Today's Management Technique and Tools: Are We Missing Something?

by Ernest M. Hahne

For decades, bookstore shelves have been filled with all manner of business guidance and management philosophy. Today we can choose from hundreds of software programs that, taken together, claim to be the solution to any management problem in any manner of approach. Every month our mailbags are overloaded with offers for better training or more effective consultation and business reengineering support.

With this much help, why do so many of us and our organizations continue to perform below par? Have the gurus of management science missed something that we all need to know? Is problem solution just "too hard," given the complexity of modern business and government requirements?

I do not believe it's "too hard." I do believe something has been overlooked. This paper describes an approach that was used to uncover this missing link, formulate a solution approach and test solution validity against in-process program needs rather than in a rarified laboratory environment.

Test/demonstration results indicate that we do not need to develop any new management principles. Rather, we need only to change our technique and some processes we use for application of existing, well-known principles. This rearrangement of technique and process application does require some modification and addition to our management tool set. However, revolutionary change is not called for and may, in fact, be counterproductive.

Identifying the Missing Link

Several study reports concerning numerous program failures within NASA, the DoD and industry in general prompted a search in the late 1980s for a missing management process link.[1] Based on the author's personal experiences as a program and systems management practitioner and consultant, an obvious question arises: Why do so many ventures that appeared sound at startup continue to report "surprising" indications of pending or actual failure? How can this be, given industry's significant investments in employee training, skills, hiring and acquisition of the "latest" in management information system (MIS) capability? What, specifically, goes wrong?

A similar question was asked in the mid-1960s by a small government team tasked to improve the existing program acquisition and management practices.[2] This team (with the author as a participant) reviewed numerous programs such as the FB-111, C5A and MinuteMan. We developed a lessons learned list of common reasons for major program problems. The list (unpublished at that time) was used as a guide for the creation of the MIL STD-499 Systems Engineering Management and early versions of the DoD 7000.2 Cost/Schedule Control Systems Criteria. The similarity between the data reported in the 1980s and in the 1960s list was very evident.

A direct correlation yielded surprising results. The only difference between the two was the increased length of the 1980s list.[3] The 22 new items, resulting in a new total of 59 Failure Lessons Learned, related primarily to software development and integration, and the rest to funding issues. In the 1960s relatively few programs had significant software content, and funding was not the issue it is today. However, what was the explanation for the rest of the list? A sample of the expanded list is illustrated in Figure 2.

Two approaches were addressed to explain the repeatability. The first, involving a validation review of existing techniques and processes, was rejected as time consuming and probably fruitless. Too many of us have "been there, done that." The approach taken was to search for a root cause, starting with the fundamentals of the overall program/production and operations management process: specifically, how organizations convert input data and raw materials into products and services that are economically useful to an end user. Fundamentally, this was a repeat of a 1966 study (conducted by the author) that resulted in a principle of management entitled "System Duality."[4]

Program Failure Lessons Learned	
1	Inadequate requirement specifications as part of the RFP severely compromise the overall acquisition effort and the quality of the delivered product.
2	Complete Interface Control Specifications between hardware and software and software are critical.
3	Adding manpower is rarely a solution to development schedule problem correction.
4	Training contractor and user personnel is essential.
5	Program management cannot specify good development criteria and just expect good development to happen.
6	Inadequately defined user requirements result in inadequate system/software specifications that lead to a contractually acceptable product that is operationally deficient.
7	Close and continuous monitoring at detailed schedule levels is essential. Risk Management needs should drive the level of detail.
8	Senior Management must be knowledgeable and involved in contract performance.
9	Communications and related documentation is critical to effective program configuration control and completion, i.e., ICWGs, minutes, telephone logs, Product Development Handbooks, etc.
10	Key personnel and management turnover causes critical problems.

Figure 2. *Program Failure Lessons Learned (Early 1990 Compilation).*

The System Duality concept states that management always deals with two interrelated systems, as illustrated by Figure 3. One is the organizational system (O) responsible for product production; the other is the product system (P) itself that is intended to satisfy the end user needs.

The concept also states that the key element of management control over the process was the Transform Function, as illustrated by the overlap of the O and P systems. Thus, management control metrics would encompass planned versus actual cost, schedule and technical performance data, describing the (O) system conversion of inputs to deliver a product (P) to the user.

The author's re-evaluation of the concept supported its validity as described, and as applied by practice. Industry has reams of processes available to address all elements of Figure 3, with two exceptions. These are highlighted in Figure 3 by the items contained within the dashed boxes. These two items appear to be the missing link within our management processes. Specifically, the absence of predictive and integrated risk analysis concerning the probability that our plans will fail at some significant cost and, also, our failure to assure timely review and feedback on developing results to the end user. In today's common practice, user feedback usually comes too late for easy design change. Essentially, the risks have already been incurred.

Risk Management Planning

Risk may be defined as the exposure to some likelihood of experiencing some loss. A loss can be expressed in many ways, such as a capability, economically, in terms of time, politically, socially, etc. The operative word in the definition is *some*. Loss magnitude can range from trivial to catastrophic. Loss occurrence can range from low to very high probability. Losses that do occur are usually additive.

There is *always* a likelihood of experiencing some loss. For previously demonstrated things, both the loss likelihood and magnitude may be known with

