

CHAPTER 17

PRODUCT IMPROVEMENT STRATEGIES

17.1 INTRODUCTION

Complex systems do not usually have stagnant configurations. A need for a change during a system's life cycle can come from many sources and effect the configuration in infinite ways. The problem with these changes is that, in most cases it is difficult, if not impossible, to predict the nature and timing of these changes at the beginning of system development. Accordingly, strategies or design approaches have been developed to reduce the risk associated with predicted and unknown changes.

Well thought-out improvement strategies can help control difficult engineering problems related to:

- Requirements that are not completely understood at program start,
- Technology development that will take longer than the majority of the system development,
- Customer needs (such as the need to combat a new military threat) that have increased, been upgraded, are different, or are in flux,
- Requirements change due to modified policy, operational philosophy, logistics support philosophy, or other planning or practices from the eight primary life cycle function groups,
- Technology availability that allows the system to perform better and/or less expensively,
- Potential reliability and maintainability upgrades that make it less expensive to use, maintain, or support, including development of new supply support sources,

- Safety issues requiring replacement of unsafe components, and
- Service life extension programs that refurbish and upgrade systems to increase their service life.

In DoD, the 21st century challenge will be improving existing products and designing new ones that can be easily improved. With the average service life of a weapons system in the area of 40 or more years, it is necessary that systems be developed with an appreciation for future requirements, foreseen and unforeseen. These future requirements will present themselves as needed upgrades to safety, performance, supportability, interface compatibility, or interoperability; changes to reduce cost of ownership; or major rebuild. Providing these needed improvements or corrections form the majority of the systems engineer's post-production activities.

17.2 PRODUCT IMPROVEMENT STRATEGIES

As shown by Figure 17-1, these strategies vary based on where in the life cycle they are applied. The strategies or design approaches that reflect these improvement needs can be categorized as planned improvements, changes in design or production, and deployed system upgrades.

Planned Improvements

Planned improvements strategies include evolutionary acquisition, preplanned product development, and open systems. These strategies are not exclusive and can be combined synergistically in a program development.

