Chapter 5

System Life Cycle and Methodologies

Contents

5.1 Life Cycle Process and Decision Making

Making decisions without full knowledge of the situation may at times be a necessity of battle, but it can prove costly in terms of quality, safety, and performance when procuring weapons to go to war. The Office of Management and Budget Circular A-109, published in 1976, provides direction for the acquisition process and defines a decision mechanism based on quantitative assessments, reviews, and audits of the life cycle process. It established policies, methods, procedures, a life cycle, and milestone decision process to increase effectiveness in decision making for all major system acquisitions. For weapon systems, C2, and AIS programs, Milestone Decisions mark the completion of one phase of the life cycle and entry into the next. Peer reviews, completion of measured process activities, the production of defined work products, audits, and other evaluation procedures throughout each phase support exit and entry criteria for the milestone decisions.

5.1.1 System-of-Systems View

To understand the system life cycle and its acquisition phases, it is important to realize how they relate to the systems and software engineering processes [discussed in Chapter 9, *Engineering Software-Intensive Systems*]. Because software must always interface with the other elements that make up the total system, a *system-of-systems* view is critical. For example, AIS systems often interface with hundreds of other independent AIS or C2 systems. Likewise, embedded avionics software often interfaces with a multitude of internal/external sensors and flight control systems. Given the relationships and interdependencies among all system components, it is vital to maintain a big picture, *systems-view*, as expressed by Field Marshall Viscount Montgomery.

> *"It is absolutely vital that a senior commander should keep himself from becoming immersed in details...In battle a commander has got to think how he will defeat the enemy. If he gets involved in details he cannot do this since he will lose sight of the essentials which really matter; he will then be led off on side issues which will have little influence on the battle. No commander whose daily life is spent in the consideration of details…can make a sound plan of battle on a high level or conduct large-scale operations efficiently."* [MONTGOMERY58]

To optimize total system performance and minimize Total Cost of Ownership, acquisition managers must employ a total system approach. According to **DoDD 5000.1**, a total system includes the following subsystems, as illustrated in Figure 5-1.

- The prime mission equipment (i.e., hardware, software, and documentation);
- The people who operate and maintain the system;
- System security procedures and practices;
- Operational procedures (tactics), practices (doctrine), limitations (rules of engagement), and characteristics (mission);
- Performance capabilities required respond to operational environment unique effects (e.g., nuclear, biological and chemical (NBC) or information warfare);
- Deployment procedures and requirements;
- Compatibility, interoperability, and integration capabilities with other systems;

- Operational and support infrastructure (including command, control, communications, computers and intelligence (C4I));
- Training and training devices;
- Any required operational data; and
- The system's potential impact on the environment and environmental compliance.

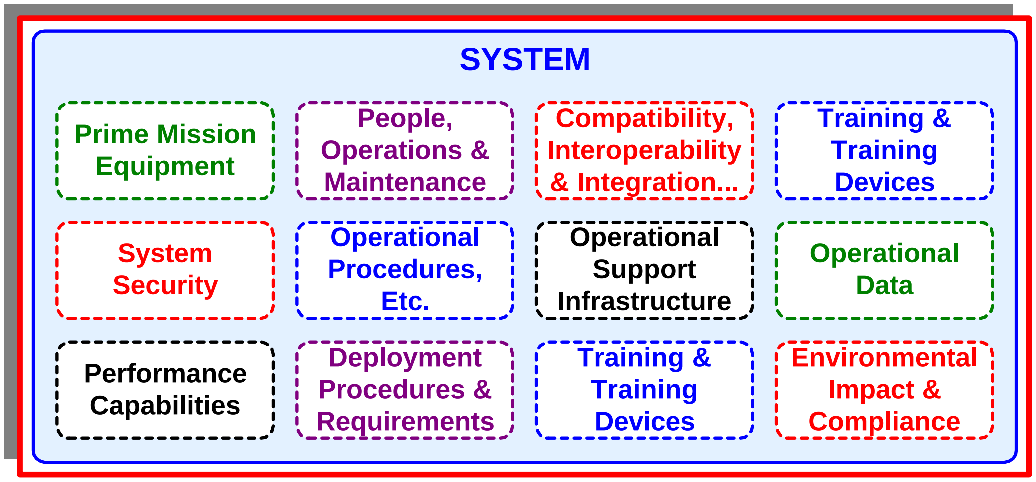

Figure 5-1. System of Systems

The life cycle phases for software-intensive weapon systems and Major Automated Information Systems (MAIS) domains are similar. For both domains, software is always on the critical path (whether it is developed independently, concurrently, and/or purchased separately).

For all systems, it is important to consider how the software will work within the system. This can only be accomplished with a comprehensive, robust system architecture (or blueprint). [Architecture is discussed in Chapter 11, *Understanding Software Development*.] The system architecture is the definition of hardware and software components and their interfaces that establish a framework for the system's development. Well-constructed interfaces are necessary to achieve cohesive, interoperable components early in the acquisition cycle. Proper hardware and software integration is only assured through carefully defined interface requirements, prudently planned prototype demonstrations, and system/subsystem tests and evaluations. Such techniques improve accuracy, currency, and quality of decision-critical information. Interfaces are of three types:

1. Software-to-software,
2. Software-to-hardware, and
3. Hardware-to-human.

5.2 DoD Decision Support Systems

The acquisition life cycle is an integral part of DoD's three main decision support systems, illustrated in Figure 5-2.

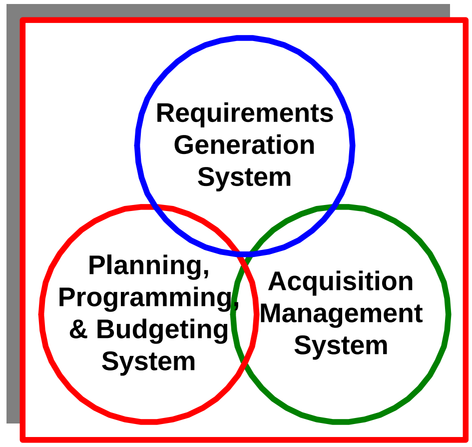

Figure 5-2. DoD Decision Support Systems [PIPLANI94]

The policies stated in the **DoDD 5000.1**, *Defense Acquisition*, and **DoD 5000.2-R**, *Mandatory Procedures for Major Defense Acquisition Programs*, forge a close relationship among these three systems, which operate continuously and concurrently to assist the SECDEF and other senior DoD officials in making critical national security, acquisition, and budgeting decisions. Early in the acquisition life cycle, the Milestone Decision Authority (MDA) [discussed in Chapter 4, *DoD Software Acquisition Environment*] establishes tailored Milestone Decision criteria for ACAT I and ACAT IA programs. In compliance with the Government Performance and Results Act (GPRA), and DoD's Strategic Plan [the Quadrennial Defense Review (QDR), discussed in Chapter 3, *Statutory Framework Governing Software Acquisition*] the MDA determines whether a major acquisition program is progressing satisfactorily at each milestone decision/program review. [DoD 5000.2-R]

5.2.1 Requirements Generation System

The Requirements Generation System, governed by **CJCS Instruction 3170.01**, *Requirements Generation System*, produces decision-critical information on projected mission needs requiring joint Major Defense Acquisition Programs (MDAPs) and MAISs to support the warfighter. Complementary guidance for MAIS functional areas is provided in **DoD 8000.1**, *Defense Information Management Program*. Requirements generation is a continuing process of assessing the capabilities of the current force structure to meet projected threats. It takes into account opportunities for technological advancement, cost savings, and changes in national policy or doctrine. Support plans are also addressed during requirements generation. They focus on issues of interoperability, system-of-systems initiatives, MAIS, and Operational Requirements Document (ORD) (discussed below) compliance. Figure 5-3 illustrates this process.

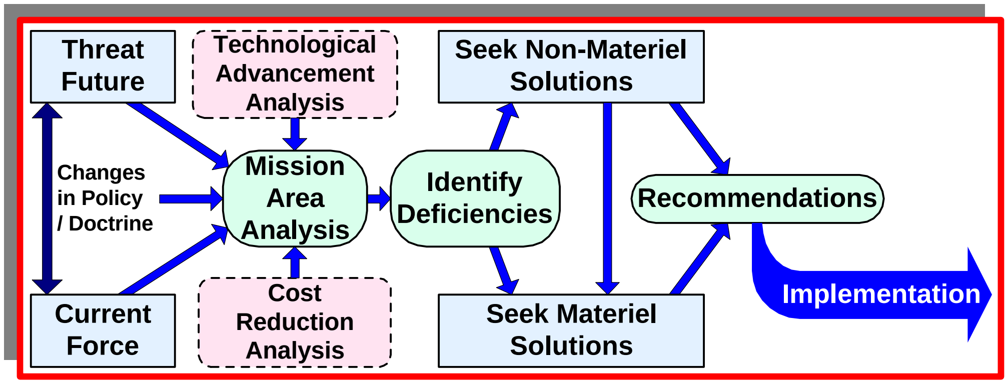

Figure 5-3. Requirements Generation System [PIPLANI94]

5.2.2 Acquisition Management System

The acquisition management system, governed by DoD Directive 5000.1, provides a streamlined, event-driven management structure that emphasizes risk management and affordability. It explicitly links milestone decisions to demonstrated accomplishments. The activities that are managed by this system are illustrated in Figure 5-4.

Figure 5-4. Acquisition Management Activities [ANDERSON98]

5.2.3 Planning, Programming, & Budgeting System (PPBS)

The Planning, Programming, and Budgeting System (PBBS) provides the basis for making informed affordability assessments and resource allocation decisions on defense acquisition programs. It is governed by **DoDD 7045.14**, *The Planning, Programming, and Budgeting System (PPBS)*, and is discussed in Chapter 7, *Acquisition Planning*.

5.3 Life Cycle Phases, Decisions, and Activities

DoD's system acquisition process was designed to manage a program through sequential phases. Each phase is followed by a major Milestone Decision in which decision-makers approve/ disapprove the acquisition strategy and its evolution into the next phase based on program progress reported by the Program Manager (PM). DoD 5000.2-R divides the life cycle into the following phases and milestone decision points, as illustrated in Figure 5-5.

1. Concept exploration;
2. Program definition and risk reduction;
3. Engineering and manufacturing development (EMD) which includes software engineering and development for software-intensive systems, and
4. Production, fielding/deployment, and operational support. [ANDERSON98]

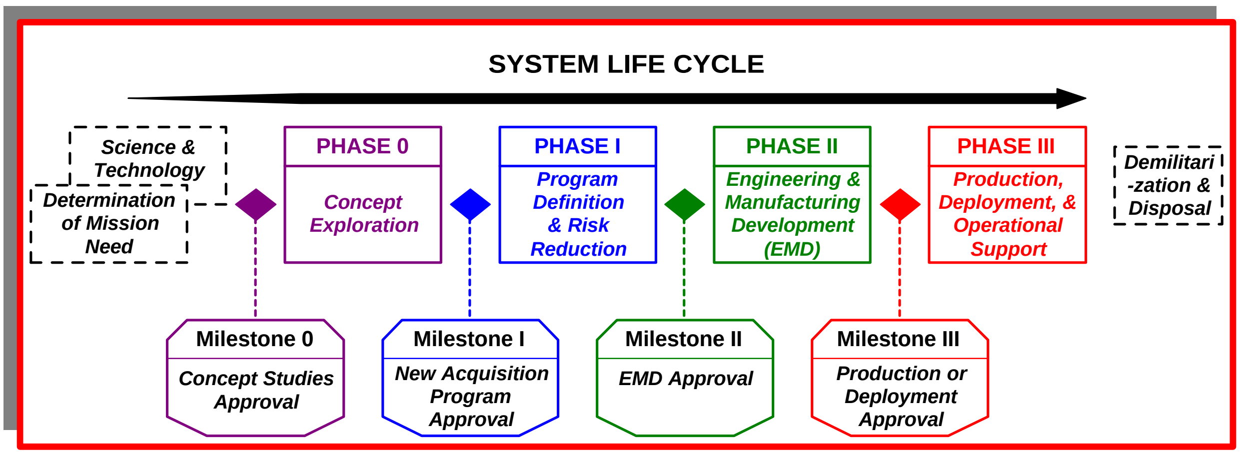

Figure 5-5. System Life Cycle [ANDERSON98]

At each Milestone Decision, assessments on program status and the plans for the next phase and the remainder of the program are made. The risks associated with the program and the adequacy of risk management planning are explicitly addressed, as illustrated in Figure 5-6. [See Chapter 6, *Risk Management*.] Additionally, program-specific results required in the next phase, called *exit criteria*, are established and approved. [ANDERSON98] Figure 5-7 summarizes DoD 5000.2R life cycle phases, Milestone Decisions, and activities discussed below.

