Introduction and Objectives

Welcome to the Consider Risk and Uncertainty lesson. After completing this lesson, you will understand this fourth of the five major steps of developing a software cost estimate.

Lesson Objectives

- Consider the application of sensitivity analysis to a software estimate by varying key parameters one at a time.
- Summarize the key risk areas for software and how they might be reflected in the estimate.
- Compare the point estimate to the probabilistic estimate (S-curve) and support the reasonableness of the relative position of the former and range of the latter.

D

Long Description

Graphic illustrates the steps of the Cost Estimating process. The steps from left to right are:
Develop Scope and Approach, Collect and Analyze Data, Develop Estimate Methodology, Consider Risk and Uncertainty (highlighted), and Document and Present Estimate.

Conduct Sensitivity Analysis

After the cost estimate is developed, it must be validated. The purpose of testing the estimate is to ensure reasonableness and completeness. The analyst should test key cost elements for sensitivity to the cost-estimating techniques used and to key ground rules and assumptions.

As a prelude to Uncertainty, we usually want to conduct Sensitivity Analysis on our cost estimates. This explores the cost-driving relationships across the total estimate by varying key input parameters one at a time, holding all else equal ("ceteris paribus").

The range across which we vary a given input should reflect the degree of uncertainty surrounding that number. For example, if software development has not yet started, and we have only a vague (and no doubt optimistic) notion of what our software reuse will be, we could vary reuse from a very high degree to none at all.

However, if we are in detail design and have identified all the specific code (including reuse libraries) we will be reusing, then we needn't vary our reuse by as much.

Conduct Sensitivity Analysis, Cont.

The basic question we are asking in sensitivity analysis is, if we are wrong on the inputs to our cost estimate, how wrong might we be on the outputs (the estimate itself)?

Again, while we are focused on the estimate of cost (effort), we can use the same process to understand impacts on schedule (duration) as well.

Conduct Sensitivity Analysis - Sensitivity Analysis of Estimate Inputs

Sensitivity analysis is a technique used to treat uncertainty surrounding estimate inputs.

It is used to evaluate the effects of changes in system parameters on the system cost.

The generic steps in sensitivity analysis are:

1. Compute the life-cycle cost (LCC).
2. Select the elements for analysis.
3. Determine the range of values for each element selected for analysis.
4. Re-compute the life-cycle cost by using the high and low values of the element in question.
5. Graph or tabulate the results.
6. Analyze for reasonableness and compare to the baseline.

Popup Text

Sensitivity Analysis

The repetition of an analysis with different quantitative values for selected parameters or assumptions for the purpose of comparing the results with the basic analysis. If a small change in the value of the variable results in a large change in the results, then the results are said to be sensitive to that parameter or assumption.

Conduct Sensitivity Analysis - Sensitivity Analysis of Estimate Inputs, Cont.

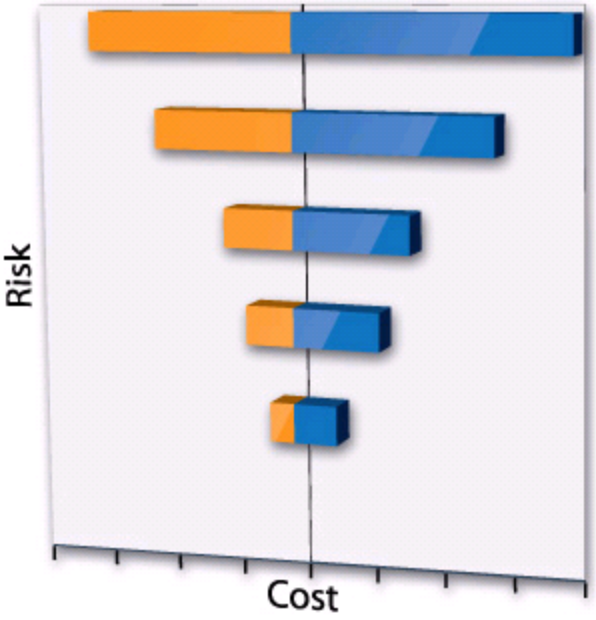
Sensitivity analysis depicts which system elements cause large changes in cost. These elements are then considered to be sensitive.

Requirements and parameters that can vary without significantly affecting cost are insensitive.

This information allows the manager to more wisely manage scarce resources by concentrating on the sensitive elements, in terms of both estimating and management.

Sensitivity analysis impacts are often depicted graphically, as in a tornado chart.

Prime candidates for sensitivity analysis for software include parameters related to the cost drivers Size, Capability and Complexity.



Assess Potential for Growth

From Sensitivity, we proceed to Risk, which is a consideration of the potential for cost growth.

Anywhere the estimate is optimistic or "aggressive," there is the potential for growth in final cost as compared with our estimate.

That is, if we have failed to consider things that can (and routinely do) go wrong, then our estimate is likely understated.

Conversely, if our estimate is conservative or "padded," there is the potential for underruns. A potential reduction in cost, while not as common as risks, is called an opportunity

Popup Text

Cost Growth

A term related to the net change of an estimated or actual amount over a base figure previously established. The base must be relatable to a program, project, or contract and be clearly identified, including source, approval authority, specific items included, specific assumptions made, date, and the amount.

