Assembling Off-The-Shelf Components: "Learn as You Go" Systems Engineering

Barry M. Horowitz and James H. Lambert
Department of Systems and Information Engineering, University of Virginia

Manuscript in preparation for IEEE *Journal of Systems, Man, and Cybernetics,* Part A

November 2001

Abstract

The process of developing new information systems has evolved from custom software development to assembly of off-the-shelf components. The change has significantly reduced both the costs and time to develop new capabilities, and as a notable result e-business systems have been implemented at a very rapid pace. An assembly sequence (components to be assembled, corresponding dates and costs) has several risks including:

1) Technical risk: successful (or not) function of assembled components by planned schedule milestones.
2) Operational risk: achieving (or not) the desired business value by using the new system of assembled components
3) Programmatic (schedule and cost) risks: accomplishing the assembly within time and budget constraints

As assembly proceeds, estimates of technical performance and operational value at the time of system completion can be adjusted, and one should consider what early milestones of component assembly suggest about later milestones. The technical community can be both hesitant to reveal and uncertain of the results of combining off-the-shelf products into a working system, and it is typical to have significant cost and schedule overruns due to technical problems that are discovered late in system assembly. The operational community can be surprised by the results achieved in applying new capabilities, causing significant changes to what was originally desired from a new system. This paper presents a framework for planning and adjusting milestone sequences in assembling off-the-shelf software components. The framework balances technical and operational risks within established cost and time constraints.

Key Words: system assembly, commercial off-the-shelf software (COTS), risk management, programmatic and technical risk, project management, multiobjective optimization

1. Introduction

Over a period of twenty years, the hardware and software technology for developing information systems has changed drastically. In the 1980's, based on the evolution of computer hardware technologies, the major costs for developing a new system shifted from hardware to software. Accordingly, system engineering methods started looking at "Software First" as the approach for developing new systems. At the same time, in order to reduce cost and to control development risk, off-the-shelf application software started to gain greater acceptance compared to developing customized applications. By the mid-1990's, with the strong desires to develop Internet-based distributed information systems, major costs and risks in developing new capabilities were related to accomplishing system integration. A major increase in demand emerged for off-the-shelf software products for accomplishing integration, called "middleware". However, in order for such products to be fully effective, a corresponding need for integration standards that would be supported by off-the shelf application software products was also required. The dual need was satisfied through the use of such standards as CORBA (Common Object Request Broker Architecture) and EJB(Enterprise Java Beans) and the introduction of numerous integration-focused software products from a number of young software companies (e.g., BEA, Iona, Active, Neon, Tibco, etc.) as well as from traditional suppliers such as IBM and Microsoft. The technology directed at software integration for Internet-based distributed systems was labeled as Enterprise Application Integration (EAI) technology in the 1998 timeframe. With the availability of EAI products, all or almost all aspects of a new system could be created by using already available components, whether they were hardware components, new or existing off-the-shelf software application or EAI components or legacy custom-built application components.

The process of creating a new information system was converted into a process of assembling components into a coherent information system. A significant side-effect of this set of changes was the ability to quickly and economically create rapid operational prototypes which could be utilized to gain experience with using a new information system as the basis for a new concept of operations. With the exploding interest in trying out new Internet-based concepts, and the growing ability to quickly assemble an off-the-shelf system, rapid prototyping became a normal step along the path of creating a new system. The important set of changes in the tools for rapidly creating a new distributed information system and, in turn, the practical ability to use a sequence of operational prototypes along the path of creating a new system, creates a corresponding demand for new system engineering approaches.

Lozinsky (1998), Meyers and Oberndorf (2001), Basili and Boehm (2001), and Alford (2001) describe issues of managing software acquisition. Shields (2001), Curry et al. (1998), and O'Leary (2000) address integration of enterprise software systems. DiUbaldo (2000), Vigder and Dean (2001), and Stavridou (1997) describe COTS in major system acquisitions. Maiden and Ncube (1998), Maiden et al. (1997), Finkelstein et al. (1996), and Rolland (1999) describe selection requirements for COTS. French (1982) describes optimization algorithms for scheduling in systems engineering. Sugimura et al. (2001) describe integrated process planning and scheduling in manufacturing. Schneidewind (1999) gives a cost estimation framework for COTS. Grey (1995) describes risk assessment for project management, including the estimation of cost overrun and time delay.

This paper addresses the need for systems engineering approaches focused on system assembly. The organization of the paper is as follows. The next section contrasts system engineering methods required for development projects with methods required for assembly projects. Next, an approach for more effectively dealing with system assembly projects is presented. Next, an example of applying the suggested approach is presented. Lastly, we provide a summary and recommendations for future effort.

2. Background: System Assembly and System Development

2.1. Overview

In this section, a comparison is made of system engineering for a development project with system engineering for an assembly project. For making this comparison, the role of system engineering is viewed as one of orchestrating and harmonizing the numerous efforts involved in creating a new information system.

A development project is defined as one whose technical risks and costs are dominated by the need for developing new customized software. An assembly project is defined as one where all, or almost all, of the software components required for creating the desired new system are available through purchase or as already existing legacy packages. As a result, the major technical risks for an assembly project are related to the ability: 1) to select appropriate software packages for accomplishing the desired system functions and, 2) to successfully integrate the selected software that the system will be comprised of.

In the case of both the development and the assembly project there is a risk that the operational capability to be supported by the new information system will not be as effective as hoped for. This risk can be due to either user interface issues or an inaccurate forecast of the value that will result through use of the new system.

2.2. System Engineering for a Development Project

Some of the key system engineering processes involved in orchestrating and harmonizing a development project are:

1. Establishing overall system development requirements. This includes determining the set of components that together will define the system to be developed, as well as establishing the division of component design and development work that will result in the creation of the desired system. This step includes technology evaluation and systems analysis to assure that the definition of the individual components both adequately serves the overall needs, and results in technically feasible and practical components to be developed (This is the initial Orchestration product of system engineering);

2. Determining the design requirements for the individual components to meet so that, when integrated, the entire system will meet its requirements. (This is the harmonizing role of the system engineering process). This step permits separate component development teams to create components for later integration;

3. Freezing individual component requirements as early as possible. This is critical in order to avoid expensive ripple effects that cause changes to component requirements after design and development are already underway;

4. Continuing analysis and test of components as they are developed, to either confirm that component requirements will be met or that it will be necessary to make changes. The system engineering efforts related to component requirements are based on analysis, simulation, and test and the freedom to control design;

5. Should changes be required, determining which components must be changed as a consequence of any other given component's failure to meet its requirements (This is the re-orchestration role of system engineering). The efforts to reallocate component requirements once a given component is discovered as being unable to meet its requirements are based on the ability to evaluate the economics of alternative reallocations. Performance of this economic analysis assumes full visibility into component designs and the state of progress on development for all components; and

6. Integration testing and component modifications as determined from the test results. This is the worst time to establish the need for changes since the amount of required rework could be large.

Some of the system engineering processes used to deal with operational risks are:

1. Early demonstrations and real-time simulations of the human interaction design for the new system;

2. Value analysis and operational simulations to improve the understanding and confidence in the ultimate value to be expected from using the new system in a new operational configuration; and

3. Establishing a sequence of sub-system deliveries that permit operational use prior to the entire system being available. The timing of these deliveries is driven by the design, development and test time for components, as well as the required time for sub-system integration testing. Typically, the flexibility to choose the sequences that may be of greatest interest and also be available at an early time is very limited. This is due to the fact that the most difficult aspects of development are usually driven by the assumption that they will create the greatest value. In

turn, since they are the most difficult, the ability to get an early viewing is usually not feasible.

Based on many years of experience with information system development, two important observations are made related to development projects for later comparison with system assembly projects:

1. The resources applied to system engineering in a large development project are typically between 10 and 15% of the total development costs; and

2. The development time required for providing early operational subsystems is usually between fifteen and eighteen months and the choice of early deliverable subsystems, as indicated earlier, is usually not as desirable as one would like it to be.