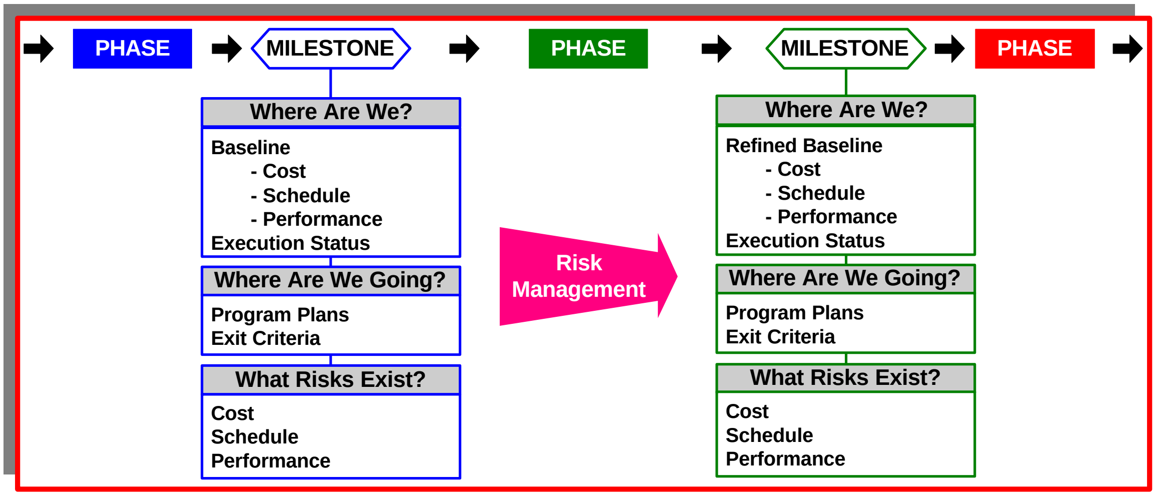

Figure 5-6. Risk Management and the Life Cycle Process [ANDERSON98]

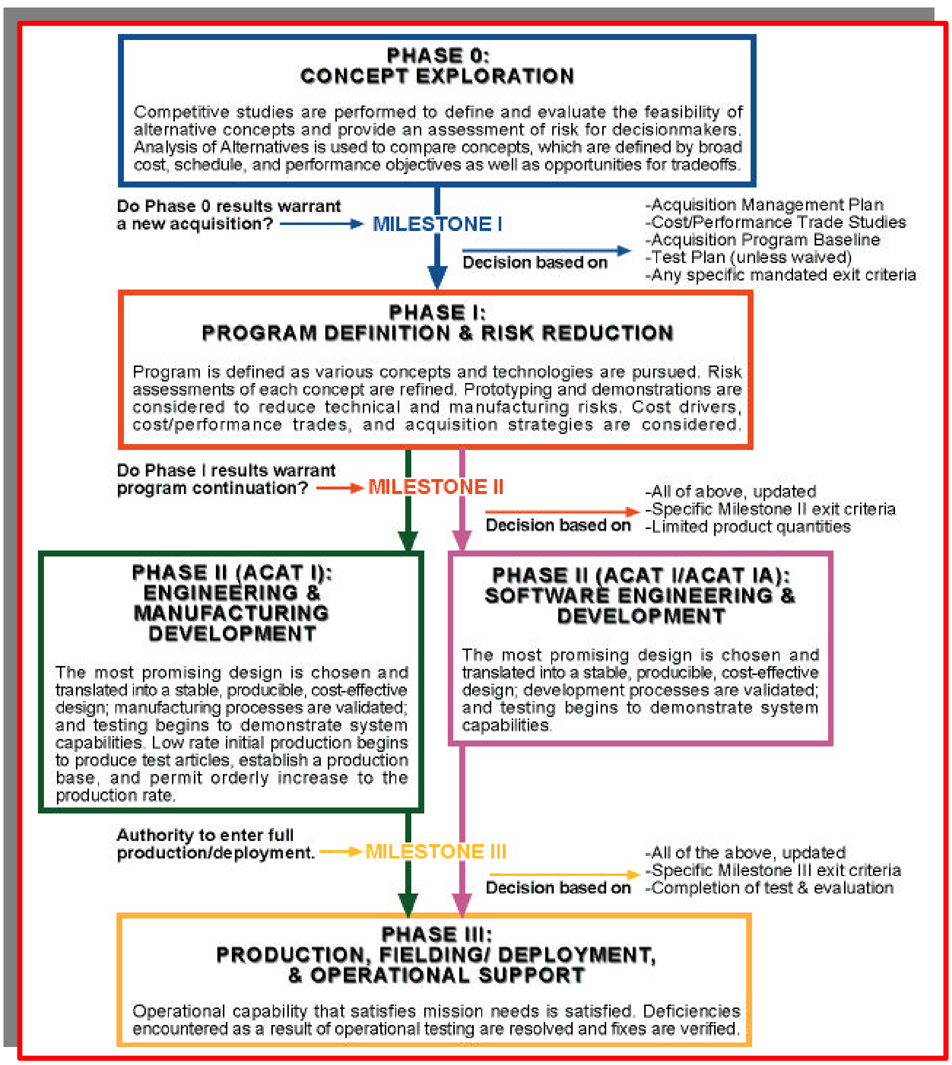

Figure 5-7. Summary of Life Cycle Phases and Acquisition Milestones Decisions for ACAT I and ACAT IA Programs [HINTON98]

ACAT I and ACAT IA Programs. For automated information systems (AIS) programs, the MDA should determine the appropriate acquisition phase for AISs designated to evolve to migration systems. AISs, designated as *migration systems* by an OSD Principal Staff Assistant (PSA), often require validation or revalidation of previous milestone decisions at the appropriate acquisition review.

5.3.1 Pre-Phase 0: Mission Need Determination

All DoD acquisition programs are based on identified, documented, and validated mission needs. Mission needs result from ongoing assessments of current and projected Defense capabilities. The mission need determination process is shown in the top half of Figure 5-8. Mission needs are identified to accomplish the following:.

- Establish a new operational capability;
- To improve an existing capability; or
- To exploit an opportunity to reduce costs or enhance performance. [DoD 5000.2-R]

Pre-Phase 0 activities and documentation for ACAT I programs are summarized in Figure 5-8.

NOTE: The activities and documentation are the same for ACAT IA programs except the memo goes to ASD (C3I) and instead of the Defense Acquisition Board (DAB) it is the Information Technology Overarching Integrated Product Team (IT OIPT).

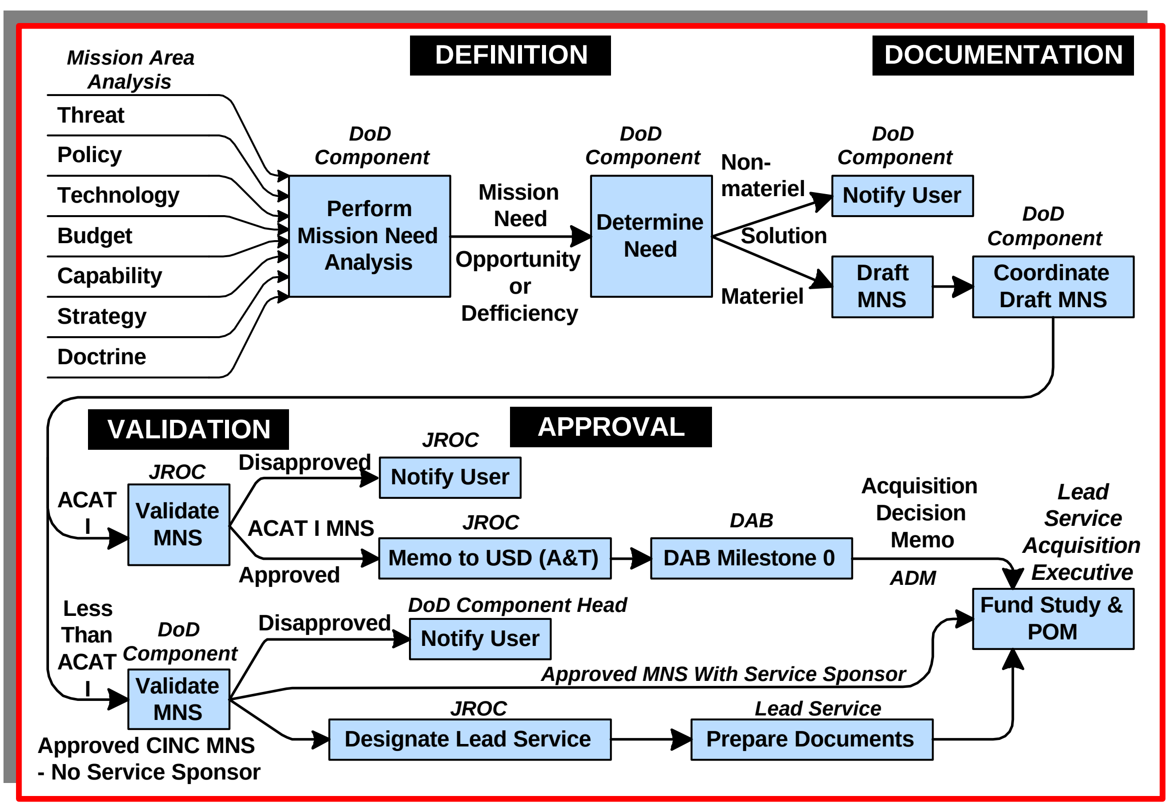

Figure 5-8. Summary of Pre-Phase 0 Activities [CJCSI 3170.01]

5.3.1.1 Mission Need Documentation

During this Pre-Milestone 0 Phase, the need for an acquisition program is studied and recorded in the following documents.

- **Mission Area Assessment** (MAA). The MAA identifies mission needs using a *strategy-to-task* process, which links the need for military capabilities to the strategy provided by the Chairman of the Joint Chiefs of Staff (CJCS).
- **Mission Need Analysis (MNA)**. The ability to accomplish the *tasks* from the *strategy-to-task* process using current and programmed systems is evaluated in the MNA. This process is called "*task-to-need*."
- **Mission Area Plan (MAP)**. The products of MAAs and MNAs are used to develop a MAP, a strategic planning document covering approximately 25 years. It records the proposed plan for correcting identified mission deficiencies. It expresses nonmateriel solutions, including changes in force structure, system modifications or upgrades, science and technology applications, and new acquisition programs.
- **Mission Need Statement (MNS)**. Upgrade, modification, and new acquisition programs are established when nonmateriel solutions will not adequately fulfill an identified mission deficiency. The MNS is a brief statement that identifies and documents mission deficiencies that require materiel and/or software solutions:
  – To define an operational need,
  – To officially validate an operational need, and
  – To furnish implementation and support to OT&E activities.
- **ACAT IA MNS**. For command, control, communications, and intelligence (C3I) systems, the Mission Need Statement (MNS) is submitted for validation and approval in accordance with **DoDD 4630.5**, *Compatibility, Interoperability, and Integration of Command, Control, Communications, and Intelligence (C3I) Systems*. In the case of automated information system (AIS) migration systems, the complete MNS is validated and approved at Milestone 0 and updated (if appropriate) at the time the AIS is designated a migration system. [ANDERSON98]
- **Operational Requirements Document (ORD)**. Once an acquisition program is approved, operational requirements for selected concept(s) progressively evolve. Broad operational capability needs identified in the MNS become system-specific performance requirements documented in the ORD, as illustrated in Figure 5-9 The ORD is prepared along with life cycle cost estimates, logistic support analysis, and producibility engineering assessments. The ORD is solution-oriented and becomes the basis for the following:
  – Program direction.
  – Program baselines.
- **Integrated Master Plan (IMP)**. The IMP is an event-based program plan that documents all the tasks required to deliver a high quality product and facilitate success throughout the product's life cycle. [Cost, schedule (specific dates), and non-essential tasks are not included in this plan.]
- **Integrated Master Schedule (IMS)**. The IMS begins as an IMP with dates — the starting points are the events, accomplishments, and criteria that make up the plan. Under acquisition reform initiatives, the dates in the IMP usually are not made contractually binding to allow flexibility in taking advantage of event-driven scheduling.

- **Test and Evaluation Master Plan (TEMP)**. The TEMP describes the program's overall test and evaluation strategy. It is prepared as early as possible in the acquisition process (normally prior to Milestone I). It is designed to identify and integrate objectives, responsibilities, resources, and schedule for all test and evaluation to be accomplished prior to key decision milestones.
- **Capstone Requirements Document (CRD)**. The CRD contains performance-based requirements to facilitate development of individual ORDs by providing a common framework and operational concept to guide their development.

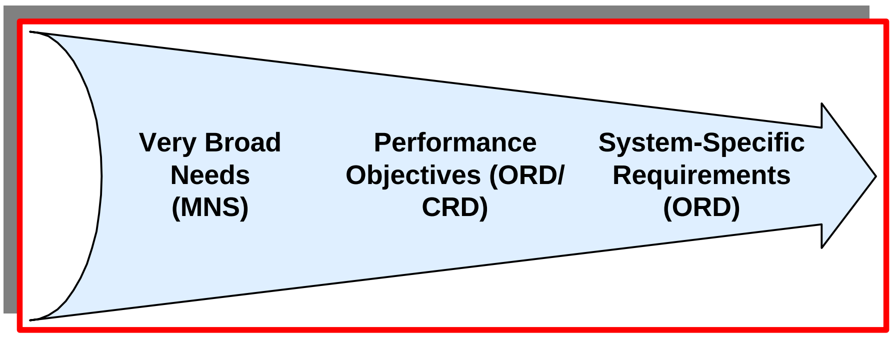

Figure 5-9. Evolution of Acquisition System Requirements Documents [CJCS3170.10]

5.3.1.2 Mission Need Validation

Nonmaterial solutions to mission needs (such as changes in doctrine or tactics) are analyzed first. If a nonmaterial solution is not feasible, all considerations are documented, and a determination is made as to whether the potential materiel solution could result in an ACAT I or ACAT IA program.