Assess Potential for Growth, Cont.

Though the terms are often used interchangeably, it is important to distinguish between Risk and Uncertainty.

Risk tries to capture the potential for cost growth relative to the point estimate. A risk-adjusted estimate should represent a true mean or average, the expectation for cost across the entire range of possible outcomes.

Uncertainty, by contrast, attempts to characterize that range of outcomes. Think of Risk as a net adder to the estimate, and Uncertainty as the "plus or minus" about the estimate.

Because risks themselves are uncertain events, an inadequate consideration of Risk will tend to lead to an understatement in Uncertainty as well.

Correctly modeling Risk and Uncertainty is arguably the biggest challenge in cost estimating. But first it is important to focus on the identification of Risk, as that is usually more than half the battle.

Questions Managers Should Ask

D

Long Description

Risk "Net Adder" and Uncertainty "Plus or Minus" flowing down to Cost Estimate.

Questions Managers Should Ask

Are the estimated costs and schedule consistent with demonstrated accomplishments on other projects?

- Did the values used for cost and schedule input parameters appear valid when compared to values that accurately predicted past projects?

Have the factors that affect the estimate been identified and explained?

- Has historical code growth been applied to the sizing estimate, including a correction for reuse optimism?
- Does the estimate include costs associated with modifying and integrating any planned COTS software?
- Was a risk analysis performed, and risks that affect cost or schedule identified and documented?

Have steps been taken to ensure the integrity of the estimating process?

- Have cost and schedule risk been added to the estimate to account for requirements volatility, consistent with the software life cycle methodology and acquisition strategy?
- If a dictated schedule has been imposed, is the estimate accompanied by an estimate of both the normal schedule and the additional expenditures required to meet the dictated schedule?
- Were any adjustments made to parameter values in order to meet a desired cost or schedule documented and accompanied by management action that makes the values realistic?
- Do estimators independent of the performing organization concur with the reasonableness of the parameter values and estimating methodology?

Assess Potential for Growth - Types of Risk

Risk and Uncertainty in cost estimates arises from many sources. Although these different types of risk occur simultaneously, in analysis they may be treated separately or together in various combinations. Keep in mind that these risks as described are forward-looking. The instantiation of these risks present in historical data is known as (cost) growth. Select each tab to read more.

Requirements | Technical | Cost Estimating

Threat Risk arises from imperfect knowledge about the state of the world, from mischaracterization or shifting nature of the threat.

A classic example of threat risk from the Iraq war is the initial underestimation of the prevalence and effectiveness of improvised explosive devices (IEDs).

Popup Text

Threat Risk

Threat Risk arises from imperfect knowledge about the state of the world, from mischaracterization or shifting nature of the threat.

A classic example of threat risk from the Iraq war is the initial underestimation of the prevalence and effectiveness of improvised explosive devices (IEDs).

Threat Risk

Risk that occurs when new threats are revealed in the System Threat Assessment Report (STAR) or threat assessment. The solution was incorrect because the needs of the system were incorrect.

Requirement Risks

Requirement Risk arises from failure to document all the true requirements of the system in view of an unchanging threat. Requirements risk causes changes in the system configuration over the course of its development.

These changes are usually deliberately introduced and, historically, have made a prime contribution to cost growth, especially in software development.

For example, a land-based unmanned aerial vehicle (UAV) is being adapted for maritime use, and requirements related to some of the additional modes needed in the control software for landing on the deck of a ship were omitted or incomplete.

Requirements Risk

The risk of unforeseen design shifts from the current Cost Analysis Requirements Description (CARD) or System Specification due to shortfalls within that description. The proposed design's inability to meet the mission or misinterpretation of the solution are causes for requirements risk.

Popup Text

Technical

Technical Risk arises from engineering difficulties in developing a solution to address the documented requirements of the system.

This is the most common use of “Risk” in acquisition and is often paired with Schedule (Sked/Tech Risk).

For example, difficulties in coding one of the ProRad waveforms, which might manifest as code growth or rework.

Technical Risk

The risk that arises from activities related to technology, design and engineering, manufacturing, and the critical technical processes of test, production, and logistics.

Popup Text

Cost Estimating (CE) Risk

Cost Estimating (CE) Risk relates to the application of estimating techniques, independent of the technical inputs thereto.

One would expect this risk to be symmetric, i.e., a net adder of zero, but there seems to be a small systematic bias, in part due to subtle technical issues related to certain analytical techniques.

CE Risk is focused on the “irreducible noise” in estimates, the statistical uncertainty arising from natural variation in the data.

For example, uncertainty in productivity for a given size CSCI, as captured in the statistical properties of the CER.

Cost Estimating (CE) Risk

The risk arising from cost estimating errors and the statistical uncertainty in the estimate. Mathematical error, omission, and double counting are all examples of possible mistakes.

Assess Potential for Growth - Root Cause Analysis (RCA)

The Weapon Systems Acquisition Reform Act of 2009 (WSARA), established the Performance Assessments and Root Cause Analyses (PARCA) organization within the Office of the Assistant Secretary of Defense for Acquisition (OASD(A)) to recognize the importance of conducting "post-mortems" to understand how a program's performance (including cost and schedule) was impacted by various factors.