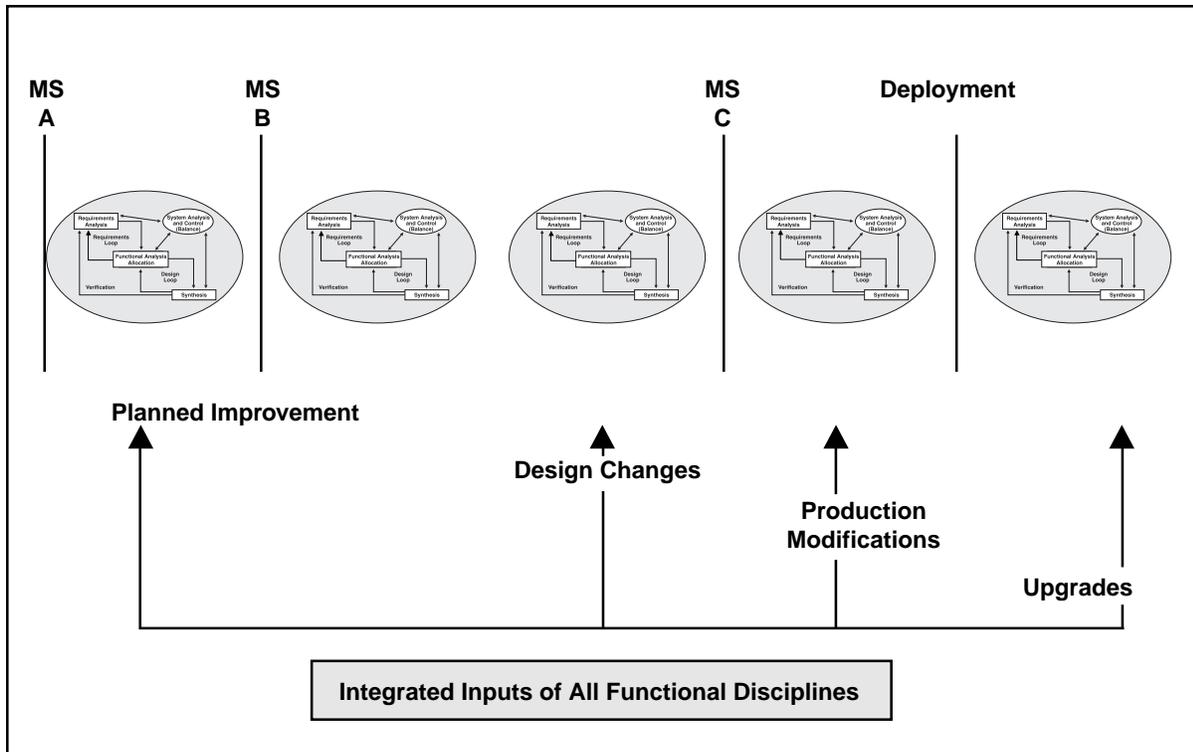


Figure 17-1. Types of Product Improvement Strategies

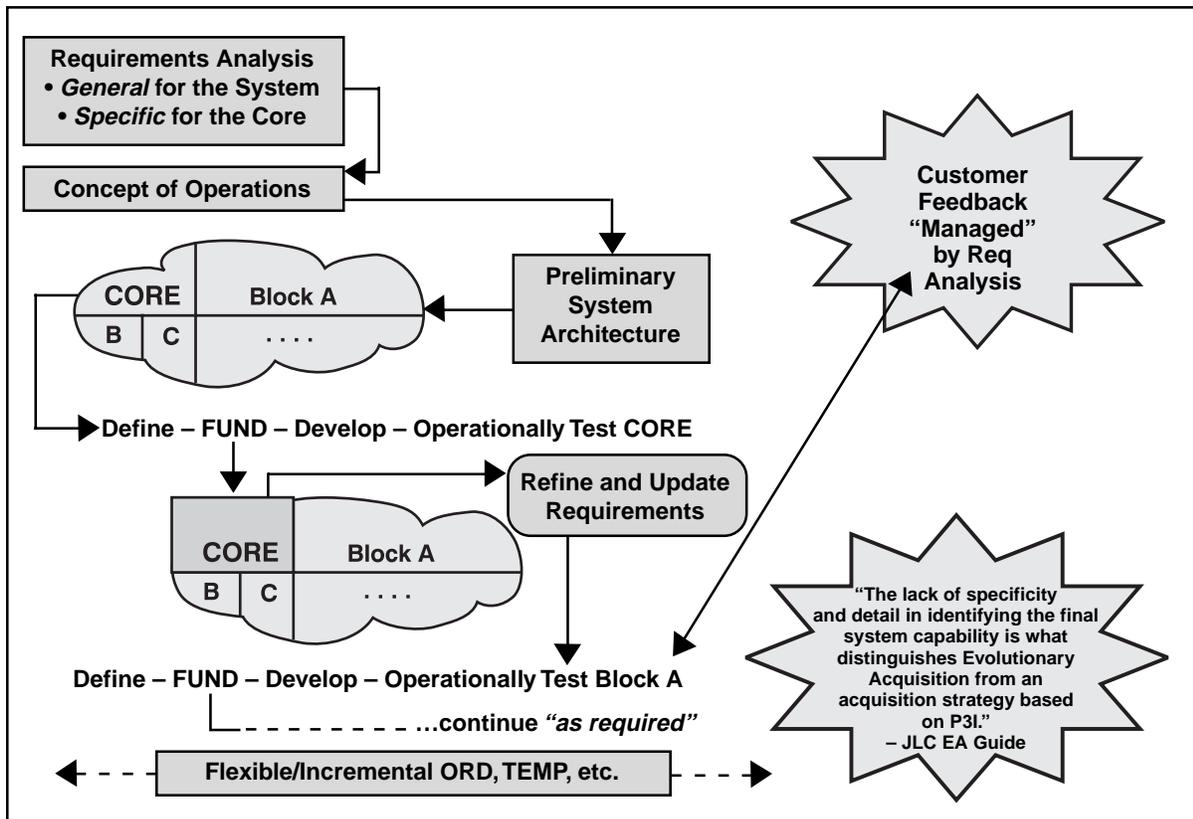


Figure 17-2. Evolutionary Acquisition

Evolutionary Acquisition: Evolutionary acquisition is the preferred approach to systems acquisition in DoD. In an environment where technology is a fast moving target and the key to military superiority is a technically superior force, the requirement is to transition useful capability from development to the user as quickly as possible, while laying the foundation for further changes to occur at later dates. Evolutionary acquisition is an approach that defines requirements for a core capability, with the understanding that the core is to be augmented and built upon (evolved) until the system meets the full spectrum of user requirements. The core capability is defined as a function of user need, technology maturity, threat, and budget. The core is then expanded as need evolves and the other factors mentioned permit.

A key to achieving evolutionary acquisition is the use of time-phased requirements and continuous communication with the eventual user, so that requirements are staged to be satisfied incrementally,

rather than in the traditional single grand design approach. Planning for evolutionary acquisition also demands that engineering designs be based on open system, modular design concepts that permit additional increments to be added over time without having to completely re-design and re-develop those portions of the system already fielded. Open designs will facilitate access to recent changes in technologies and will also assist in controlling costs by taking advantage of commercial competition in the marketplace. This concept is not new; it has been employed for years in the C4ISR community, where system are often in evolution over the entire span of their lifecycles.

Preplanned Product Improvement (P3I): Often referred to as P3I, preplanned product improvement is an appropriate strategy when requirements are known and firm, but where constraints (typically either technology or budget) make some portion of the system unachievable within the schedule required. If it is concluded that a militarily

