What Do You Mean, You Can't Tell Me How Much of My Project Has Been Completed?

Joe Kasser
The Anticipatory Testing Corporation
POB 3419
Silver Spring
MD 20918
Phone (301) 593 3316
E-mail: jkasser@iee.org

ABSTRACT

All the measurements made during the Systems Development Life Cycle do not provide an accurate answer to the question. This paper describes the development of a tool that can provide an answer to the question. In addition, when used with conventional budgetary data may be used to identify management and technical problems in their early stages.

INTRODUCTION

The Software and Systems Development Life Cycle (SDLC) for large systems can take several years to complete. During this time, the:

- Customer makes periodic progress payments to the supplier. In this situation, since the acceptance tests are only made at the end of the SDLC, the suitability of the product for its mission is unknown for the time in which the bulk of the payments are made.
- Supplier provides the customer with information to demonstrate the risk of non compliance with the Statement of work (SOW) is minimal. The information is provided in the form of:
  - **Management** - i.e., budget (estimated and actual), GANTT and PERT Charts, conformance to "best practices".
  - **Intermediate products** - i.e., documents, lines of code produced, defects found, number of requirements satisfied.
  - **Process** - i.e., degree of compliance to the Capability Maturity Model (CMM) and International Organization of Standards (ISO) models.

The intermediate reports are produced to reduce the risk of non delivery and non compliance to the requirements in the SOW. Now in spite of all the measurements being made, the supplier is unable to tell the buyer the exact percentage of completeness of the system under construction anytime during the SDLC. Consequently, according to [Cuppan 1995], of the medium to large software projects executed within the Department of Defense (DoD) in 1995, approximately:

- Eighty percent were 100% over budget, and,
- Ninety percent were at least one year behind schedule.

REQUIREMENTS

A definition of **a completed system is one that meets its requirements**. However, measuring completed requirements does not provide a measurement of completeness of the system for several reasons, including:

- **Nature of the requirements** - different requirements have different complexities, resulting in different implementation times and costs.
- **Changes in requirements over the SDLC** - the customer either does not state the full requirements for a system in the Request for Proposal (RFP), or changes them for various reasons during the SDLC.

The ideal SDLC is shown in Figure 1. A set of requirements for the system is developed based on the need. The implementation phase of the SDLC is then supposed to take place across several milestones until the system is completed.

The real world SDLC shown in Figure 2 is one in which the vision changes during the implementation phase. Consequently, the requirements change. Thus, while the delivered system may meet its original requirements, it recognized that the system will not meet the requirements in effect at the time of delivery. This situation leads to changes in requirements during the implementation phase, which in general are poorly controlled. And, the major consequences of failing to control changes are moving baselines and confusion leading to cost escalation and schedule delays [Kasser, 1994].

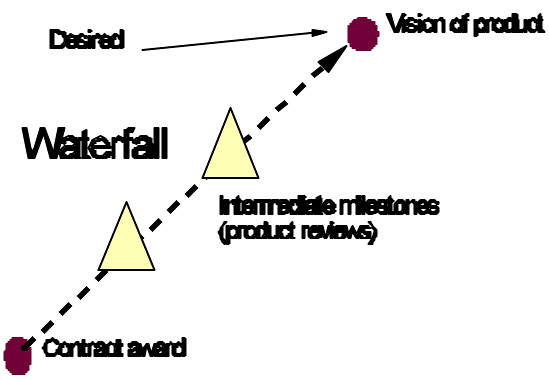

Figure 1The Ideal SDLC

Recognizing that this situation was inevitable, the cataract approach (a series on mini waterfalls) shown in Figure 3 was proposed to control change [Kasser, 1995]. The approach is best implemented using a budget-tolerant SDLC methodology [Denzler and Kasser, 1995] based on the traditional waterfall SDLC model, but with significant enhancements.

The budget tolerant methodology categorized requirements by cost (to implement) and priority. Tracking the implementation of the categorized requirements has led to a measurement approach that has **the potential of providing a measurement of completeness of the product at any of the milestones in the SDLC**. This approach is called *categorized requirements in process* (CRIP).

CATEGORIZED REQUIREMENTS IN PROCESS

The four step CRIP approach is:

C Categorize the requirements.
C Quantify each category into ranges.
C Place each requirement into a range.
C Monitor the differences in the state of each of the requirements at the SDLC reporting milestones

The first part of the approach avoids the problem of comparing requirements of different complexities. **The last step is the key element in the CRIP approach.**

Categories. Typical categories are:

C **Priority** of the requirement to the customer.
C **Complexity** of the requirement., i.e. the difficulty of implementing the requirement.