By examining such root cause analyses (RCAs) for analogous historical software-intensive programs, we attempt to apply lessons learned to the current program we are estimating and managing.

There is the temptation to fall into the mindset that "history is something that happens to other people," but even when specific pitfalls can be avoided based on others' experiences, other unanticipated problems – the proverbial "unknown unknowns" – will inevitably crop up, and these must be accounted for.

RCA is still an emerging field, but it is an important source of Risk information we cannot afford to ignore. It is particularly important to conduct RCA on programs with new technology or those that involve critical life and death decisions, such as "friend or foe" detection algorithms.

Popup Text

Root Cause Analysis (RCA)

With respect to a major defense acquisition program, an assessment of the underlying cause or causes of shortcomings in cost, schedule, or performance of the program, including the role, if any, of—

(1) unrealistic performance expectations;
(2) unrealistic baseline estimates for cost or schedule;
(3) immature technologies or excessive manufacturing or integration risk;
(4) unanticipated design, engineering, manufacturing, or technology integration issues arising during program performance;
(5) changes in procurement quantities;
(6) inadequate program funding or funding instability;
(7) poor performance by government or contractor personnel responsible for program management; or
(8) any other matters.

Assess Potential for Growth - Root Cause Analysis (RCA), Cont.

One key source of historical program cost growth is the Selected Acquisition Reports (SARs) that are submitted to Congress on an annual basis for Major Defense Acquisition Programs (MDAPs). They have a standard set of cost variance categories, including:

- Economic: If labor rates for specialized programming skills rise more steeply than anticipated, then software development labor costs will grow (even if labor hours do not).
- Engineering: If a signal processing algorithm turned out to be more complex than predicted, then software design, cost, and test hours will increase.
- Schedule: If external interface definitions were late, the resultant delay and disruption will cause increased cost of the software development effort.

Assess Potential for Growth - Requirements Creep

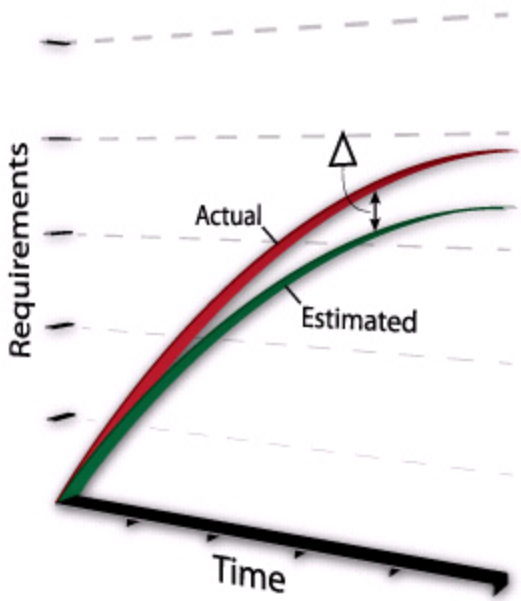
Complex projects are often prey to a phenomenon called "requirements creep," wherein additional requirements are levied after the initial requirements phase, leading directly to growth in project scope (and therefore cost and schedule).

Software development is notoriously susceptible to this problem, with an inability or reluctance to fully define requirements up front ("I don't know what I want until I see it"). The powerful but intangible nature of software is a contributing factor.

For example, in some cases hardware issues may be remedied with software fixes, but this leads to additional unplanned software development effort.

Requirements creep for software-intensive systems should be considered in conjunction with the Life Cycle Development Approaches (Waterfall, Incremental, Evolutionary, Spiral).

Evolutionary and Spiral in particular embrace requirements uncertainty with their "build a little, test a little" approach, while attempting to keep a lid on requirements creep by conducting trade-off analyses and focusing on the most immediately needed functionality at any given time while deferring "nice-to-haves" to future versions or updates wherever possible.



Δ = Requirements Creep

D

Popup Text

Requirements Creep

The tendency of the user (or developer) to add to the original mission responsibilities and/or performance requirements for a system while it is still in development.