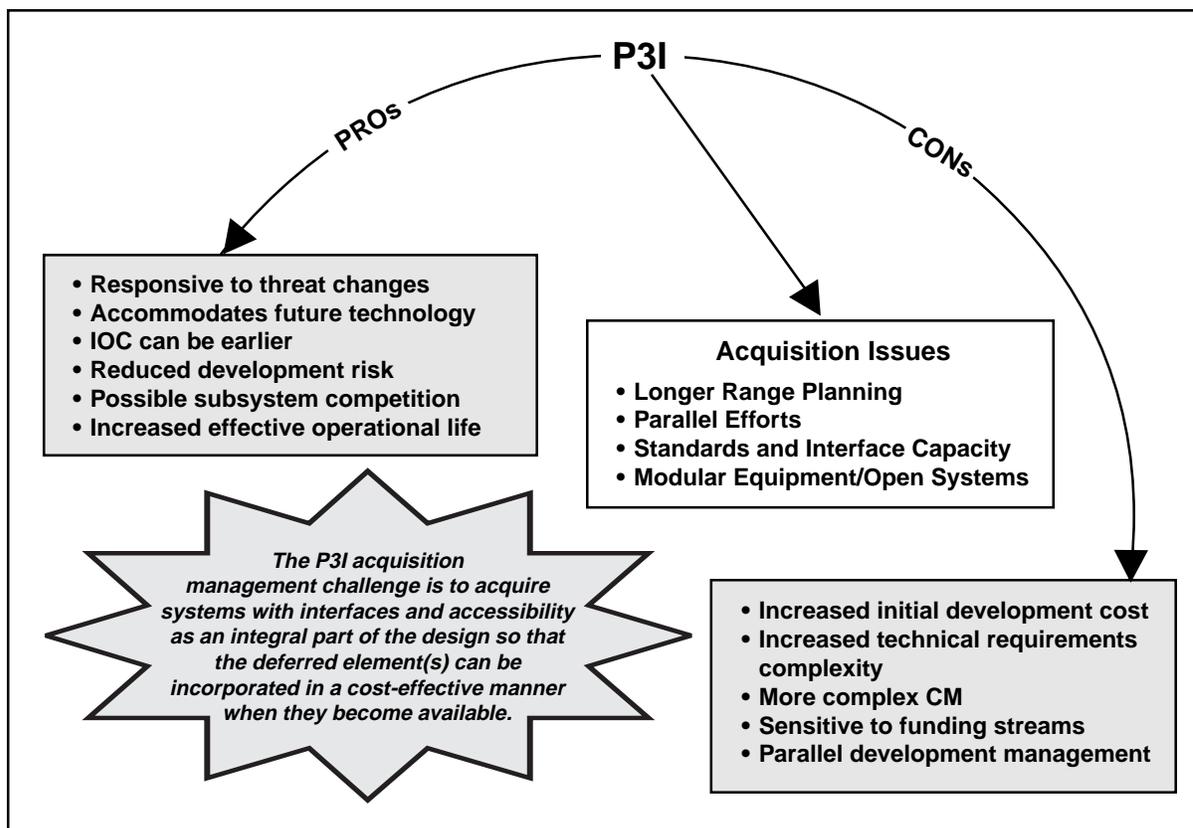


Figure 17-3. Pre-Planned Product Improvement

useful capability can be fielded as an interim solution while the portion yet to be proceeds through development, then P3I is appropriate. The approach generally is to handle the improvement as a separate, parallel development; initially test and deliver the system without the improvement; and prove and provide the enhanced capability as it becomes available. The key to a successful P3I is the establishment of well-defined interface requirements for the system and the improvement. Use of a P3I will tend to increase initial cost, configuration management activity, and technical complexity. Figure 17-3 shows some of the considerations in deciding when it is appropriate.

Open Systems Approach: The open system design approach uses interface management to build flexible design interfaces that accommodate use of competitive commercial products and provide enhanced capacity for future change. It can be used to prepare for future needs when technology is yet not available, whether the operational need is known or unknown. The open systems focus is to design the system such that it is easy to modify using standard interfaces, modularity, recognized interface standards, standard components with recognized common interfaces, commercial and nondevelopmental items, and compartmentalized design. Open system approaches to design are further discussed at the end of this chapter.

Changes in Design or Production

Engineering Change Proposals (ECPs): Changes that are to be implemented during the development and production of a given system are typically initiated through the use of ECPs. If the proposed change is approved (usually by a configuration control board) the changes to the documentation that describes the system are handled by formal configuration management, since, by definition, ECPs, when approved, change an approved baseline. ECPs govern the scope and details of these changes. ECPs may address a variety of needs, including correction of deficiencies, cost reduction, and safety. Furthermore, ECPs may be assigned differing levels of priority from routine to emergency. MIL-HDBK-61, Configuration Management Guidance, offers an excellent source of

advice on issues related to configuration changes.

Block Change before Deployment: Block changes represent an attempt to improve configuration management by having a number of changes grouped and applied such that they will apply consistently to groups (or blocks) of production items. This improves the management and configuration control of similar items substantially in comparison to change that is implemented item by item and single change order by single change order. When block changes occur, the life cycle impact should be carefully addressed. Significant differences in block configurations can lead to different manuals, supply documentation, training, and restrictions as to locations or activities where the system can be assigned.

Deployed Systems Upgrades

Major Rebuild: A major rebuild results from the need for a system that satisfies requirements significantly different or increased from the existing system, or a need to extend the life of a system that is reaching the end of its usable life. In both cases the system will have upgraded requirements and should be treated as basically a new system development. A new development process should be started to establish and control configuration baselines for the rebuilt system based on the updated requirements.

Major rebuilds include remanufacturing, service-life extension programs, and system developments where significant parts of a previous system will be reused. Though rebuilding existing systems can dramatically reduce the cost of a new system in some cases, the economies of rebuild can be deceiving, and the choice of whether to pursue a rebuild should be done after careful use of trade studies. The key to engineering such systems is to remember that they are new systems and require the full developmental considerations of baselining, the systems engineering process, and life cycle integration.

Post-Production Improvement: In general, product improvements become necessary to improve the system or to maintain the system as its components

reach obsolescence. These projects generally result in a capability improvement, but for all practical purposes the system still serves the same basic need. These improvements are usually characterized by an upgrade to a component or subsystem as opposed to a total system upgrade.

Block Upgrades: Post-production block upgrades are improvements to a specific group of the system population that provides a consistent configuration within that group. Block upgrades in post-production serve the same general purpose of controlling individual system configurations as production block upgrades, and they require the same level of life-cycle integration.

Modifying an Existing System

Upgrading an existing system is a matter of following the system engineering process, with an emphasis on configuration and interface management. The following activities should be included when upgrading a system:

- Benchmark the modified requirements both for the upgrade and the system as a whole,
- Perform functional analysis and allocation on the modified requirements,
- Assess the actual capability of the pre-upgrade system,
- Identify cost and risk factors and monitor them,
- Develop and evaluate modified system alternatives,
- Prototype the chosen improvement alternative, and
- Verify the improvement.

