# The Cost Risk/Uncertainty Exposure Determination (CRED) Model – A New Approach

Cheryl L. Jones, Army Futures Command

Robert Charette, PhD, ITABHI Corp.

Bradford Clark, PhD, Galorath Federal

Sariyu Marfo, Senior Operations Research Analyst, Office of the Deputy Assistant Secretary for Cost and Economics (ODASA-CE)

James Judy, NISEC Division Chief, ODASA-CE

Paul Janusz, Army Futures Command

## Abstract

The objective of this model is to improve the credibility and trust in software sustainment project cost estimate by:

1) Identifying, characterizing, and accounting for different cost performance factors that may be sources of risk/uncertainty and can result in creating material impacts on software sustainment/maintenance cost estimate.
2) This approach makes visible the "knowledge gap" (if any) between "what is known" and "what should known" about the system under assessment. The "knowledge gap" is an input used to aid in the assessment of the degree of uncertainty associated with the estimate.
3) It also fully documents the key program issues, assumptions and related performance factors that may influence the cost estimate and why.

## Introduction

*"We don't know what we don't know"*

This well-known adage expresses the potential for risk and uncertainty in every program/project. The Cost Risk/Uncertainty Exposure Determination (CRED) model attempts to account for risks/uncertainties that may be overlooked or understated. The CRED model documents the key program issues and related performance factors derived from analysis of scores of DoD program risk assessments that may influence the quality of a cost estimate.

Contributing to why these factors may be unknown is that program identification of risk and uncertainty is often perceived as being due to managerial weakness or incompetence, as observed by the late J. Ronald Fox, professor emeritus in business administration at Harvard University, past assistant secretaries to both the Air Force and Army, and an expert in defense acquisition and management.

"The project manager often views risks as mere expected challenges, which can be addressed using normal management techniques and processes. Therefore, project managers tend not to highlight risks. Managers also fear that highlighting risks may make their project more costly, and possibly increase the likelihood of its postponement or cancellation." [Fox 2011] Similarly, project managers do not like to highlight their assumptions or constraints for the same reasons.

In trying to capture the impact of the unknown, cost estimators seek to capture risks and uncertainty originating from two sources.

1. The first source of risks/uncertainties is errors found in the estimation methodology. Regarding the first error source, estimation by analogy, for example, has error in the degree of disparity of similar systems to the one being estimated. Estimation by a work breakdown structure bottoms-up summation of cost, on the other hand, has the error of missing tasks and activities. Estimation using parametric models has error from the model itself as well as error in model inputs (often derived from the project context). These estimation methodology errors are important to consider, have been addressed extensively in historical literature, and are not discussed in this paper. [Smart 2020, Garvey 2000, Boehm 1981]
2. The second source of error or risks/uncertainties is in the context of the estimate, i.e., what is the current situation in which the estimate is being made. For instance, is it a major system modernization, an extensive upgrade, or a yearly update? Differing context will determine the types of information that an estimator needs to know. Context sources of error can be minimized when project managers provide the required information. However, as noted earlier, project managers view risk differently, which for the estimator, can mean having few risks identified. The CRED model assists the cost estimator (and project manager) in filling the knowledge gaps in the project context.

   Cost estimators are rarely software experts yet are asked to assess software intensive systems that even experts would have some difficulty trying to understand. The CRED model tries to assist estimators to dig deeper into a program to identify these program's "soft or blind spots." In unprecedented systems, this is critical, since the program itself may not recognize where the knowledge gaps are, or where the assumptions made are weak. Using CRED, estimators are directed to ask these two fundamental questions: "What should we know?" and "What do we know?" Implicit in these two questions for the estimator, of course, is "Can we find out?" If not, then the certainty with the cost estimate created should be adjusted accordingly.

The CRED model provides guidance on what should be known reducing the "We don't know what we don't know" conundrum. The next sections will describe the model and its application.

## Some Working Definitions

There are four important terms that need to be defined when applying the CRED modeling approach: risk, uncertainty, material information, and exposure. While the usage of the terms is fairly standard, there exist subtleties that require explanation for those not experienced in risk analysis or cost estimation.

When the term "risk" is used, the working definition is, "A risk is a potential future event or condition that may have a negative effect on cost, schedule, and/or performance" and "Risk is an event not in the project's baseline that is an undesirable outcome". [NASA 2017, OSD 2020] There is also the concept of "positive risks", i.e., risks that create beneficial opportunities, for example, by creating events that can achieve cost reductions. In this paper, we are only concerned about negative risks or what issues gleamed from past program experiences may keep a program or project from succeeding if not addressed.