Long Description

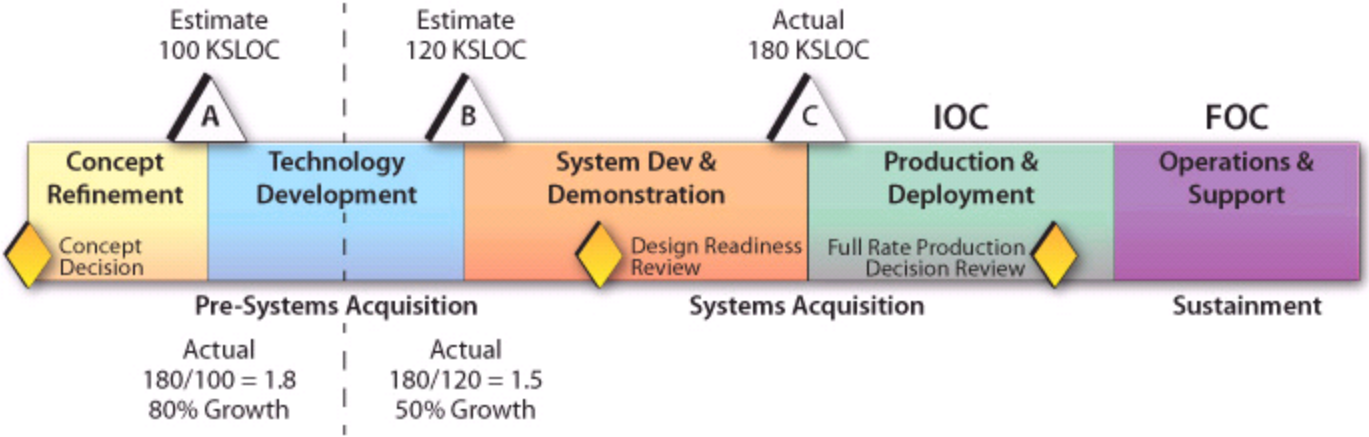
Chart illustrating requirements creep with Time on the X-axis and Requirements on the Y-axis. Two data lines moving left to right are separated with the area of difference or delta between the two representing requirements creep.

Assess Potential for Growth - Code Growth

Whether driven by Requirements Risk, Technical Risk, or both, there is an undeniable and inevitable tendency of software code to grow in size from initial estimates to final delivered code. Since code size is a key driver for both cost and schedule estimates, if code growth is not accounted for, estimates can be understated, usually significantly. Code growth on the order of dozens of percentage points or more is not uncommon.

A proven approach to handling code growth is to maintain a database of historical programs, each with size estimates at various milestones together with final size, as empirically determined by a code counter. A growth factor can then be determined for application in future estimates.

For example, suppose the sizing estimate at MS A was 100 KSLOC, and at MS B 120 KSLOC. If the actual code ended up being 180 KSLOC, then the growth factor from MS A is 180/100 = 1.8 (or 80% growth), and from MS B 180/120 = 1.5 (or 50% growth).



D

Long Description

Acquisition time line with Estimate 100 KSLOC above and Actual 180/100 = 1.8, 80% Growth beneath the time line at Milestone A. Estimate 120 KSLOC above and Actual 180/120 = 1.5, 50% Growth beneath the time line at Milestone B. Actual 180 KSLOC over Milestone C.

Assess Potential for Growth - Reuse Optimism

Assumptions on reuse can drastically reduce the number of Equivalent SLOC (ESLOC), which in turn can lead to significant "savings" in effort and schedule. The problem is that such assumptions tend to be optimistic in two ways:

1. There is a tendency to overestimate the amount of code that can be reused, with or without modification; and
2. There is a tendency to overestimate the effectiveness of reusing code.

The consequence of the first assumption is that the development team ends up having to write from scratch code originally thought to be Reuse or Modified. The consequence of the second assumption is that ESLOC conversion factor (and hence ESLOC) are higher than anticipated. In other words, even though we were able to reuse code, it required a greater degree of redesign, recode, and/or retest.

As with code growth in general, reuse optimism is best addressed by developing a track record based on historical data and applying factors derived therefrom to future estimates. Similar optimism can apply to commercial off-the-shelf (COTS) software, where there may be a tendency to overestimate the degree of functionality provided by COTS, as well as the ease of integration.

D

Long Description

Three arrows: green upward arrow with (text) Reuse, red downward arrow with (text) ESLOC, green upward arrow with (text) Saving.

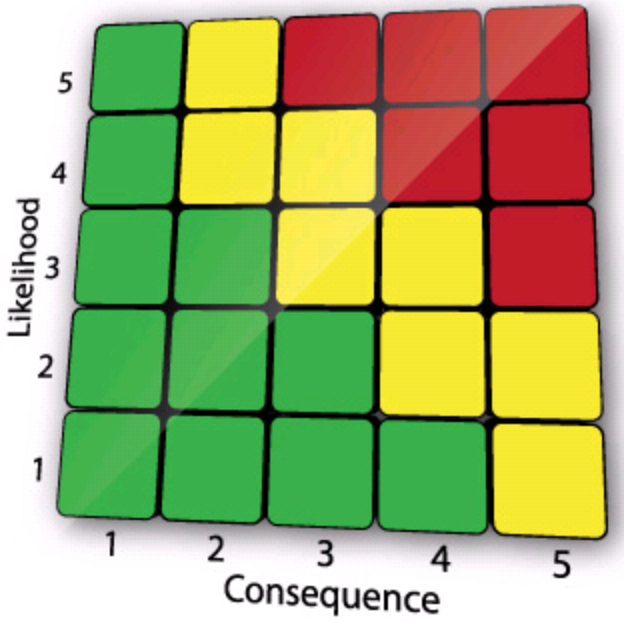
Assess Potential for Growth - Discrete Risks and Opportunities

To a cost analyst "Risk Analysis" means Uncertainty Analysis, but in the context of DoD Acquisition it often means the identification and rating of discrete risks and opportunities using a risk reporting matrix, the so-called "Risk Cube."