- **ACAT I Programs**. If the potential materiel solution results in a new ACAT I program, the Joint Requirements Oversight Council (JROC) reviews the documented mission need, determines its validity, and establishes joint potential. The mission need validation process for ACAT I programs is summarized in Figure 5-8. [NOTE: The process is the same for ACAT IA, except that the memo goes to ASD(C3I).]
- **ACAT IA Programs**. If the potential solution results in a new ACAT IA, the appropriate OSD PSA or the JROC reviews the documented need, determines its validity, establishes joint potential, and confirms that the requirements [defined in **DoD 8000.1**] have been met. [DoDD 5000.1]

5.3.1.3 Milestone 0 Decision: Approval to Conduct Concept Studies

The Concept Studies Approval (Milestone 0 Decision) is made when the Milestone Decision Authority (MDA) determines the following:

- The MNS is validated,
- The need cannot be satisfied by a nonmateriel solution,
- The need is sufficiently important to warrant funding study efforts to explore and define alternative concepts;
- Command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR) support requirements can be met; and
- For an ACAT IA program, an Analysis of Alternatives (AOA) has been considered. [DoD 5000.2-R]

- **Analysis of Alternatives (AOA)**. An evaluation of the advantages and disadvantages of alternatives being considered to satisfy a requirement, to include the sensitivity of each alternative to possible changes in key assumptions or variables. The analysis aids decision-makers in judging whether or not any of the alternatives offer sufficient benefit to be worth the cost.

[NOTE: A favorable Milestone 0 Decision does not mean a new acquisition program has been initiated.]

- **ACAT 1 Programs**. After the JROC validates the mission need for an ACAT I program, USD(A&T) convenes a Milestone 0 Defense Acquisition Board (DAB) to review the MNS, identify possible materiel alternatives, and authorize concept studies, if necessary.
- **ACAT IA Programs**. The JROC (or cognizant OSD PSA), validates the mission need and acquisition process integrity in compliance with DoDD 8000.1. ASD(C3I) convenes a Milestone 0 Information Technology Overarching Integrated Product Team (IT OIPT). For C3I systems, the MNS should be submitted for validation and approval in accordance with DoDD 4630.5. In the case of AIS migration systems, the complete MNS is validated and approved at Milestone 0 and updated, if appropriate.

5.3.2 Phase 0: Concept Exploration and Definition

Following a successful Milestone 0 Decision, Concept Exploration and Definition involves a series of studies to define and evaluate the feasibility of alternative concepts and their relative merits (i.e., advantages, disadvantages, degree of risk, etc.). The most promising system concepts are defined in terms of the following initial, broad objectives:

- Cost
- Schedule
- Performance
- Software requirements
- Tradeoff opportunities
- Overall acquisition strategy
- Test and evaluation strategy
- Readiness objectives

These concepts are often explored through competitive, parallel, short-term contracts, which provide the basis for assessing the merits of alternative concepts at the Milestone I Decision. [DoD 5000.2-R]

NOTE - A list of recommended items to include in a draft ORD is found in Volume 2, Appendix T, *Automated Information Systems (AIS) Operational Requirements Documents (ORDs) Recommendations.*

A product of this phase is the selection of a proposed Acquisition Strategy [discussed in Chapter 7, *Acquisition Planning*]. Demonstration program(s) are designed, coded, tested, and implemented to provide basic (or elementary) capabilities across the full range of requirements. During Phase 0, the following activities are normally performed:

PHASE 0 ACTIVITIES	ACAT I	ACAT IA
1. A validated assessment of the military threat is created.	X	
2. Potential environmental consequences are identified.	X	
3. Major technology and industrial capability issues are analyzed.	X	
4. Cooperative opportunities are identified.	X	
5. Compliance with international arms control agreements is assured.	X	
6. Technology and technical risk is considered.	X	X
7. Advantages and disadvantages of alternative concepts are assessed.	X	X
8. An Acquisition Strategy is identified.	X	X
9. Cost, schedule, and performance for approval are defined.	X	X
10. Program-specific objectives for the next phase are defined.	X	X
11. A proposed oversight and review strategy to include a description of mandatory program information and when this information needs to be submitted for the next milestone decision is developed.	X	X
12. System requirements in terms of measures of effectiveness (MOE), measures of performance (MOP), and C4ISR support requirements are defined.	X	X

Table 5-1. Phase 0 Activities for ACAT I and ACAT IA Programs

Rapid prototyping should be used to support analyses performed during this and the next phase (and throughout the life cycle, as appropriate). Rapid prototyping can also be used to develop a subset of functional capabilities to be released to a limited user community for shakedown. Rapid prototyping is approved at the milestone decision point before its use. Rapid prototyping, modeling, and simulation are discussed in Chapter 11, *Understanding Software Development*.

5.3.2.1 Milestone I Decision: Approval to Begin a New Acquisition

The purpose of the Milestone I Decision is to determine whether the results of Phase 0 warrant establishing a new acquisition program and to approve entry into Phase I, Program Definition and Risk Reduction. At Milestone I Decision, the MDA normally considers the following.

MILESTONE I CONSIDERATIONS	ACAT I	ACAT IA
Threat assessment.	X	
1. Environmental consequences	X	
2. Hierarchy of materiel alternatives	X	
3. Acquisition Strategy	X	X
4. Phase 0 exit criteria status and Phase I exit criteria plans	X	X
5. Acquisition Program Baseline (APB) (CAIV-based objectives)	X	X
6. AOA and concept studies supporting the need for a new program	X	X
7. Adequacy of resources (manpower and funding)	X	X
8. Affordability assessment	X	X
9. Updated C4ISR support requirements. [DoD 5000.2-R]	X	X

Table 5-2. Milestone I Decision Approval for ACAT I and ACAT IA Programs

5.3.3 Phase I: Program Definition and Risk Reduction

This phase [formerly Demonstration/Validation (Dem/Val)] follows a successful Milestone I Decision to proceed. One or more concepts, design approaches, and/or parallel technologies are pursued (as warranted) and advantage/disadvantage assessments of alternative concepts are refined. Prototyping, demonstrations, and early operational assessments are analyzed to reduce risk so that technology, production, and support risks are well in hand before the next milestone decision. The following are also analyzed:

- Cost drivers,
- Life cycle cost estimates,
- Cost-performance tradeoffs,
- Interoperability, and
- Alternative acquisition strategies are considered to include evolutionary and incremental software development. [DoD 5000.2-R]

An important aspect of this phase is the early integration of *supportability considerations* into the system design concept. As illustrated in Figure 5-10, the decisions made during this phase impact approximately 60% of total life cycle costs. As shown in Figure 5-11, much of these costs are incurred during the operations and support phase.

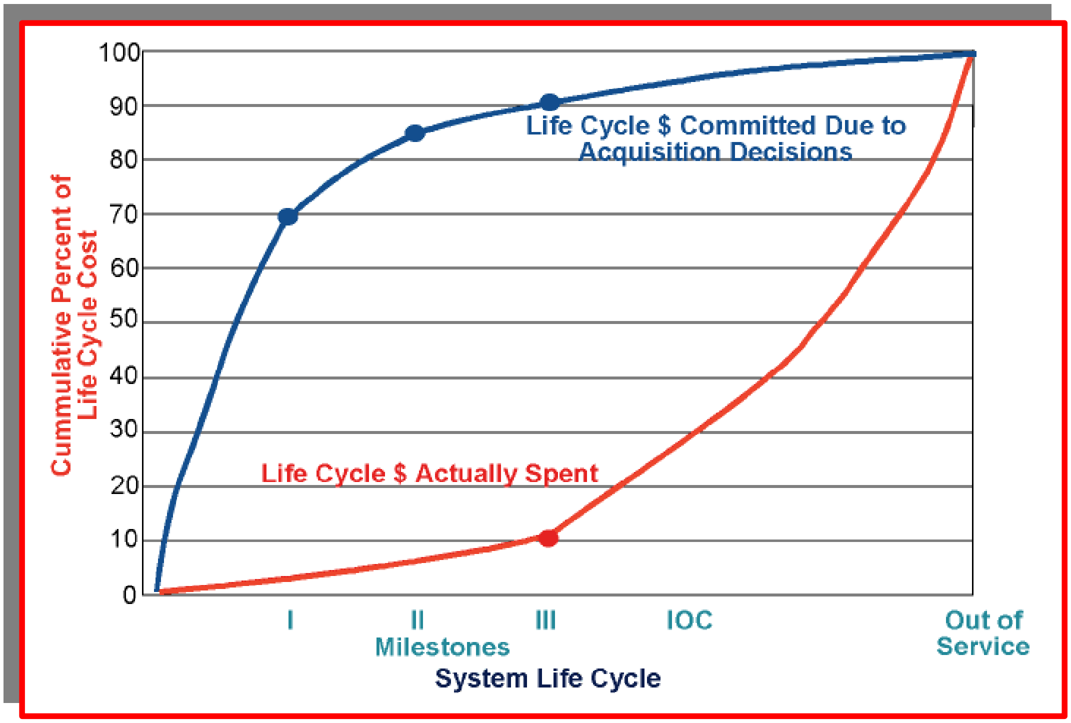

Figure 5-10. Effect of Early Decisions on Life Cycle Cost [DSMC90]

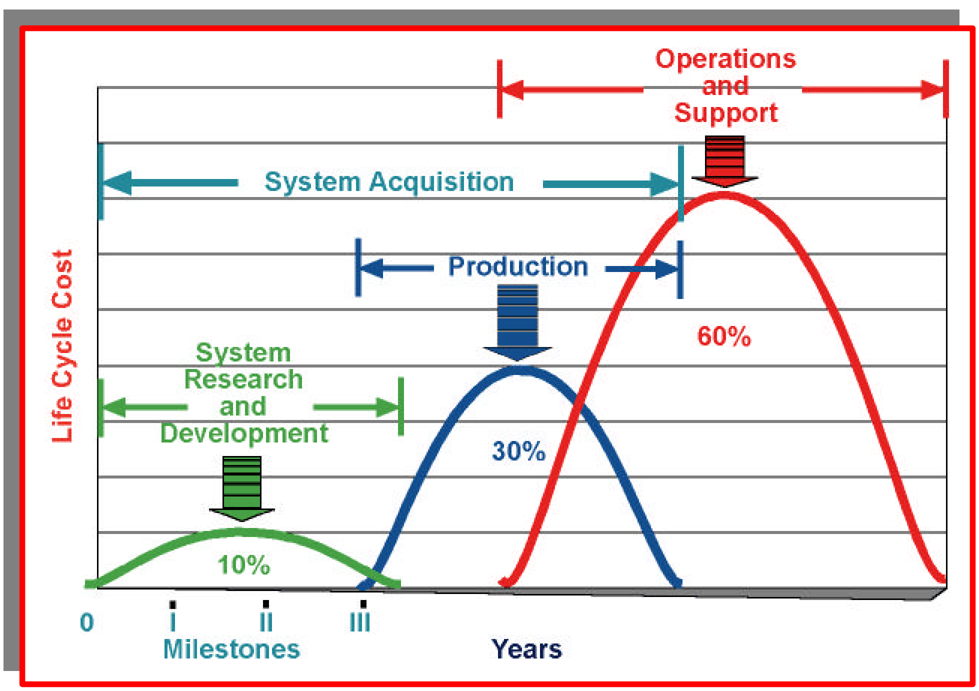

Figure 5-11. Nominal Cost Distribution of a Typical DoD ACAT 1 / ACAT IA Program [DSMC90]

Prototyping, testing, and *early user involvement* in operational assessments of critical components cannot be overemphasized as risk and cost reduction methods. As a function of risk, the costs of alternative design approach(es) must be evaluated against performance capabilities. [See Chapter 6, *Risk Management*.] The ORD, TEMP, and Acquisition Strategy are updated to reflect the work performed during Phase I. During Phase I, the following activities are normally performed:

PHASE I ACTIVITIES	ACAT I	ACAT IA
1. Assessment of the military threat is updated.	X	
2. Acquisition Strategy & low rate initial production (LRIP) quantities refined.	X	
3. Industrial capability to support the program is assessed.	X	
4. Potential environmental impacts are assessed.	X	
5. Cooperative opportunities are identified.	X	
6. Compliance with international arms control agreements is assured.	X	
7. Technology and technical risks are considered.	X	X
8. Cost objectives and affordability assessment are refined.	X	X
9. Major cost, schedule, and performance tradeoff opportunities are identified.	X	X
10. Test and evaluation strategy and appropriate testing requirements identified.	X	X
11. Proposed cost, schedule, and performance objectives and thresholds for approval are identified.	X	X
12. That adequate resources have been programmed to support production, deployment, and support is verified.	X	X
13. A proposed oversight and review strategy to include a description of mandatory program information and when this information needs to be submitted for the next milestone is developed.	X	X
14. CAIV objectives are refined.	X	X
15. Major technology and industrial capability issues are analyzed.	X	X
16. Independent Cost Estimate (ICE) and Manpower Estimate (ME) developed.	X	X
17. C4ISR support requirements are refined. [DoDD 5000.2R]	X	X

Table 5-3. Phase I Activities for ACAT I and ACAT IA Programs

5.3.3.1 Milestone II Decision: Engineering & Manufacturing Development/ Software Engineering & Development

At the Milestone II Decision, *Engineering & Manufacturing Development (EMD)/Software Engineering & Development* (for software-intensive programs), the MDA rigorously assesses the affordability of the program and establishes a development APB. Defense Planning Guidance, long-range modernization and investment plans, and DoD Component planning documents form the basis for this decision. Because there is a significant resource commitment associated with this decision, program risks and risk management plans are rigorously analyzed. The development APB involves effective interaction among the Requirements Generation, Acquisition Management, and PPBS systems. The Milestone II Acquisition Decision Memorandum (ADM) states the following:

- Approval of entry into Phase II, the proposed (or modified) APB, and the Acquisition Strategy,
- Life-cycle cost objectives and exit criteria have been established, and
- LRIP quantities have been identified (if appropriate for ACAT I).