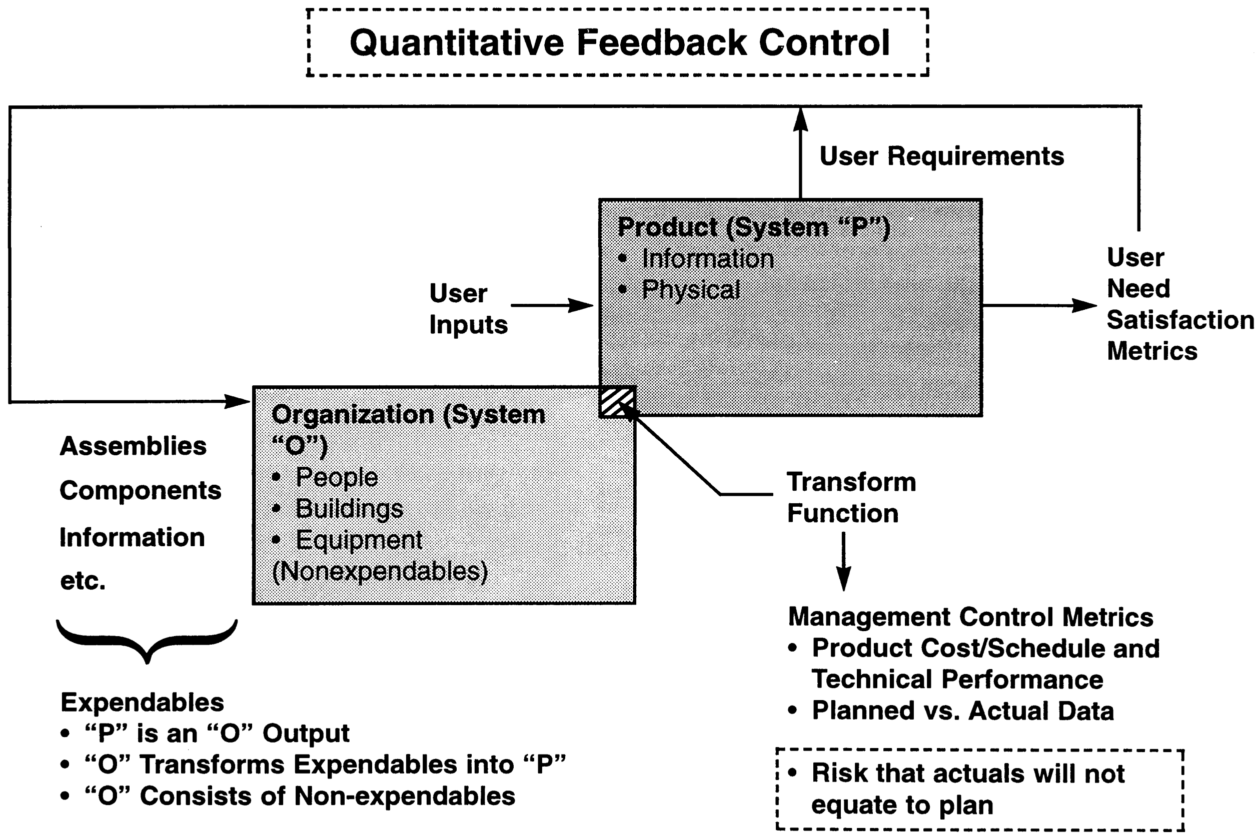


Figure 3. *System Duality Concept.*

reasonable accuracy. For things not demonstrated, a significant range of uncertainty may exist concerning not only both parameters, but also the mechanism responsible for the exposure.

Another finding from the Figure 3 study was that many causes for program failures appeared as the result of planning errors of omission. Items 1 and 6 on the Figure 2 list exemplify this. The complete list provides several additional examples of planning inadequacy that suggest the need to change our basic planning concepts.

First, we must admit that our biggest planning problem is that we don't know what we don't know at process startup. The author and others call this the (I DON'T KNOW)^2 problem. If we don't know that an issue exists, how can we possibly plan to avoid it?

Fortunately, there are many tools available that, if used properly, would surface critical planning questions. Unfortunately, too many of us do not use them or are unaware of their existence.

One such tool is the list represented by Figure 2. Its use as a checklist is extremely valuable for risk avoidance planning. Several other similar tools will be described later.

Another concept we should embrace involves the notion that in the absence of risk, management becomes basically unnecessary. Stated another way, we should conclude that the primary purpose of management planning is to provide a roadmap and measurements for avoidance and/or control of risks that attend development of any new product. On average, most of us currently practice reactive risk management. We must change our practice to emphasize preplanned or predictive risk management.

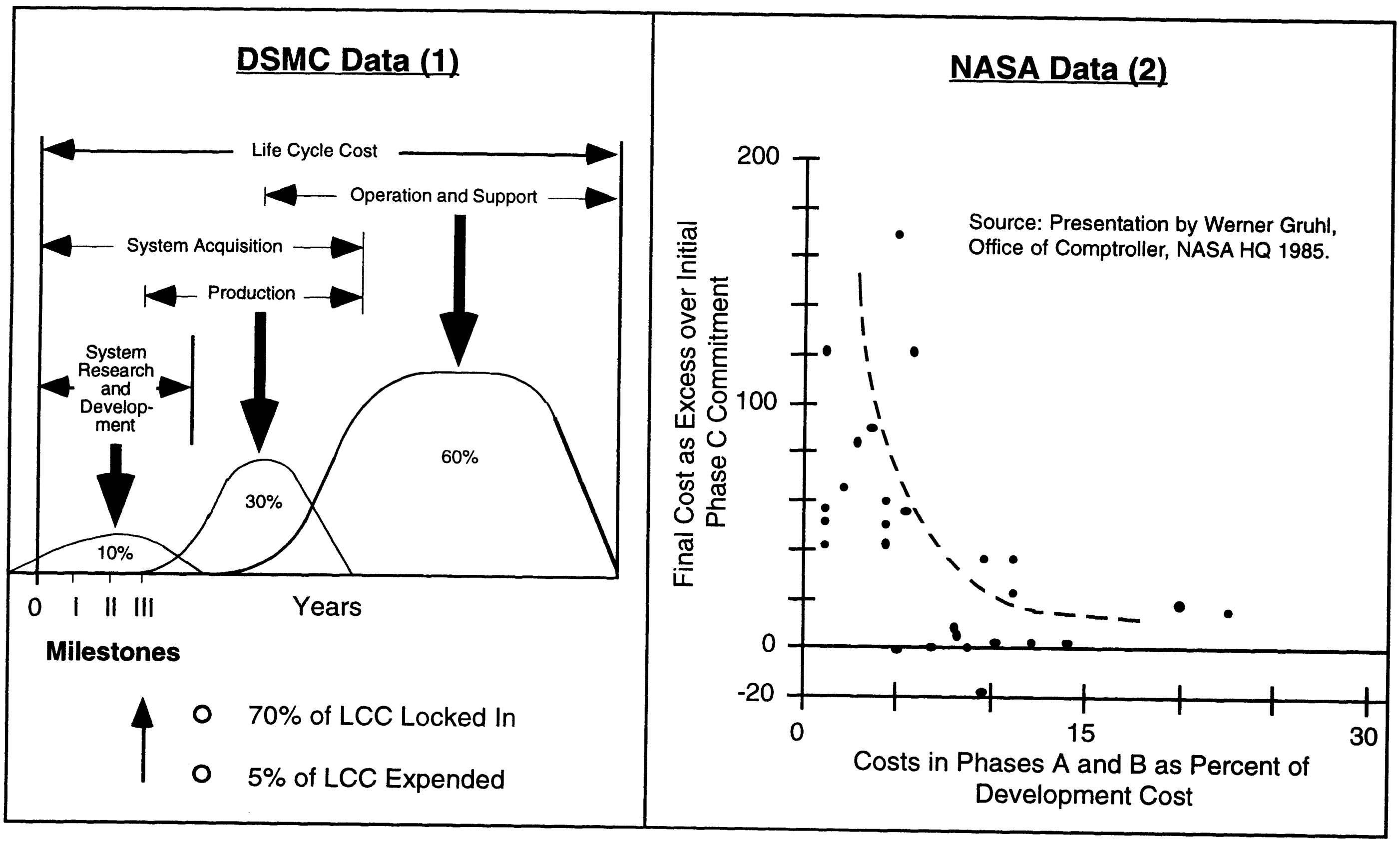
Another challenge to conventional thinking is that risk taking is bad. We can advance only by taking risk. The key here is that any risk taken must be affordable.

Finally, we must all realize that predictive risk planning requires a greater investment of time, skills and experience. Numerous studies show that significant payback can result when an upfront investment is dedicated to more detailed planning. Figure 4 illustrates data from two such studies.

The above recommended changes to our conceptual planning approaches are illustrated pictorially in Figure 5. Our current approach is illustrated at the top left. We have a plan for concept A with no upfront risk assessment. Implementation results are illustrated at the right. Note that "surprise" risk losses are a significant part of total cost, that total cost exceeds what was planned, and that a part of planned value was lost due to risks having occurred.

Just below we show the same Concept A plan but have included risk assessment. Note that total cost now includes the risk cost. Of course, the advertised cost is higher than one that did not include risk costs. Would the second plan and price be a winner?

An alternative plan (Concept B) including its risk costs is shown at the bottom of Figure 5. Note that total cost as illustrated is physically smaller than A above it and also that the risk budget is smaller. Planned value results remain approximately the same. (B results from trade studies that improve the baseline of A.) This figure illustrates the objectives that management techniques and tools are intended to achieve.



(1) A stretchout of MS 1 schedules of 25% or more, at added cost, would result in significant Life Cycle Cost (LCC) payback.
(2) An increase in cost expenditure during phases A/B results in a significant reduction of phase C overrun.

