	ACPMT & ACPMT
ACIPC & AXCIP	ACPMT & AAREN
ACIPC & APCON	AAREN & AXCIP
ACIPC & ACPTD	AAREN & ACPMT
ACIPC & AACPX	AAREN & ASPRT
ACPMT & AXCIP	APCON & AXCIP
ACPMT & ACPMT	APCON & ACPTD
APCON & ACPTD	

**Figure 5.** Risk Situation List for Project A (left) and B (right)

Step 3: Identifying risk probability, severity, and quantify overall risk. Firstly, risk probability level rating values are assigned for each risk combination, and the result is showed in Table 8 (the number 4 in the parenthesis refers to Severe risk, 2 for Significant risk, and 1 for general risk):

**Table 8.** Risk Probability Ratings for Project A

Risk	Probability
ACREL & ACIPC	0.2
ACREL & ACPMT	0.1
APCPX & ACIEP	0.1
APCPX & ACIPC	0.1
ACIPC & AXCIP	0.2
ACIPC & APCON	0.1
ACIPC & ACPTD	0.4
ACIPC & AACPX	0.1
ACPMT & AXCIP	0.1
ACPMT & ACPMT	0.1
APCON & ACPTD	0.1

Secondly, COCOTS cost driver risk level rating values and their productivity range values are plotted into the equation as shown in Figure 3, and the final normalized overall project risk for project A is 21. Similarly, we get the final normalized overall project risk for project B is 22. The conclusion we can make from our approach is project B is about at the same risk level as project A in CBD perspective, though their risk profiles differs largely. Project A has 4 risk situations involving ACIPC, integrator personal capability; while project B has 4 risks involving AAREN, architecture reengineering.

A further study is to compare these risk analysis results with the projects’ weekly progress reports. In project A’s progress reports, from week 1 to week 4, the top two risks for project is “ Personal shortfall: Some of the team members are not familiar with the online payment mechanism and the programming language needed” and “ Team members are not familiar with the shopping cart component and are not able to test it”. The first risk is reflected by APCPX\_ACIPC and ACIPC\_AACPX in Figure 5. The second risk is reflected by APCPX\_ACIEP in Figure 5. Advices for these two risks from the knowledge base in our approach could be “ Provide more training sessions for developers” and “ Ask COTS vendor for support”. Another risk appeared from week 7 is “ Shopping cart component may not be stable enough”. This is also indicated by ACREL\_ACPMT from the risk situation list. In this case, advice from the knowledge base could be “ More detailed assessment on COTS through testing”. In summary, about 85% COTS related project risks in the reports are reflected in our risk situation list. The only risk that is not caught in the risk situation list is “ Slow implementation on the component development due to environment factors”. The major reason is because the current COCOTS cost drivers do not address the issue of system environment.

### 4.2 Prototype of COCOTS risk analyzer

Since the manual analysis is labor intensive and error-prone, we develop a prototype of COCOTS risk analyzer to automate the

risk assessment method discussed in Section 3. It is currently implemented on top of MS Access Database.

The knowledge base is implemented through 3 tables: Risk Rule table, Risk Probability Scheme table, and Cost Driver Productivity Range table. There is also a Project Parameter table used to store project parameters that user submits for COCOTS estimation.

Figure 6 and Figure 7 show the screenshots of the input and output interface of the prototype.