The term "uncertainty" is defined as, "Uncertainty is the indefiniteness of the outcome of a situation" as well as in a larger meaning as, "Uncertainty is the indefiniteness about a project's baseline". [NASA 2017, OSD 2020] The CRED model investigates both types of uncertainty.

Two more terms need to be defined, the first being "material information." For CRED modeling purposes, "Information is material if omitting, misstating or obscuring it could reasonably be expected to influence the decisions that the primary users of general-purpose financial statements make on the basis of those financial statements, which provide financial information about a specific reporting entity." [IFRS 2018] The CRED model does not try to assess every type of project risk but concentrates on those that may definitively impact the cost estimate. Omission or misstatement can be assessed by asking *What should I know?* and *What do I know?*

Typically, for risk analyses, the term "exposure" is defined as the probability of a risk multiplied by the consequence of the risk. For CRED modeling purposes, we use a slightly different concept: exposure is a quantification of the gap between what should be known and what is actually known. This knowledge gap implicitly incorporates the classical definition of risk exposure. Exactly what this means will be discussed in the following sections.

## Assessing Risk & Uncertainty

Assessing risk/uncertainty is related to "material information" — namely those events, situations or circumstances that could influence a decision maker when making an affordability decision concerning a program or project's sustainment/maintenance costs. In the CRED modeling approach, there are four material information categories: 1) cost environment, 2) internal software, 3) program and project management, and 4) external program environment. Each category has a list of attributes, based on Tri-Service Assessment Initiative Systemic Analysis [McGarry 2003] that examined scores of DoD's largest software intensive system projects and the documented historical experience gained from developing the software sustainment database, that can be used to review potentially missed or omitted material information. The software sustainment database, the largest database of its kind, collected cost and effort data on 267 systems with 1,499 software releases over a period of 8 years. Review of the data in the database validated the practicality and coverage of the material information categories presented in this paper.

Each of these *material information categories* are discussed in the following sections.

### *Cost Environment Attributes*

A program/project's "cost environment" reflects the general confidence or credibility level a cost estimator has in information they are using in their maintenance or sustainment cost estimate. This is garnered from assessing the Software Sustainment (SWS) Work Breakdown Structure (WBS), Figure 1, against what we do know versus what we should know. Each WBS element has a series of questions to help the cost estimator understand what information may be missing or should be taken into consideration in their cost estimate.