Figure 4. *Cost of Poor Planning.*

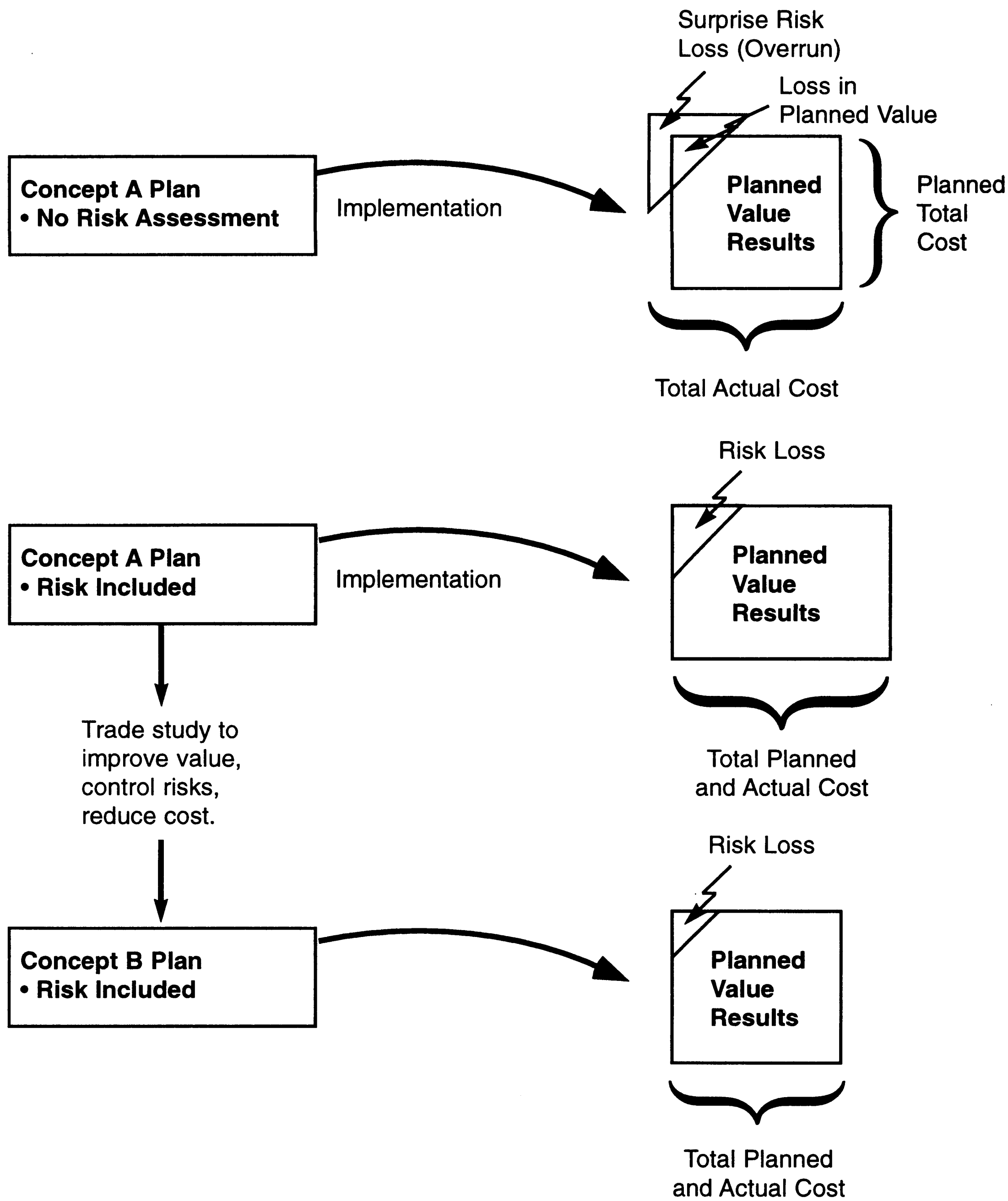


Figure 5. *Risk Planning Concept.*

Systems Engineering: A Primary Risk Analysis Technique

Risk analysis has a clearly definable starting point. Specifically, that point includes complete and quantified definition of end user needs, related constraints and measures of effective user results. This is an iterative process. The existing classical processes for systems engineering provide the foundation for performing predictive risk analysis and planning. This process starts with the end user needs and concludes with the assured delivery of an acceptable end product.

This paper does not address systems engineering process applications for resolution of all risk analysis needs. The applications that are addressed focus on how risks within a design concept are surfaced and how *relative* measurements can be made concerning their probability of occurring and the magnitude of loss if they occur. These relative measurements will serve as a flag and guide to management for their investment of resources and attention to avoid or control each identified risk.

An overview of the systems engineering process as commonly discussed in most publications[5] is shown in Figure 6. Note that risk analysis is one of many supporting functions to the centralized functions of system evaluation, trade studies and optimization.

Conclusions concerning the repeatability of the Figure 2 Program Failure Lessons Learned suggest that Figure 6 should be revised as shown in Figure 7. These revisions should aid future system engineering practice as needed to achieve predictive risk planning and more certain risk control. All suggested revisions can be correlated to one or more Program Failure Lessons Learned.

Revision 1: Insert risk analysis within the centralized function block. As a supporting function, many interpreted it to be a standalone requirement.

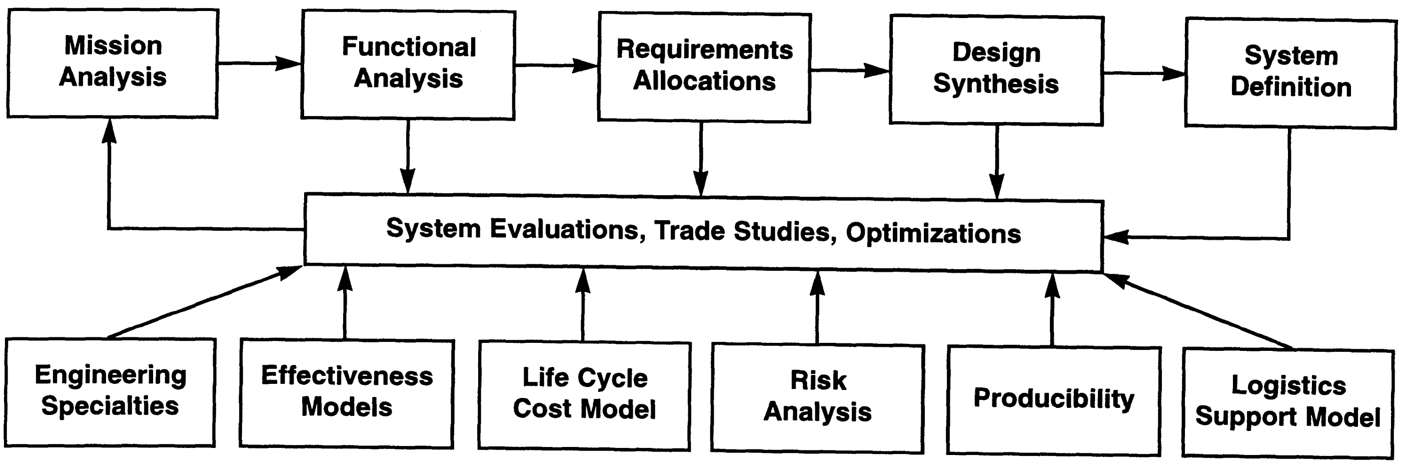


Figure 6. The Classical System Engineering Process.