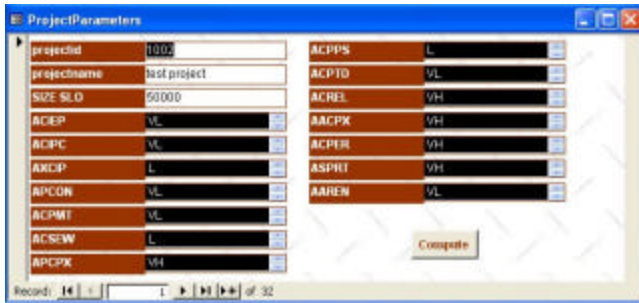


Figure 6. COCOTS risk analyzer – Input interface

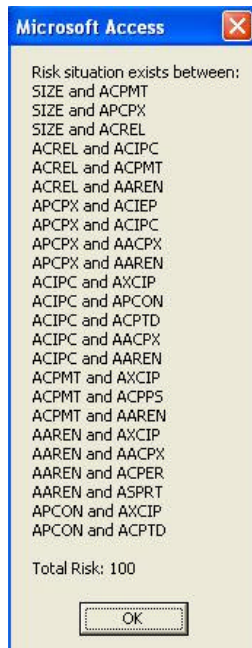


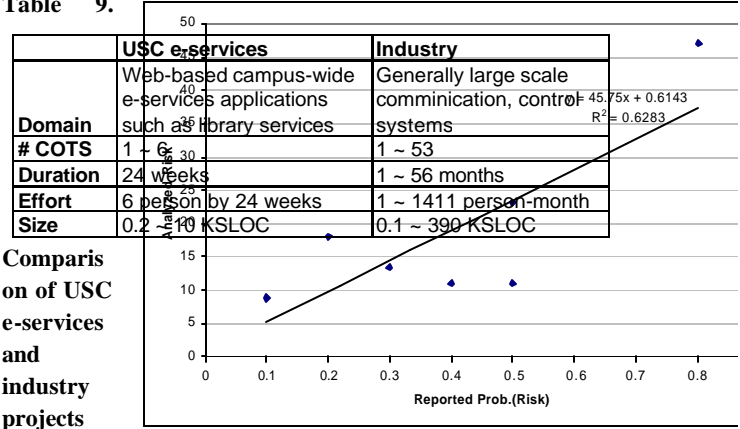
Figure 7. COCOTS risk analyzer – Output interface

As shown in Figure 5, the total risk of 100 refers to the worst case project where each cost factor input is at its worst case rating (i.e. either VH or VL rating in Figure 4). The list of risk situations is very helpful especially for inexperienced project manager or projects in early inception phase to identify risks that an experienced software manager might recognize, quantify, and prioritize during project planning.

## 5. Evaluations

To evaluate our risk assessment method, we have analyzed 9 small USC e-services CBD projects and 7 large industry CBD projects to apply the method. Some characteristics of both groups are compared in Table 9.

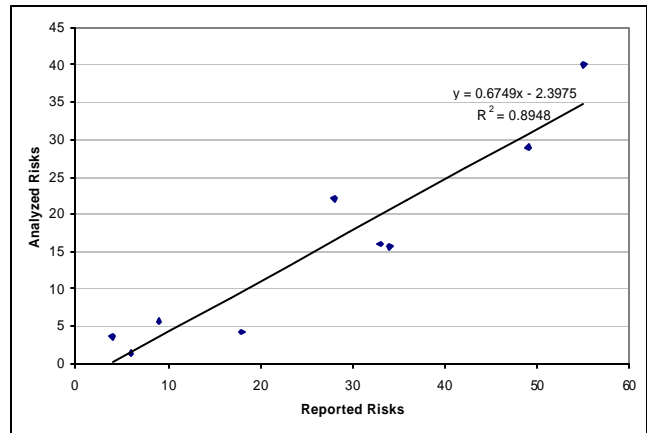
Table 9.



Comparison of USC e-services and industry projects

### 5.1 USC e-services projects

We are able to access to the COCOTS estimation reports of the 9 USC e-services projects, and run the COCOTS risk analyzer using



the 9 projects' COCOTS inputs. On the other hand, we examined the weekly risk reports of the 9 projects to get the actual risks reported by the development team.

The x-axis in Figure 6 is the reported number of risks from the teams' weekly risk report; and the y-axis is the analyzed risk by applying our risk assessment method. The trendline in Figure 8 shows that a strong correlation exists between the team reported risks and the analyzed project risks from our COCOTS risk analyzer tool.

Figure 8. Evaluation results of USC e-services projects

### 5.2 Industry projects

To evaluate the performance of COCOTS risk analyzer on large-scale projects, 7 large industry projects from COCOTS model

calibration database are independently analyzed by two of the authors. One author runs the COCOTS risk analyzer on the project data to calculate the project risks; and the other author who has been involved in the COCOTS data collection provides her evaluation of the relative risk for each project based on her early interviewing notes with key project personnel during the data collection process.