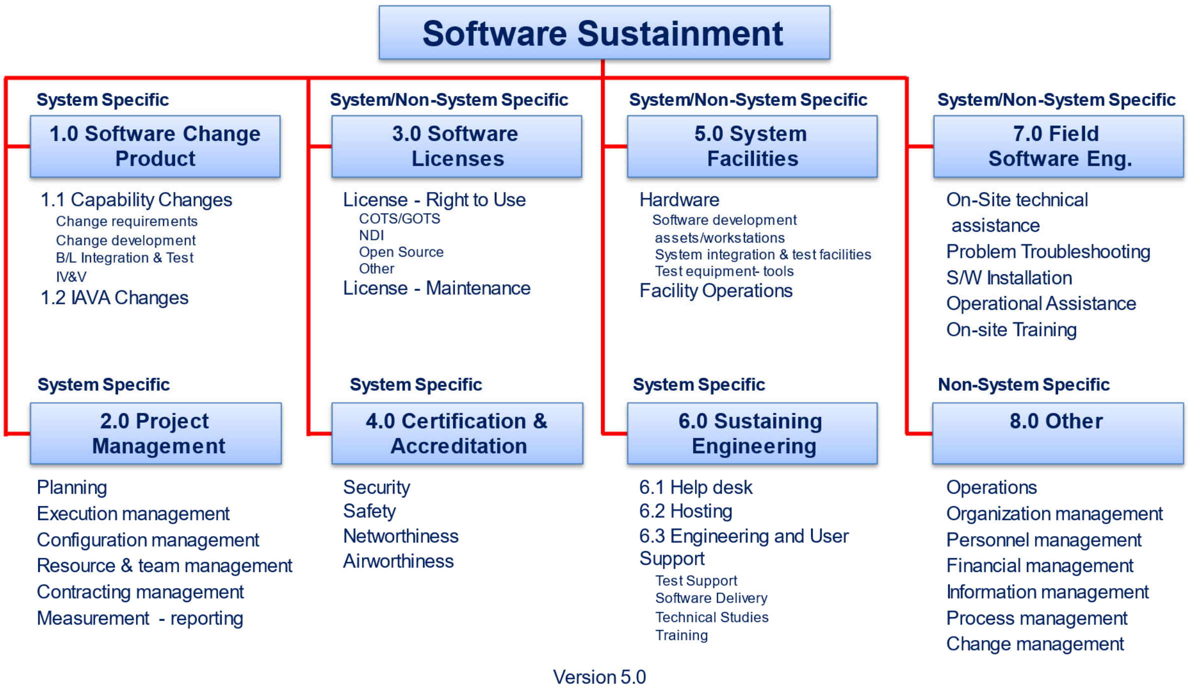


Figure 1. Software Sustainment Work Breakdown Structure

1. Software Change Product.
   This attribute addresses what is known about the amount of anticipated change to the software. The volume of changes could be expressed using several different measures: number of requirements, number of function points, number of RICE-FW[1] objects, number of external interfaces, number of software change requests, number of source lines of code, etc. Is the amount of change known for this estimate?

2. Project Management.
   This attribute is significant, and it is treated as its own materiality category. It will be discussed in a later section on Program/Project Management Attributes.

3. Software Licenses.
   Are the number and cost of each Commercial Off The Shelf (COTS) product used in development and operations known? Which products are renewed annually? If the software does not use COTS (or other types of Non-Developed Items), this attribute would not be considered.

4. Certifications and Accreditations (C&A).
   Will the software have to be certified or accredited for things such as security, safety, networthiness, or airworthiness? Are the numbers and cost of each C&A known? Sometimes this attribute is eliminated from the estimate because the software does not require C&A.

5. System Facilities.
   This attribute addresses the hardware used to change, maintain, and test the software such as workstations, test facilities, and test equipment as well as keeping the lights on. Are

[1] Reports, Interface, Conversion, Enhancements, Forms, and Workflows

the costs for hardware and utilities known? Sometimes this attributed is eliminated from the estimate because facility costs are covered by a different organization.

6. Sustaining Engineering.
   This attribute addresses costs associated with running a help desk, using a server hosting environment, engineering activities such as studies, design, test support, deployment, and training. For costs that are germane to the estimate, are the costs known?
7. Field Software Engineering.
   This attribute addresses on-site or field technical assistance, problem troubleshooting, software installation, operational assistance, and training. Are the costs for this service known?
8. Other Costs.
   This attribute address organizational costs usually in the form of a tax. Are these costs known?

The assessment of the Cost Environment uses a table based on “What Should We Know” versus “What Do We Know” as illustrated in Table 1. Each attribute is rated between 0 (nothing is known) and 10 (an adequate amount is known) for “What Should We Know” versus “What Do We Know.” It should be noted that any rank scoring scheme could be used if it is consistently applied.

In the Software Sustainment phase, full understanding of the attribute is assigned a value of 10 in the “What Should We Know” row. In the “What Do We Know” row, a subjective assessment is made by the cost estimator on what is known about that attribute. A rating of 10 means full understanding, a rating of 5 means a partial understanding, and a rating of zero (0) means major uncertainty. For example, if the project provides the estimator with detailed information in a WBS element, a high rating may be given; similarly, if the project can provide only little to no information, a low rating will result. Obviously, estimator judgement and experience play a part on determining a specific rating, as does honesty on part of the project. The estimator must trust the information the project is providing is accurate.

The difference between what should be known and what is known is the Individual Exposure value for each attribute, the third row. For instance, if a cost estimator does not have an anticipated change amount for the software (1.0 Software Change Product, column 2) but has historical data, the rating for “What Do We Know” might be rated as a 7. The Individual Exposure in this instance would be 10 minus 7, or 3.

Table 1. Cost Environment Assessment Table

| Operations and Sustainment | 1.0 Software Change Product | 3.0 Software Licenses | 4.0 C&A | 5.0 System Facilities | 6.0 Sustaining Engineering | 7.0 Field Software Engineering | 8. Other Costs |
|---|---|---|---|---|---|---|---|
| What Should We Know? | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| What Do We Know? | 7 | | | | | | |
| Individual Exposure | 3 | | | | | | |

For exposures greater than zero (0), a rationale should be recorded for what additional information is needed. The table and associated rationales become documentation for recording the risk/uncertainty in the Cost Environment material information category.