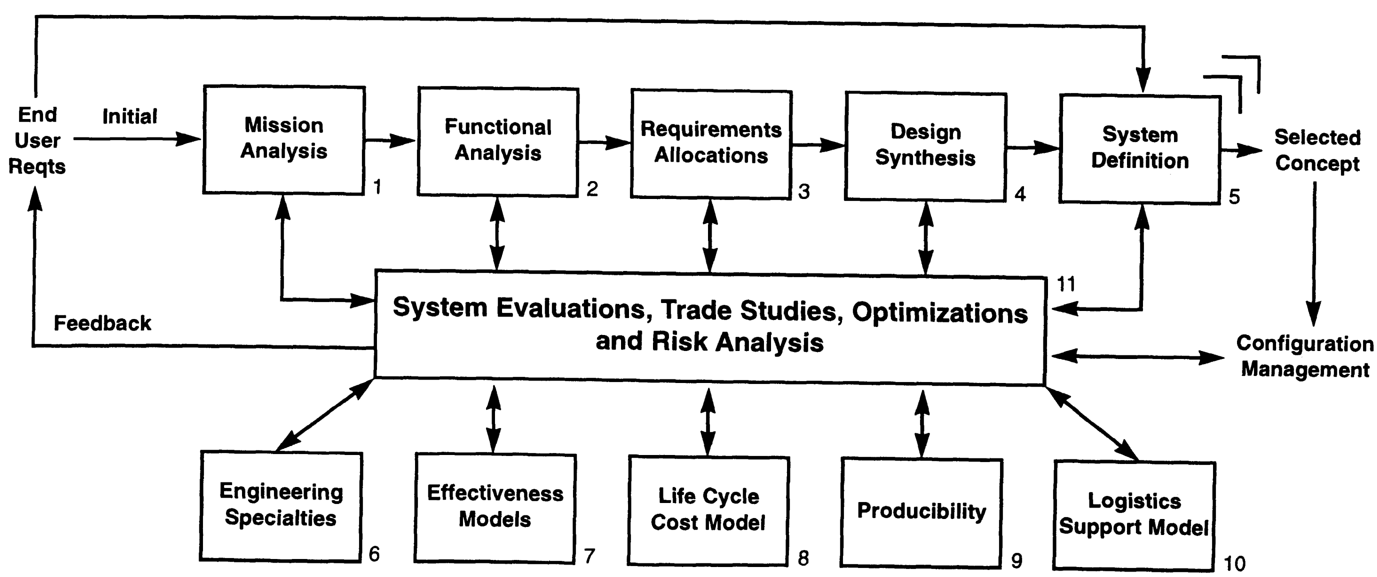


Figure 7. Suggested Revisions to the System Engineering Process.

Factually, that is how it is treated throughout today's DoD 5000 Series, NHB-7120, and most other publications. The transform function, previously shown in Figure 3, requires that risk analysis must be integrated within and across all functions.

Revision 2: Add the end user as a major function at the process beginning. Most of us overlook the criticality of this function to systems engineering success. Initial user inputs should only be introduced to block 1, Mission Analysis. Future inputs should be introduced into both block 1 and block 5, System Definition. Feedback should only emanate from block 11.

Revision 3: Clarify that all communications between the centralized and supporting functions are two-way, and for new problems, real time. Use double ended arrows. These paths contain the data for process direction, authorization and reporting of process problems and results. Real time communication control is critical to effective conduct of the Successive Refinement Process of Systems Engineering. (Avoid surprises at major progress reviews.)

Revision 4: Annotate block 5, System Definition, to emphasize that the purpose of the entire process is to select a best alternative based on a trade study among alternatives. Too many programs fail because the trade study was inadequate or not conducted.

Revision 5: Add Configuration Management (CM) as an administrative support process within systems engineering. CM should not function as a decision authority for change or approval. Reserve this role for the centralized authority of block 11. Also, all trade study data should be controlled under CM. Trade study results and decisions are totally dependent on the assumptions made and the analytical technique used. If these data are not available for future change analysis, chaos can result.

Trade Study and Risk Planning

Effective risk management depends on trade study performance and trade study is the heart of systems engineering. Systems engineering and risk management are totally intertwined.

While the actual performance of a trade study is usually complex and difficult, the fundamental concept is easy. (See Figure 8.)

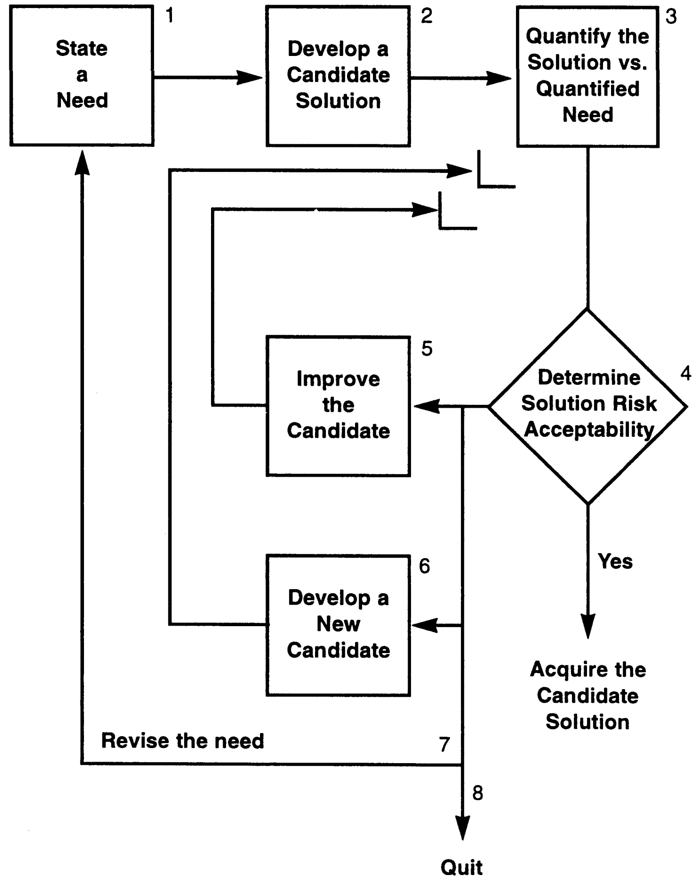


***Figure 8.** A Simple Trade Study Process.*

If a first pass through steps 1 to 4 don't yield a yes (it usually won't), exercise paths 5, 6 and 7 singly or in parallel. At this point block 4 becomes the trade study function where the best of all available choices is tested for acceptability. If a yes is not obtained, repeat 5, 6 and 7 or decide you have no acceptable solution approach and go to path 8 "Quit" or No Bid.

