SWE-151 - Cost Estimate Conditions

1. Requirements

3.2.2 The project manager's software cost estimate(s) shall satisfy the following conditions:

- Covers the entire software life cycle.
- Is based on selected project attributes (e.g., assessment of the size, functionality, complexity, criticality, reuse code, modified code, and risk of the software processes and products).
- Is based on the cost implications of the technology to be used and the required maturation of that technology.
- Incorporates risk and uncertainty, including cybersecurity.
- Includes the cost of the required software assurance support.
- Includes other direct costs.

1.1 Notes

In the event of a decision to outsource, it is a best practice that both the acquirer (NASA) and the provider (contractor/subcontractor) be responsible for developing software cost estimates. For any class of software that has significant risk exposure, consider performing at least two cost estimates.

1.2 History

Click here to view the history of this requirement: SWE-151 History

1.2 Applicability Across Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applicable?</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

Key: ✓ - Applicable | ✗ - Not Applicable

2. Rationale

Credible software cost estimates are a key activity performed as part of a project and software life-cycle management (NPR 7150.2, NASA Software Engineering Requirements, Chapter 3). Software cost estimates are required to meet NASA policies and have specific characteristics that make them credible and provide programs and projects with confidence in the cost of work to be performed.

3. Guidance

Credible software cost estimates for NASA are required to meet specific conditions. When preparing cost estimates, the software acquirer and the software provider work together to develop cost estimates that meet the conditions described below.

The software cost estimate(s) and associated estimation data should be documented in a software cost driver sheet (a sample is available in Software Processes Across NASA (SPAN), accessible to NASA users from the SPAN tab in this Handbook) that supports the development of a parametric cost estimate.

Life Cycle Coverage

A credible software project cost estimate covers the entire software life cycle from concept to retirement, including maintenance, support (SWE-015) and associated project attributes such as annual percent lines of code changed. A breakout of cost by life cycle phase and activity is a good estimating practice and cost estimates should be updated to reflect new assumptions.

Project Attributes

Software cost estimates are based on project attributes, including the estimated size of the software, functionality, complexity, criticality, and risk. Other attributes to consider include the use of reused and/or modified code, database size, documentation requirements, and required reusability. Many of these attributes apply to the software being created, the processes used to create and verify that software, and the products created throughout the life cycle such as requirements documentation, architecture, detailed design, test plans, and procedures, etc. (SWE-015).

Technology Implications
Given the need to use the latest technology to achieve its missions, NASA cost estimates take into account the cost implications of the technology to be used and the required maturation of that technology. When developing a software cost estimate, maintain all assumptions and data associated with the technology to be used on the software project, and maintain good data and records associated with any cost-saving identified from the use of software technology. Make sure that the software estimate maintains data showing the cost required for the maturation of any planned use of new technology. New technology can be defined as any new software development environments or tools or software development associated with new or unproven technology.

Risk and Uncertainty

Risk and uncertainty both relate to the same underlying concept – randomness. Risk is quantifiable; uncertainty arises from imperfect knowledge.

When developing a cost estimate, it is important to identify common software risks, assess their impact on the cost estimate, and make revisions to the estimate based on these impacts, e.g., incorporate allowances for uncertainty in sizing and other cost estimation model inputs.

A risk is an event that has the potential to cause a significant impact on technical, cost and/or schedule performance.

Compile known major risks.

As you generate the software risk list, consider the following:

- WBS elements are affected.
- When it would occur.
- Likelihood of occurrence.
- Impact (consequences).
- What it would cost to mitigate the impact.
- Stakeholder and user perspectives.

Risk and uncertainty can be estimated and analyzed in various ways:

- Risk matrices (e.g., 5x5).
- Expected risk (likelihood of impact).
- Monte Carlo methods (Monte Carlo methods use random numbers to obtain numerical solutions when analytical methods are too difficult to use) to produce distributions.

Monte Carlo methods use random numbers to obtain numerical solutions when analytical methods are too difficult to use. When using Monte Carlo methods with cost models, they are used to simulate the estimated cost distribution.

The project team may add the results of the expected risk impact or identified mitigations to the bottom-up estimate. The cost models have Monte Carlo algorithms that produce a cost distribution and the identified risks are used to determine which model inputs should be entered with a wider range.

Software Assurance Support

Support efforts such as software assurance and, when applicable, Independent Verification and Validation (IV&V) also have costs associated with them that are important to include in the software cost estimation. Including software assurance in the cost estimate helps ensure that the project fully understands the software assurance activities that should be performed on the project and provides the appropriate level of funding for those activities. It is important for software assurance and the project to discuss the risks associated with reducing or limiting the software assurance effort on a project which may be caused, in part, by limited funding.

Other Costs

Other factors to consider when developing the cost estimate include:

- Procurement of equipment such as workstations, testbeds, and simulators.
- Procurement or shared costs of tools such as compilers, licenses, and configuration management tools.
- Travel for reviews, customer interfaces, and vendor visits.
- Training.

NASA-specific software cost estimation process information is available in Software Processes Across NASA (SPAN), accessible to NASA users from the SPAN tab in this Handbook.