The LRIP strategy, required information, and the following are normally considered by the MDA at this milestone.

MILESTONE II DECISION CONSIDERATIONS	ACAT I	ACAT IA
1. Validated threat assessment	X	
2. Waiver from full-up, system-level live fire T&E (LFT&E)	X	
3. Potential environmental consequences	X	
4. LRIP quantities	X	
5. Acquisition Strategy	X	X
6. APB (including CAIV-based objectives)	X	X
7. Status of Phase I exit criteria status and Phase II exit criteria plans	X	X
8. Prototyping/demonstration results	X	X
9. Adequacy of resources (manpower and funding)	X	X
10. ICE and Manpower Estimate	X	X
11. Updated C4ISR support requirement	X	X

Table 5-4. Milestone II Decision Approval for ACAT I and ACAT IA Programs

With the Milestone II Decision, the affordability of the program is assessed and a decision is made on whether the activities of this Phase I warrant continuation to the next phase. A Development Baseline is established reflecting cost, schedule, and performance requirements. [DoD 5000.2-R]

5.3.3.1.1 Milestone II Decision: Low Rate Initial Production Decision

A favorable LRIP Decision only authorizes the program manager (PM) to commence LRIP. The PM is authorized to commence full-rate production with further approval of the MDA.

5.3.4 Phase II: Engineering Manufacturing & Development/ Software Engineering & Development

Following a successful Milestone II Decision, *Engineering & Manufacturing Development (EMD)/ Software Engineering & Development* (for software-intensive programs), actual product development and/or manufacturing begins. Note that this phase may be repeated if the life cycle methods are incremental or evolutionary [discussed below]. Effective risk management is especially critical during this phase. To assist in managing risk, resources are only committed commensurate with the reduction and closure of risk elements. Configuration management control is established for the design, engineering, and management processes. Development and test activities focus on high-risk areas, address the operational environment, and are phased to support internal decision making and the Milestone III Decision review. The primary objectives of this phase include:

- Translating the most promising design approach into a stable, interoperable, producible, supportable, and cost-effective design;
- Validating the development process; and
- Demonstrating system capabilities through testing (i.e., verifying the fulfillment of requirements). [DoD 5000.2-R]

When possible, developmental testing supports and provides data for operational assessment before initial operational test and evaluation by the operational test activity. Cost of an Independent Variable (CAIV) analyses from earlier phases are refined and continued through the Critical Design Review (CDR). System-specific performance requirements are developed for contract specifications in coordination with the user (or the user's representative).

Planning for Phase III addresses design stability, development, industrial base capacity, configuration management control, deployment, and support (including, as appropriate, the transition from interim contract to in-house support). Developmental test and evaluation (DT&E) (testing to development specification) is performed. This supports and provides data for an operational assessment prior to operational test and evaluation (OT&E) (testing to operational requirements), also performed during Phase I.

Program budget status is periodically reviewed by both the PBBS and acquisition management systems during this phase. Changes to the program that result in the actual or projected breach of an established program baseline parameter are identified. Such changes may require a formal notification to the MDA. During Phase II the following activities are normally performed:

PHASE II ACTIVITIES	ACAT I	ACAT IA
1. Update assessment of the military threat is	X	
2. An updated test program with required lethality and survivability testing is developed	X	
3. Potential environmental impacts are assessed	X	
4. Cooperative opportunities are identified	X	
5. Compliance with international arms control agreements is assured	X	
6. Design stability is achieved	X	X
7. Technology and technical risk are considered	X	X
8. Software design, coding, integration, and testing is performed	X	X
9. Initial operational test and evaluation (IOT&E) results that realistically portray operational performance are produced	X	X
10. Technological and industrial capability to support the program is assessed	X	X
11. The Acquisition Strategy to include the support concept is refined	X	X
12. The program cost estimate, independent cost estimate, cost objectives and Manpower Estimate are refined	X	X
13. An updated affordability assessment is developed	X	X
14. Proposed cost, schedule, and performance objectives and thresholds for approval are identified	X	X
15. That adequate resources have been programmed to support production, deployment, and support is verified	X	X
16. A proposed oversight and review strategy to include a description of mandatory information and when this information must be submitted for the next milestone is developed	X	X
17. CAIV objectives are refined	X	X
18. C4ISR support requirements are updated	X	X

Table 5-5. Phase II Activities for ACAT I and ACAT IA Programs

5.3.4.1 Phase II: Software Engineering and Development Activities

For all major software development programs, the following activities should occur:

PHASE II SOFTWARE ENGINEERING & DEVELOPMENT	ACAT I	ACAT IA
1. Plans are made for the development and use of reusable software assets.	X	X
2. Security specifications are based on identified security requirements and the consideration of potential threats and vulnerabilities.	X	X
3. DoD-approved software metrics are used to provide a quantitative framework from which to evaluate and control software development or integration. • A common core set of software management metrics are developed early in the development cycle and approved at Milestone II.	X	X
4. ACAT IA performance objectives and measures are established and supported by program evaluations and cost/benefit analyses that are refined in later phases and prepared, in accordance with DoDI 7041.3, *Economic Analysis and Program Evaluation for Resource Management.*	X	X
5. Standards planning, including identification of information technology standards profiles, is accomplished in accordance with the Technical Architecture Framework for Information Management (TAFIM).	X	X
6. The AIS human computer interface is developed in accordance with the Human Computer Interface Style Guide.	X	X
7. DoD standard data elements are designed, development, registered, and implemented in accordance with DoDD 8320.1, DoD *Data Administration*.	X	X
8. Government-off-the-shelf (GOTS), commercial-off-the-shelf (COTS), or nondevelopmental item (NDI) products are certified as meeting appropriate standards.	X	X
9. C3I systems are reviewed for compliance with compatibility and interoperability policy in accordance with DoDD 4630.5. [ANDERSON98]	X	X

Table 5-6. Activities for Phase II Software Engineering and Development

5.3.4.2 Low Rate Initial Production (LRIP)

Low Rate Initial Production (LRIP) (for ACAT I programs) occurs during EMD. Design fixes or upgrades based on test results are incorporated into the initial assets. The objective of LRIP is to produce the minimum number of systems necessary to:

- Provide production configured or representative articles for operational tests,
- Establish an initial system production base; and
- Upon successful completion of operational testing, permit an orderly increase in the system production rate sufficient to lead to full-rate production.

At the LRIP decision, the MDA normally considers the following:

LRIP DECISION CONSIDERATIONS	ACAT I	ACAT IA
1. Acquisition Strategy	X	
2. APB	X	
3. Phase II exit criteria	X	
4. Threat assessment	X	
5. Test results	X	
6. Initial production experience	X	
7. Environmental consequences	X	
8. CAIV progress	X	
9. Adequacy of resources (manpower and funding)	X	
10. Updated C4ISR support requirements	X	
11. ICE and Manpower Estimate	X	

Table 5-7. LRIP Design Considerations [ANDERSON98]

NOTE - LRIP is not applicable to ACAT IA programs; however, They may employ a limited deployment phase. [DoD 5000.2-R]

5.3.4.3 Milestone III Decision: Production or Fielding/Deployment Approval

The purpose of the Milestone III Decision is to authorize the transition into production for an ACAT I program, or deployment for an ACAT IA program. It represents a commitment to build, deploy, and support the system. Particular attention is placed on assessing DOT&E and OT&E results. The most economic production rate that can be sustained (given affordability constraints) is established. Planning is performed for a possible transition to contingency support or reconstitution. Establishing the production APB requires effective interaction among the three major decision support systems. This is particularly critical for establishing economic production rates.

Criteria are established for determining when the operational capability is attained and that planning for deployment and support is complete and adequate. The completion of engineering drawings, the system and software architecture, and their release to engineering organizations signify that program managers are confident that they are mature and perform adequately. These documents reflect the results of prototyping and testing, describe the hardware and software, and define engineering processes. The Critical Design Review (CDR) is a major event that represents a point of departure from detailed design to system development.

The risks of proceeding with CDR and the rest of development as planned are increased without a mature design. Thus before production begins, the process of discovery, the accumulation of knowledge, and the elimination of risks or unknowns must be complete. Because system development must be a clearly defined, stable, statistically controlled process, the Milestone III Decision must be based on achieving the following criteria.

- The technological solution is mature and fulfills all system requirements;
- The design performs as expected; and
- The system can be developed on time and at a reasonable cost.

Be advised, if an acquisition program cannot meet these criteria, it possesses an unacceptable level of cost, schedule, and technical risk. Immature or undeveloped software-intensive solutions must be managed separately until they can meet these criteria before proceeding to the next life cycle phase. [HINTON98] See discussion on Advanced Concept Technology Demonstrations (ACTDs) in Chapter 4, *DoD Software Acquisition Environment*. In addition, frequently changing designs lead to unstable development processes. Late design changes cause serious development problems that necessitate costly process changes. Unstable designs force developers to perform problem workarounds, which lead to labor inefficiencies and result in high scrap and rework rates. According to Watts Humphrey,

> *"As long as programmers are writing code, they are making design decisions, just at a more detailed level. Many of these details will impact the usability and performance of the system, just not at a high enough level for the people who wrote the requirements to be aware of them. The field users of such systems, however, will almost always find that systems developed blindly from requirements documents are inconvenient and unwieldy in operational use. Truly superior usability can only be obtained when the developers have an in-depth knowledge of actual field conditions. While suppliers should start from official requirements, these must be recognized as a starting point and that much more detailed knowledge is required before the system can actually be built. The key is to make the supplier responsible for devising, defining, and using a process that uncovers true operational requirements."* [HUMPHREY95]

The Milestone III ADM approves entry into Phase III (Production, Fielding/Deployment, and Operational Support), approves the proposed or modified Acquisition strategy and production APB, and establishes exit criteria. At the Milestone III Decision, the MDA normally considers the following:

MILESTONE III DECISION CONSIDERATIONS	ACAT I	ACAT IA
1. Threat assessment	X	
2. Initial production experience	X	
3. Environmental consequences	X	
4. Acquisition Strategy	X	X
5. APB (including CAIV-based objectives)	X	X
6. Phase II exit criteria	X	X
7. Test results	X	X
8. Provisions for evaluating post-deployment performance [compliance with GPRA, Cohen Act (ACAT IA only), and PRA] (see Table 5-10 below)	X	X
9. Adequacy of resources (manpower and funding)	X	X
10. ICE and ME		
11. Updated C4ISR support requirements	X	X

Table 5-8. Milestone III Decision Approval for ACAT I and ACAT IA Programs [ANDERSON98]

Once a Milestone III Decision has been approved, a Production Baseline is established that reflects the cost, schedule, and performance assessment requirements for the next phase.

5.3.5 Phase III: Production, Fielding/ Deployment, and Operational Support

Following a successful Milestone III Decision, system performance and quality are monitored by FOT&Es. Cost, schedule, and performance are reviewed and compared to the Production Baseline. User feedback and field experience results (including operational readiness rates) are continuously monitored. Support plans are implemented to ensure sufficient support resources are acquired and deployed with the system. The objective of this phase is to achieve an operational capability that satisfies mission needs. This includes:

- Assessing the ability of the system to perform as intended,
- Identifying and incorporating minor engineering change proposals into production lots to meet required capabilities; and
- Identifying the need for major upgrades or modifications.

Deficiencies encountered in DT&E and initial operational test and evaluation (IOT&E) are resolved and fixes verified. During fielding/deployment and throughout operational support, opportunities for system improvements through upgrades, enhancements, and modifications are continuously assessed. System performance and quality is monitored by follow-on OT&E (FOT&E). Program budget status is periodically reviewed by the PPBS and Acquisition Management Systems. The results of field experience (including operational readiness rates) are continuously monitored, particularly during the early stages of deployment. Support plans are implemented to ensure support resources are acquired and deployed with the system. During Phase III the following are normally performed:

PHASE III ACTIVITIES	ACAT I	ACAT IA
1. The military threat assessment is updated and validated	X	
2. Full rate production experience is used to verify development processes, confirm the design stability and producibility, and develop realistic development cost estimates.	X	
3. The configuration management program is developed	X	X
4. Life cycle cost estimates are updated	X	X
5. Operational and support plans, to include transition from contractor to in-house support (if appropriate) are executed	X	X
6. Operational and support problems are identified	X	X
7. System deficiencies discovered during DOT&E and FOT&E are verified and resolved (as appropriate)	X	X
8. C4ISR support requirements are updated	X	X

Table 5-9. Phase III Activities for ACAT I and ACAT IA Programs [ANDERSON98]

NOTE - The production requirement of this phase does not apply to ACAT IA acquisition programs or software-intensive systems with no developmental hardware components. [DoD 5000.2-R]

5.3.5.1 Operational Support

The objective of this activity is to implement a support program that meets performance and sustainment threshold values in the most cost effective manner. An FOT&E program assesses performance, quality, compatibility, and interoperability, and identifies operational deficiencies. Operational support plans are implemented to include the transition from contractor to organic support, as appropriate.