Obviously, a first step is to define an initial candidate solution that demonstrates feasibility for satisfaction of end user needs. Since this paper is primarily about techniques that avoid or mitigate risk, three major recommendations must be made concerning step one. First, obtain every scrap of detail available concerning user needs, related constraints and measures of minimally acceptable performance of the

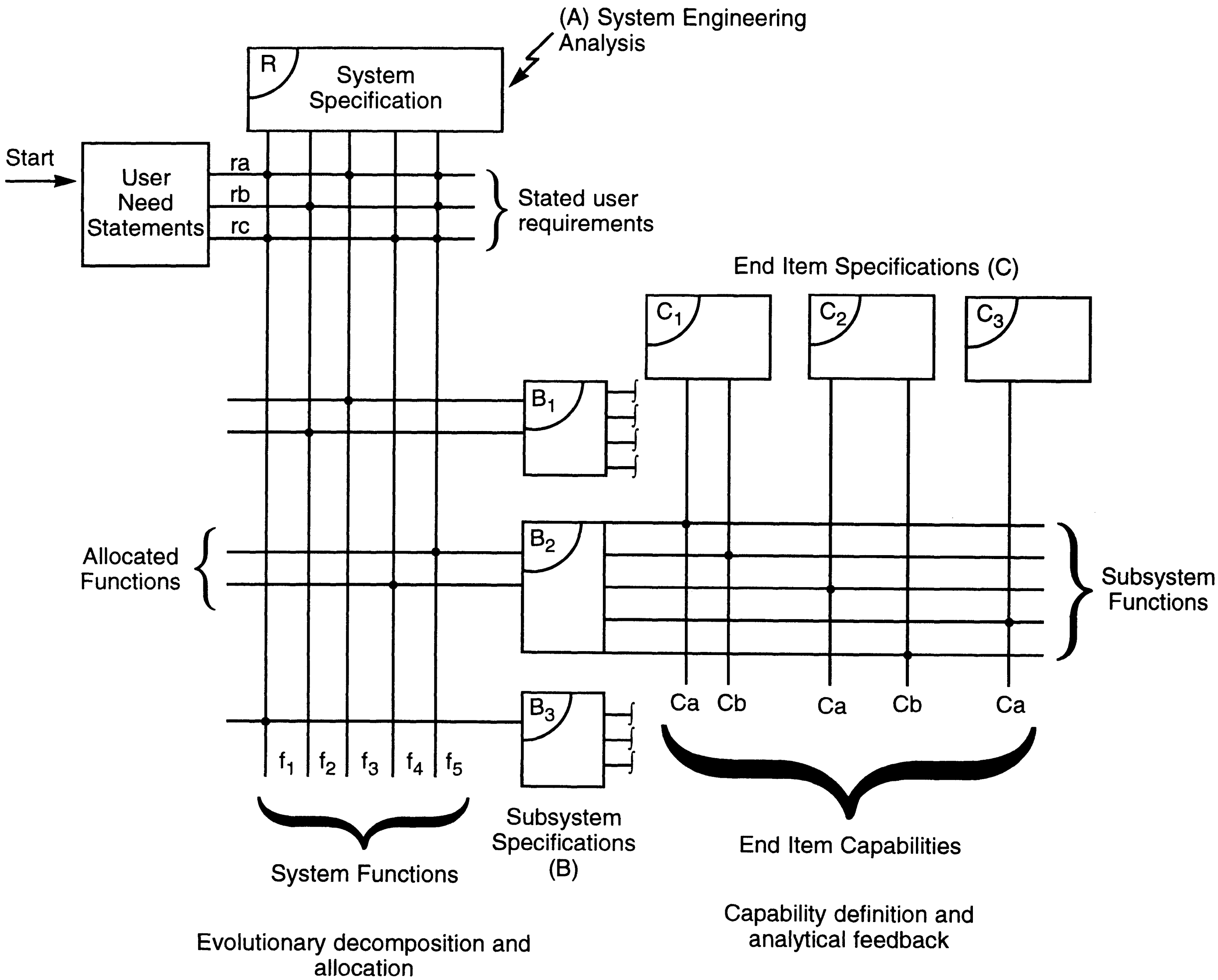


Figure 9. *A Typical Candidate System Design Matrix.*

product/system to be delivered. Help the user create this data if it is inadequate. Second, make sure that only highly skilled engineers experienced in the disciplines needed for initial solution definition are assigned. Third, avoid elegance in first cut approaches. Emphasize substance of need and why off-the-shelf solutions may be inadequate for user need satisfaction. Failure to adhere to the above recommendations will increase startup cost and may result in unforeseen life cycle risk in resulting program plans.

Design Synthesis

Creating the initial system solution candidate requires most of the functions of the System Engineering process illustrated by Figure 7. Initially, blocks 1, 2, 3, 4 and 6 are most critical. Difficulty in creating their data products suggests that team experience may be inadequate or that for block 4 the existing technological art is too limited. The latter issue represents a major risk that is discussed later.

Without belaboring how the Figure 7 processes are performed, the synthesized system concept that results from block 4 could result in a model as shown in Figure 9. At the top left are stated user needs (ra to rc) that initiate the analysis process and definition of specific system functional requirements, f_1 through f_5. These functions are allocated to subsystem B_1, B_2 and B_3. Further functional decomposition occurs and, as shown for B_2, these subsystem functions are allocated to end items C_1, C_2 and C_3. They provide the capabilities to perform the system and subsystem functions: for example, Ca and Cb for end item C_1.

A basic feasibility test of this synthesized design is conducted by asking the following questions:

1. Can end item capabilities, as identified, be reasonably satisfied by existing or new equipment known to be undergoing development?

2. Are there obvious reasons why the end items within the model would be difficult to produce or support logistically?

3. Are there difficult and perhaps unacceptable engineering specialty issues related to reliability, maintainability, human factors, safety, etc., concerning any of the end items or their integration?

4. Are the end items high cost? Is the schedule for their availability reasonable?

A negative response to one or more of the questions requires repetition of the Figure 7 process until two or more alternative synthesis models that demonstrate feasibility of satisfaction of end user needs are defined.

Note: If at least one feasible candidate cannot be defined, stop work. If this is due to unavailable technology, consider initiating an R&D project.

Establishing plausibility of each feasible design follows the Figure 7 process but emphasizes efforts through blocks 7, 8, 9 and 11. These activities are complex, time consuming and relatively expensive. The Program Failure Lessons Learned List items (Figure 2) suggest they are among the most poorly performed systems engineering activities. However, without some reasonable data input from them, effective performance of the block 11 trade study is hopeless.

Experience has shown that designing for perfection is infinitely costly and time consuming. Also, given the rapid growth of technology while we are designing, it's probably impossible. We need to change our selection and approval paradigm from a search for what's best, to a search for what is "least bad" but acceptable for satisfaction of known needs.

I do not suggest eliminating classical system effectiveness and life cycle cost analysis processes. I do advocate doing them only in areas where user need satisfaction would be significantly impaired by their absence. For any other purpose they tend to waste resources and time.

The following sections present a "poor person's approach" to resolution of these measurement needs.

Risk Management Decision Making

The proposed poor person's approach emphasizes the drawing of management decision attention to what most of us call grey areas.

Critical issues are usually obvious early on. (They can be enhanced by the judicious use of past lessons learned checklists.) Once known, they are sometimes given more attention than deserved.

Small issues are often set aside, as they should be, unless their impacts can be shown to grow.

The vast majority of issues are somewhat vague and, unless prioritized relative to their potential contribution to end user need and risk, consume vast amounts of management time and "self-protection" funding.

In addition to prioritization, another concept drives implementation of the poor person's approach. Rigorous mathematical analysis is often no better than relative magnitude estimation by an expert. Management decision making requires a "go/no-go" approach to metrics, not the precision that results from sophisticated and often computerized methods. The latter are usually costly and add little extra value.

Following is an application example of the poor person's approach to decision making, using an example solution candidate matrix based on the simple model shown previously in Figure 9. This measurement application example is shown in Figure 10.