Product improvement requires special attention to configuration and interface management. It is not uncommon that the existing system's configuration will not be consistent with the existing configuration data. Form, fit, and especially function interfaces often represent design constraints

that are not always readily apparent at the outset of a system upgrade. Upgrade planning should ensure that the revised components will be compatible at the interfaces. Where interfaces are impacted, broad coordination and agreement is normally required.

Traps in Upgrading Deployed Systems

When upgrading a deployed system pay attention to the following significant traps:

Scheduling to minimize operational impacts: The user's operational commitments will dictate the availability of the system for modification. If the schedule conflicts with an existing or emerging operational need, the system will probably not become available for modification at the time agreed to. Planning and contractual arrangements must be flexible enough to accept unforeseen schedule changes to accommodate user's unanticipated needs.

Configuration and interface management: Configuration management must address three configurations: the actual existing configuration, the modification configuration, and the final system configuration. The key to successful modification is the level of understanding and control associated with the interfaces.

Logistics compatibility problems: Modification will change the configuration, which in most cases will change the supply support and maintenance considerations. Coordination with the logistics community is essential to the long-term operational success of the modification.

Minimal resources available: Modifications tend to be viewed as simple changes. As this chapter has pointed out, they are not; and they should be carefully planned. That planning should include an estimate of needed resources. If the resources are not available, either the project should be abandoned, or a plan formulated to mitigate and control the risk of an initial, minimal budget combined with a plan for obtaining additional resources.

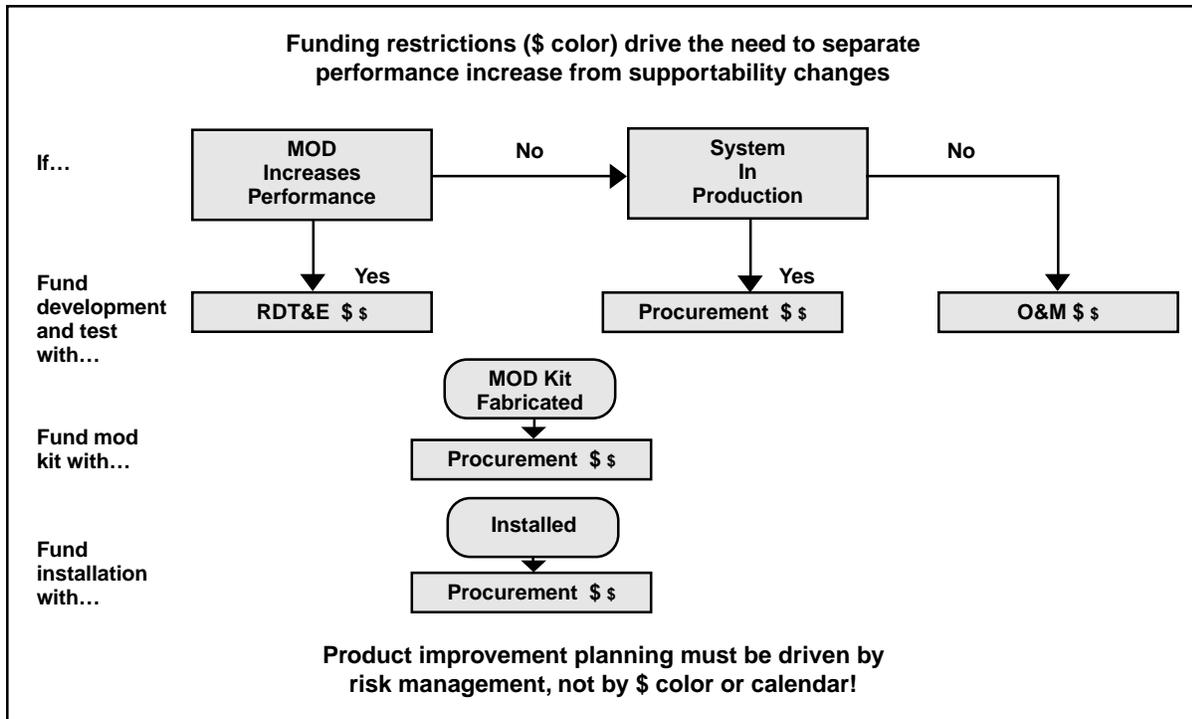


Figure 17-4. Funding Rule for DoD System Upgrades

Limited competitors: Older systems may have only a few suppliers that have a corporate knowledge of the particular system functions and design. This is especially problematic if the original system components were commercial or NDIs that the designer does not have product baseline data for. In cases such as these, there is a learning process that must take place before the designer or vendor can adequately support the modification effort. Depending on the specific system, this could be a major effort. This issue should be considered very early in the modification process because it has serious cost implications.

Government funding rules: As Figure 17-4 shows the use of government funding to perform system upgrades has restrictions. The purpose of the upgrade must be clear and justified in the planning efforts.

17.3 ROLES AND RESPONSIBILITIES

Modification management is normally a joint government and contractor responsibility. Though any

specific system upgrade will have relationships established by the conditions surrounding the particular program, government responsibilities would usually include:

- Providing a clear statement of system requirements,
- Planning related to government functions,
- Managing external interfaces,
- Managing the functional baseline configuration, and
- Verifying that requirements are satisfied.

Contractor responsibilities are established by the contract, but would normally include:

- Technical planning related to execution,
- Defining the new performance envelope,
- Designing and developing modifications, and

- Providing evidence that changes made have modified the system as required.
- Ensuring operations, support activities, and early field results are considered in planning.