It may be that closing the knowledge gap may not occur until the project completes several tasks in the future. Thus, the gap highlights both an assumption that is being made (i.e., the assumption that the future tasks will in fact be successfully completed at the predicted cost and schedule) as well as the current uncertainty (or dependencies) that are present in the project. Future cost reviews can quickly examine where these knowledge gaps exist and see whether they have indeed been closed.

The scope of a cost estimate may not include all the elements in the SWS WBS depending on the project context. Only those elements that are within the estimation scope of the program or project should be assessed. The CRED model is flexible enough to be tailored to the scope of the estimate. Tailoring will be shown in the section on CRED Model Application Example.

### *Internal Software Attributes*

The Internal Software material information category are the intrinsic material attributes of the software-intensive system (product) that make it "easier" or "harder" to develop/implement and therefore likely to successfully maintain/sustain. Again, there are questions to aid the cost estimator.

1. Number of External Interfaces and Ownership.
   This attribute addresses the number of interfaces to other systems. The uncertainty is the unforeseen changes in an interface causing unplanned software changes. How many external interfaces are there and who owns the interfaces? What is the percentage of system data received and/or passed to other systems?
2. Execution Timing Constraints.
   This attribute addresses the percentage of system key performance parameters that are dependent on real-time execution. Real-time software is challenging to sustain because of the need for guaranteed execution times, e.g., a flight controller that misses a cycle may lead to an uncontrollable air vehicle. What percentage of system key performance parameters depend on non-stop processing?
3. COTS Product Incorporation.
   This attribute addresses the percentage of system key performance parameters met by COTS software. What percentage of the system functionality depends on COTS software?
4. Critical Technology (for enhancements).
   This attribute addresses the software technology readiness level required to meet system key performance parameters. What percentage of software system components in use are approaching obsolescence? What is the complexity resulting from security or legal mandates?
5. Data Rights.
   Often overlooked, this attribute addresses the rights, including copyright and other intellectual property, the program/project has to the technical data and computer software delivered as part of a contract. What is the extent of the program's or project's various

license rights (e.g., unlimited, restricted, or limited)? Does (non)access to data rights affect the long-term cost of sustainment/maintenance?

The assessment of the Software Internal Attributes uses a table based on "What Should We Know" versus "What Do We Know" as explained above in Cost Environment. Table 2 shows the attribute table.

Table 2. Software Internal Assessment Table

| Operations and Sustainment | External Interfaces | Execution Timing Constraints | COTS Product Incorporation | Critical Technology | Data Rights |
|---|---|---|---|---|---|
| What Should We Know? | 10 | 10 | 10 | 10 | 10 |
| What Do We Know? | 5 | | | | |
| Individual Exposure | 5 | | | | |

Again, each attribute is rated between 0 (major uncertainty), 5 (partial understanding) and 10 (full understanding of the attribute) in the "What Do We Know" row. The difference between what should be known and what is known is the individual exposure.

For exposures greater than zero (0), a rationale should be recorded for what additional information is needed. The table and associated rationales become documentation for recording the risk/uncertainty in the Software Internal material information category.

### *Program & Project Management Attributes*

The Program/Project Management material information category are the activities (i.e., ones that are generally within a program or project's trade space or direct control) that add risk or uncertainty to the planned/in-work sustainment/maintenance effort.

1. Management Personnel Capability.
   This attribute addresses the capability of management personnel. What percent of management is moderately to highly experienced in software maintenance? What percent of management has worked on this type of system before?

2. Technical Personnel Capability.
   This attribute addresses the capability of technical personnel. What percent of the personnel is moderately to highly experienced in software maintenance? What percent of the personnel is moderately to highly experienced with this type of system?

3. Technical Processes Capability.
   This attribute looks at process capability. What percent of the maintenance processes are useful/effective? What percent of software trouble reports are traceable to process shortfalls?

4. Facilities & Infrastructure Support.
   Unlike the System Facilities attribute in Cost Environment, this attribute speaks to the adequacy of facilities and infrastructure to get the work done. What percent of the support tools is considered useful? What percent of needed planned capital equipment (e.g., for SILs, simulators, and emulators) is available?

5. Sustainment/Funding Rhythm.
   This attribute looks at what drives software maintenance. What percent of the maintenance work is discretionary as opposed to legally mandated? What percent of funding has been changed in the past fiscal year? What percent of current maintenance work is attributed to backlog? What percent of current maintenance work is attributed to technical debt?