Measurement begins at the upper left with user definition of value for each stated requirement. (For this discussion, limit this to value statements concerning mission functional requirements shown as ra, rb and rc.) Measurement ends at the lower right of the figure. This is where the engineer ranks the ability of available or soon to be available end items (hardware or software) proposed to provide the capabilities required to support satisfaction of mission functional requirements. In between are subsystem alloca-

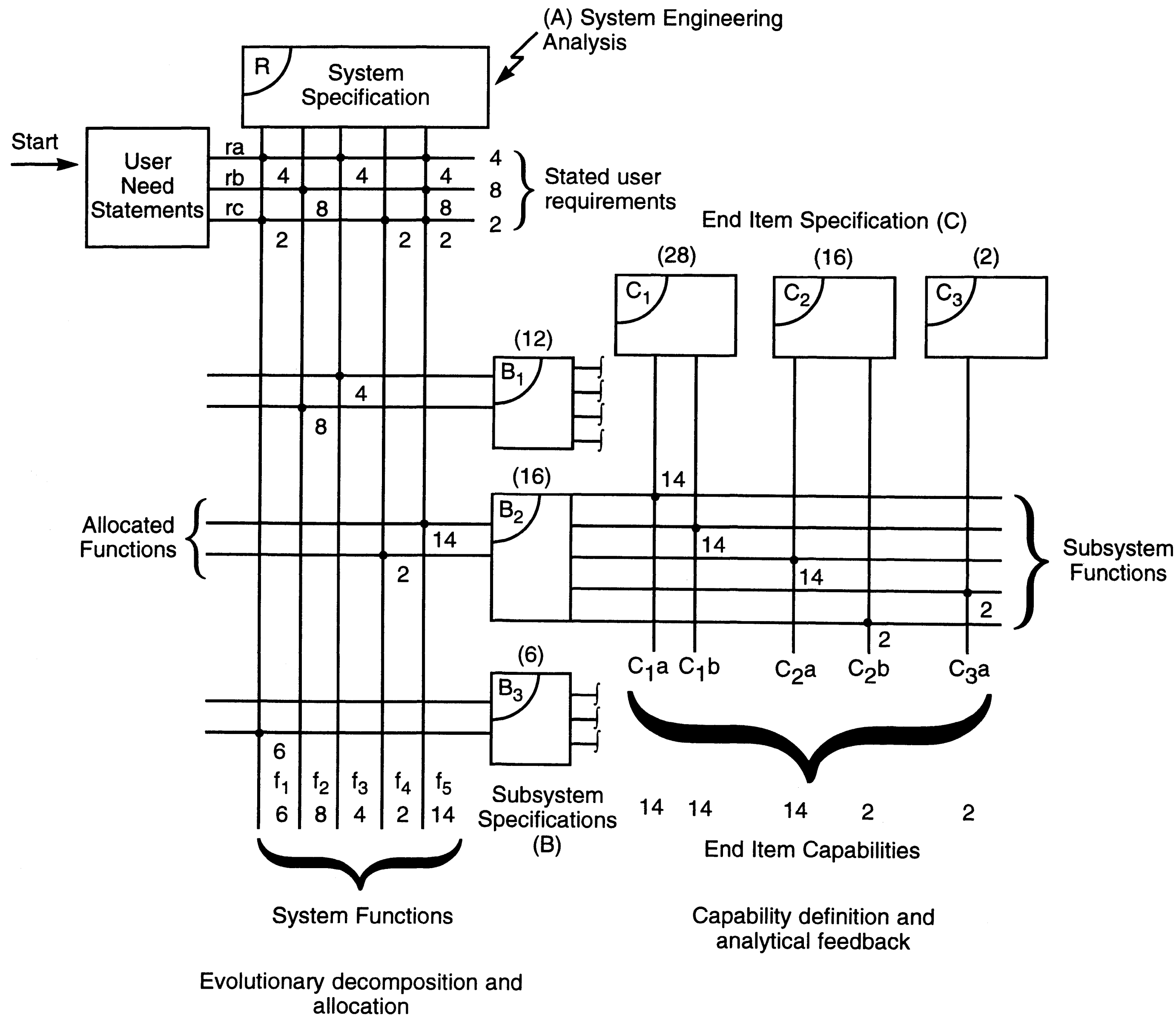


***Figure 10.** A Typical Candidate System Design Matrix with Value Measurements Annotated.*

tion concepts which serve as design control mechanisms. They allocate superior needs downward, to end items intended to serve these needs. Thus, the sum of user needs must be satisfied by the sum of end item capabilities. Management is concerned with the risk that this equation may not be met unless they exercise decisions to assure they will be. Simple step function metrics can effectively point the way.

As shown in the Figure 10 example, the user states a value rating for each defined need, using a scale of 10 for highest value and 1 for lowest value. Intermediate values fall in between. In the example shown requirement (ra) is valued at 4, (rb) at 8 and (rc) at 2.

Based on systems analysis, the engineer has identified five major functions (f_1 through f_5) as needed to satisfy the user requirement. How these functions contribute to user requirement satisfaction are shown by the dots at intersections of the (ra) (rb) and (rc) lines with the vertical function lines.

To assign values to functions, each dot is given the value of its source requirement. To establish a functions value, add up its vertical dot values. Thus:

$f_1 = 6$
$f_2 = 8$
$f_3 = 4$
$f_4 = 2$
$f_5 = 14$

One reason that f_5 is so high could be that its design represents a centralized computation function that contributes to performance of all other functions.

Subsystems of the synthesized design are shown as B_1, B_2 and B_3. Each is allocated the subrequirement to perform all or part of the system functions f_1 through f_5. Again, allocated functional values are added and the relative subsystem value: ecome:

$B_1= 12$
$B_2= 16$
$B_3= 6$

Repeating the process for subsystem decomposition and allocation the end items making up B_2 have the following values:

$C_1 = 28$ where $C_1a = 14$
$C_1b = 14$
$C_2 = 16$ where $C_2a = 14$
$C_2b = 2$
$C_3 = 2$ where $C_3a = 2$

At this point a *relative* value for all synthesized capabilities of a given design concept are established. All originate from stated user needs and values. Notice that the arithmetic method used amplifies the value numerics that flow downwards from the user mission requirements. Based on these value assignments, management attention should emphasize end item C_1 of B_2 over C_2 of B_2. However, until the risk associated with the acquisition and delivered performance of each end item is understood, management attention based on value alone may be misdirected.

While system value analysis is performed "Top Down," system risk analysis is performed from the bottom up. Consider the following axioms.

Axiom 1: Functional and physical performance of systems and subsystems is only limited by the capabilities of their end items.

Axiom 2: Systems and subsystems don't fail. Only their end items do.

Axiom 3: End item risk is a function of its maturity and past performance history. If an end item's capability has not been demonstrated previously *within* its intended operating environment, it is risky.

Axiom 4: Planning granularity is the most critical requirement for early surfacing and assessment of risk. End items must be understood.

Given the above, the author suggests the use of data as shown in Figure 11 as a tool for assigning a Risk Index to the capabilities of end items as synthesized for a new system. Note that the highest end item risk

Risk Index*	Risk Characteristic
10	New Technology Required
7-10	New development: Technology exists, but unproven for this use
5-8	New Design: Similar equipment in use. None directly applicable to this need.
3-6	Design Upgrade: Similar equipment in use: > 40% change required.
2-4	Shelf Modification: < 40% change required.
1-2	Shelf Equipment: COTS: Only changes as required for integration.
Note: The risk represents your resources expenditure to achieve the user requirement. The more you must invest, the greater your risk of loss.	