2.3. System Engineering for an Assembly Project

The key system engineering steps in carrying out a system assembly project are quite different than for a development project. Since there is little development required, the efforts are focused on:

1. Selection of available software based on functional capability and interoperability with other packages, including legacy custom software,

2. Integration of diverse software packages using EAI technology and integration standards which must be selected, and

3. Carrying out value analysis with the ability to conduct operational experiments using operational prototypes as a key ingredient of the effort.

Unlike the development project the major costs are in selecting, buying and integrating hardware and software and based on experience, systems engineering consumes 25%-40% of total project costs. Since the overall project costs have been reduced significantly by the use of off-the-shelf application software and EAI technology, the overall systems engineering effort is smaller than for a development project, but larger in the percentage it consumes. Since there is no control of, or visibility into, existing vendor provided software packages, the analysis oriented systems engineering efforts on a development project are replaced with comparing vendor stated specifications and carrying out test evaluations of the potential system components. Some very significant consequences of the lack of ability to perform design analysis include the limited ability to predict before testing:

1. System integration problems,
2. System capacity,
3. System performance,
4. System failure modes and responses, and
5. System security shortfalls.

The risk of significant problems occurring in these areas is reduced if the components being integrated have been combined before to perform the same or a similar function. Having an experienced information technology staff to evaluate this situation provides a good start. However, the ability to judge whether a similar function has concealed a problem that the desired new system will reveal remains an open area of risk. The system engineering approach for dealing with this risk is to first establish where the greatest integration risks may lie, and then sequence integrations of components to reveal in a timely manner the extent of the problems that actually emerge. Should a problem emerge, there are a limited set of responses, since the design and development of the components is not under control. The available responses include:

1. Requesting support from the vendors, such as changing their design in a future release. If a significant relationship exists, vendors will be responsive if the request is a practical one that other customers may benefit from;
2. Selecting a different product. This is often difficult to do, since the original choice has certain benefits that provoked its selection to start with and there is a sunk cost in licenses and training to overcome. Nonetheless, if the problem is significant, this choice may be necessary;
3. Working around the problem by modifying the intended capability to one that is close to equally acceptable; and
4. If the problem is catastrophic and nor readily correctable, carrying out focused custom software development or, in the worst case, canceling the project.

The choices for sequencing the integration of components are large in number because the components are already developed. With a wide range of sequencing possibilities, the selection of the most desirable sequence for integration and evaluation becomes a critical systems engineering function. This is quite different from a development project where sequence is to a great extent controlled by the time for developing components.

As indicated earlier on in this section, in addition to technical risks, operational risks must also be considered. In an assembly project, since the components are available at the start of the project, there is a great deal of flexibility in using operational prototypes as the basis for evaluation. However, selecting the sequence for providing subsystems that permit operational evaluations can be difficult. This is due to different objectives that the operational evaluations can have. Some examples are:

1. Gain organizational support by demonstrating how quickly, how risk free and how inexpensively an initial capability can be brought into operation (frequently referred to as picking the low hanging fruit),
2. Start down the path of a natural sequence of evolutionary operational capabilities that the using organization can most readily absorb, eventually leading to a desired end result, independent of the technical risk or operational value entailed in each step along the way (organizationally driven sequence), and

3. Conduct early evaluations of some questionable operational capabilities that could result in a very high return in value.

In all cases, experience gained during early use will bring with it a set of desired changes to the objective system. These changes can make the technical parts of the system assembly either more or less complex. In any case, the ability to choose the type of operational objective to be achieved with a prototype is relatively unconstrained when compared with a development project. From experience, the typical time required to establish an initial operational prototype tends to be between six and nine months.

In the material presented in this section, there were two fundamental reasons provided for operational prototyping. One was to sort out and hopefully reduce technical risks in an operational setting. The other was to sort out operational risks in terms of expected value. The sequence that might best serve one objective, might be a poor candidate for serving the other. For example, the picking of low hanging fruit for gaining operational momentum purposely avoids technical challenges.

The fact that the objectives for selecting a sequence can be quite different is accentuated by the fact that the people working on the technical aspects of a project tend to have a very different value system than those who are paying for the system and may not understand the technical risks as well as they do the operational risks. It is not uncommon for the technical people to be cautious about revealing gaps in their knowledge about the ability to successfully integrate existing software packages because they are fearful that operational people might over-react to the degree of risk this imposes on the project. A possible risk to the technical team is that this over-reaction could lead to a decision of not funding an effort. In practice, this fear of revealing risk can eliminate technical risk reduction as a significant factor in the selection of the sequence of integration and operation. As a result, experience indicates that more frequently than not, significant technical problems occur on assembly projects in an untimely and unexpected fashion.

3. Example: Wireless E-Commerce Services

Suppose an e-commerce company that is already providing customers with the ability to purchase retail items over the web decides to also support customers using wireless PDA's. The company concept might be to provide three different services to customers who are suitably equipped:

1. A service to compare brands and prices while shopping. This would be a low cost subscription service built on the company's current customer base and used to establish a new customer base for expanded wireless services. The customer, upon seeing an item of interest for purchase at a store, would access the new service to discover potential alternative sources at a lower price as well as alternative brands that might be more desirable.
2. A service to locate the nearest alternate locations for purchasing the originally identified product at a better price or a desired alternate product. The consumer would be required to have a GPS position location capability with the wireless device in

order for the company to provide this service. This service would be provided at the expense of a small increase to the basic subscription price, and would principally be viewed by the company as a necessary accommodation to those customers equipped with GPS.

3. A service where the customer could purchase the desired product from the company on the spot, using the wireless system. The company would hope to offer lower prices and wider selection, with quick delivery. The company would earn income from this service by virtue of the margins between its cost for the products and the customer's price for buying the product. In order to receive this service the customer would need to have a secure wireless configuration. This service would be expected to be the major revenue source from wireless services.

In order for the company to support the wireless services it must modify its existing e-commerce system. Figure 1 shows a system configuration that supports the new requirements.

The WAP (Wireless Access Protocol) Gateway is the interface between the company's e-commerce system and the wireless user population. It converts the wireless WAP protocol into the standard Internet http protocol for communicating over a wired infrastructure. This is necessary due to wireless bandwidth limitations. The Gateway also handles security from the wireless devices using the standard wireless security protocol WTLS, and conducts corresponding secure interactions with the company's e-commerce system using TLS, the Internet standard. The Gateway services will be obtained from a service provider, of which there are many.

The company's Web Server must be augmented to interact with the wireless user population. The communications from the Web Server are to be encoded using WML, the wireless mark-up language counterpart to the HTML standard for the Internet. The produced web pages must be usable on the small screen configurations of PDA's.

The company's DBMS must be augmented to identify wireless users so that records can be analyzed to determine the returns from the wireless initiatives. New records on price versus store must be established to provide users with the described Comparison Shopping services. The DBMS must also be modified to include physical locations of stores in order to support the GPS-based service described above.

The company's OLAP (On line Analytical Processing) package must be modified to provide needed business reports on the success of the wireless activities.

The technical work required to create the desired system can be divided into three tasks.

1. Task A – Provide the basic Comparison Shopping Service. This includes integration of the company's e-commerce system with the WAP Gateway, modifying the DBMS and OLAP systems for Comparison Shopping and creating the desired wireless web page capabilities on the Web Server.
2. Task B – Add the GPS-based services on top of the Comparison Shopping service. This requires added changes to the DBMS, OLAP and Web Server, including the ability to locate the nearest alternative stores to the current user's location and support

for the presentation arrangement that provides the new information to the user. It is assumed that B must follow A.

3. Task C – Provide the on-line purchasing service. While this service does require the WAP Gateway and the WML-based interactions through the Web Server, it does not require many of the capabilities that the comparison shopping service needs (e.g., identification of alternate brands or alternate stores). However, it does require many new features such as support of secure sessions, support of financial transactions, coupling to the DBMS in the areas related to delivery of purchased goods, etc.

The company wants to consider two alternative strategies for introducing the new service. One strategy is to start with the Comparison Shopping service, build a user base and then add the Purchasing service. The second strategy is to introduce the purchasing service first and expand to the Comparison Shopping services as a way to get more customers for the purchasing service (since most users may not have the security capability options with their PDA).

Strategy 1 – A, AB, ABC;
Strategy 2 – C, AC, ABC.