The x-axis in Figure 9 represents the data collection author's rating of relative project risks evaluated from interviewing notes; and the y-axis is the analyzed risk by applying our risk assessment method. Though the correlation line in Figure 7 is not as strong as that in Figure 6, it still shows a nice trendline and R-sqr value of 0.6283 between the predicted risk level and reported risk level.

**Figure 9.** Evaluation results of Industry projects

## 6. CONCLUSIONS AND FUTURE WORK

COTS-based development brings a host of unique risk items different from those in traditional software development. Many reported problems and lessons learned associated with CBD are consequences of poor appreciation of the risks involved and their management.

Unlike most risk analysis tools and techniques which require the user to enter a good deal of information before they can provide useful diagnoses, this paper presents an automated, heuristic risk assessment approach and its prototype implementation, the COCOTS Risk Analyzer, to assess the COTS integration risks using the cost factors defined in COCOTS to detect patterns of COTS integration risk.

By leveraging on expert knowledge in both cost estimation and risk management, this approach enables the user to obtain a COTS glue code integration risk analysis with no inputs other than the set of glue code cost drivers the user enters to obtain a COCOTS glue code integration effort estimate. It identifies a list of risk situations and assesses these risks in conjunction with COCOTS estimation to help create mitigation plans based on the relative risk severities and provide mitigation advice. The initial evaluation results illustrate that it has done an effective job of estimating the relative risk levels of 17 COTS-based e-services applications, both small and large projects.

The work presented here is an ongoing research. The future work includes:

- Explore specific risk mitigation advices for identified risks and incorporate these in future implementation to provide stronger support for risk management planning;
- Perform further expert Delphi analysis of the relative risks involved in the most critical combinations of COCOTS cost driver ratings.
- Extend the COCOTS risk analyzer to the other two sub-models of COCOTS, COTS Assessment sub-model and

COTS Tailoring sub-model to support risk assessment during the whole lifecycle of COTS-based development.

- Collect project data and calibrate the COCOTS risk analyzer model to improve its accuracy and effectiveness.

## 7. ACKNOWLEDGMENTS

Our thanks to Brad Clark, Gary Thomas, Dick Stucktz for sharing their great insights with us during the expert Delphi.

## 8. REFERENCES

- [1] B. Boehm, C. Abts, A.W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. Reifer and B. Steece. Software Cost Estimation with COCOMO II. Prentice Hall PTR, July 2000.
- [2] C. Abts, "Extending the COCOMO II Software Cost Model to Estimate Effort and Schedule for Software Systems Using Commercial-Off-The-Shelf (COTS) Software Components: the COCOTS Model," Ph.D. Dissertation, Oct. 2001.
- [3] R.J. Madachy, Heuristic Risk Assessment Using Cost Factors. IEEE Software, May/June 1997.
- [4] K. Kansala. "Integrating Risk Assessment with Cost Estimation," IEEE Software, vol. 14, no. 3, pp. 61-67, May/June 1997.
- [5] A. Rashid, and G. Kotonya (2001) Risk Management in Component-Based Development: A Separation of Concerns Perspective. ECOOP Workshop on Advanced Separation of Concerns (ECOOP Workshop Reader). Springer-Verlag Lecture Notes in Computer Science.
- [6] D. Carney, E. Morris, and P. Place. Identifying Commercial Off-the-Shelf (COTS) Product Risks: The COTS Usage Risk Evaluation. September 2003. TECHNICAL REPORT. CMU/SEI-2003-TR-023.
- [7] A. Minkiewicz, The Real Costs of Developing COTS Software. <http://www.pricystems.com/downloads/pdf/COTSwhitepaper3-31-04.pdf>.
- [8] D. Port, and Z.H. Chen. "Assessing COTS Assessment: How Much Is Enough?" Proceedings, ICCBSS 2004, Feb. 2004, pp. 183-198.
- [9] B. Boehm, D. Port, Y. Yang, and J. Buhta, "Not All CBS Are Created Equally: COTS-Intensive Project Types," Proceedings, ICCBSS'03, Ottawa, Canada, Feb. 2003, pp. 36-50.
- [10] D. Port, and Y. Yang, "Empirical Analysis of COTS Activity Effort Sequences," Proceedings, ICCBSS'04, Los Angeles, California, USA, Feb. 2004.