Figure 11. *Risk Index.*
End Item Maturity/Characteristics vs. Risk.

characteristic is assigned a 10 while the lowest is assigned a 1 or 2. A Risk Index equal to zero is never used. End items assigned a value of 8 or higher should be considered a candidate for R&D or Pre-Planned Productivity Improvements (P_3I).

Apply the Risk Index data of Figure 11 to the exampled synthesis in Figure 10. Sample results are shown in Figure 12, and are explained as follows.

1. The value of a capability (V) multiplied by its Risk Index (RI) equals the Management Concentration Index (MCI). Management should focus on capabilities that have highest value and risk combinations, i.e., V x RI = MCI.

2. Based on technology status, a Risk Index (RI) is assigned to each capability. (See lower right of figure.)

3. The capability value assignment (V) and (RI) are multiplied to obtain each end item capability (V x RI).

4. Add the capability (V x RI) totals to obtain the end item (V x RI).

5. Add the end items (V x RI) to obtain the subsystem (V x RI).

The resultant data per Figure 12 could be normalized to suggest that management attention for allocation and control of resources for Subsystem B_2 be applied as follows: $C_1 = 43\%$; $C_2 = 54\%$; $C_3 = 3\%$.

The same processes could be applied to Subsystems B1 and B3 end items. Normalizing all data across subsystems would result in relative ranking of all end items to prioritize management concentration across subsystems.

In a similar manner, subsystems could be ranked. By continuing the flow upwards to the system level, a system (V x RI) or MCI metric results. Given that, alternative syntheses can be compared to determine which one has a best change of being "least bad." Also, the detailed metrics data provides an indication of the plausibility of continuing with efforts for detailed design of the least bad alternative.

The reader should understand that the above numerics have only addressed technical needs risk assessment. The process can be expanded to encompass both cost and schedule parametrics as necessary to support more robust management decision making guidance. Economic rather than engineering decision theory provides the basis to such expanded application.

Supporting Tools and Training

On average, no new tools are required to perform what has been described. Most of the arithmetic processes presented can be aided by basic spreadsheets and a simple relational database.

Extending the technical risk assessment process to encompass economic issues requires tools and techniques that are generally unfamiliar to most systems engineers.

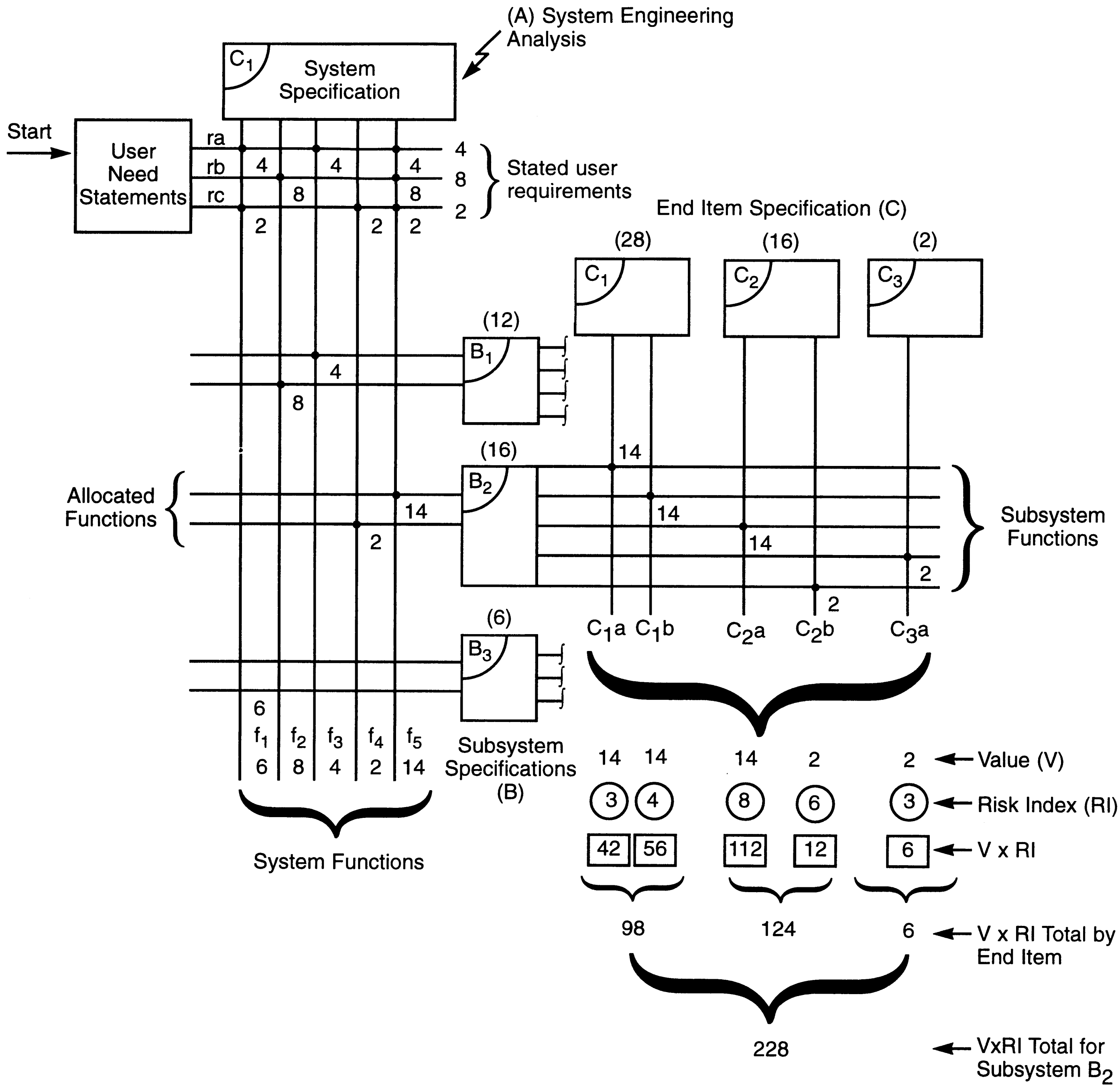


Figure 12. *A Typical Candidate System Design Matrix with Value and Risk Measurements.*

While new tool requirements are not a major issue, the failure or inability by most of us to use existing tools properly is a major issue. Some examples:

Checklists: Dozens exist in the literature that are rarely used. Using them can reduce risk that derives from errors of omission. They will jog the experienced person's memory. For inexperienced people, they stimulate questions and thought. Once an issue is surfaced, resolution will be addressed. Most checklists have been developed because of recurring failures.

Specification Formats: When combined with their descriptive instructions, they are a checklist. Don't modify their content. Tailor your response detail.

Mark unapplicable items as N/A. That is a useful data element to your reviewer.

Data Requirement Lists: Same value as the above, with one additional thought. If an item of data is necessary for decision making and future product/system maintenance or change, produce it. All else should be avoided.

Software Systems: Don't buy the latest because it's there. The cost of training and equipment upgrade can be prohibitive. Stay with what is "least bad," that with lowest risk.

Training is or should be a major concern, but most organizations continue to regard training from viewpoints that do not and cannot satisfy today's business and program management needs. Some specific issues of concern are:

Formal Training: Too many organizations continue to provide training from a "Square Filing" viewpoint. A person must participate in so many classroom hours per year to be considered for advancement. As an alternative, we should be training people to help them make decisions about things they are accountable for. Can it be that we don't know what their accountabilities are or should be? We should test every student in terms of how job performance was improved (risk reduction) because of classroom attendance.