6. Project & Program Management.
   This attribute looks at the stability of the management team. How experienced is the project/program management? Is the project/program management team stable or changing?

Table 3 is the assessment table for Program & Project Management. Recall each attribute is rated between 0 (major uncertainty), 5 (partial understanding) and 10 (full understanding of the attribute) in the "What Do We Know" row. The difference between what should be known and what is known is the individual exposure. For instance, if management personnel have been working on this system for 5 years, the rating for Management Personal Capability "What We Do Know" would likely be 10 and the individual exposure would be zero (0).

Table 3. Program & Project Management Assessment Table

| Operations and Sustainment | Management Personnel Capability | Technical Personal Capability | Technical Process Capability | Facilities & Infrastructure Support | Sustainment / Funding Rhythm | Project & Program Management |
|---|---|---|---|---|---|---|
| What Should We Know? | 10 | 10 | 10 | 10 | 10 | 10 |
| What Do We Know? | 10 | | | | | |
| Individual Exposure | 0 | | | | | |

Again, for exposures greater than zero (0), a rationale should be recorded for what additional information is needed. The table and associated rationales become documentation for recording the risk/uncertainty in the Software Internal material information category

### *External Program Environment Attributes*

The External Program Environment material information category are the extrinsic activities (i.e., ones that are outside a program or project's trade space or direct control) that add risk or uncertainty to the planned/in-work sustainment/maintenance effort. These may require extensive discussion between the cost estimator and the project/program management team to understand their impact.

1. External Stakeholders.
   This attribute addresses the chaotic impact external stakeholders can have on a project. How many external stakeholders are there? Do external stakeholders provide funding, set requirements, or both? Is there agreement or conflict among different stakeholders as to the system's mission priorities?

2. Mandated Requirements.
   This attribute looks at the impact of non-negotiable requirements. How stable are

mandated policies and guidelines and do they conflict? Are mandated policies, like security, fully funded?

3. Policy-driven Maintenance/Funding Rhythm.
   This attribute addresses the impact of funding shortfalls and unfunded plans. How long does the project/program funding stream look secure? Are there planned and funded system upgrades?

The assessment table, Table 4, follows the same pattern as the previous tables.

Table 4. External Program Environment Assessment Table

| Operations and Sustainment | External Stakeholders | Mandates | Policy-driven Sustainment / Funding Rhythm |
|---|---|---|---|
| What Should We Know? | 10 | 10 | 10 |
| What Do We Know? | 7 | | |
| Individual Exposure | 3 | | |

As mentioned previously, the assessment tables can be modified to reflect the material information that is relevant to the scope of the cost estimate. For instance, if the project has no external stakeholders, that column can be eliminated in Table 4. Additionally, if there is additional material information that should be in the table, it can be expanded to accommodate that new information, e.g., Contractor Volatility (meaning there is a change in contract support, or a high turnover of contractor personnel) if that is historical material information.

In fact, the ability to create material information categories and add/remove attributes makes the CRED model adaptable to other cost estimation domains beside software sustainment, such as project development. All that is needed is a list of material attributes in a domain that identify typical project "soft or blind spots" and questions to help the cost estimator. This list could be the result of conducting retrospectives on past projects, lessons learned from prior cost estimates that have underperformed, or a brainstorming session with experienced cost estimators. How the CRED model would change for different project development milestone phases have been explored.

## Assessing the Knowledge Gap

### *Individual gap exposure*

What do we do with the self-assessments for each of the material information categories: cost environment, software internal, program & project management, and external program environment? Each attribute table presents a list of attributes with individual exposures for each attribute.

To quantify the individual exposure relative to "What Should We Know", add up the should-we-know values for each attribute and do the same for the individual exposure. Divide the individual exposure sum by the should-we-know sum to derive a value between 0 and 1. This value provides an assessment of the attributes that may increase or decrease the overall risk/uncertainty for that material information category. A low value would indicate a decrease in risk/uncertainty and a high value would indicate major risk/uncertainty.

$$Category\ Exposure = \frac{\sum Individual\ Exposure\ values}{\sum Should - We - Know\ values}$$

### *Total exposure*

Total exposure is the sum of the individual exposures in each material information category divided by the sum of the should-we-know values in each category. This presents an overall indication of risk/uncertainty: a low value indicates reduced risk/uncertainty, and a high value indicates major risk/uncertainty.

$$Total\ Exposure = \frac{\sum Individual\ Exposure\ sums}{\sum Should - We - Know\ sums}$$

The total assessment exposure is only a guideline. It is important to look at each individual attribute to see if there is a large risk/uncertainty knowledge gap that requires further investigation.