It focuses on Schedule and Technical Risk, as previously described, and translates each potential risk item into a consequence of failure (cost impact in dollars) if the risk manifests itself and a probability of occurrence.

The product of these two is the expected value, or "factored" impact, of the risk, the amount by which it increases the overall average estimate. Often, the consequence and probability are translated to a 1-to-5 scale and plotted on the risk reporting matrix, which is divided into regions of high (red), medium (yellow), and low (green) risk. This matrix is shown to the right.

When both probability and consequence are low, risk is low. As these factors increase, risk increases. If probability of occurrence nears one (i.e., the event is certain to happen), the risk becomes an "issue" and should be dealt with outside the risk analysis



Popup Text

Risk Analysis

The activity that examines each identified risk to refine the description of the risk, isolate the cause, and determine the effects in setting risk mitigation priorities. It considers the likelihood of root cause occurrence; identifies possible consequences in terms of performance, schedule, and cost; and identifies the risk level in terms of high (red), medium (yellow), and low (green) on a Risk Reporting Matrix.

Popup Text

Risk Reporting Matrix

A matrix that displays five levels of likelihood versus five levels of consequence with likelihood increasing along the vertical y-axis and consequence increasing along the horizontal x-axis from a common point of origin. Nominally, each level of likelihood/probability of occurrence is defined as follows:

- Level 1: not likely/10 percent
- Level 2: low likelihood/30 percent
- Level 3: likely/50 percent
- Level 4: highly likely/70 percent
- Level 5: near certainty/90 percent

A nominal definition of schedule consequence by level is as follows:

- Level 1: minimal or no impact
- Level 2: able to meet key dates
- Level 3: minor schedule slip
- Level 4: Program Critical Path (CP) affected
- Level 5: cannot meet key program milestones

Definitions of cost or performance consequence levels are devised in a similar manner, depending on the program. The intersection points of the likelihood and consequence levels for future root causes (risk events) are displayed on the Risk Reporting Matrix. For example, a future root cause assessed as Level 1 likelihood/Level 1 consequence would be rated green (low risk), while one rated Level 3 likelihood/Level 3 consequence would be rated yellow (medium risk), and one rated Level 5 likelihood/Level 5 consequence would be rated red (high risk). Assignment of a risk color to future root causes requires judgment in the context of the particular program being assessed.

Assess Potential for Growth - Discrete Risks and Opportunities, Cont.

The Risk Cube approach is geared to risk management, focused on identifying risks and mitigating them (where possible). It is an important source for our cost risk analysis, but be careful not to double-count. Cost analysts tend to prefer continuous ranges for risks instead of a binary it-happens-or-it-doesn't.

For example, if the risk reporting matrix contains a risk item for inability to hire highly-skilled developers as planned, but both the Capability of the development team and the associated composite labor rate are already being modeled as continuous distributions in the Uncertainty Analysis, it would be duplicative to include the discrete risk item separately.

However, the average net impact of the two risks should be cross-checked to make sure they are consistent.

Popup Text

Risk Management

The overarching process that encompasses identification, analysis, mitigation planning, mitigation plan implementation, and tracking of future root causes and their consequences.

Assess Potential for Growth - Indirect Risk

It is important not to overlook Indirect Risk, the phenomenon wherein when the core effort grows, supporting functions tend to grow along with it. In this case, the core effort is generally the requirements-to-test software development activity, and supporting functions include SE/PM, CM, and IV&V. If these below-the-line costs (BTLs) are estimated as related costs using cost-on-cost CERs, then their proportional growth will be properly reflected in the cost model. This is known as Functional Correlation.

For example, if a development effort is estimated at $10M with an additional $2M in SE/PM (estimated as 20%), but the effort grows to $15M, it is not unreasonable to expect that the supporting SE/PM activity will grow to about $3M, which can be viewed either as 20% of the new base effort, or a growth of $1M, which is 20% of the $5M base growth. It is preferable to use the actual cost-on-cost CER, with incipient uncertainty, in the risk model instead of a simple factor, but the most important thing is to be aware of the peril of overlooking indirect risk.

D

Long Description

Two tables containing the following data:

Initial
DE + SE/PM = Total
10M + 20% = 12M

Growth
DE + SE/PM = Total
15M + 20% = 18M

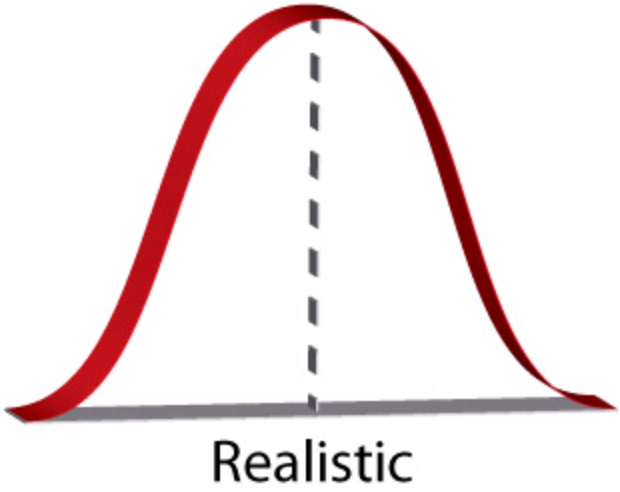
Conduct Uncertainty Analysis

A point estimate for cost or schedule should ideally be a middle-of-the-road number, neither optimistic nor pessimistic, but realistic.