The company's entry into providing wireless services comes with many risks and uncertainties. Technical risks include:

1. The wide variation of PDA products includes differences in their presentation software, resulting in different outputs for the same Web Server provided input. This variation could include cases of unacceptable presentations. In addition, the input capabilities for different devices could make certain interactions unacceptable for a given set of users. The time to discover and correct potential problems could be significant.
2. The security software for wireless devices is immature and could cause a variety of security and interoperability problems, causing schedule delays and assembly cost increases.
3. The team of company people that have the complete knowledge of the existing DBMS and OLAP systems are involved with current operations, and may not be available when needed for creating the necessary new features.

Operational risks include:

1. The rate of people purchasing wireless PDA's is uncertain, potentially reducing (or increasing) the hoped for user population to draw customers from.
2. The subscription services may not be appealing enough to gather customers at the hoped for rate.
3. The security of the purchase service may be questionable to people based on prior wireless experiences, resulting in reduced demand.

Given the opportunity, the risks and the uncertainties, the Company's management team decides to establish a project to be completed in about 1 year at a cost of about $1 million. They will look at sales of the full set of services over a six month period as the basis for evaluating the merit of funding additional wireless services. The Company's System Engineering team is asked to look at the two possible strategies indicated above and to organize a set of information that will provide the basis for selecting the strategy for implementation.

4. Methodology for Sequencing of System Assembly

System engineering methods are needed to help deal with the problem that is typified by the above example: selecting sequences of assembly and operational prototypes to account for operational value and technical risk among other factors. This section provides an analytical and graphical framework for the system engineering problem of sequencing an assembly project.

Consider the assembly of off-the-shelf components *A, B,* and *C*. Each member of {*A, B, C*} is itself an assembly. *AB* denotes the assembly of components *A* and *B*. Figure 2 describes Plans 1 and 2 of the example introduced above: two alternative sequences of assembly of components A,B, and C. Let ω denote a completed assembly ABC of all components {A, B, C} under consideration. The status of an assembly is given by dashed, thin, and bold segments, representing *preliminary integration*, *final integration*, and *operational status*, respectively, as are defined in Table 1.

Define the *operational value o(i)* to be the planned operational value (e.g., sales revenue, customers, number of purchases) of the assembly *i*. The operational value *o*(*i*) is the planned value of the assembly *i*. Thus *o* represents the evolution of the operational value through an assembly sequence. Intermediate assemblies (e.g., AB, AC) with higher operational value *o* are most predictive that the completed assembly will provide its planned operational value. Uncertainty in the planned operational value *o* can be characterized by optimistic o_u, pessimistic o_l, and nominal o_n estimates.

Define the cumulative, or sunk, *cost c(i)* to be the planned cost of the assembly *i*, and *s(i)* to be the planned time of completion of the assembly *i*.

Define the *cost risk* $s_c(i)$ to be the semi-standard deviation of the cost of the completed assembly given the on-plan completion of an intermediate assembly *i*. The semi-standard deviation $\sigma_c(i)$ (Kim and Wallace 1998; Fishburn 1977) of the estimate of final cost is a measure of the possibility of the final cost $C(\omega)$ exceeding a planned (target) level $c(\omega)$ as follows

$$\sigma_c^2\,(i) = E\,[\max(0, (C(\omega;\, i) - c(\omega))^2\,]$$

where E[•] denotes the expectation operator, ω denotes the completed assembly, and $C(\omega; i)$ denotes the random variable of *sunk cost* at the completed assembly given the on-plan completion of the intermediate assembly *i*. Other measures of risk than the semi-standard deviation can be substituted, including other conditional moments and "worst-case" estimates.

Similarly, define the *schedule risk* $s_t(i)$ to be the semi-standard deviation of the time (measured from the initiation of assembly) of completion of the full assembly given the on-plan completion of the intermediate assembly *i*.

The cost and schedule risks $s_c(i)$ and $s_t(i)$ diminish through the execution of a plan until achievement of the completed assembly ω, when $s_t(\omega) = 0$ and $s_c(\omega) = 0$.

Table 2 summarizes the parameters used in specifying an assembly plan *k*.

Tables 3 and 4 describe Plans 1 and 2 for the wireless-deployment example. The data are obtained by expert elicitation, the details of which are not a topic for the current paper.

Figures 3 and 4 describe the intermediate and completed assemblies of the Plans 1 and 2 over time in terms of *cost risk*, *schedule risk*, *operational value*, and sunk *cost*. Figure 5 juxtaposes Plans 1 and 2 for purpose of comparison. Notice that the operational value of intermediate milestones is greater for Plan 2, but that the operational value of the completed assembly ABC is higher for Plan 1 than for Plan 2. Thus there exists a tradeoff between Plans 1 and 2 in terms of intermediate and completed operational values.

Figure 6 describes the sunk costs and ratio of sunk costs over time and shows that the sunk cost of Plan 1 alternately leads and lags that of Plan 2 before and after six months. Plans 1 outspends Plan 2 until month 6, when Plan 2 begins to require about 25% more cumulative investment relative to Plan 1. The cumulative sunk costs of the two plans are equal after twelve months.

Figure 7 describes the uncertainty in the planned operational values of the two plans over time. Plan 1 lags Plan 2 in all of the pessimistic, nominal, and optimistic forecasts of operational value until the completion of the full assembly. At the completion of the assembly, Plan 1 dominates Plan 2 in pessimistic, nominal, and optimistic forecasts of operational value. However, the full assembly of Plan 1 lags that of Plan 2 by a few months. One must wait longer to realize the higher ultimate operational value that is anticipated under Plan 1. Another perspective is that the operational value of Plan 2 is less predictable at the start of assembly, but more predictable by the end of assembly.

Figure 8 indicates that the cost risk is reduced more by Plan 2 than by Plan 1 as sunk cost grows.

Figure 9 indicates that schedule risk is reduced more by Plan 2 than by Plan 1 as sunk cost grows.

Figure 10 shows (i) the cost-risk reduction efficiency (which is defined as the ratio of cumulative reduction in cost risk to cumulative cost) over 18 months; and (ii) the ratio of risk-reduction efficiencies of the two plans. Notice that Plan 2 is more efficient at reducing *cost risk* throughout the twelve months of assembly.

Figure 11 shows (i) the schedule-risk reduction efficiency (the ratio of cumulative reduction in schedule risk to cumulative cost) over 18 months; and (ii) the ratio of risk-reduction efficiencies of the two plans. Notice that Plan 1 is providing more schedule risk reduction per dollar spent early in the assembly, particular by months 6 to 9. While Plan 1 is more efficient at reducing schedule risk, Plan 1 initially exhibited greater schedule risk.

Figure 12 demonstrates that replanning can occur before the completion of an assembly. Plan 1 was adopted at the start of assembly. By the completion of assembly A at six months, it was clear that more operational value had been achieved at less sunk cost than had been planned. Plans 3 and 4 are then considered in replanning from the intermediate milestone A. Plan 3 aggressively continues to spend sunk cost (in a sequence AD to ACD) to grow the operational value dramatically exceeding the value that was anticipated under the old Plan 1. Plan 4 proceeds conservatively by expending little or no sunk cost to reach the next milestone AB. By completion of the assembly ABC under Plan 4, there is less sunk cost and less operational value than under Plan 3, but the operational value expected under Plan 4 is equal to the operational value that was expected under the old Plan 1. Plans 1, 3, and 4 all expect to achieve a full assembly (either ABC or ACD) by twelve months. As the assembly proceeds, an adjustment from the planned sequence may be justified by knowledge gained at an intermediate milestone, or assembly may proceed as in an earlier plan.

The construction of Figures 3 through 12 implies a multi-stage, multiobjective assembly process in which the choice among planning and replanning options entails the minimization of cumulative costs, costs-to-go and cost risks, schedule time-to-go and schedule risks, and cost- and schedule-risk reduction efficiencies; and the maximization of operational values under uncertainty. Decision theory and engineering economic analysis could be applied to the computation of a numerically optimal plan at any planning milestone, e.g., the choice of a plan that maximizes an expected net present value and/or expected value of a multiattribute utility function of operational value and cost. The above methodology can support a calibration of numerically optimal planning against experience that will address additional decisional criteria. Such additional criteria include: reassurance of investors at intermediate planning milestones, variable rates of credit for capital and operating expenses, value of innovation in an assembly sequence, and the distribution of risks across multiple assembly projects.