This phase overlaps Phase III and begins after initial systems, increments, or capabilities have been fielded. It is marked by the declaration of an operational capability or the transition of management responsibility from the developer to the maintainer. Operational support continues until the system is retired from the inventory or a decision is made to commit to a major upgrade or modification (which causes the program to re-enter Phase I, II, or III, as appropriate). Quality, safety, performance, and technological obsolescence are corrected as identified. Post deployment supportability/readiness reviews are periodically conducted to resolve operational and supportability issues. [DoD 5000.2-R]

5.3.5.2 Modifications

The objective of this activity is to undertake modifications and updates to extend the system's useful life; however, the proliferation of system configurations must be minimized. For management purposes, any system modification of sufficient cost and complexity that qualifies as an ACAT I or ACAT IA program is considered a separate acquisition effort. System modifications that do not meet the ACAT I or IA thresholds are considered part of the program being modified, unless the system is no longer in production. In that case, the modification is considered a separate acquisition effort. [DoD 5000.2-R]

5.4 Life Cycle Compliance with Statutory and Regulatory Requirements

In the past several years, Congress has enacted legislation intended to improve the management and performance of Federal Agencies. These laws [discussed in Chapter 3, *Statutory Framework Governing Software Acquisition*] include the Cohen Act GPRA, and the PRA. Further guidance from the OMB places added emphasis on managing investments, *to include weapon systems*. DoD programs must also comply with other statutory and DoD regulatory acquisition requirements. These requirements are applied (as appropriate) to each increment of incremental and evolutionary programs (discussed below) at the following life cycle milestone decisions, as illustrated in Figure 5-12.

- **Pre Milestone 0**. Some requirements (those that address the need for information systems and the processes they support) are the responsibility of the user or the functional proponent. Responsibility for ensuring compliance with these requirements before MDA Milestone 0 approval belongs to the appropriate user or functional proponent in coordination with the JROC process, the Component, or the PSA.
- **Milestones 0 through III**. Many of these requirements are similar to those discussed above and are appropriate for MDA review at each major milestone. For software-intensive NSS programs subject to DAB review, the DoD CIO provides the MDA with an assessment of compliance with regulatory requirements. This is accomplished through the DAB IPT process.
- **Post Milestone III**. Milestone III ADMs include post-deployment performance evaluations and compliance with other performance measurement guidance, as appropriate. The ADM ensures that the user or functional proponent performs post-deployment evaluations and provides the results to the DoD CIO. [KAMINSKI97]

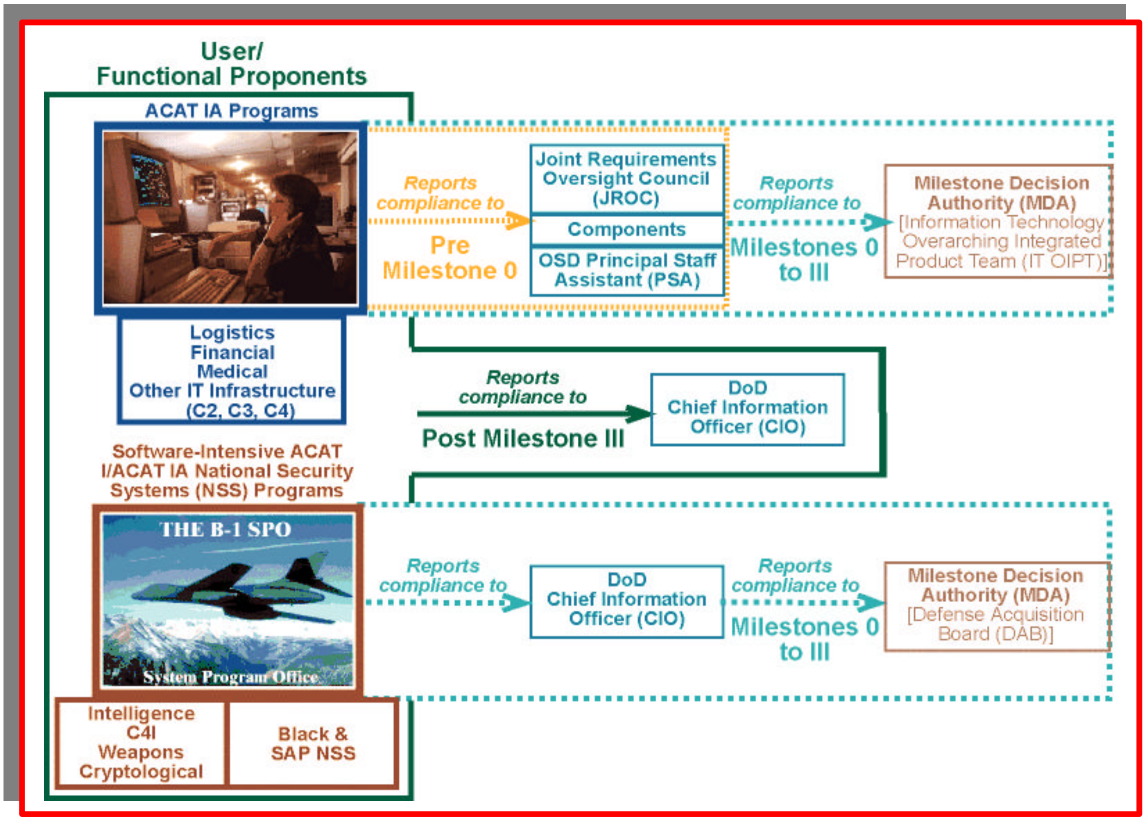

Figure 5-12. Milestone Review Process for Compliance with Statutory and other Regulatory Requirements

Table 5-10 correlates GPRA, PRA, and Cohen Act requirements with the other statutory and DoD regulatory acquisition requirements. To ensure program success, Integrated Product Team (IPT) members should consider these requirements as programs progress through the acquisition process. To the maximum extent possible, these requirements should be addressed by incorporating them into existing acquisition processes, procedures, and documents. [*The document is explicitly approved by the official indicated.] [KAMINSKI97]

ACAT IA Program Regulatory Requirements		Milestone Applicability				Document Source	Prepared By	Approval	Consider-ed By
		0	I	II	III				
Core Mission	Does it support DoD Core/ Primary mission functions?	X				MNS (with linkage to DoD Strategic Plan)	Component	JROC*	PSA*
			X	X	X	ORD	Component		
Outsourcing	Is it an inherently government function?	X	X			MNS	Component	JROC*	PSA*
		X	X			ORD	Component		
		X	X			AOA	PSA/ Component	MDA	
Business Process Reengineering/ Benchmarking	Have work processes been redesigned to reduce costs and improve effectiveness (including benchmarking against comparable processes in other public or private organizations)?	X	X			MNS	Component,	JROC*	PSA*
		X	X			AOA	PSA/ Component	MDA	
COTS Solution	Does it maximize COTS technology use?	X	X			MNS	Component	JROC*	PSA*
		X	X			AOA	PSA/ Component	MDA	
Return on Investment	Does the projected ROI support this alternative? [ROI includes improvements to mission performance, resource savings, or qualitative mission benefits.]	X	X			AOA	PSA/ Component	MDA	
			X	X	X	Life-Cycle Cost (and Benefit) Estimate	PM	MDA	
Strategic Goals	Are work processes, information flows, and technology integrated to achieve DoD Strategic Goals?	X	X			ITM Strategic Plan	PSA/ Component	MDA	
		X	X			PSA/ Component Strategic Plans	PSA/ Component	MDA	
Technology	Does it reflect DoD's technology vision?		X	X	X	ITM Strategic Plan	DoD CIO	MDA	
Joint Technical Architecture (JTA)			X	X	X	Acquisition Strategy	PM	MDA	
			X	X	X	C4I Support Plan	PM	MDA	
Technical Architecture for Information Management (TAFIM)		X				Acquisition Strategy	PM	MDA*	

Table 5-10. ACAT IA Program Requirements [KAMINSKI97]

ACAT IA Program Regulatory Requirements		Milestone Applicability				Document Source	Prepared By	Approval	Consider-ed By
		0	I	II	III				
Year 2000	Is it Year 2000 compliant?		X	X	X	Defense Integration Support Tools (DIST)	PM	MDA*	
Standards/ Flexibility	Does it incorporate standards that enable information exchange and resource sharing while retaining flexibility?								
Defense Information Infrastructure Common Operating Environment (DII COE)			X	X	X	Acquisition Strategy	PM	MDA	
Automated information collection/Continuous Acquisition and Life-Cycle Support (CALS)			X	X	X	Acquisition Strategy	PM	MDA*	
Software engineering **--Software reuse** **--Software language** **--DoD standard data**			X	X	X	Software Engineering Strategy	PM	MDA	
Information assurance			X	X	X	Information Assurance Strategy	PM	MDA	
Electromagnetic Environmental (E2) **-- Effects (E3) and Spectrum** **– Management**			X	X	X	C4I Support Plan	PM	MDA	
Open Systems	Does it avoid/isolate custom-designed components?		X	X	X	Acquisition Strategy	PM	MDA*	

Table 5-10 , continued. ACAT IA Program Requirements [KAMINSKI97]

ACAT IA Program Regulatory Requirements		Milestone Applicability 0	I	II	III	Document Source	Prepared By	Approval	Consider-ed By
OT&E	Will it have fully tested pilots, simulation, or prototypes before production/ deployment?			X	X	TEMP	PM	DOT&E*	
Performance Measures	Are there clearly established measures and accountability for program progress? Are these measures linked to strategic goals?		X			MNS	Component	JROC*	PSA
				X	X	ORD	Component	JROC*	PSA*
Acquisition Program Baseline (APB)		X	X			APB	PM	MDA*	
Cost as An Independent Variable (CAIV) **objectives**				X	X	Acquisition Strategy	PM	MDA*	
			X			APB	PM	MDA*	
Milestone Exit Criteria				X	X	ADM	PM	MDA*	
			X	X	X	IT OIPT Quarterly Report	PM	MDA	
Software measures			X						
Full Funding	Is it supported by all intended users or sponsors?		X	X	X	ORD	Component	JROC*	PSA*
						Affordability Assessment	PA&E	MDA	
Incremental	Will it be implemented in phased, successive chunks?		X	X	X	Acquisition Strategy	PM	MDA*	
Contract Risk Management	Does the acquisition strategy allocate risk between government and contractor?		X	X	X	Acquisition Strategy	PM	MDA*	
Competition	Does it effectively use competition?		X	X	X	Acquisition Strategy	PM	MDA*	
Earned Value	Are contract payments tied to accomplishments?		X	X	X	Acquisition Strategy	PM	MDA*	
			X	X	X	Integrated Baseline Review	PM	MDA*	
COTS Components	Does it take maximum advantage of commercial technology?		X	X	X	Acquisition Strategy	PM	MDA*	

Table 5-10 , continued. ACAT IA Program Requirements [KAMINSKI97]

5.5 Life Cycle Management

Performance-based and modeling & simulation-based acquisition management should be folded into your life cycle management methodology and process. They are proven risk reduction methods, which are firmly endorsed by the Congress and the DoD.

5.5.1 Performance-Based Life Cycle Management

Performance-based life cycle management is the application of the performance-based paradigm [discussed in Chapter 3, *Statutory Framework Governing Software Acquistion*] to the DoD 5000-series life cycle process. According to Lt. Col. Dennis Drayer (USAF), DoD acquisition practices have produced the best military systems in the world. However, our requirements allocation process is often flawed. Requirements are flowed down without allocation at lower levels, resulting in incomplete requirements definition at the user/maintainer level. Testing often dominates the design evolution process and program and product teams fail to identify and control critical system features and processes. In many cases, the causes for a system's behavior are not understood or controlled as the design evolves. The result is design by *trial and error*. [DRAYER98] Performance-based life cycle management attacks these shortcomings. As you learned in Chapter 3, *Statutory Framework Governing Software Acquisition*, Performance-based management includes the following key steps:

- Define clear missions and desired outcomes,
- Measure performance to gauge progress, and
- Use performance information as a basis for decision-making. [BOWSHER96]

5.5.1.1 Performance-Based Systems Definition

Design is often a point solution that does not tolerate normal variations, which makes it hard to transition from the laboratory to development. Incorporating changes or adding new technology is often difficult. Such conditions limit the ability to apply innovative concepts (such as competitive sourcing through open system architectures and migration to common processes). Drayer explains that a good performance-based system definition must include three information categories:

- **Category 1**: System Performance Definition
- **Category 2**: System Design
- **Category 3**: Software Engineering and Development Definition (for Software-Intensive Systems)

5.5.1.1.1 Category 1: System Performance Requirements Definition

Derived operational requirements are translated into specific technical engineering language stated in performance terms, which provides the basis for a design solution and design qualification. Similar to traditional development specifications, in a disciplined systems engineering process the contractor develops and verifies this data as top-level requirements filter down.

The Government conveys its needs through high-level specifications, limiting military-unique specifications and standards. Although the prime contract may include some requirements allocation items, most are under contractor control. The aim of the system definition process is to decrease the amount of deliverable technical data. Based on technical and capability risk assessments, the DoD program team decides to include or exclude contractually required data. [DRAYER98]

5.5.1.1.2 Category 2: System Design

Engineering and development environments are linked by translating Category 1 requirements into the system design. Design-specific performance requirements define key system engineering design and development characteristics and enable efficient technology insertion at a minimum requalification cost.

The system design defines how the user implements a given function. This avoids high, non-recurring costs that result from growing designs to meet new requirements, technology insertion/ obsolescence, and service-life extension. In developing the system definition, key interface requirements are specified that drive interoperability with other platforms and systems/subsystems (such as armament and C3I systems).

System acceptance criteria are defined for functional and physical attributes measured by the developer and used for system acceptance. Where interchangeability and interoperability issues are complex (such as in avionics and electronics design), it is important that the Program Manager capture *as installed/ as integrated* characteristics within the subsystem design, as well as total weapon system (or AIS) designs.