Training Curriculum: Most training continues to teach the basics. While important, these are not sufficient. In today's business environment training must be tailored to fit the student's working needs. Basic theory, coupled with a generic classroom exercise, is usually too vague for timely job application subsequent to course completion. Solution of this problem involves two considerations. First, emphasize training of an Integrated Product Team (IPT) rather than a general student group. Secondly, tailor all training and classroom exercise to definition and management of the IPT's joint responsibilities and accountabilities.

Basic IPT training should emphasize teaching the overall processes of Program and Systems Management as required to meet IPT needs. This basic training should be followed up with specialty courses for the team after unique needs are determined as part of on-the-job training (OJT).

On-the-job Training: Tailored formal training without the provision of OJT has been shown to be ineffective. The classroom exercise should be developed as the OJT start-up exercise. Essentially it should be the "plan for the plan" of the IPT to develop an integrated IPT Project Plan after formal training. This planning effort identifies the need for follow-on specialty training courses. The earlier discussions of this paper outline a "plan for the plan" approach, resulting in a capability for risk management decision making.

Mentor Support: All but absent in most organizations today, mentor support is proving to be a costly issue for many organizations. It represents a form of training that is impossible to formalize for two reasons. When it's needed, it's needed *now*. And, what is needed can only be derived from combining previous experiences. There are two approaches to serve this need: retain some top quality "oldtimers" for this purpose, or, be sure that the selected IPT/OJT instructors can provide the service. A little of both may be the best choice. Consultants are not usually effective in this role.

Industry Lessons Learned

Over the past two years, the processes described in this paper have been applied to several NASA, DoD and commercial projects. In each case, formal training, OJT and mentor support was provided to an IPT. Descriptive experience concerning each project's results are beyond the scope of this paper.[6] However, the following lessons learned are typical of each.

1. A young team can follow the requirements of the NMI-7120 and DoD 5000 series processes with adequate training, OJT and mentor support.

2. You can start process application in the middle of a project.

3. Positive results are achieved within six to 12 weeks of start-up; that is, by a next-scheduled review.

4. At the next-scheduled review, there is more information on the scope of the effort and potential risk identification than by following the "usual" process, for the same or less effort cost.

5. Processes can force identification of risk areas that need to be addressed early on.

6. Specification/product trees can define an analytical baseline for planning, even if initially incorrect.

7. Help is essential in determining appropriate process tailoring.

8. The process holds people accountable and relies on hard data and metrics to determine performance acceptability.

9. The process provides high visibility over issues that affect interfacing projects.

10. The Planning/War Room process provides an effective means for evaluators and management to review work in process rather than waiting for a scheduled review. Reviews are shorter and fewer discrepancies are noted.

11. Planning/War Room data appears more complex and labor intensive than the usual process. It's not!

12. Resource-Loaded Schedule and Life Cycle Costing is not hard. It forces one to think about what is being done versus what should be done, and it surfaces uncertainty for early risk planning.

13. System/concurrent engineering is critical. End users must be involved at start-up.

14. In the beginning, some false starts will be made, but that is part of the learning process.

15. Management must provide proactive support to process implementation.

Failure Lessons Learned

Comparison of the Failure Lessons Learned in the 1980s with those from the 1960s showed them to be basically the same. I concluded that something was missing in how we were performing management.

A new analysis of the very basic requirements underlying the management process revealed that little if any emphasis was given to the management of risk. In general, it was observed that risk management was conducted to fill a square. Risks were only treated seriously when they had already been incurred. Few if any programs addressed predictive risk management.

A subsequent analysis of the Failure Lessons Learned List in the light of predictive risk management objectives revealed that some modest changes to existing practice could yield significant return. Following are some specific changes that have been presented.

1. Risk must be taken in order to advance or improve. The purpose of management is to surface and avoid *unacceptable* risk.

2. Early and in-depth planning is the only tool that can surface risk and thereby avoid reactive risk management. You must plan to a level of granularity that assures all remaining risk is affordable.

3. If remaining risk is not affordable, but the goal is valuable, consider an R&D or P_3I program in place of a Development/Production Program.

4. Management must redefine their decision criteria to choose the alternative that is "least bad" yet still meets overall end user system requirements.

5. Systems engineering must be recognized as the primary discipline that provides a common thread among all program management disciplines.

6. Simple mathematical processes can serve to support most value/risk decisions involved in the trade study analyses.

7. Process application rigor is essential to performance of predictive risk management.

8. Front end planning should be assigned only to experienced and skilled personnel.

9. All personnel should be trained to understand their role in the systems engineering "Big Picture." Such training provides the foundation to the Integrated Process Team's performance as required to carry out the Systems Engineering process. The use of checklists should be a major training thrust.

The results of the two studies have been presented for independent assessments of why so many major government programs are behind schedule, over budget and often deliver products that fall short of required operational capability. The studies were conducted more than 20 years apart, yet the failure reasons were basically the same.

Based on early study results, many changes were made to existing management policy, practice and procedures. Based on the more current study, similar changes are being made.

Comparative review of these new requirements versus the old revealed that the new practices are more clear and streamlined, but that no substantive differences are evident. Thus it appeared questionable that the next 10 or 20 years would produce any more improvements than the last 20 years. Better training did not appear to be the answer *per se*. Since 1970, industry and the government have invested heavily for this purpose. I felt something was still missing from our approach.

A return to basic analysis of fundamental business practice suggested this to be true. It was established that the primary need for management was to avoid risk in the Program Development and Acquisition process. A review of old and new practice through NMI 7120, the DoD 5000 series and other similar policies, showed that risk was addressed poorly, if at all.

This paper described a relatively simple approach towards solution of the risk management problem. The process is founded on the practices of our current systems engineering processes. Field testing has shown that predictive risk management is practical and not too hard to perform by a young team, given some simple checklist tools and minimal training in their use.

References

1. J.S. Gansler, "Program Instability: Causes, Costs and Cures." Defense Acquisition Study, Center for Strategic and International Studies, Georgetown University, March 1, 1986. December 1992 GAO Report Summary. SPF Management Office. Reconfiguration Management Division. GAO/OCG - 93-27 TR.

2. Team organized Sept. 1966 by Colonel Donald H. Heaton, USAF, Assistant for Systems Management, DCS/Systems. Objective: Develop Systems Analysis/Engineering Output Specification. Result: Initial Release of Mil Std 499, Systems Engineering Management, 1967.

3. The initial Program Failure Lessons Learned List was published by E.M. Hahne in the mid-1980s. An updated version was developed in 1989 for incorporation in all training programs presented internationally by E.M. Hahne International.

4. Presented in a paper entitled "Hypothesis with Applications for Total Systems Management," page 137. Published in *A Forum on Systems Management*, School of Business Administration, Temple University, June 1967.

5. Defense System Management College *Systems Engineering Management Guide*, Fort Belvoir, VA 22060-5426. (Several Releases) See also NASA System Engineering Process for Programs and Projects, JSC 49040 Oct 1994. *System Engineering*. Goode & Machal, McGraw-Hill, 1957. *NASA Systems Engineering Handbook*, Sept 1992. (Draft) NASA Headquarters. (Revised June 1995).

6. The reader may contact the following persons for applications project detail.

NASA Projects: Tom Diegleman and Susie Mauzy at NASA JSC, Houston, Texas.

DoD (Navy) Projects: Captain R. Kollmorgan, PMA-299, NAVAIR, Washington, D.C.

Commercial Applications: L&R Nursery Business Analysis: E. Hahne, Plant City, Florida (813) 759-8719.

Hospital Business Analysis: J. Tucker, Houston, Texas (713) 334-4868.