5. Conclusions and Future Work

A methodology to aid in the comparison of alternative sequences of assembly has been developed and demonstrated in a realistic example. The methodology is based on experience and straightforward analysis and is promising in its ability to address real dimensions of the assembly-planning problem. The methodology needs to be tested on problems of larger scale in order to ascertain its scalability and ability to address additional complexity. A case study will be useful to assess the success of project teams that use the methodology relative to teams that proceed in other approaches. Moreover, a case study will provide a testbed for iterative replanning, at intermediate milestones, as the assembly proceeds. The case study will enable the refinement of methods for expert elicitation of the needed data, in particular the forecasts of operational value and risks of cost and schedule.

In addition, the elicitation of expert evidence must evolve as planners learn from experience in the assembly process. The evidence of experienced engineers must be treated differently from that of less experienced engineers, requiring the ability to evaluate and determine the relationship between past experience and current project risks. Such an evaluation must be of suitable detail so as not to give too much credit for experience that is generally related, but is not going to have specific value. For example, Beroggi and Wallace (2000) apply multiattribute utility theory to address situations of multiple risk managers in project management.

6. Acknowledgments

The authors appreciate the support of colleagues and students in the Department of Systems and Information Engineering at the University of Virginia.

References

Alford, Lionel D., Jr., *The Problem with Aviation COTS* IEEE Aerospace and Electronics Systems Magazine 16 (2) Feb. 2001 pp. 33-37.

Basili, Victor R. and Barry Boehm, *COTS-Based Systems Top 10 List* Computer 34 (5) May 2001 pp. 91-95.

Beroggi, G.E.G. and W.A. Wallace 2000. Multi-expert operational risk management. *IEEE Transactions on Systems, Man, and Cybernetics—Part C*. 30(1): 32-44.

Curry, Patrick A., Ray Morgan, Brenda Jones, Steven Onley and Norman Casto, *COTS Contribution to the ERS Success Story* AUTOTESTCON '98. IEEE Systems Readiness Technology Conference 24-27 Aug. 1998 pp. 198-203.

DiUbaldo, John A., *NASA Earth Observing System Mission Operations Center Development Using COTS Products* IEEE Aerospace Conference Proceeding 18-25 vol.2 Mar. 2000 pp. 243-247.

Finkelstein, G. Spanoudakis and M. Ryan, *Software Package Requirements and Procurement, Proceedings of the Eighth International Workshop on Software Specification and Design*, IEEE Computer Society Press, Washington DC 1996 pp. 141-145

Fishburn, P.C. 1977. Mean-risk analysis with risk associated with below-target returns. The American Economic Review. 67.

French, S. 1982. *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*. New York: John Wiley and Sons. 245 pp.

Grey. S. 1995. *Practical Risk Assessment for Project Management*. New York: John Wiley and Sons. 140 pp.

Kim, J. and D. Wallace 1998. Mean-semi-variance analysis: risk and opportunity. Manuscript submitted to the 10th International Conference on Design Theory and Methodology.

Lozinsky, Sergio. *Enterprise Wide Software Solution: Integration Strategies & Practices, 1st edition* Addison-Wesley Pub Co (Jan. 1998)

Maiden, Neil A. and Cornelius Ncube, *Acquiring COTS Software Selection Requirements* IEEE Software 15 (2) March-April 1998 pp. 46-56.

Maiden, Neil A.M., C. Ncube, and A. Moore, *Lessons Learned During Requirements Acquisition for COTS Systems* Communications of the ACM 40 (12) Dec. 1997 pp. 21-25.

Meyers, Craig and Patrick Oberndorf, *Managing Software Acquisition: Open Systems and COTS Products, 1st edition* Addison Wesley Professional (Jun. 2001)

O'Leary, Edmund Daniel. *Enterprise Resource Planning Systems: Systems, Life Cycle, Electronic Commerce, and Risk* Cambridge Univ. Pr (Aug. 2000)

Rolland, C., *Requirements Engineering for COTS based systems* Information and Software Technology 41 Nov. 1999 pp. 985-990.

Schneidewind, Norman F., *Cost Framework for COTS Evaluation* Computer Software and Applications Conference. Proceedings. The Twenty-Third Annual International 27-29 Oct. 1999 pp. 100-101.

Shields, Murrell G.. *E-Business and ERP: Rapid Implementation and Project Planning, 1st edition* John Wiley & Sons (Apr. 2001)

Stavridou, V., COTS, *Integration and Critical Systems* IEE Digest, 97/013 Jan. 1997.

Sugimura, N.; Hino, R.; Moriwaki, T. Integrated process planning and scheduling in holonic manufacturing systems, Proceedings of the IEEE International Symposium on Assembly and Task Planning, 2001, pp. 250 -255

Vigder, Mark R. and John Dean, *System Implementation Using Commercial off-the-shelf Software* NRC-CNRC Software Engineering 2001 http://wwwsel.iit.nrc.ca/projects/cots/COTSpg.html.