System Engineering Role

The systems engineering role in product improvement includes:

- Planning for system change,
- Applying the systems engineering process,
- Managing interface changes,
- Identifying and using interface standards which facilitate continuing change,
- Ensuring life cycle management is implemented,
- Monitoring the need for system modifications, and

17.4 SUMMARY POINTS

- Complex systems do not usually have stagnant configurations.
- Planned improvements strategies include evolutionary acquisition, preplanned product development, and open systems.
- A major rebuild should be treated as a new system development.
- Upgrading an existing system is a matter of following the system engineering process, with an emphasis on configuration and interface management.
- Pay attention to the traps. Upgrade projects have many.

SUPPLEMENT 17-A

OPEN SYSTEM APPROACH

The open system approach is a business and technical approach to system development that results in systems that are easier to change or upgrade by component replacement. It is a system development logic that emphasizes flexible interfaces and maximum interoperability, optimum use of commercial competitive products, and enhanced system capacity for future upgrade. The value of this approach is that open systems have flexibility, and that flexibility translates into benefits that can be recognized from business, management, and technical perspectives.

From a management and business view, the open system approach directs resources to a more intensive design effort with the expectation of a life cycle cost reduction. As a business approach it supports the DoD policy initiatives of CAIV, increased competition, and use of commercial products. It is a technical approach that emphasizes

systems engineering, interface control, modular design, and design for upgrade. As a technical approach it supports the engineering goals of design flexibility, risk reduction, configuration control, long-term supportability, and enhanced utility.

Open Systems Initiative

In DoD the open system initiative was begun as a result of dramatic changes in the computer industry that afforded significant advantages to design of C4ISR and IT systems. The standardization achieved by the computer industry allows C4ISR and IT systems to be designed using interface standards to select off-the-shelf components to form the system. This is achieved by using commercially-supported specifications and standards for specifying system interfaces (external and internal, functional and physical), products, practices, and tools. An open system is one