As a general guideline, if a total exposure value is between 0.0 and 2.0, that is minimal exposure and considered satisfactory. If the value is between 0.2 and 0.5, collect more information or increase the estimate uncertainty range. If the value is above 0.5, collect more information and significantly increase the estimate uncertainty range.

What estimate uncertainty range should be used? At present, there is no historical data from which to derive values. Cost groups may have their own derived values which could be used. Based on a Delphi process of several highly experienced defense project cost experts, a table of values was proposed for estimates in the Operations and Sustainment phase, Table 5. Total exposure is Low when the exposure value is between 0.0 and 0.2. Exposure is Medium when the value is between 0.2 and 0.5. Exposure is High when the value is above 0.5.

Table 5. O&S Estimation Uncertainty Ranges

| Total Exposure | Best Case | Most Likely | Worst Case |
|---|---|---|---|
| Low | 0.95 | 1.05 | 1.10 |
| Medium | 1.00 | 1.10 | 1.15 |
| High | 1.10 | 1.15 | 1.25 |

It may be puzzling to see that in the "most likely" case, the multiplier is always higher than one. The reason for this is to account for the ongoing problem of "optimism bias" that Frank Kendall, former Under Secretary of Defense for Acquisition, Technology, and Logistics (AT&L). mentions as a cause of many acquisition-cost "overruns" and is part and parcel of all human decision making (i.e., humans tend to optimistically estimate their ability to accomplish some task) [Kendall 2013].

Finally, the weighting of the material information used to determine "low – medium – high" risk/uncertainty level may also need to be tailored if deemed appropriate. Currently, each material information component is weighed equally. However, this weighting approach is likely to be unrealistic in practice. For example, internal system attributes tend to influence the risk/uncertainty level early in the acquisition process more than later when the system is deployed, and operational experience has been gained. Similarly, external drivers may exert more influence during war time situations than during peacetime operations in O&S.

An example of this are the CRED estimation uncertainty ranges for Milestone A (Concept Development) estimates that were, as previously mentioned, explored. Table 6 shows that the ranges are much higher because of the higher degree of unknowns that exist in this early lifecycle phase. This will be apparent in the material information category assessment tables. The values in this table are based on a Delphi of experienced defense cost experts and have not been validated with data.

Table 6. Milestone A Estimation Uncertainty Ranges

| Total Exposure | Best Case | Most Likely | Worst Case |
|---|---|---|---|
| Low | 1.00 | 1.25 | 1.50 |
| Medium | 1.00 | 1.50 | 1.90 |
| High | 1.00 | 1.75 | 2.50 |

# CRED Model Application Example

To illustrate the use of the CRED model, a fictitious case study is presented. The Chimera Helicopter is an ACAT I Aviation program. The program and associated flight avionics are in the operations and support (O&S) phase. The software is in the Real Time domain. The engineering opinion is that software changes are driving the cost of sustainment. The Program Office has provided the anticipated number of software changes for the next major release.

Given its size and potential cost, a sustainment cost estimate is needed for the next release. Estimators will use the CRED model as an adjunct to their normal estimation process to better highlight the risk and uncertainty in the estimate. Below are the tables for each material information category.

**Cost Environment**

Table 7 provides an example of a cost environment assessment. The Program Office has provided the number of anticipated software changes. The Software Change Product attribute is rated 10 for what-do-we-know resulting in zero (0) for individual exposure. The number and cost of software licenses is not known but there is data on cost from analogous helicopter programs. The Software Licenses attribute is rated 5 for what-do-we-know resulting in a 5 for exposure. The cost for C&As is known but there are expected to be a few more additional certifications. The C&A attribute is rated 8 for what-do-we-know resulting in a 2 exposure. The remaining attributes are out of scope for this estimate.