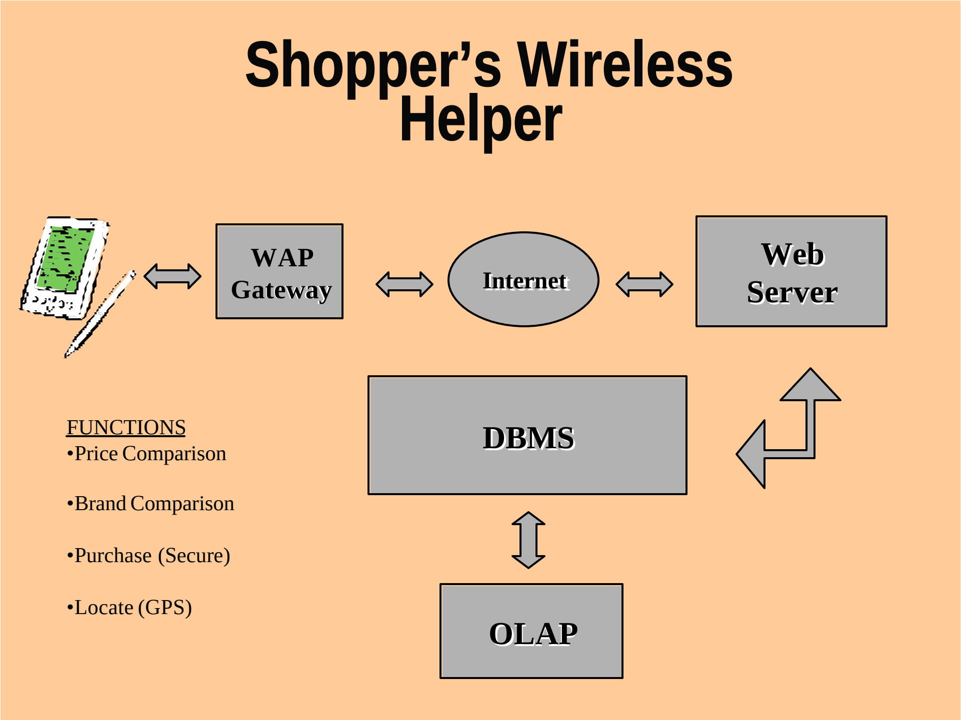

Figure 1. Example of system assembly for providing wireless services

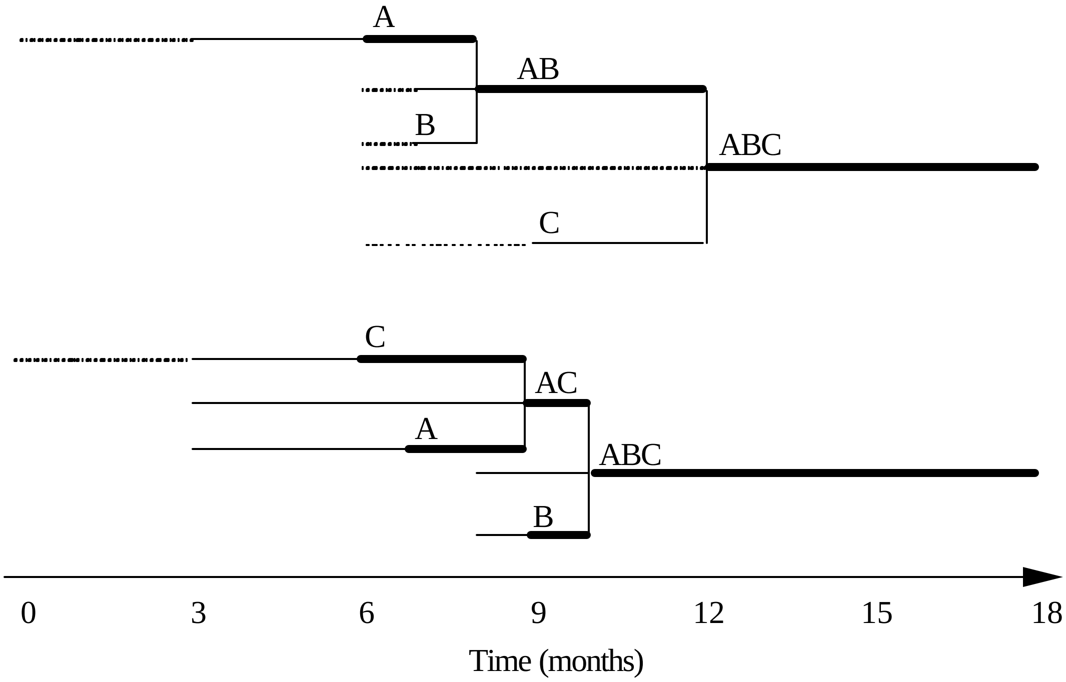

Figure 2. Assembly sequence for Plan 1 (top) and Plan 2 (bottom); thick lines represent operational capability and thin lines represent integration (in the case of assemblies) and development (in the case of single components)

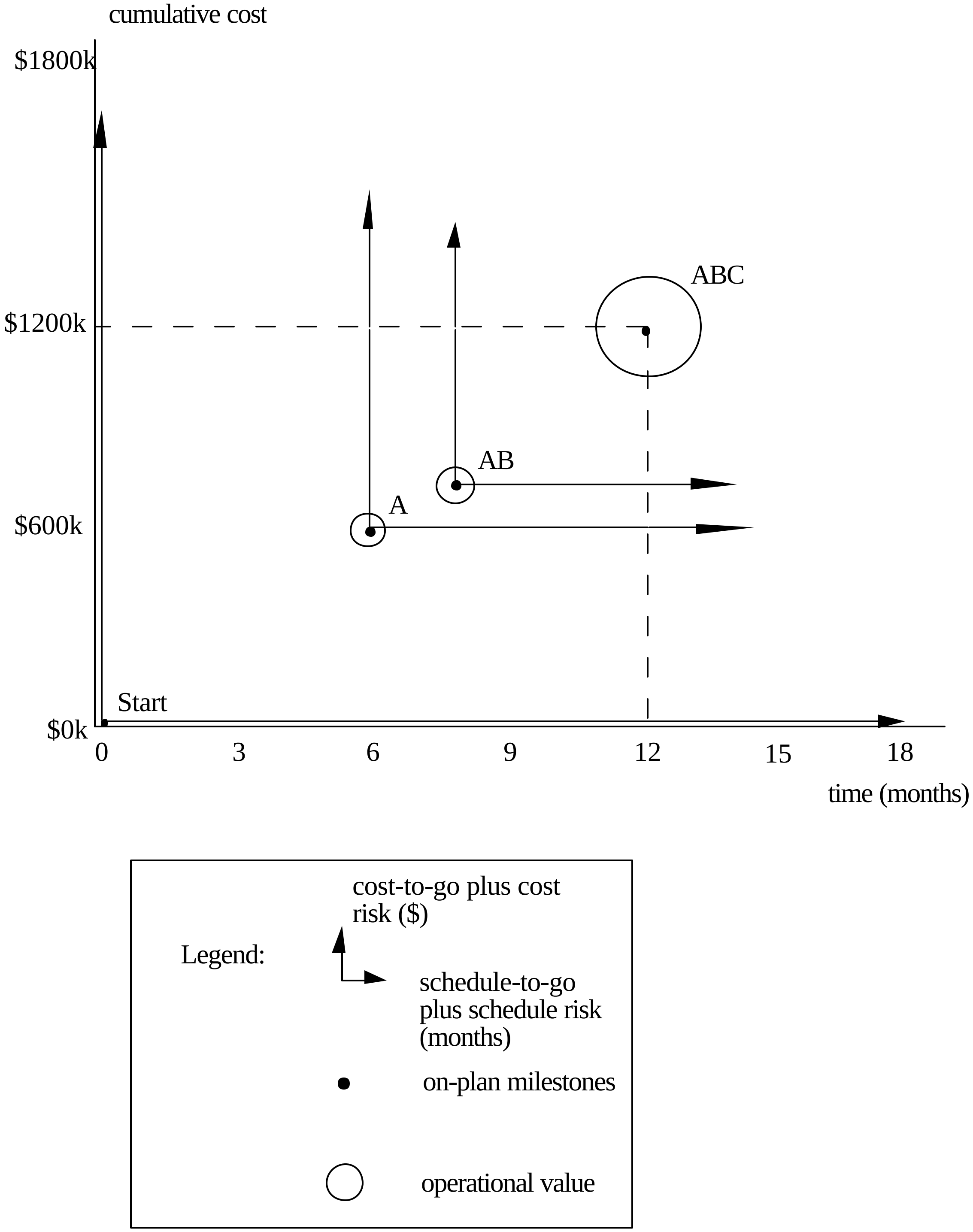

Figure 3. Plan 1 cumulative cost, cost risk, schedule risk, and operational value (lower, upper, and nominal estimates) over time

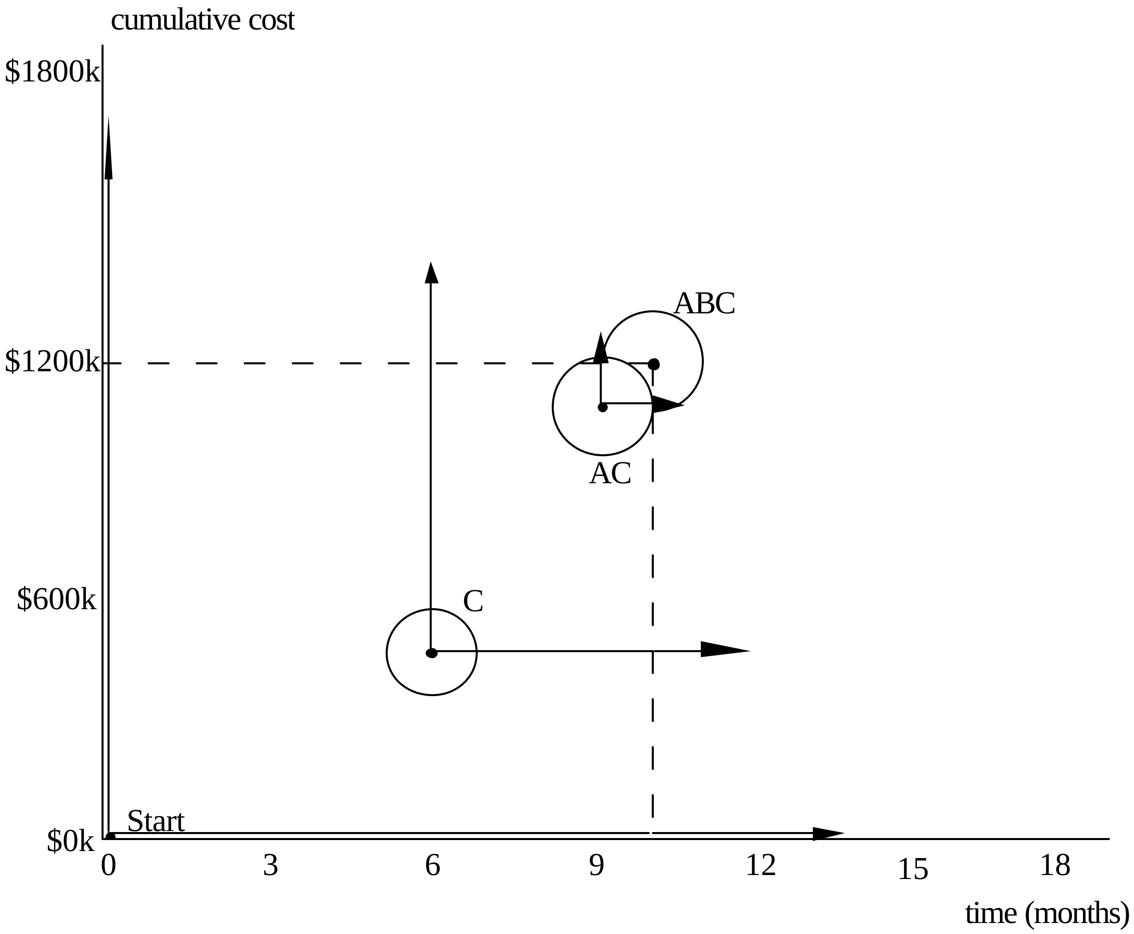

Figure 4. Plan 2 cumulative cost, cost risk, schedule risk, and operational value (lower, upper, and nominal estimates) over time