The most useful point estimate reflects a mean, the average expectation of all possible outcomes. Other possible mathematical interpretations are a mode, the "most likely" value where the probability density is highest; or a median, the 50th percentile value for which overruns and underruns are equally likely. (The geometric interpretation of these three are the balancing point of the distribution, the peak of the distribution, and the point that divides the distribution into equal areas, respectively.)

It becomes clear that a definitive interpretation of the point estimate is not possible without describing the range of possible outcomes. This is where uncertainty analysis comes in.

It goes beyond sensitivity analysis by taking into account how all relevant parameters might vary together, and it includes the inherent uncertainty ("plus or minus") of the estimating relationships used.



Questions Managers Should Ask

Popup Text

Questions Managers Should Ask

Are the estimated costs and schedule consistent with demonstrated accomplishments on other projects?

- Has the consistency achieved when applying cost and schedule estimating techniques to historical data been measured and reported?

Have the factors that affect the estimate been identified and explained?

- Are uncertainties in parameter values identified and quantified?

Conduct Uncertainty Analysis - Typical Approach

Though there are other options for Uncertainty Analysis, perhaps the most common approach is Inputs Risk using Monte Carlo Simulation. In this approach, key input parameters are varied just as is done sensitivity analysis, except that:

1. Inputs are varied together, including consideration of correlation, instead of one at a time; and
2. Inputs are varied according to probability distributions, preferably based on historical data, rather than a simple plus-or-minus, or low and high.

When the simulation is run, the program takes a random value for each input or cost element according to its specified distribution. These values are combined using the cost model to find the total system cost. This procedure is repeated as many as 5,000 or 10,000 times.

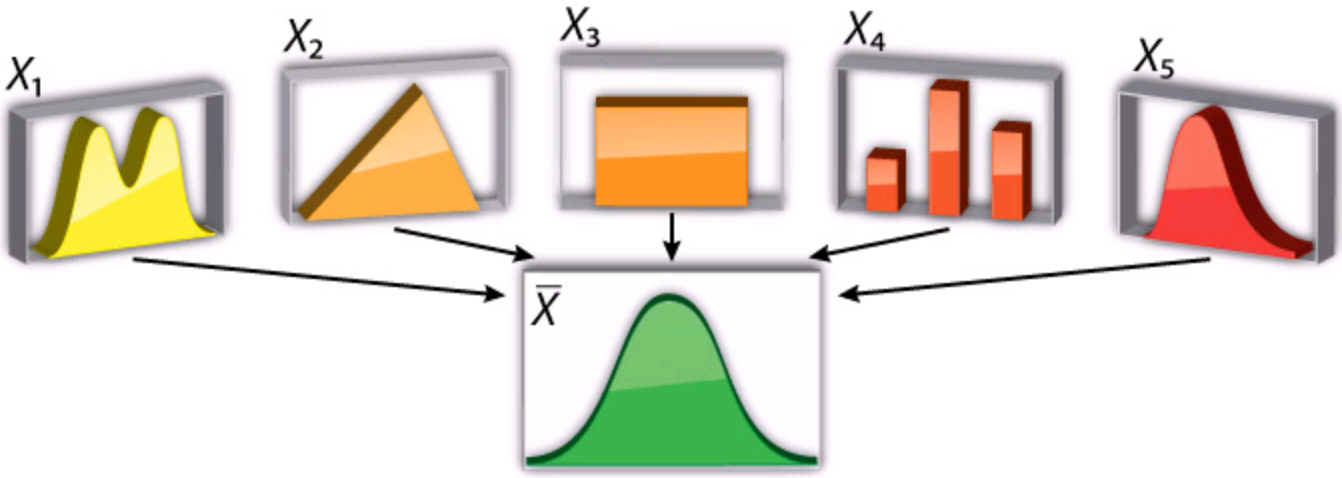
The distribution of the outcomes is an approximation of the distribution of the cost of the total system. The values can be used to calculate statistics for that distribution, such as average cost, standard deviation, and various percentiles of interest, such as the 50th (median) and 80th.

Conduct Uncertainty Analysis - Typical Approach, Cont.

With advances in simulation technology, the Monte Carlo method is cheaper, faster, and more accessible than ever. Its challenge lies in the mathematical sophistication needed to correctly model all the inputs. It allows great flexibility in choice of distributions.

A traditional alternative to Monte Carlo is Method of Moments, which relies on the Central Limit Theorem to calculate the mean and standard deviation of total cost analytically. A simplified version, Symmetric Approximation, uses only symmetric distributions such as normal, uniform, and beta.

Symmetric approximation is amenable to hand calculation and is a good sanity check, but does not easily accommodate correlation, nor does it allow the skew-right distribution of total cost. Another alternative designed to appeal to decision-makers is the Scenario-Based Method (SBM).