Additional guidance related to software cost estimates may be found in the following related requirements in this Handbook.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWE-015</td>
<td>Cost Estimation</td>
</tr>
<tr>
<td>SWE-033</td>
<td>Acquisition vs. Development Assessment</td>
</tr>
<tr>
<td>SWE-086</td>
<td>Continuous Risk Management</td>
</tr>
<tr>
<td>SWE-142</td>
<td>Software Cost Repository(ies)</td>
</tr>
</tbody>
</table>

4. Small Projects
The conditions for cost estimates apply to small as well as large tasks but should be scaled to fit the size and scope of the task. All estimates are made using some combination of the basic estimation methods of analogy or Cost Estimating Relationship (CER). Whatever method is used, it is most important that the assumptions and formulas used to be documented to enable more thorough reviews and to make it easier to revise estimates at future dates when assumptions may need to be revised. A documented estimate is the first line of defense against arbitrary budget cuts. Small projects nominally use the analogy estimates methods if the development organization has data to support them. The use of at least two estimates is always considered to be a best practice for small and large software estimation activities.

5. Resources

5.1 References

- **(SWEREF-116)**

- **(SWEREF-136)**

- **(SWEREF-166)**
  Cost Analysis Data Requirement (CADRe), Database of cost metrics for NASA missions: http://www.nasa.gov/offices/ooe/CADRe_ONCE.html#.U0Qc-BC9Z8E. (Accessed December 09, 2014). This URL is a website that lists multiple resources for download.

- **(SWEREF-184)**

- **(SWEREF-306)**

- **(SWEREF-419)**

- **(SWEREF-572)**
  Flight Software Engineering Lessons Public Lessons Learned Entry: 2218.

5.2 Tools

Tools relative to this SWE may be found in the table below. You may wish to reference the Tools Table in this handbook for an evolving list of these and other tools in use at NASA. Note that this table should not be considered all-inclusive, nor is it an endorsement of any particular tool. “Check with your Center to see what tools are available to facilitate compliance with this requirement.”

6. Lessons Learned

6.1 NASA Lessons Learned

The NASA Lessons Learned database contains the following lesson learned related to cost estimation:

- **Flight Software Engineering Lessons. Lesson Number 2218**: The engineering of flight software is a major consideration in establishing JPL project total cost and schedule because every mission requires a significant amount of new software to implement new spacecraft functionality. Constraints to the development and testing of software concurrent to engineering the rest of the flight system have led to flight software errors, including the loss of some missions. The findings of several JPL studies and flight mishap investigations suggest a number of recommendations for mitigating software engineering risk.

6.2 Other Lessons Learned

Other Lessons Learned from experience (Hihn, Jairus M. JPL. 818.354.1248) and the JPL Cost Estimation Handbook available in SPAN include the following concepts:

- Common costs left out of cost estimates.
  - Review preparation.
  - Documentation.
  - Fixing Anomalies and Engineering Change Requests (ECR)’s.
7. Software Assurance

<table>
<thead>
<tr>
<th>SWE-151 - Cost Estimate Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.2 The project manager’s software cost estimate(s) shall satisfy the following conditions:</td>
</tr>
<tr>
<td>a. Covers the entire software life cycle.</td>
</tr>
<tr>
<td>b. Is based on selected project attributes (e.g., assessment of the size, functionality, complexity, criticality, reuse code, modified code, and risk of the software processes and products).</td>
</tr>
<tr>
<td>c. Is based on the cost implications of the technology to be used and the required maturation of that technology.</td>
</tr>
<tr>
<td>d. Incorporates risk and uncertainty, including cybersecurity.</td>
</tr>
<tr>
<td>e. Includes the cost of the required software assurance support.</td>
</tr>
<tr>
<td>f. Includes other direct costs.</td>
</tr>
</tbody>
</table>

7.1 Tasking for Software Assurance

1. Assess the project's software cost estimate(s) to determine if the stated criteria listed in "a" through "f" are satisfied.

7.2 Software Assurance Products

- Assessment of the project's cost estimate, including any issues or risks identified.

7.3 Metrics

- See SWE-15: Planned SA resource allocation versus actual SA resource allocation.

7.4 Guidance

For this requirement, software assurance needs to assess the project cost estimate to determine whether all the conditions stated in the requirement are satisfied:

a. Covers the entire software life cycle.

b. Is based on selected project attributes (e.g., assessment of the size, functionality, complexity, criticality, reuse code, modified code, and risk of the software processes and products).

c. Is based on the cost implications of the technology to be used and the required maturation of that technology.

d. Incorporates risk and uncertainty, including cybersecurity.

e. Includes the cost of the required software assurance support.

f. Includes other direct costs.

a. Verify that the project cost estimate covers all the life cycle phases that the project is participating in from the project conception stage and approval stages, through requirement development and analysis, design, implementation, testing, acceptance, operations, and maintenance through retirement. The life cycle phases should be defined in the project or software development plan.

b. Verify that the attributes considered important for this project have been accounted for in any grassroots estimate, as well as in the inputs to any models used for estimation.

c. If new technology is to be used on this project, verify the cost estimate has been adjusted to account for learning curves, potential immaturity of tools and processes and any other issues that might affect the project's costs and schedules.

d. Verify that any potential risks and uncertainties have been documented and incorporated into the uncertainty factors for the cost.

e. Verify that the software assurance effort and the safety assurance effort, if applicable, have been incorporated into the cost estimate. If these are not included in the models used to estimate the project costs, add a factor of 5 to 6% of the software cost or a grassroots software assurance estimate for this support.

f. Verify that the cost estimate includes other direct costs including the following "typical forgot items":

- Procurements: Might include development and test environments, project management and development tools (CM system, issue tracking system, risks management tool, software diagnostic tools like static analyzers, etc., COTS software, hardware (computers, lab set-ups, etc.)
- Training for the project team as well as assurance personnel, operators, customers
- Travel for meetings, training
• Project activities that are often forgotten: management activities, development, and production of documentation, preparation for reviews, fixing anomalies

NASA-specific software cost estimation process information is available in Software Processes Across NASA (SPAN), accessible to NASA users from the SPAN tab in this Handbook.