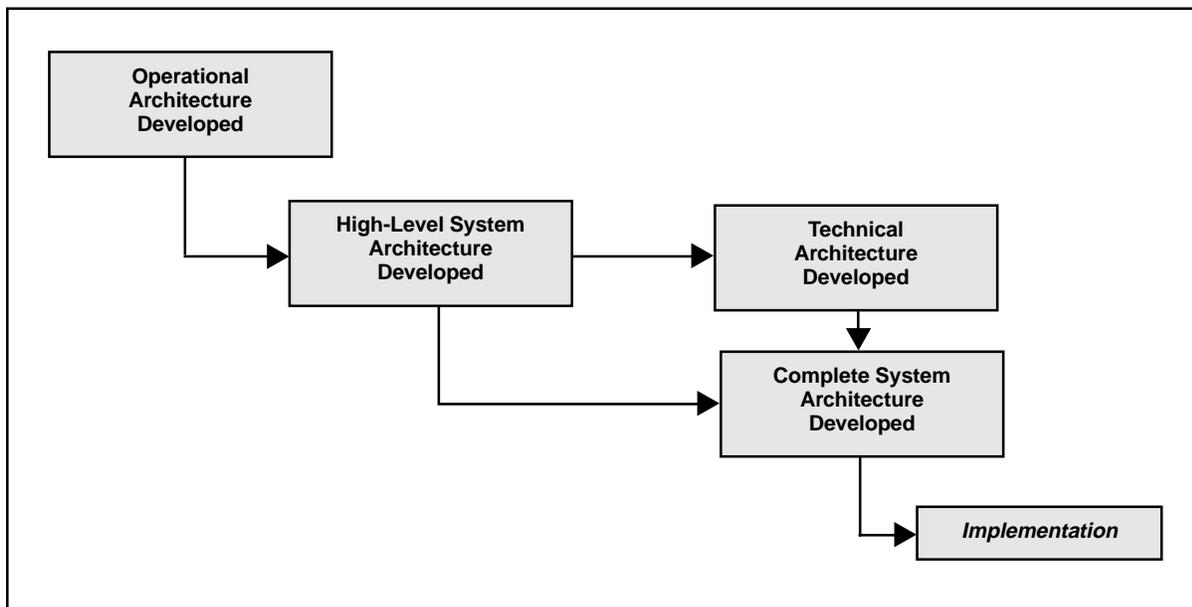


Figure 17-5. C4I and IT Development

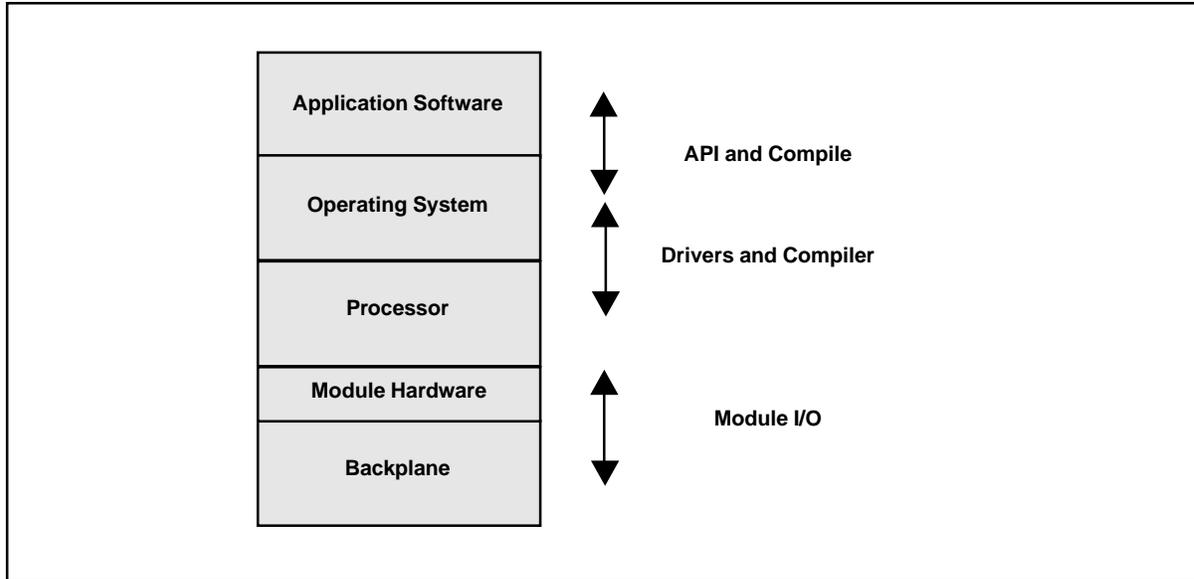


Figure 17-6. Simplified Computer Resource Reference Model

in which interfaces are fully described by open standards.¹ An open system approach extends this concept further by using modular design and interface design to enhance the availability of multiple design solutions, especially those reflecting use of open standards, competitive commercial components, NDIs, and future upgrade capability.

As developed in the C4ISR and IT communities, the open system approach requires the design of three architectures: operational, technical, and system.

As shown in Figure 17-5, the first one prepared is an operational architecture that defines the tasks, operational elements, and information flows required to accomplish or support an operational function. The user community generates the operational concepts that form an operational architecture. The operational architecture is allusive. It is not a specific document required to be developed by the user such as the ORD; but

because of their operational nature, the user must provide the components of the operational architecture. It is usually left to the developer to assemble and structure the information as part of the system definition requirements analysis. Once the operational architecture has clearly defined the operational need, development of a system architecture² is begun.

The (open) system architecture is a set of descriptions, including graphics, of systems and interconnections supporting the operational functions described in the operational architecture. Early in the (open) system architecture development a technical architecture is prepared to establish a set of rules, derived from open consensus-based industry standards, to govern the arrangement, interaction, and interdependence of the elements of a reference model. Reference models are a common conceptual framework for the type of system being designed. (A simple version for computer resources is shown in Figure 17-6.)

¹ Open Standards are non-proprietary, consensus-based standards widely accepted by industry. Examples include SAE, IEEE, and ISO standards.

² This system architecture typically describes the end product but not the enabling products. It relies heavily on interface definitions to describe system components.

The technical architecture identifies the services, interfaces, standards, and their relationships; and provides the technical guidelines upon which engineering specifications are based, common building blocks are built, and product lines are developed. In short, the technical architecture becomes a design requirement for developing the system. (The purpose, form, and function of the technical architecture is similar to building codes.)

The system architecture is then further developed to eventually specify component performance and interface requirements. These are then used to select the specific commercial components that form the system under development. This process, called an *implementation*, envisions the production process as consisting primarily of selecting components, conformance (to the interface and performance requirements) management, and assembly, with little or no need for detailed design fabrications.

The process described above has allowed significant achievements in computer-related developments. Other technical fields have also used the open system design approach extensively. (Common examples are the electrical outlets in your home and the tire-to-wheel interface on your car). In most cases the process is not as well defined as it is in the current digital electronics area. A consistent successful use of the open design concept, in and outside the electronics field, requires an understanding of how this process relates to the activities associated with systems engineering management.

Systems Engineering Management

The open system approach impacts all three essential elements of systems engineering management: systems engineering phasing, the systems engineering process, and life cycle considerations. It requires enhanced interface management in the systems engineering process, and requires specific design products be developed prior to engineering-event milestones. The open systems approach is inherently life-cycle friendly. It favorably impacts production and support functions, but it

also requires additional effort to assure life-cycle conformance to interface requirements.

Open Systems Products and SE Development Phasing

A system is developed with stepped phases that allow an understanding of the operational need to eventually evolve into a design solution. Though some tailoring of this concept is appropriate, the basic phasing (based on the operational concept preceding the system description, which precedes the preliminary design, which precedes the detailed design) is necessary to coordinate the overall design process and control the requirements flow-down. As shown by Figure 17-7 the open system approach blends well with these development phases.

Concept Studies Phase

The initial detailed operational concept, including operational architectures, should be a user-community output (with some acquisition engineering assistance) produced during the concept exploration phase that emphasizes operational concepts associated with various material solutions. The operational concept is then updated as necessary for each following phase. Analysis of the initial operational concept should be a key element of the operational view output of the system definition phase requirements analysis. An operational architecture developed for supporting the system description should be complete, comprehensive, and clear; and verified to be so at the Alternative Systems Review. If the operational architecture cannot be completed, then a core operational capability must be developed to establish the basis for further development. Where a core capability is used, core requirements should be complete and firm, and the process for adding expanded requirements should be clear and controlled.

System Definition Phase

System interface definitions, such as the technical architecture, and high-level (open) system architecture should be complete in initial form at the