Table 7. Cost Environment Assessment Example

| Operations and Sustainment | 1.0 Software Change Product | 3.0 Software Licenses | 4.0 C&A | ~~5.0 System Facilities~~ | ~~6.0 Sustaining Engineering~~ | ~~7.0 Field Software Engineering~~ | ~~8. Other Costs~~ |
|---|---|---|---|---|---|---|---|
| What Should We Know? | 10 | 10 | 10 | | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| What Do We Know? | 10 | 5 | 8 | | | | |
| Individual Exposure | 0 | 5 | 2 | | | | |

The sum of the individual exposures is 7. The sum for what-should-we-know is 30. This results in an overall cost environment risk/uncertainty exposure is 7 / 30 = 0.23 which is considered in this case to be satisfactory.

**Software Internal**

In Table 8, the number and ownership of avionics software interfaces are known; individual exposure is zero (0). The software is generally considered real time but there is uncertainty in the proposed enhancements; exposure is rated by the estimators as 2. COTS products are used in the software, but it is unclear how much of the system's performance relies on them; exposure is set as 5. The enhancements rely on new critical technology, but the readiness of this technology has not been determined; its exposure is set to 10. There are presently no information data rights issues, however, the Program Office desires to transition the maintenance of the avionics software to the Aviation Life Cycle Management Center which will lead to a question of who owns the data rights, resulting in an exposure of 10.

Table 8. Software Internal Assessment Example

| Operations and Sustainment | External Interfaces | Execution Timing Constraints | COTS Product Incorporation | Critical Technology | Data Rights |
|---|---|---|---|---|---|
| What Should We Know? | 10 | 10 | 10 | 10 | 10 |
| What Do We Know? | 10 | 8 | 5 | 0 | 0 |
| Individual Exposure | 0 | 2 | 5 | 10 | 10 |

The sum of the individual exposures is 27. The sum for what-should-we-know is 50. This results in an overall software internal risk/uncertainty exposure of 27 / 50 = 0.54. It is clear that more information is needed especially for the critical technology and data rights attributes to understand how much credibility should be given the cost estimate.

**Program & Project Management**

In Table 9, the project management team has five years of experience in managing this program although there are some shortfalls in knowing about the cost of COTS products, the degree of critical technology readiness, and data rights; exposure 2. The technical personnel are the same as those that developed the Chimera avionics; exposure zero (0). The technical processes use by the contractor are unknown possibly causing quality and deliver acceptance issues; exposure 10.

The remaining attributes are out of scope for this estimate.

Table 9. Program & Project Management Assessment Example

| Operations and Sustainment | Management Personnel Capability | Technical Personnel Capability | Technical Process Capability | ~~Facilities & Infrastructure Support~~ | ~~Sustainment / Funding Rhythm~~ | ~~Project & Program Management~~ |
|---|---|---|---|---|---|---|
| What Should We Know? | 10 | 10 | 10 | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| What Do We Know? | 8 | 10 | 0 | | | |
| Individual Exposure | 2 | 0 | 10 | | | |

The total individual exposure is 12. The sum of what-should-we-know is 30. The overall program & project management uncertainty is 12 / 30 = 0.4. Again, more information is needed on the contractor's technical processes before trust in the cost estimate is assured.

**External Program Environment**

Table 10 presents the results of an assessment of the external environment. Besides the Army, the Marine Corp are a participating stakeholder. They have worked well with the management team, and they provide maintenance funding; individual exposure zero (0). The mandated policies and guidelines have not changed and are fully funded; exposure zero (0). While the funding stream for both services look secure, there has been changes in funding priorities in the past; exposure 8.

Table 10. External Program Environment Assessment Example

| Operations and Sustainment | External Stakeholders | Mandates | Policy-driven Sustainment / Funding Rhythm |
|---|---|---|---|
| What Should We Know? | 10 | 10 | 10 |
| What Do We Know? | 10 | 10 | 2 |
| Individual Exposure | 0 | 0 | 8 |

The sum of the individual exposures is 8. The sum for the what-should-we-know is 30. The overall exposure is 8 / 30 = 0.26, which again is seen as a satisfactory result.

**Total Exposure**

Total exposure is the sum of the individual exposures in each material information category divided by the sum of the should-we-know values in each category. This presents an overall indication of risk/uncertainty: a low value indicates reduced risk/uncertainty, and a high value indicates major risk/uncertainty.

$$Total\ Exposure = \frac{\sum Individual\ Exposures\ sums}{\sum Should - We - Know\ sums} = \frac{7 + 27 + 12 + 8}{30 + 50 + 30 + 30} = \frac{54}{140} = 0.39$$

As a general guideline, if a total exposure value is between 0.2 and 0.5, collect more information or increase the estimate uncertainty range. In this example, Critical Technology, Data Rights, and Technical Process Capability had high individual exposures (10) indicating the need for more information.