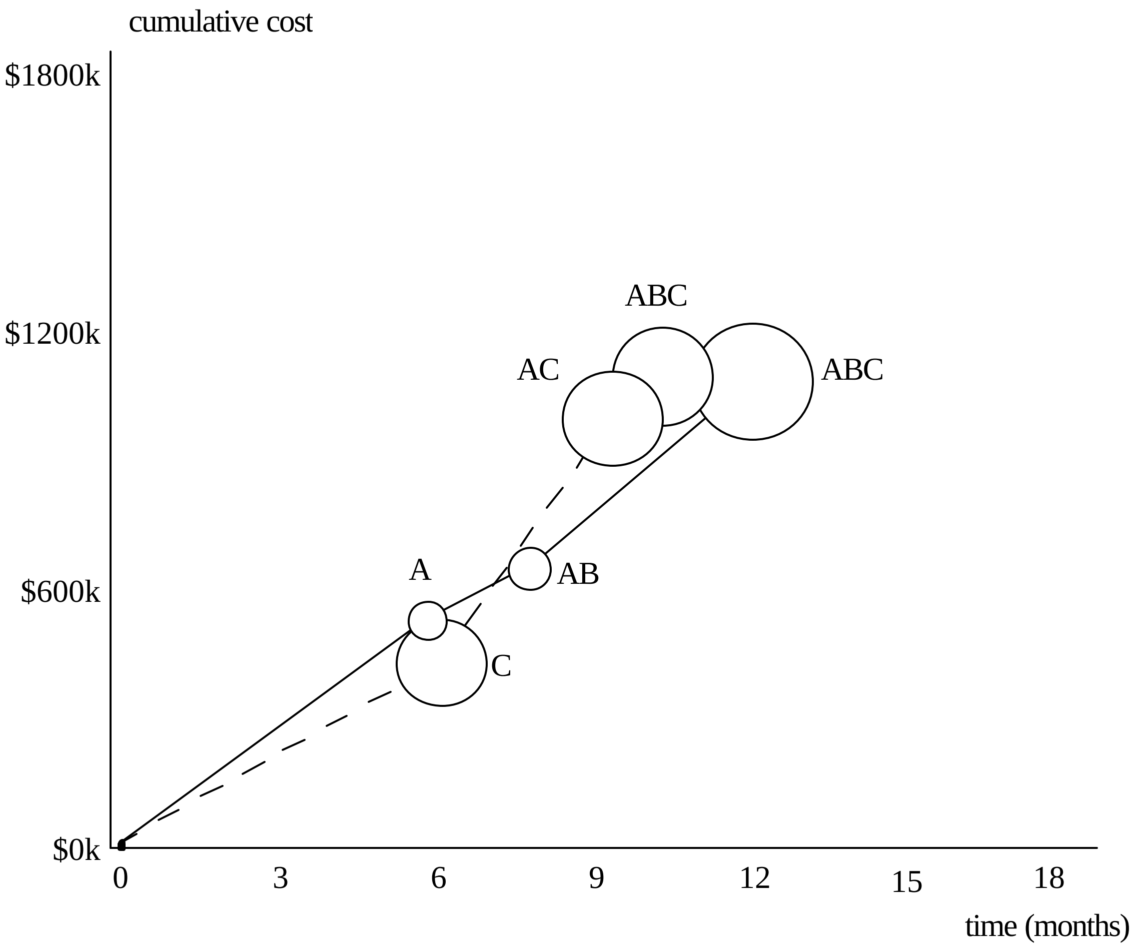

Figure 5. Juxtaposition of plan1 (solid line) and plan 2 (dashed line)