5.5.1.1.3 Category 3: Software Engineering and Development Definition (for software-intensive systems)

This includes everything the build package needs to develop the system as defined by Category 2 requirements (including detailed software development process capability requirements). The detailed system definition includes development-level information (in contractor format) applicable to the *as built* condition and industry-wide software engineering standards, which form the basis for system quality assurance. The data required to develop the system efficiently drive the level of detail — not the Government's intent to control the developer's process.

5.5.1.2 Performance-Based Systems Engineering

The performance-based approach for maintaining system integrity draws upon lessons learned without dictating a solution. For example, past methods for defining critical flight safety components and systems imposed prescriptive military specifications and standards. This did not always capture the critical information needed to develop and sustain the system. The thorough definition of what is used to develop and support the system over its life cycle results in a robust system engineering process. Rather than prescribing a new, rigid format, this method is flexible and carefully tailored to the contractor's specific engineering and technical processes. The system description quantifies required performance parameters and defines system characteristics, key processes, critical interfaces, and system acceptance criteria. [DRAYER98]

5.5.1.2.1 Statistical Process Control (SPC)

Performance-based system acceptance uses statistical process control (SPC) rather than extensive testing and inspections. All key processes must be under statistical control when production begins. An SPC program is established that monitors processes to ensure they consistently produce output that is within the quality standards and tolerances set for the overall system. Statistics concerning the quality of each process output are analyzed, and when the output is out of tolerance, process owners search for causes.

Once a process is producing consistently high-quality output, the process is considered to be in statistical control, and inspections are reduced. The knowledge gained using SPC is significant in transitioning from development to production because it helps ensure that cost, schedule, quality, and reliability targets satisfy user requirements. The ability to establish SPC for key processes before production begins is the culmination of all the practices employed to identify and reduce risk. The criteria established throughout the system development process forces Program Managers to prove that the system design is capable and producible early in the process. [HINTON98]

5.5.1.3 Performance-Based Management Benefits

Performance-based requirements identify safety critical components, define special development requirements or tolerances, and quantify critical software functions or life cycle management requirements. Because system integrity can be maintained, this management method offers considerable cost savings. Combined with a rigorous systems engineering approach to system design and development, the performance-based life cycle method provides the following benefits:

- Acquisition operations are more efficient;
- Performance and quality requirements are fulfilled at minimum cost;
- Robust design solutions are facilitated that tolerate development variations; and
- Technology insertion is accommodated in a cost-effective manner. [DRAYER98]

5.5.2 Modeling & Simulation-Based Life Cycle Management

> *"Simulation and modeling technology can be applied to every major DoD weapon development program to reduce design and production cost, improve performance, improve diagnostics and maintenance, assist in better and faster training of personnel, and improved command and control on the battlefield."* — Colleen Preston (Deputy Under Secretary of Defense Acquisition Reform) [PRESTON94]

Modeling & simulation (M&S) is a powerful tool for helping acquisition Program Managers to optimize their processes. DoDD 5000-1 states that simulations/models (sim/mods) are to be used to reduce the time, resources, and acquisition process risks and increase systems quality. Representations of proposed systems (virtual prototypes) should be embedded in realistic, synthetic environments to support various acquisition life cycle phases. Sim/mods can be used from requirements determination to initial concept exploration, manufacturing, new system testing, and related training.

M&S are often viewed as a potential solution to DoD's many acquisition problems. It can be used to prove concepts through simple chalkboard mathematical calculations to full-scale system replicas that operate in controlled environments. Sim/mods are tools that minimize acquisition costs, schedule, performance, and supportability risks, as illustrated in Figure 5-13.

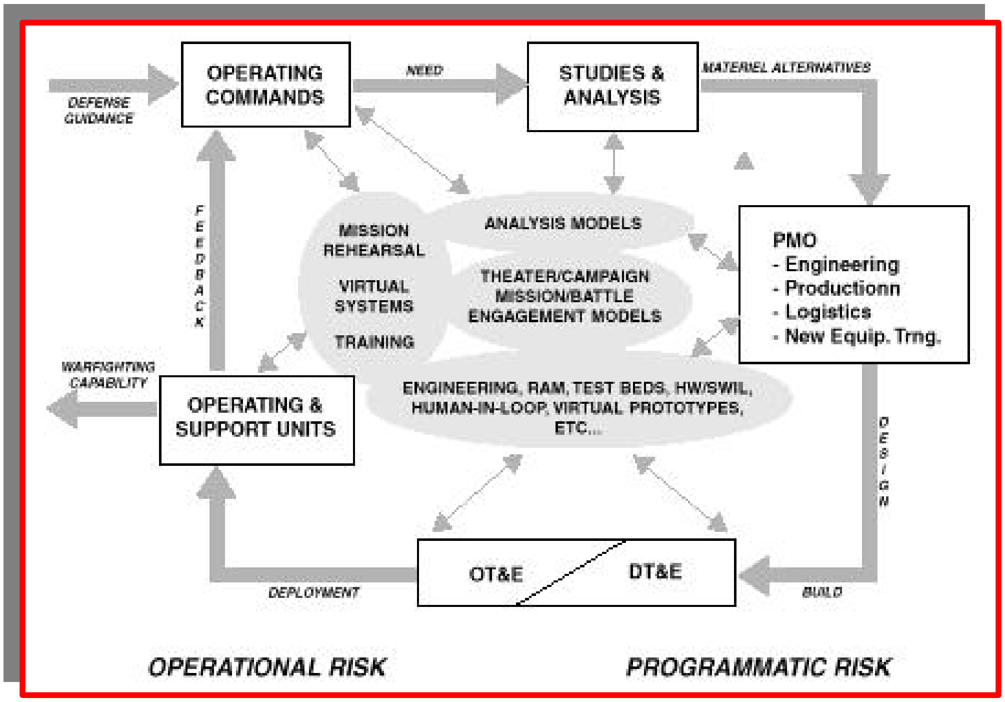

Figure 5-13. Risk-Based Management of Systems Acquisition Life Cycle Process [PIPLANI94]

M&S can be used to support in each phase of the acquisition life cycle. M&S is the application of those tools to support decisions. An efficient and effective source of valuable information for new defense system development and evaluation. When used in a well-defined, integrated manner, M&S reduces the expenditure of resources, accelerates understanding through early insight, shortens cycle times, and improves system quality.

The full potential of M&S-based life cycle management is realized through Integrated Product and Process Development (IPPD) and IPTs. Their effectiveness is improved by implementing state-of-the-art M&S for planning, design, analysis, management, and testing throughout the acquisition process.

5.5.2.1 M&S-Based Pre-Phase 0: *Mission Area Analysis*

The modeling and simulation (M&S) tools used during this phase provide insight into operational risk and are used to identify nonmaterial or materiel approaches to mitigate risks. Suites of sim/mods, along with supporting data (including threat, environment, tactics, doctrine, etc.), are used to perform MAAs. In accordance with DoDI 5000.2-R, M&S are to be used to analyze tactics and concept of operation changes with existing baseline systems before new systems evaluations. Campaign/theater level sim/mods, used in conjunction with the results of lower level sim/mods, produce data for identifying warfighting needs documented in the MNS. [PIPLANI94]

5.5.2.2 M&S-Based Phase 0: Concept Exploration and Definition

The same sim/mods used to define requirements are employed to examine the capabilities of specific materiel solutions. Initial program planning and key program documents are developed. M&S data ensure consistency among planning and other program documents. M&S supports the implementation of the Cost as an Independent Variable (CAIV), early risk reduction, the establishment of consistent measures of effectiveness (MOEs), and measures of performance (MOPs). Engineering sim/mods project performance and requirements tradeoffs. Engagement and mission/battle level sim/mods determine mission effectiveness, support cost/performance tradeoffs, and ORD development. Theater and campaign level sim/mods evaluate conflict outcomes in support of the same documents. Human interactive sim/mods are used to develop tactics. Virtual sim/mods evaluate concepts, technologies, and tactics in realistic synthetic environments.

The TEMP identifies M&S resources required to support development and operational testing. How sim/mods will be used across all IPPD functional disciplines is determined as the appropriate plans (e.g., integrated logistics support plan, systems engineering management plan, etc.) are developed. As illustrated in Figure 5-14, M&S aids in establishing and maintaining consistent relationships among MOEs, MOPs, and program documentation. [PIPLANI94]

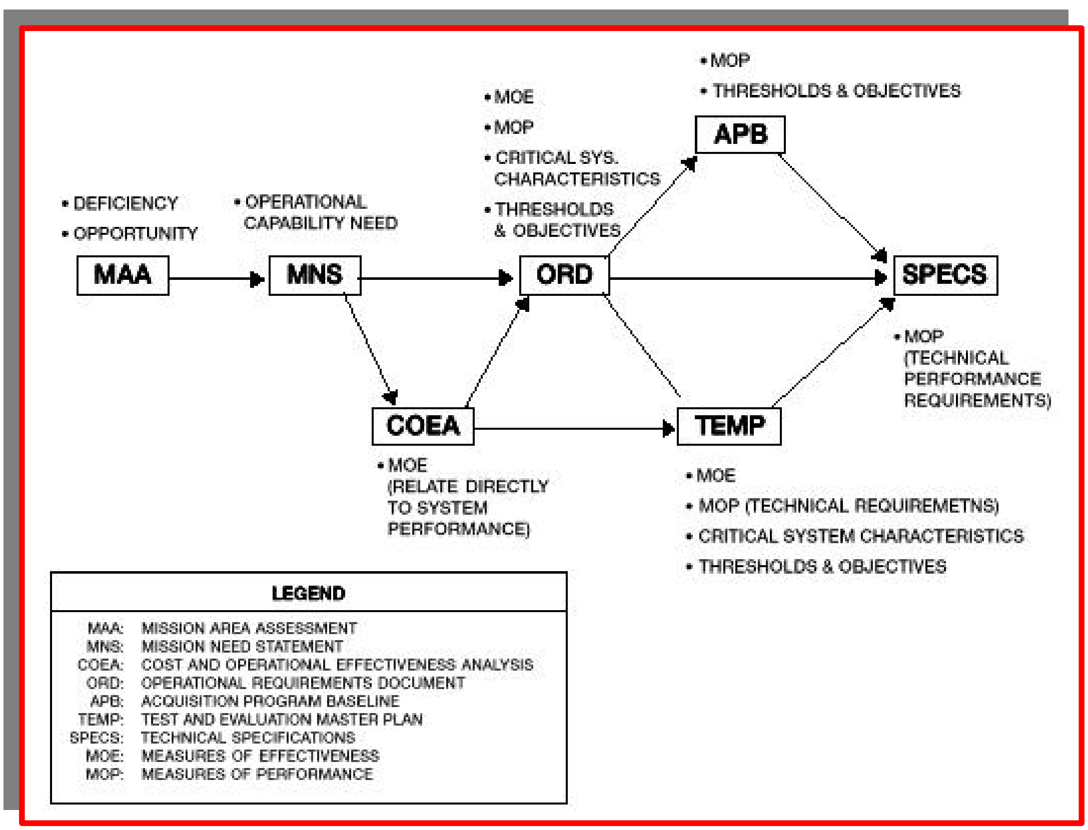

Figure 5-14. M&S Provides Consistency among Acquisition Program Documents, Needs, and Measures [PIPLANI94]

During this phase, the groundwork is established for continued M&S life cycle application. This includes sim/mod use/reuse, integration and interoperability, and common databases. Sim/mod development planning also addresses future compatibility with synthetic battlefields through Distributed Interactive Simulation (DIS) communication standards and eventual transition of developed sim/mods to the training environment. The main objective of this phase is to allow later builds of already developed sim/mods. This reduces duplication and provides consistency throughout the life cycle phases and among the documents and activities within a given phase. [PIPLANI94]

5.5.2.3 M&S-Based Phase I: Program Definition and Risk Reduction

During this phase, system definition and alternative concept assessments are refined. Critical technologies are demonstrated and prototyping is conducted. M&S applications include hardware/ software-in-the-loop (HWIL/SWIL), computer-aided design/manufacturing (CAD/CAM), engineering and mission level sim/mods for interoperability analyses, and continued theater/ campaign level support to cost/performance studies and the ORD.

Human-in-the-loop sim/mods are used to evaluate the developing human-machine interface. Virtual prototyping can include actual prototype hardware and software, with the possible linking of sim/mods to a virtual battlefield. These sim/mods are be used to support early operational assessments (EOA), and to assist the source selection process. M&S continues to support risk reduction with added emphasis on technical solutions and system integration. Figure 5-15 illustrates how M&S is used for requirements definition.