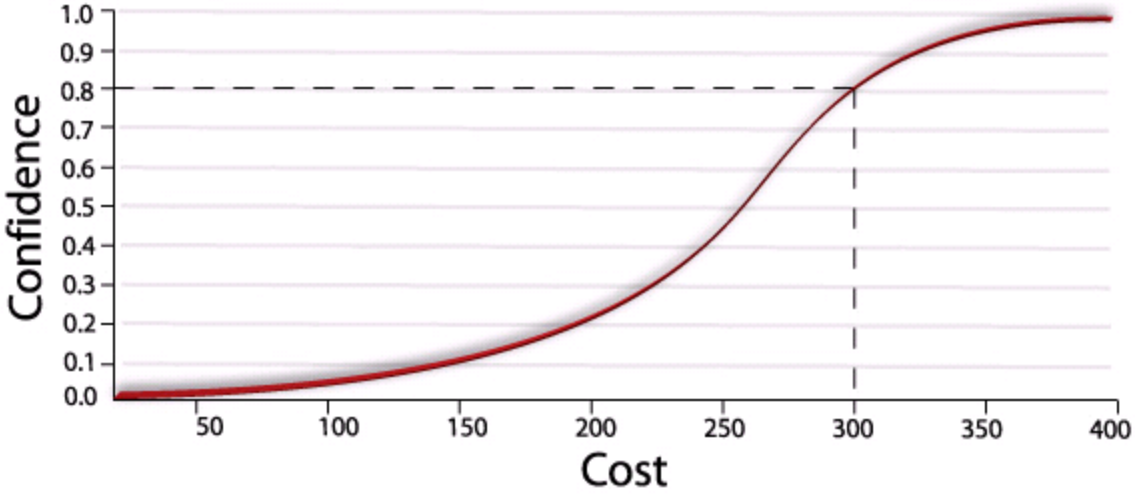


Conduct Uncertainty Analysis - S-Curve and CV

The main benefit of conducting uncertainty analysis on your software cost estimate is that you can give decision-makers answers about the level of confidence that can be associated with it. The key output that enables this is the so-called S-curve, which gives the cumulative probability distribution of cost.

It is crucial that cost analysts both learn to read and interpret S-curve and explain them to decision makers. The range of costs is shown on the x-axis, and the cumulative probability or "confidence" associated with each cost is shown on the y-axis.

For example, if the point ($300M, 0.8) is on the S-curve, it means the probabilistic cost estimate shows there is an 80% chance of final cost coming in at or under $300M. Mathematically, this is the 80th percentile of the estimate, though you may often hear it described as the "80% confidence estimate."



D

Long Description

Graph reflects the example presented in the page content. Graph with x-axis labeled Cost and y-axis labeled Confidence. The S-curve on the graph intersects at Cost of 300 and Confidence level of 0.8.

Conduct Uncertainty Analysis - S-Curve and CV, Cont.

The most common shortcoming of faulty uncertainty analysis is that it produces too narrow a range of outcomes, thereby understating the cost of higher percentiles. The key measure to capture this is the coefficient of variation (CV), which is not to be confused with the cost variance from EVM. It is a unitless measure, usually expressed as a percent, calculated as the standard deviation divided by the mean, and gives a fairly intuitive sense of the "plus or minus" in an estimate.

Software estimates, especially early in acquisition, are expected to have significant uncertainty, so a CV of 40 or 50 percent would not be unreasonable. Where possible, historical benchmarks for CV at various stages of maturity should be used to cross-check the S-curve. (Note that the example is referring to the CV of the distribution of total cost, which is different from but related to the CVs of the individual CERs used to estimate components of that cost.)

In addition to the S-curve for cost (effort), it is good to show an S-curve for schedule (duration) and give confidence levels for achieving a certain schedule. The emerging field of joint cost and schedule risk analysis attempts to capture the probability of simultaneously meeting certain goals for cost and schedule.

$$CV = \frac{\sigma}{\mu}$$

Popup Text

Coefficient of Variation (CV)

The ratio of the standard deviation to the mean or of the sample standard deviation to the sample mean. The coefficient of variation (CV) is a popular measure of the variability of the element or variable because it expresses the range within which (plus or minus one CV) about 68.3% of the data will be found, or, loosely, a 68.3 percent confidence interval. It is not exactly a CI for the sample because it fails to take into account the difference between the sample and the parent distributions, but to consider it to be one will not introduce a great deal of error.

Conduct Uncertainty Analysis - ProRad Uncertainty

The table below illustrates how the Joint ProRad Program Office described their treatment of cost estimating uncertainty in the documentation accompanying their Program Office Estimate (POE). It provides a summary of the distributions and their parameters for evaluating ProRad software cost risk. It lists the expected value (Baseline Cost Estimate) for each major software category being developed by the Joint ProRad Program Office. Click here to view the Cost Estimate Table referred to in the table below.

Cost Category	Distribution	Spread	Skew	Baseline Cost Estimate
Test Software	Triangular	Medium	Center	Not Shown in Case Study
Waveform Software	Triangular	Medium	Center	Cost Estimate Table
Crypto Software	Normal	Medium	Center	Not Shown in Case Study

A triangular distribution was used for test and waveform software based on results provided by the OTS cost model, based on "Least Likely" to "Most Likely" parameter sets in the model's database. The triangular distribution form assumes a cost probability distribution based on the three points: a minimum cost, a most likely cost, and a maximum cost.