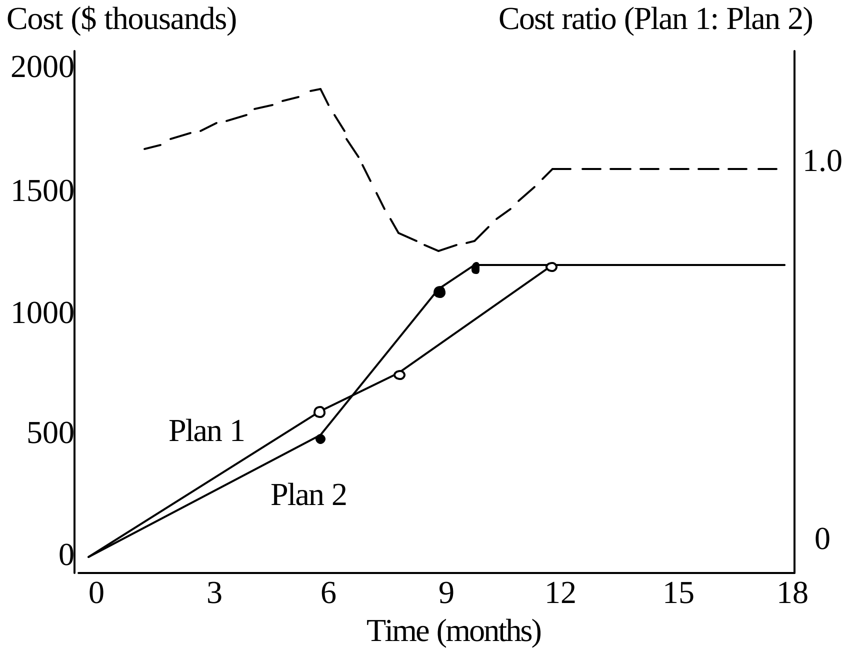

Figure 6. Cost and cost ratio versus time

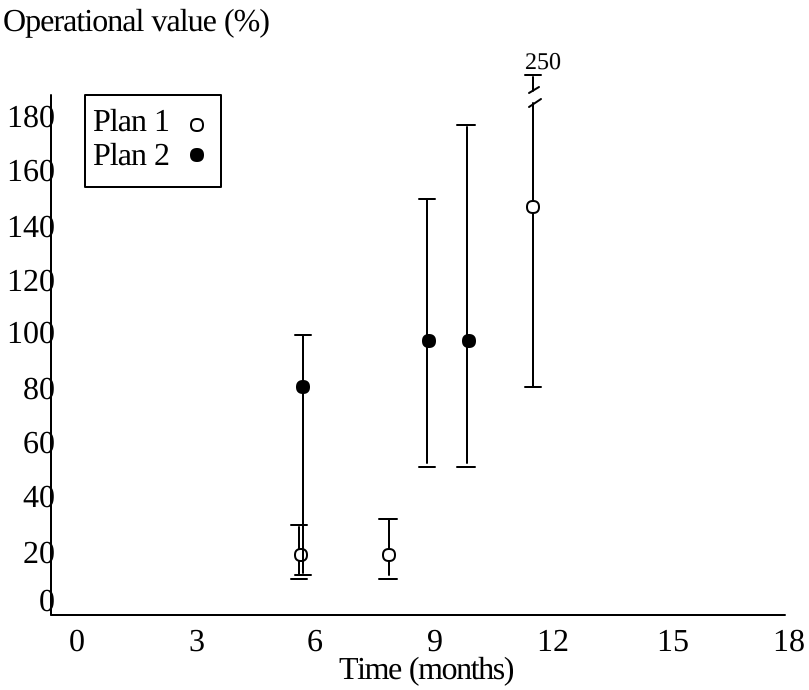

Figure 7. Operational value and operational value ratio versus time

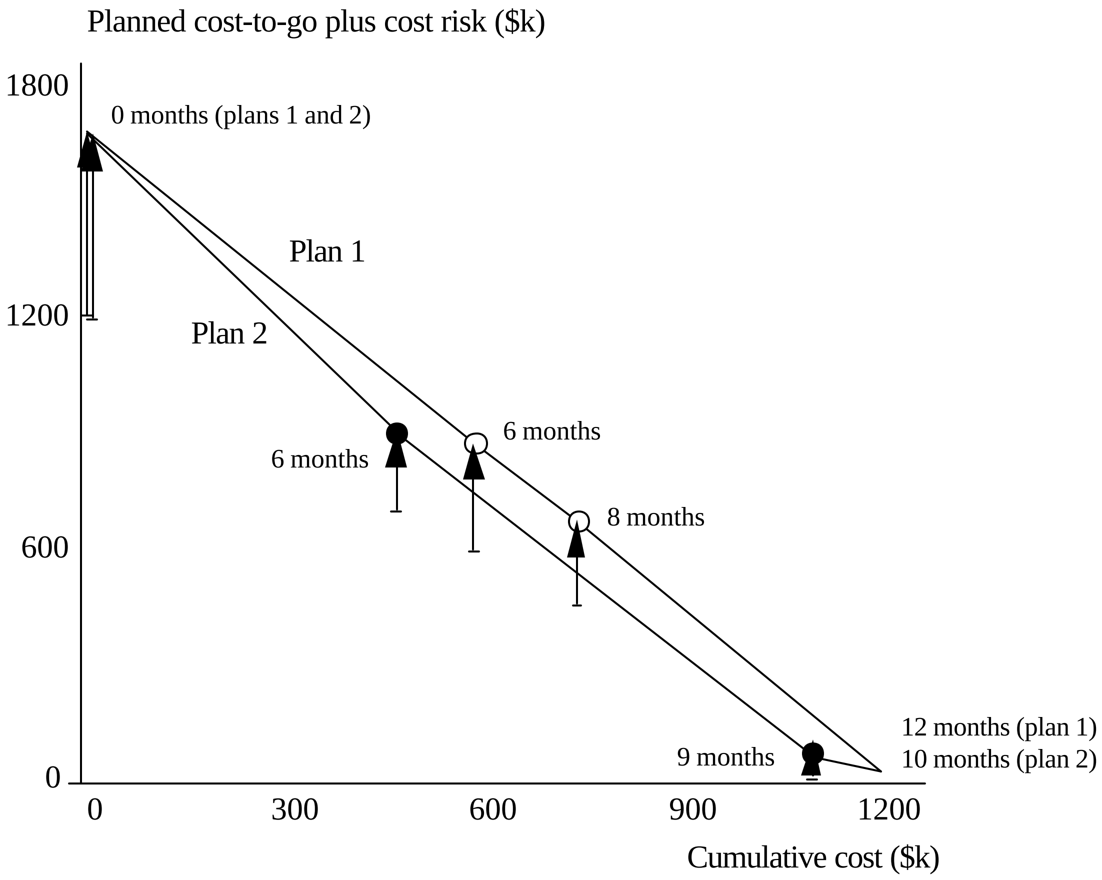

Figure 8. Planned cost-to-go plus cost risk versus cumulative cost; cost risk is vertical height of arrow-tipped segments

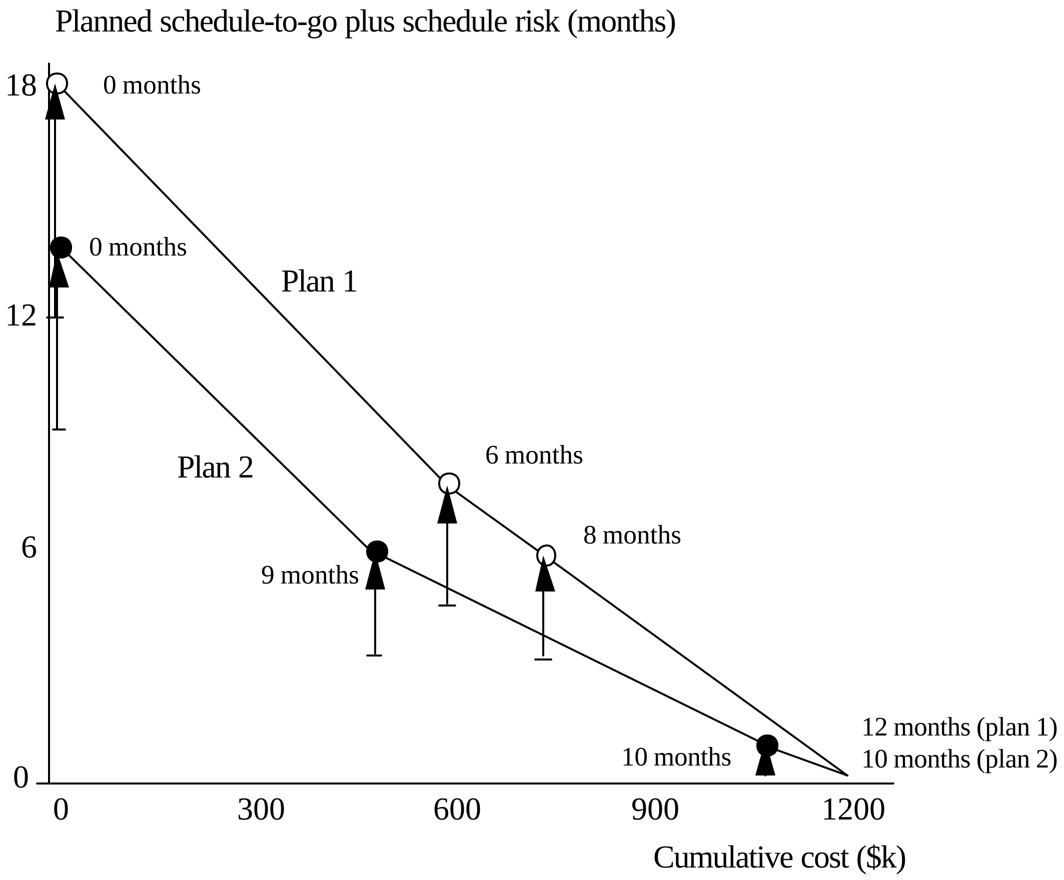

Figure 9. Planned schedule to go and remaining schedule risk versus cumulative (sunk) cost

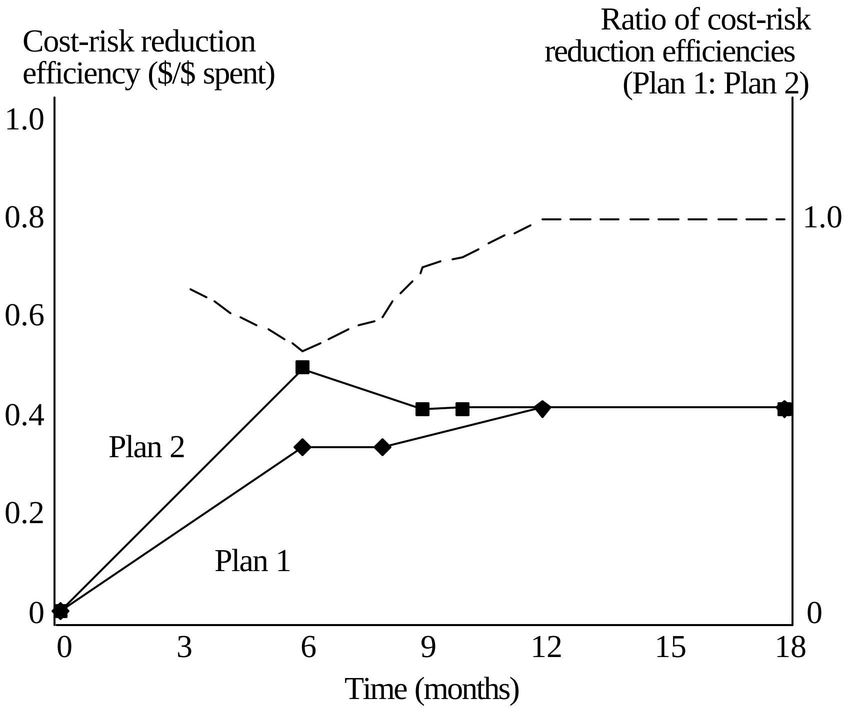

Figure 10. Cost-risk reduction efficiency and ratio (Plan 1: Plan 2) of efficiencies versus time