As demonstrated, the CRED model highlights where the credibility of the cost estimate could greatly be increased by having additional information available to the cost estimator. In addition, the model has the flexibility to adapt to the scope of the estimate and context of the project. It also documents the estimation context unknowns that will be useful for retrospective analysis of software cost estimates that proved to be less than accurate.

## CRED Modeling Limitations

The CRED model is still a paper-based model and while experimentally applied on several test cases, it has not been verified and validated in the field. As discussed in the Assessing the Knowledge Gap section, research still needs to be conducted to recommend a cost estimate uncertainty range, i.e., how much to increase/decrease an estimate.

The weighting of the material information attributes between zero (0) and 10 may need to be tailored. Currently, each material information attribute is weighed equally. However, this weighting approach may be unrealistic in practice.

There is a possibility that the material information categories or their attributes do not cover all the unknowns. The model was based on the assessment of scores of software intensive programs during the Tri-Service Assessment Initiative as well as the information gleamed from 267 programs conducted under the Software Sustainment Initiative over a period of 8 years. Specific or unique program vulnerabilities may not be represented in this paper. The model is flexible enough to be expanded to include additional questions for each attribute. The questions presented in this paper are sample questions and are not comprehensive.

There is a small possibility of double counting or discounting the level of risk/uncertainty present in an estimate due to how attributes interact. For instance, lack of material information in the cost environment category could impact the assessment of the management capability attribute in the program & project management category.

Furthermore, we do not yet fully understand how to connect sensitivity profiles (Tables 5 and 6) with mission readiness, which is the ultimate objective of the CRED model.

## Conclusions

The CRED Model attempts to provide insight into *We don't know what we don't know*. The model makes visible the "knowledge gap" (if any) between "what should be known" and "what is known" about the system under estimation. This model examines in detail the "context" surrounding the estimate. This context may influence the parameter settings in parametric model estimation.

By using the assessment tables, the CRED Model provides documentation on what is known and unknown. This is valuable in justifying wide estimation ranges and in investigating past estimates. The model, as its name implies, also highlights how much credibility and trust a given cost estimate should be given. A cost estimate where large knowledge gaps exist should be treated with extreme caution.

While this presentation focuses on the defense software intensive project domain, it is highly adaptable to other domains. The ability to create material information categories and add/remove attributes makes the CRED model adaptable to other cost estimation domains beside software sustainment or development. Any domain where knowledge of risk/uncertainty exists is a candidate for use. All that is needed is a list of material attributes in a domain that identify typical project "soft or blind spots." This list could be the result of conducting retrospectives on past projects, lessons learned from prior cost estimates that have underperformed, or a brainstorming session with experienced cost estimators.

By applying the CRED model, cost estimators will be able to start shrinking the scope of what they don't know.

## References


Boehm 1981 — Barry Boehm, "Software Engineering Economics," Prentice-Hall, Englewood Cliffs, NJ, 1981.

Fox 2011 — J. Ronald Fox, "Defense Acquisition Reform 1960-2009: An Elusive Goal," Center of Military History, United States Army, CHM Pub 51-3-1, 2011.

Garvey 2000 — Paul Garvey, "Probability Methods for Cost Uncertainty Analysis," Marcel Dekker, Inc., New York, 2000.

IFRS 2018 — International Financial Reporting Standards, "Definition of Material," https://www.ifrs.org/content/dam/ifrs/project/definition-of-materiality/definition-of-material-feedback-statement.pdf, October 2018.

Kendall 2013 — Frank Kendall, "Use of Fixed-Price Incentive Firm (FPIF) Contracts in Development and Production," Defense AT&L a publication of the Defense Acquisition University, March-April 2013. https://www.dau.edu/sites/default/files/Migrate/DATLFiles/Mar-Apr2013/DATL_mar-apr2013.pdf

McGarry 2002 — John McGarry, "System Analysis of Software Intensive System Acquisition Issues," Software Technology Conference, April 29, 2003.

NASA 2017 — "NASA Cost Estimating Handbook," Version 4.0, 2017

OSD 2020 — "DOD COST ESTIMATING GUIDE," OSD Cost Assessment and Program Evaluation, Version 1.0, December 2020

Smart 2020 — Christian Smart, "Solving for Project Risk Management: Understanding the Critical Role of Uncertainty in Project Management," McGraw Hill, 1st Ed., October 2020.