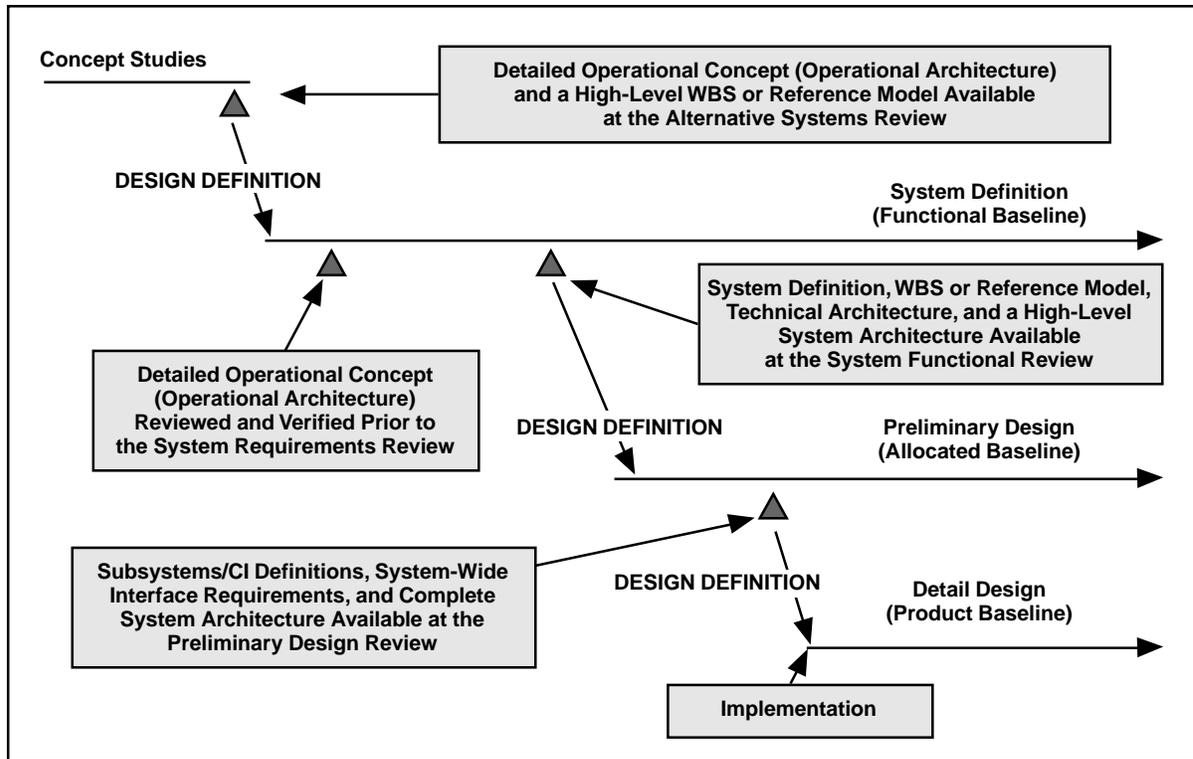


Figure 17-7. Phasing of Open System Development

end of the system definition phase (along with other functional baseline documentation). Successful completion of these items is required to perform the preliminary design, and they should be available for the System Functional Review, also referred to as the System Definition Review or System Design Review. The open system documentation can be separate or incorporated in other functional baseline documentation. The criteria for acceptance should be established in the systems engineering management plan as phase-exit criteria.

Preliminary Design Phase

Along with other allocated baseline documentation, the interface definitions should be updated and the open-system architecture completed by the end of the preliminary design effort. This documentation should also identify the proper level of openness (that is, the level of system decomposition at which the open interfaces are established) to obtain the maximum cost and logistic advantage available from industry practice.

The preliminary design establishes performance-based descriptions of the system components, as well as the interface and structure designs that integrate those components. It is in this phase that the open system approach has the most impact. Interface control should be enhanced and focused on developing modular designs that allow for maximum interchange of competitive commercial products. Review of the technical architecture (or interface definitions) becomes a key element of requirements analysis, open system focused functional partitioning becomes a key element of functional analysis and allocation, iterative analysis of modular designs becomes a key element of design synthesis, and conformance management becomes a key element of verification. Open system related products, such as the technical architecture, interface management documentation, and conformance management documentation, should be key data reviewed at the Preliminary Design Review. Again, the criteria for acceptance should be established in the systems engineering management plan as phase-exit criteria.

Detail Design Phase

The detail design phase becomes the implementation for those parts of the system that have achieved open system status. Conformance management becomes a significant activity as commercial components are chosen to meet performance and interface requirements. Conformance and interface design testing becomes a driving activity during verification to assure an open system or subsystem has been achieved and that components selected meet interface requirements and/or standards.

Systems Engineering Process

The systems engineering problem solving process consists of process steps and loops supported by system analysis and control tools. The focus of the open systems engineering process is compartmentalized design, flexible interfaces, recognized interface standards, standard components with recognized common interfaces, use of commercial and NDIs, and an increased emphasis on interface control. As shown by Figure 17-8, the open-system approach complements the systems engineering process to provide an upgradeable design.

Requirements analysis includes the review and update of interface standards and other interface definitions generated as output from previous systems engineering processes. Functional analysis and allocation focuses on functional partitioning to identify functions that can be performed independent of each other in order to minimize functional interfaces. Design synthesis focuses on modular design with open interfaces, use of open standards compliant commercial products, and the development of performance and interface specifications. The verification processes include conformance testing to validate the interface requirements are appropriate and to verify components chosen to implement the design meet the interface requirements. Engineering open designs, then, does not alter the fundamental practices within systems engineering, but, rather, provides a specific focus to the activities within that process.

System Engineering Control: Interface Management

The key to the open systems engineering process is interface management. Interface management should be done in a more formal and comprehensive manner to rigidly identify all interfaces and