These define the distribution's dispersion (spread) and skew. When the max cost is farther away from the most likely that the min cost is from the most likely, we say that the triangle is "skew right," meaning there more potential for growth than reduction. Such a distribution will produce a net positive shift (cost growth) in the estimate.

Click here to learn how Crypto software costs were estimated.

D

Popup Text

Program Office Estimate (POE)

A Component Cost Estimate (CCE) of life cycle costs conducted by an acquisition program office.

Crypto Software cost estimation

Crypto software costs were estimated as: the number of algorithms needed (one per waveform), times the number of host chip designs (for AIM, Sierra and Cornfield brand crypto chips), times $500K per algorithm design. The normal, a symmetric distribution with no skew, was used for the cost probability distribution.

Long Description

Table with the following data:

Cost Category	Distribution	Spread	Skew	Baseline Cost Estimate
Test Software	Triangular	Medium	Center	Not Shown in Case Study
Waveform Software	Triangular	Medium	Center	Cost Estimate Table
Crypto Software	Normal	Medium	Center	Not Shown in Case Study

Knowledge Review

Historical code growth for an organization has been shown to be 20% on average – that is, a growth factor of 1.2 – with a standard deviation of twenty percentage points. If the point estimate for software size of a new development project undertaken by that organization is 10K SLOC, which of the following is not a reasonable distribution to use for the size input to a Monte Carlo uncertainty analysis for cost and schedule?

- [x] Normal, with a mean of 10,000 and a standard deviation of 2,000
- [ ] Normal, with a mean of 12,000 and a standard deviation of 2,000
- [ ] Uniform, between 8,536 and 15,464
- [ ] Triangular, with a min of 7,101, a most likely of 12,000, and a max of 16,899

Check Answer

Normal, with a mean of 10,000 and a standard deviation of 2,000 is not reasonable, because it omits the mean shift associated with expected code growth. The last three choices all have the appropriate mean of 12,000 SLOC (growth factor of 1.2 multiplied by 10K) and standard deviation of 2,000 (0.20 multiplied by 10K).

Knowledge Review

Which of the following is not a common reason for cost growth in software estimates?

- [ ] New requirements added after coding begins
- [ ] Reused software required more redesign than anticipated
- [ ] External interfaces were more difficult than anticipated
- [ ] Early completion deadline forced overtime and led to communication breakdown
- [x] New development tools led to more efficient use of design templates and better configuration management

Check Answer

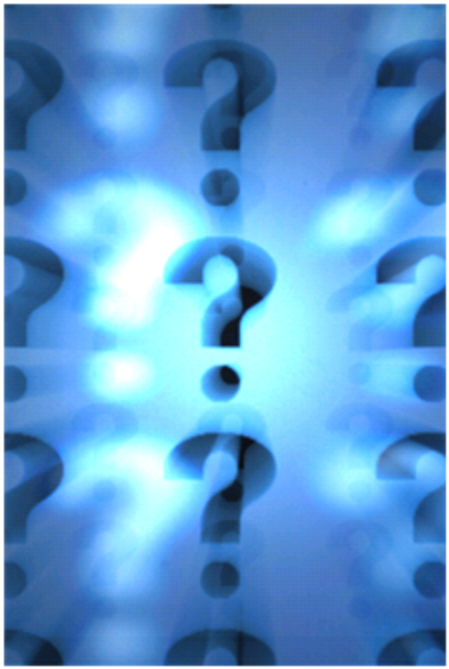
New development tools led to more efficient use of design templates and better configuration management is an opportunity (for cost reduction) not a risk (of cost growth).

Knowledge Review

Which of the following does not lend itself to traditional sensitivity analysis?

- [ ] Reuse efficiency
- [x] Programming language
- [ ] Productivity
- [ ] Software size

Check Answer



Programming language does not lend itself to traditional sensitivity analysis. Sensitivity analysis usually involves varying continuous numerical parameters. Programming language is a discrete, categorical variable (Ada vs. C++ vs. Java, for example).

Summary

This completes the Consider Risk and Uncertainty lesson. In this lesson, you learned:

- Software estimates are very sensitive to key assumptions such as sizing and reuse.
- Software estimates are prey to significant growth due to requirements creep, code growth, optimism regarding reuse and the capability of the development team, and other factors.
- Software estimates are expected to have considerable uncertainty, which should be depicted as an S-curve showing the range of possible outcomes and their associated (cumulative) probabilities.

D

Long Description

Graphic illustrates the steps of the Cost Estimating process. The steps from left to right are:
Develop Scope and Approach, Collect and Analyze Data, Develop Estimate Methodology, Consider Risk and Uncertainty (highlighted), and Document and Present Estimate.

Lesson Completion

You have completed the content for this lesson.

To continue, select another lesson from the Table of Contents on the left.

If you have closed or hidden the Table of Contents, click the Show TOC button at the top in the Atlas navigation bar.