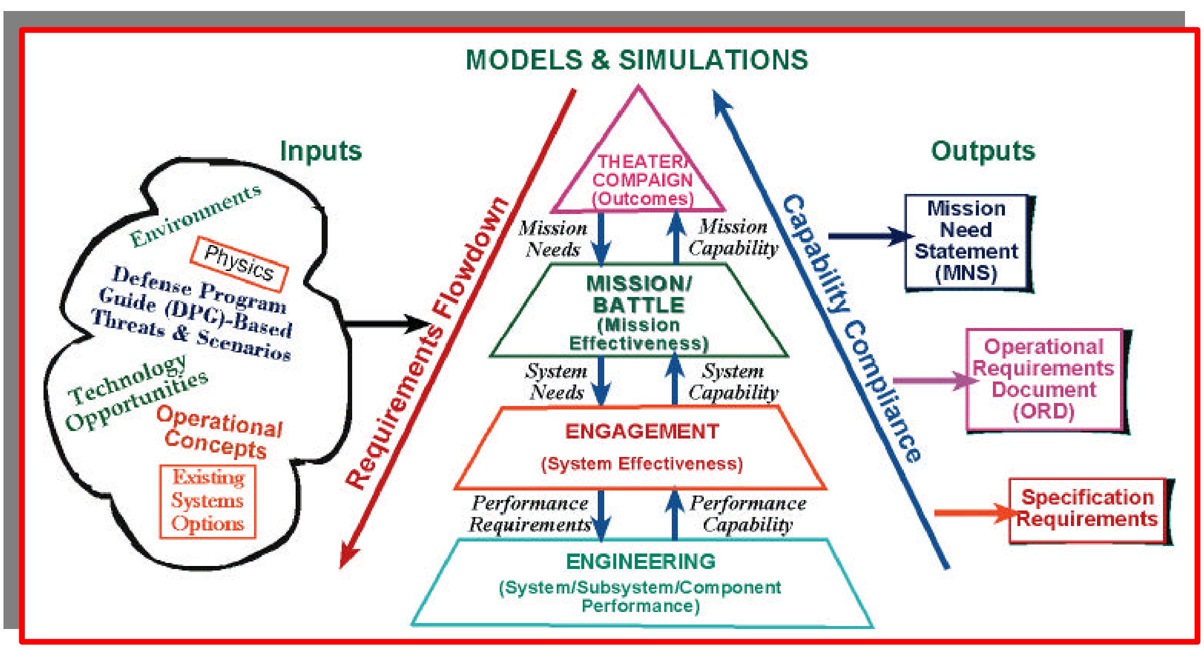

Figure 5-15. M&S in Requirements Definition [PIPLANI94]

5.5.2.4 M&S-Based Phase II: EMD/Software Engineering and Development

M&S supports the transition from design to initial production and testing. Engineering level models are used for performance analysis, test planning, and test support. As illustrated in Figure 5-16, the M&S *model-test-model* process is used to optimize testing resources and further refine hardware and software components. Theater, campaign, battle, and mission sim/mods are updated with the latest parameters and continue to support high-level conflict resolution and MOE consistency.

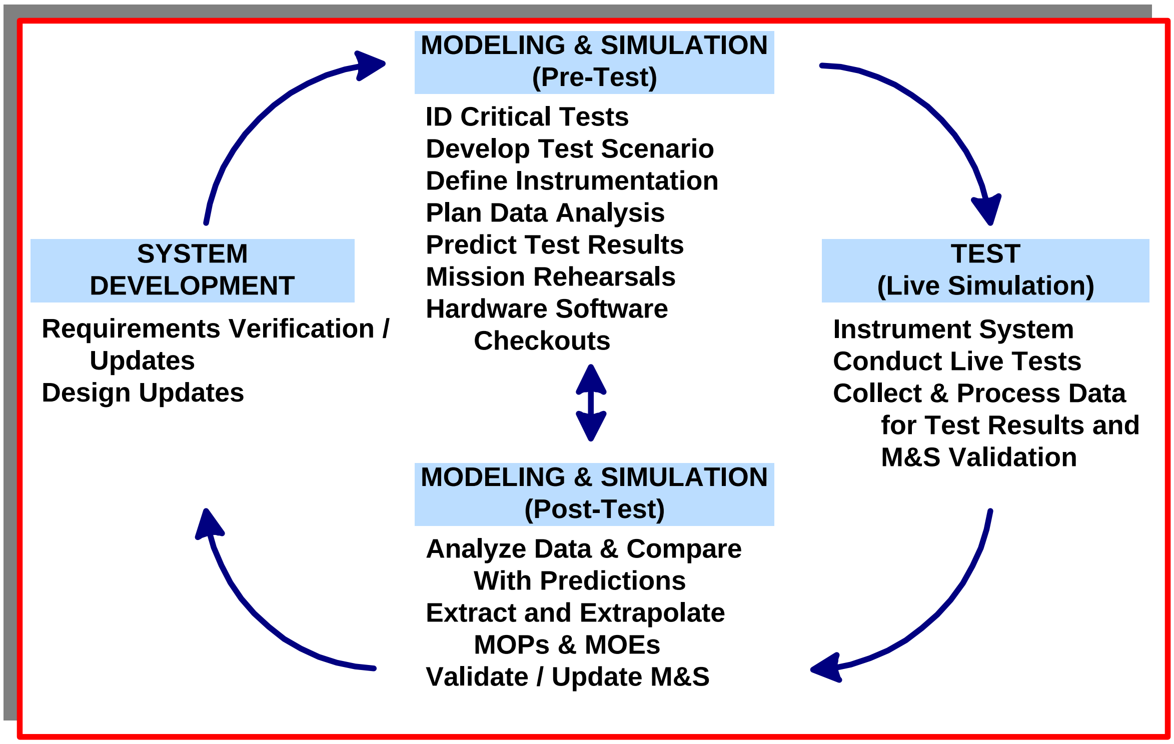

Figure 5-16. *Model-Test-Model* Development Process [PIPLANI94]

Virtual factories are used to define production schedules and physical configuration. Virtual sim/mods are used to develop user-training systems and to conduct test rehearsals. Combinations of engineering, engagement, mission, campaign, and HWIL sim/mods augment developmental and operational testing commensurate with the level of verification, validation, and accreditation (VV&A) performed. M&S serves as a tool for continued risk reduction through detailed design development and test planning. Figure 5-17 illustrates the simulation-based design process.

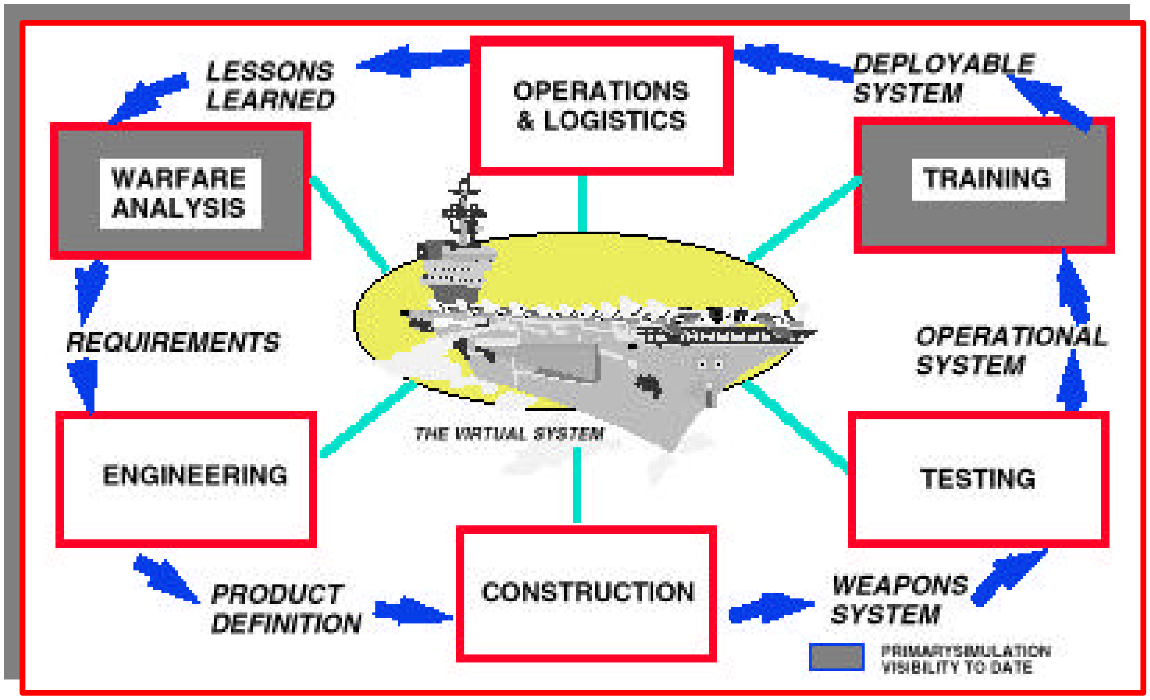

Figure 5-17. Virtual M&S Design in the System Life Cycle [PIPLANI94]

5.5.2.5 M&S-Based Phase III: Production, Fielding/Deployment, and Operational Support

In this phase, M&S supports continued testing, design changes and enhancements, training, logistics, maintenance, and system updates. The *model-test-model* process continues to refine sim/mods and improve systems. Actual data are applied to all levels of M&S for continued validation. High level sim/mods are used for training commanders and battle staffs. Virtual sim/mods are employed as for operational crew training, tactics development, and new threat evaluations.

Throughout all life cycle phases, M&S tools are used, refined, and shared to set and achieve challenging cost objectives, reduce program risk, and conserve program assets. Total acquisition cost and cycle times can be reduced. M&S tools can be reused by users for training. Other acquisition domains can build upon existing sim/mods for interoperability. VV&A can be reapplied to mitigate M&S costs.

5.6 Life Cycle Management Methodologies

A methodology refers to the standards and procedures that affect the planning, analysis, design, development, implementation, operation, support, and disposal of a software-intensive system. Thus, we use the term *software life cycle management* methodology, rather than *software development* methodology, to avoid a perception that the methodology only focuses on the design and build stages. Derived from historical program experience, methodologies provide insight into the use of candidate solutions based on program character, acceptable level of risk, and system constraints.

Methodologies present a conceptualization of the life cycle process that can be used as a communication tool among all system stakeholders. Specifically, they aid in defining the sequence of major life cycle activities, provide a better understanding of required activities, and serve as a starting point for management decision-making. Remember that software development methodologies used for weapon systems must integrate with, and be consistent with, the weapon system and systems engineering development methodologies used for the total system. The following selected software life cycle management methodologies will be discussed:

- Evolutionary,
- Incremental,
- Spiral,
- Modeling and simulation-based,
- Waterfall, and
- Other methods chosen for their applicability to your program's development or support environment. [PASSMORE94]

Fast-track methods speed up (or bypass) one or more life cycle phases or development processes. An organization can use an existing methodology or develop its own. The focus, names of components, and division of activities vary among methodologies. A methodology should be selected based on the nature of your program, software domain, the methods and tools used, and the controls and deliverables required. [PRESSMAN92] Most life cycle management methodologies include at least the following:

- **Phases**. The methodology divides the life cycle into phases, noting which activities fall in each, and includes a process for determining when each system component can move to the next phase.
- **Milestones**. The methodology defines event-driven milestones in each phase — rather than schedule- or cost-driven. Each milestone specifies appropriate deliverables (e.g., a written report, briefing, test result, portion of code or functionality, analysis or design data, etc.). The methodology also includes program office approval criteria for completion of one phase and movement to the next.
- **Content of deliverables**. The methodology defines (either by topic or outline) what each milestone deliverable should include.
- **Evaluation criteria for deliverables**. The methodology defines which criteria a deliverable must meet to satisfy a milestone for formal Government acceptance. These are also defined as *exit criteria* for completion of one phase and passage to the next. The methodology and criteria are specified in the Software Development Plan (SDP).

During software development, errors/defects are discovered, opportunities are revealed, changes are superimposed, and even changes are changed. Unless carefully controlled, the ensuing complexity makes software evolution error-prone, time consuming, and expensive. The use of life cycle management methodologies has proven extremely effective in controlling change and managing development process complexity.

> **BE AWARE! For any life cycle methodology to be effective, *it must be customized to your specific program goals.***

As useful as they are, life cycle methodologies have their limitations in that they can hide important process detail critical to program success. Life cycle management methods are often too abstract to convey the details of architecture, concept of operations, process steps, data flows, development activities, engineering roles, program constraints, etc. Program management CASE (Computer Assisted Software Engineering) tools address many of these issues. They can enhance many facets of the life cycle process, such as data modeling and normalization, graphical support for design modeling, and code testing. They also support program management, planning, estimation and control, as well as configuration management. [PASSMORE94]

While CASE tools make the program management task easier, do not forget what their name implies. They are *tools*. They cannot tell you what software system to build, what the system must do, or how it should be designed. This process must evolve from user needs and reflect improvements in development methods, techniques, standards, and available software engineering technology.

Choosing an appropriate life cycle methodology is not always an easy task. The selected methodology must be adapted and evolved, the same as the technical activities it ties together. Understanding the software process and making tradeoffs among life cycle components is crucially important for producing high quality software, on time, within budget. The methods presented here have unique advantages and limitations that must be considered. For AIS developments, either incremental or evolutionary methods [particularly the Ada Spiral Model, discussed below] provide effective risk management and earlier satisfaction of user requirements. These approaches are also recommended for weapon systems, when appropriate.

5.6.1 Evolutionary Method

The evolutionary life cycle method is a strategy for systems where future requirement refinements are anticipated to evolve. It is also for programs where there is a technical risk that discourages the immediate implementation of the needed capability. This strategy develops a system in a series of builds. It differs from the incremental approach (discussed next) by acknowledging that user needs are not fully understood and that not all requirements can be fully defined up front. Thus, requirements are only defined to the extent that they are known. As additional user needs become known through feedback from previous builds, the specific requirements for each succeeding build are defined.

The evolutionary model involves an incremental specification, design, implementation, testing, delivery, operations, and maintenance. System capabilities are increased with the delivery of each incremental release until the system is complete. Users have early access to system releases and are encouraged to provide performance feedback, which is used to shape the system as it evolves into its final form. The objective is to establish an acquisition environment sensitive and responsive to users needs. Generic characteristics of the evolutionary model include the following: [ANDERSON98]

NOTE: These must be tailored to the unique requirements of each acquisition program.