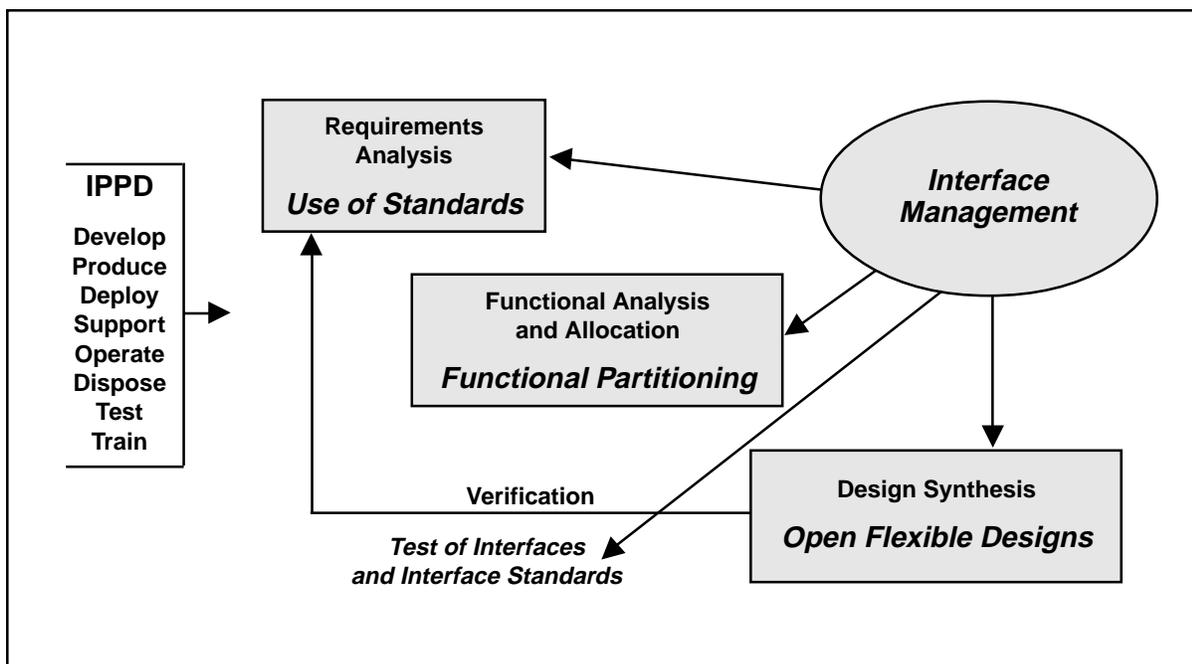


Figure 17-8. Open System Approach to the Systems Engineering Process

control the flowdown and integration of interface requirements. The interfaces become controlled elements of the baseline equal to (or considered part of) the configuration. Open system interface management emphasizes the correlation of interface requirements between interfacing systems. (Do those designing the interfacing systems understand the interface requirements in the same way?) Computer-Aided System Engineering (CASE) generated schematic block diagrams can be used to track interface design activity.

An open system is also characterized by multiple design solutions within the interfaces with emphasis on leveraging best commercial practice. The interface management effort must control interface design such that interfaces specifically chosen for an open system approach are designed based on the following priority:

- Open standards that allow competitive products,
- Open interface design that allows installation of competitive products with minimal change,
- Open interface design that allows minimal change installation of commercial or NDI products currently or planned to be in DoD use, and last,
- Unique design with interfaces designed with upgrade issues considered.

Note that these are clear priorities, not options.

Level of Openness

The level at which the interface design should focus on openness is also a consideration. Each system may have several levels of openness depending on the complexity of the system and the differences in the technology within the system. The level chosen to define the open interfaces should be supported by industry and be consistent with program objectives. For example, for most digital electronics that level is the line-replaceable (LRU) and shop-replaceable (SRU) level. On the other hand the Joint Strike Fighter intends to establish openness at a very high subsystem level to achieve

a major program objective, development of different planes using common building blocks (which, in essence, serve as the reference model for the family of aircraft). The open system approach designed segments of a larger system could have additional openness at a lower level. For example, the Advanced Amphibious Assault Vehicle (AAAV) engine compartment is an open approach design allowing for different engine installation and future upgrade capability. On a lower level within the compartment the fuel filters, lines, and connectors are defined by open standard based interfaces. Other systems will define openness at other levels. Program objectives (such as interoperability, upgrade capability, cost-effective support, affordability, and risk reduction) and industry practice (based on market research) drive the choice of the level of openness that will best assure optimum utility and availability of the open system approach.

Life Cycle Considerations

Life cycle integration is established primarily through the use of integrated teaming that combines the design and life cycle planning. The major impacts on life-cycle activity include:

- ***Time and cost to upgrade a system is reduced.*** It is common in defense systems, which have average life spans in excess of 40 years, that they will require upgrade in their life due to obsolescence of original components, threat increase, and technology push that increases economy or performance. (Most commercial products are designed for a significantly shorter life than military systems, and designs that rely on these commercial products must expect that original commercial components will not necessarily be available throughout the system's life cycle.) By using an open system approach the ability to upgrade a system by changing a single or set of components is greatly enhanced. In addition, the open system approach eases the design problem of replacing the component, thereby reducing the cost and schedule of upgrade, which in turn reduces the operational impact.

- ***An open system approach enhances the use of competitive products to support the system.*** This flexibility tends to reduce the cost associated with supply support, but more importantly improves component and parts availability.
- ***Conformance management becomes a part of the life cycle configuration process.*** Replacement of components in an open system must be more controlled because the government has to control the system configuration without controlling the detail component configuration (which will come from multiple sources, all with different detail configurations). The government must expect that commercial suppliers will control the design of their components without regard to the government's systems. The government therefore must use performance- and interface-based specifications to assure the component will provide service equivalent to that approved through the acquisition process. Conformance management is the

process that tracks the interface requirements through the life cycle, and assures that the new product meets those requirements.

Summary Comments

Open system design is not only compatible with systems engineering; it represents an approach that enhances the overall systems engineering effort. It controls interfaces comprehensively, provides interface visibility, reduces risk through multiple design solutions, and insists on life cycle interface control. This emphasis on interface identification and control improves systems engineers' capability to integrate the system, probably one of the hardest jobs they have. It also improves the tracking of interface requirements flow down, another key job of the systems engineer. Perhaps most importantly, this rigorous interface management improves systems engineers' ability to correctly determine where commercial items can be properly used.