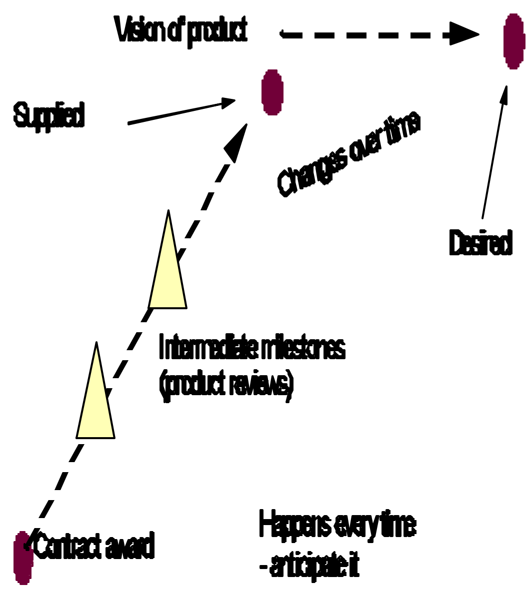

Figure 2 The Actual SDLC

C **Cost to implement** the requirement by the supplier.

Ranges. Each category is split into no more than ten ranges. Thus, for:

- **Priority** - requirements may be allocated priorities between one and 10.

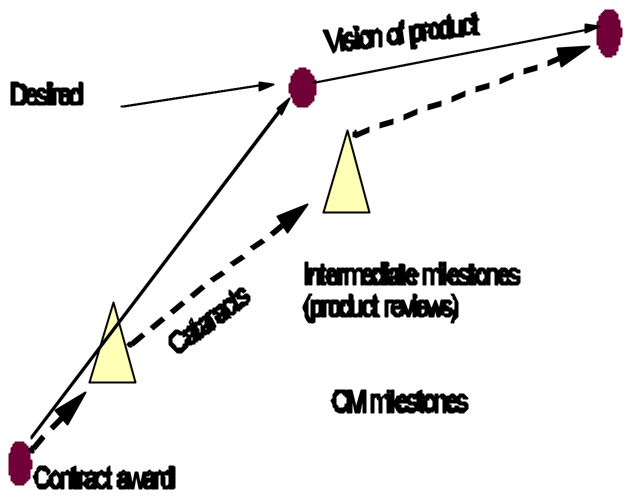

Figure 3 The SDLC - Anticipatory Testing Ideal

- **Complexity** - requirements may be allocated estimated complexities between 'A' and 'J'.
- **Cost to implement** - requirements may be allocated estimated costs to implement values between 'A' and 'J'.

The ranges are relative, not absolute. Any of the several techniques for sorting numbers of requirements into relative ranges may be used. The act of categorizing the requirements into relative ranges is in itself a beneficial act. For example if a low priority requirement has a high cost to implement, the need for the requirement in the system should be reevaluated. Cost may not always be the same as complexity. For example, the use of Commercial-off-the-shelf (COTS) products may lower the cost, but not change the complexity of a requirement.

The buyer and supplier may determine the range limits in each category. A requirement may be moved into a different range as more is learned about its effect on the development during the implementation phase. Thus, the priority of a requirement or the cost to implement may change between SDLC reporting milestones. However, **the rules for setting the range limits must not change during the SDLC.**

States. The state of implementation of each product requirement varies during the SDLC as follows:

- **Identified** - A requirement has been identified, documented and approved.
- **Working** - The supplier has begun work to implement the requirement.
- **Completed** - The supplier has completed work on the requirement.
- **In test** - The supplier has started to test the requirement.
- **Accepted** - The buyer has accepted delivery of part of the system (a Build) containing the implementation of the requirement.

The categories, ranges, and states of each of the requirements are presented in tabular format (a CRIP chart) at reporting milestones (major reviews or monthly progress meetings) as shown in Figure 4 where each cell in a range shown in the Table contains the following three elements:

- **Expected** - The number of requirements planned to be in the implementation state, based on the previous reporting milestone.
- **Actual** - The number of requirements in the implementation state.
- **Planned** - The number of requirements planned to be in the implementation state in the following reporting milestone.

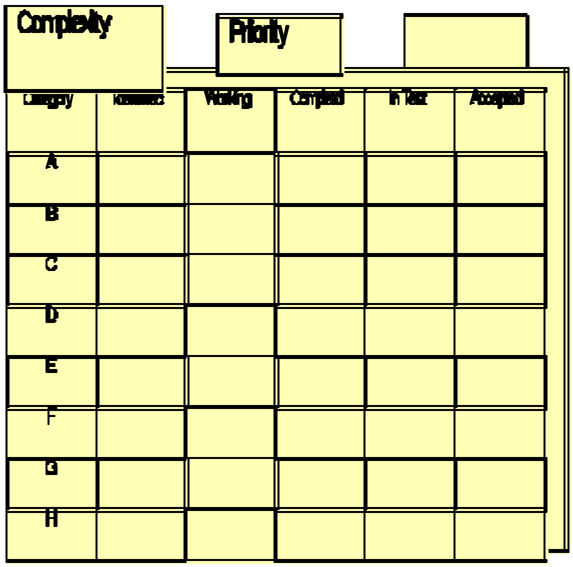

Figure 4 The CRIP Chart

EXAMPLES OF USE

The CRIP chart may be used on a standalone basis or in accordance with budget and schedule information. For example, if there is a change in the number of:

- **Identified requirements** and there is no change in the budget or schedule, there is going to be a problem. For example, if the number of requirements goes up and the budget does not, the risk of failure increases. If the number goes down, and the budget does not, there is a financial problem.