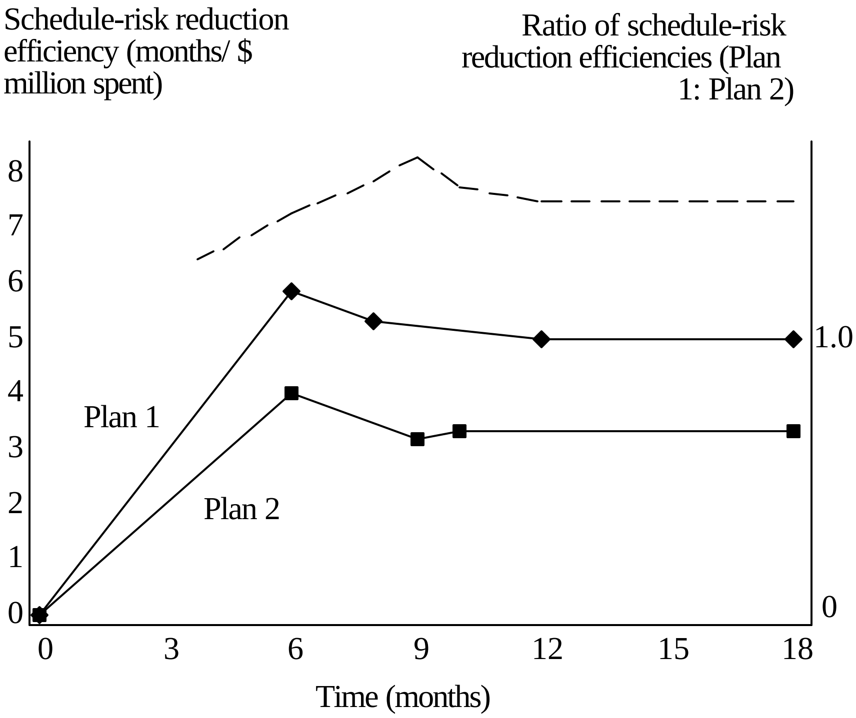

Figure 11. Schedule-risk reduction efficiency and ratio (Plan 1: Plan 2) of efficiencies versus time

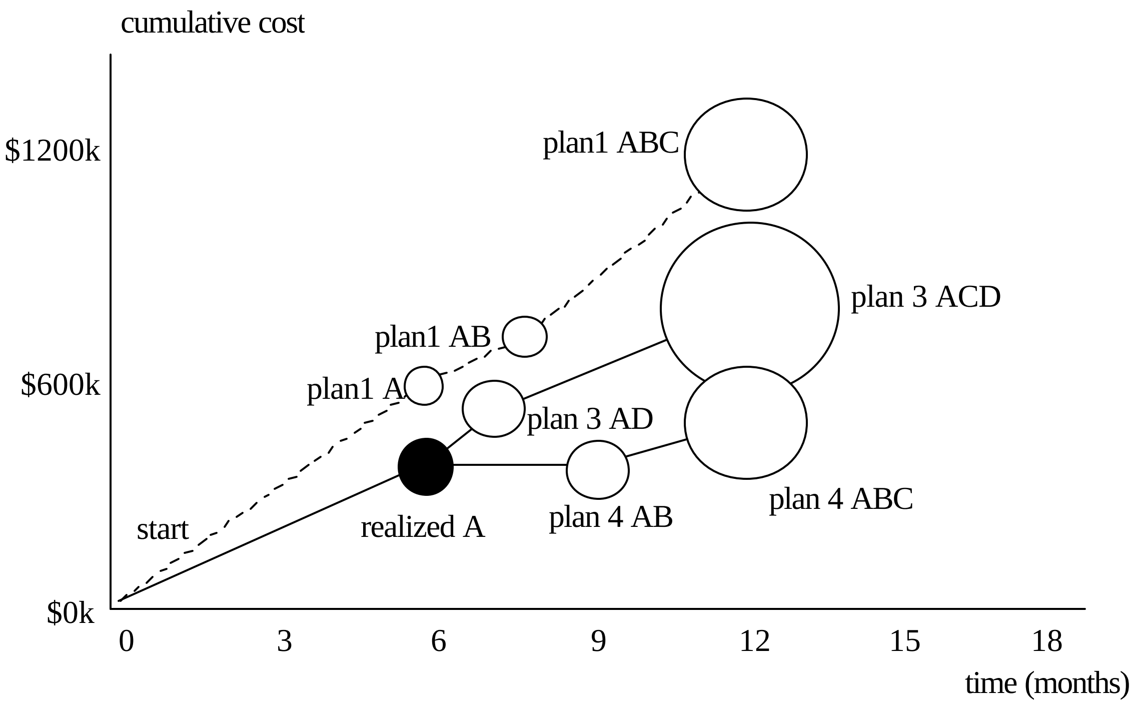

Figure 12. Re-planning example in which plan 3 is selected from among several replanning options at six months when a lower cost and higher operational value is realized from assembly A relative to the originally selected plan 1

Table 1. Component or assembly status

Assembly status	Description
Preliminary integration	Component adaptation and initial test
Final integration	Mature component or assembly that is available for final testing, available for independent operation and/or integration
Operational status	Configuration is fixed and supported by maintenance team

Table 2. Notation of parameters of assembly planning methodology

Parameter	Name	Remark
i	Assembly milestone and opportunity for replanning	--
s_p	Start of preliminary integration or adaptation (month)	--
s_f	Start final integration (month)	--
s_o	Start operation (month)	--
$c^k(i)$	Cost ($ thousands)	Planned cost to date for plan k at the milestone i
$o^k(i)$	Operational value (% of a nominal revenue, market share, or other measure of value)	Low, nominal, and high estimates of operational value for plan k at the milestone i
$s^k{}_s(i)$	Schedule risk (months)	Semi-standard deviation (beyond planned date of operation) of the time to operation of the full assembly under plan k at the milestone i
$s^k{}_c(i)$	Cost risk ($ thousands)	Semi-standard deviation (beyond planned cost of full assembly) of the cost to achieve the full assembly under plan k at the milestone i

Table 3. Data needs for assembly sequence Plan 1 with duration 18 months and expected revenue $500k per year (= 100%)

Assembly	Prel. integration (month)	Final integration (month)	Operation (month)	Schedule risk (months)	Cost ($ thousands)	Cost risk ($ thousands)	Operational value (%)
Start	-	-	-	6	-	500	-
A	0	3	6	2.5	600	300	(10, 20, 25)
B*	6	7	-	-	-	-	-
C*	6	9	-	-	-	-	-
AB	6	7	8	2	750	250	(10,20,30)
ABC	6	9	12	0	1,200	0	(80,150,250)

Three entries per cell represent a low, nominal, and high estimate; * denotes intermediate milestones not on the planning path

Table 4. Data needs for assembly sequence Plan 2 with duration 18 months and operational value $500k per year (= 100%)

Assembly	Prel. integration (month)	Final integration (month)	Operation (month)	Schedule risk (months)	Cost ($ thousands)	Cost risk ($ thousands)	Operational value (%)
Start	-	-	-	4	-	500	-
C	0	3	6	2	500	250	(10,80,100)
A*	3	7	-	-	-	-	-
B*	8	9	-	-	-	-	-
AC	3	8	9	0.5	1,100	50	(50,100,150)
ABC	8	9	10	0	1,200	0	(50,100,175)

Three entries per cell represent a low, nominal, and high estimate; * denotes intermediate milestones not on the planning path