- **Requirements being worked on** and there is no change in the number being tested, there is a potential supplier management or technical problem if this situation is at a major milestone review.

- **Requirements being tested** and there is no change in the number accepted, there may be a problem with the supplier's process.

- **Identified requirements** at each reporting milestone, the project is suffering from poor buyer requirements.

ADVANTAGES OF THE CRIP APPROACH

The advantages of the CRIP approach include:

- May be used at any level of system decomposition.

- Provides a simple way to show progress or the lack of it, at any reporting milestone. Just compare the numbers and ask for an explanation of the variances.

- Provides a window into the project for top management (buyer and supplier) to monitor progress.

- Identifies the probability of some management and technical problems as they occur, allowing proactive risk containment techniques.

- May be built into requirements management, and other computerized project and design management tools.

- May be built into Government contracts via the SOW. Falsifying entries in the CRIP chart to show progress then constitutes fraud.

DISADVANTAGES OF THE CRIP APPROACH

The CRIP approach has the following disadvantages, it:

- Is a different way of viewing project progress.
- Requires categorization of the requirements.
- Requires a process.
- Requires configuration management.

CRIP CHARTS AND CONTRACTOR PAST PERFORMANCE

The CRIP chart numbers at major milestones can provide objective past performance evaluation criteria and force cost effective behavior. Consider the following examples.

Requirements are met or they are not. Waived requirements are not accepted requirements by definition. Hence the process of 'waiving requirements that a supplier cannot meet at the end of a project' shows up in the CRIP chart when the number of accepted requirements at the pre-completion milestone is different from the number of requirements accepted at the completion milestone.

Requirements creep shows up on the CRIP chart in the number of identified requirements at major milestones. If requirements creep is a negative evaluation criterion in a cost plus contract, then it is in the supplier's interest to:

- Identify a full set of requirements as early in the program as possible. The supplier is now motivated to get it right the first time. The underlying information (reasons for the changes) will not be enclosed with the CRIP chart numbers.
- It may be possible to develop a CRIP rating based on the difference between the number of system requirements identified over the SDLC, and the number accepted at the completion of the project and the total cost of the project as a function of the number of categories and ranges within each category. However, this rating will require a "CRIP standard" for future contracts.

If the CRIP charts show that all the buyers' requirements are met, yet the subjective past performance rating is poor, there may be a problem with the buyer's project team. This is something the Agency should investigate.

If requirements can only be tested over time, such as mission effectiveness requirements and failures, the buyer can update the CRIP chart in the past performance database to reflect the status of the requirements after several months of use.

SUMMARY

The CRIP chart approach to measuring progress can provide a more accurate answer to the buyer's question than any other measurement approach in use today. It provides a high degree of visibility of the status of a project in both the buyer and supplier organizations that should discourage poor management in both organizations. However, it still does not guarantee the completeness of the system level requirements.

REFERENCES

Cuppan, C.D., *Capability Maturity Model (CMM) Characteristics and Benefits*, tutorial presentation at the Defense Mapping Agency, 2 June 1995.

Denzler, D.W.R., Kasser, J.E. "Designing Budget Tolerant Systems", *5th Annual International Symposium of the International Council of Systems Engineering (INCOSE)*, St. Louis, MO, 1995.

Kasser, J.E, "Gaining the Competitive Edge Through Effective Systems Engineering." *4th Annual International Symposium of the National Council of Systems Engineering (NCOSE)*, San Jose, CA, 1994.

Kasser, J.E. *Applying Total Quality Management to Systems Engineering*, Artech House, 1995.

BIOGRAPHY

Joe Kasser earned his doctoral degree in systems engineering in 1997. He is a recipient of NASA's Manned Space Flight Awareness (Silver Snoopy) Award for quality and technical excellence. He is also an Institute of Certified Professional Manager's (ICPM's) Certified Manager and a recipient of the ICPM's 1993 Distinguished Service Award. He is the author of *Applying Total Quality Management to Systems Engineering* published by Artech House. His paper "Systems Engineering - Myth or Reality" won the Systems Engineering Management Outstanding Paper Presentation Award at last year's symposium.