- **Incremental approach**. Programs are divided into phases and increments (discussed below). Systems are developed and acquired incrementally. Each increment results in the development of functions that progressively increase overall system capability. Each increment may involve a cycle of system development activities from specification through design to testing, fielding, and maintenance.
- **Bounded functionality**. Broad system requirements must be known prior to the start of design and development. Despite this, there is a tacit assumption that requirements may move outside agreed upon bounds, to a limited extent, during the course of the program.
- **Requirements**. A detailed set of minimal system requirements (and preferably some other early increments) is clearly defined at program outset. Other requirements are progressively refined during successive program phases.
- **Flexible architecture**. The system architecture must support the functions delivered with each release, including those for which detailed requirements have yet to be defined. Generally it must be:
- **Flexible**. Allows for changes within the existing design or implementation.
- **Scalable**. Allows increases in performance requirements through incremental system capacity changes.
- **Extensible and maintainable**. Facilitates architecture expansion and system modifications.
- **User involvement**. The dedicated involvement of users in the development process allows the continuous review of requirements. Users also contribute through their use and review of early system releases (often in the operational environment).
- **Multiple contracts**. The contract for the first phase is often for a minimal system, plus additional increments if they can be clearly specified. The capability and cost of later increments are agreed to prior to the start of each phase. It is expected (but not assured) that the same contractor will build successive phases.

An evolutionary acquisition strategy is effective where detailed functional requirements are difficult to articulate at program inception, although the desired ultimate capability is generally known and a core of initial functionality can be specified. Thus, a core capability is fielded, where the system is designed modularly, and provisions are made for upgrades and changes. Figure 5-18 illustrates Pressman's interpretation of the evolutionary model where a first generation spiral evolves into an extended second generation spiral, and so on. [PRESSMAN93]

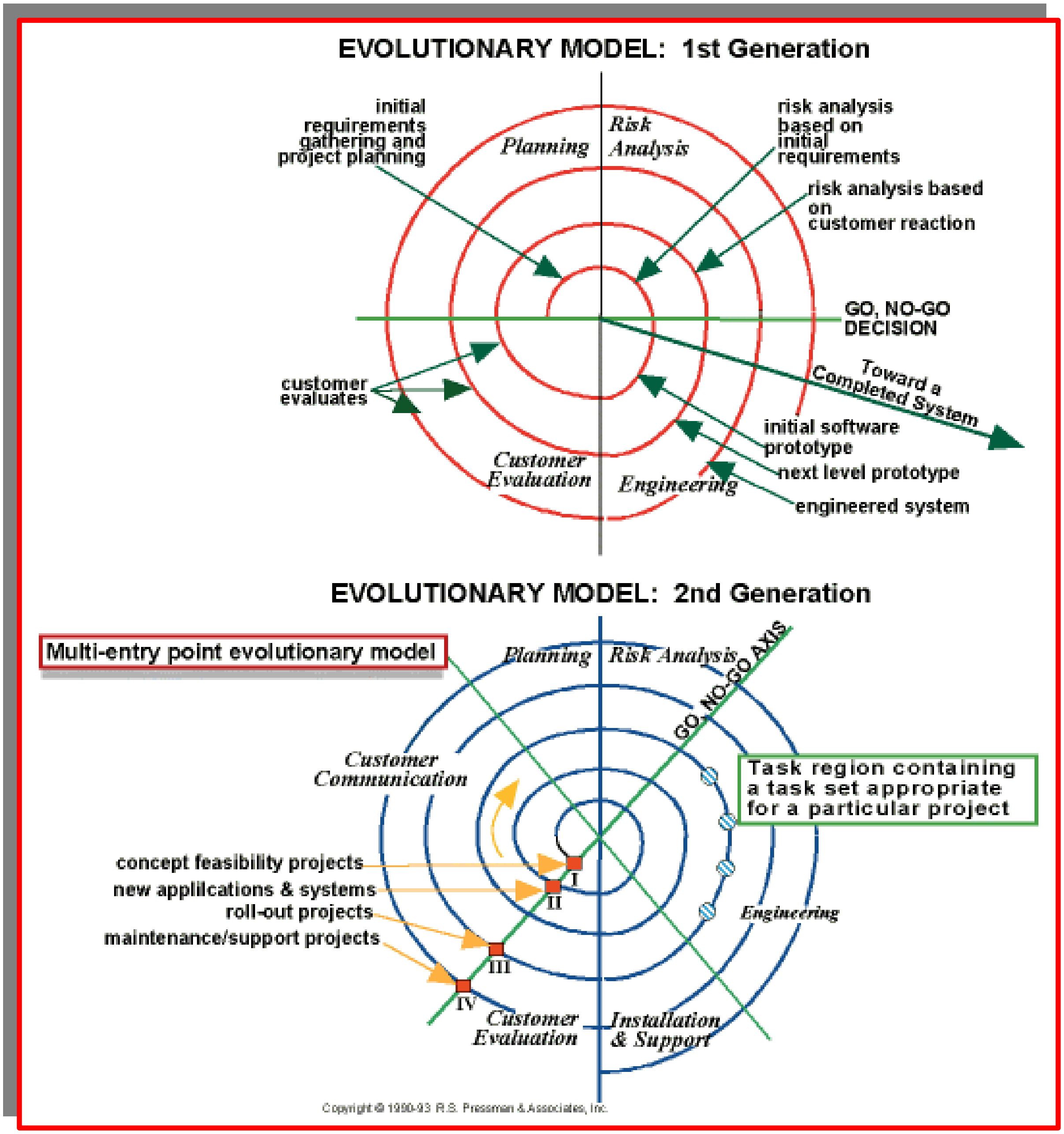

Figure 5-18. Evolutionary Life Cycle Generations [PRESSMAN93]

Evolutionary programs progress towards an ultimate capability. This strategy requires the development of increments of software that are demonstrable to the user, who is involved throughout the entire development process, as illustrated in Figure 5-19.

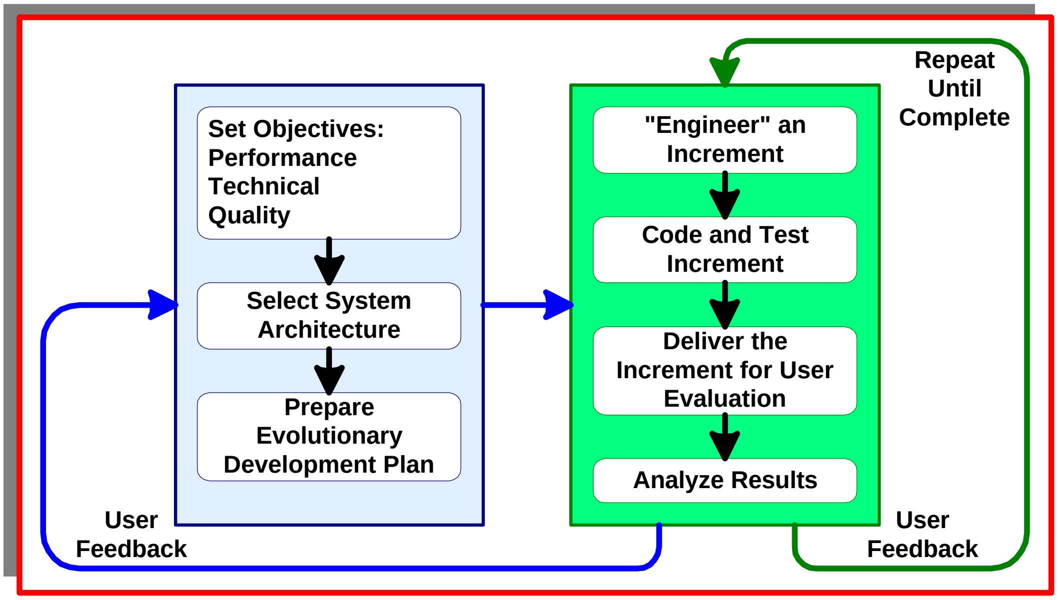

Figure 5-19. User Involvement in the Evolutionary Method

An evolutionary methodology can be employed on all types of acquisitions. However, *it is mostly used on medium to high-risk programs*. While the final version of the system often takes more time and effort to develop than other methods, additional effort and time is offset by delivery of a higher quality product with a lower maintenance cost burden. An evolutionary life cycle method is well suited for advanced technology solutions where requirements beyond the core capability can generally be identified — but not specifically. This is usually the case with highly interactive decision support systems and systems with complex human-machine interfaces. According to Watts Humphrey,

> *"There is a basic principle of most systems that involve more than minor evolutionary change: the system will change the operational environment. Since the users can only think in terms of the environment they know, the requirements for such systems are always stated in the current environment's terms. These requirements are thus necessarily incomplete, inaccurate, and misleading. The challenge for the system developer is to devise a development process that will discover, define, and develop to real requirements. This can only be done with intimate user involvement, and often with periodic prototype or early version field tests. Such processes always appear to take longer but invariably end up with a better system much sooner than with any other strategy."* [HUMPHREY95]

Acquisition programs best suited to the evolutionary method normally have some or all of the following characteristics:

CHARACTERISTICS SUITED TO EVOLUTIONARY METHOD
Software-intensive programs
Programs with rapidly changing software technology
Programs where humans are an integral part of the system
Programs with a large number of diverse users
Programs developing unprecedented systems
Programs where a limited capability is needed quickly

Table 5-11. Program Characteristics Suited to Evolutionary Method

Evolutionary acquisitions are also suited to programs implementing emergent or quickly changing COTS solutions. Additionally, because the volatile Defense environment imposes rapidly changing operational requirements for a range of systems, the evolutionary method is applicable to many weapons and sensor systems. [HENDERSON97]

5.6.1.1 Evolutionary Model Benefits

When used effectively, the evolutionary model can provide significant interrelated benefits. Examples of benefits include the following:

- **Improved requirements**. Maximum user involvement and early build delivery, provides mechanisms to obtained feedback from users and to easily make changes in later increments. Mechanisms also allow for requirements tradeoff, which results in higher quality and better validated requirements. The fielding of early releases helps users understand what is feasible and what is not, resulting in requirements that are more realistic.
- **Early operational capability.** Early system releases can be used operationally, and regularly enhanced.
- **Technology insertion.** By deferring design and component selection until late in the program, incorporation of state-of-the-art hardware and software is enabled. This results in higher performance, lower acquisition costs, and reduced **Total Cost of Ownership**.
- **Management control and program visibility**. Aspects of the evolutionary model contributing to a high level of Government control and visibility include the following:
  - Increased interaction between the Government and contractor.
  - The partitioning of development into well-defined increments.
  - Release of builds showing clear progress.
  - Testing and validation of progressive builds by the users.
  - Thus, the Government has a clearer view of progress and risks. Latent defects are discovered sooner (which often are accidentally or deliberately obscured). The number of control points is increased throughout the acquisition process, which provides better control of risks. Decisions to change program direction, or even to cancel it, can be made earlier.
- **Improved system quality**. The continuous concentration on user requirements through user involvement results in improved system quality. [HENDERSON97]

5.6.1.2 Cautions About the Evolutionary Method

The benefits listed above are not free or automatic. Evolutionary programs are inherently intense and difficult to manage. Drawbacks with the evolutionary model include the following

- **Increased cost and schedule**. The evolutionary model involves more activities, more changes, and more interaction among participants. Therefore, the projected cost and schedule of an evolutionary program may be higher than for an equivalent waterfall program (discussed below). Whether the actual cost and schedule will be more depends on the probability of success of a waterfall approach (and the subsequent need for remedial activities) and the effectiveness of program management in each case. For a program where evolutionary is the best acquisition strategy, the probability of success using other approaches may be quite low.
- **Increased management activities and resources**. An evolutionary acquisition requires close control by both the Government and the contractor throughout the program. The Government needs much more progress visibility, which involves more discussion, negotiations, and planning. An evolutionary acquisition is not *hands off*. The intense, dynamic nature of evolutionary programs requires higher skill levels and management experience than traditional models.
- **Impact of concurrent activities**. Development, test, operations, and support occur concurrently, which require interactions among staff members working on different activities. This requires greater planning and coordination that often results in challenging management problems. Software development change management can be complicated by the fielding of multiple system releases.
- **Increased configuration management**. During most evolutionary stages, at least three builds are under development. These include the most recently released build, the build currently under development, and the build being negotiated for the next increment. High quality configuration management is essential in this environment. The Government must take an active role in the configuration management of requirements, because each build usually corresponds to a different functional baseline.
- **Government technical support**. Many joint government/contractor decisions are based on tradeoffs among operational, technical, and development issues. The program office must have an understanding of requirements, technical issues, and the software development process.
- **User resources and coordination**. Evolutionary acquisition requires continuous user review and feedback. Finding the right users, ensuring their availability, and coordinating their input is not a simple task.
- **Fielded release support**. Support requirements for multiple releases with different characteristics (including operational and support staff training) is greater than for a single delivery at the end of the program.
- **Greater risks**. Evolutionary acquisitions involve higher risk levels than traditional models. These risks include the following.
- **Requirements risk**. There is a temptation to defer requirements definition.
- **Management risk**. Programs are more difficult to control and require higher systems engineering skill levels. Maintaining a close government/industry, cooperative relationship is more critical with evolutionary programs.

- **Approval risk**. The evolutionary phased approach is vulnerable to delays in funding approval, which increase schedule and can be detrimental to performance and cost estimates. Such delays can break the flow of development resulting in the loss of key engineering staff, with associated costs in winding development down and winding up again.
- **Architectural risk**. The architectural design is critical to evolutionary program success. If the initial architecture cannot accommodate later changes, the program may fail.
- **Short term benefits risk**. There is the risk of becoming driven by short term operational needs rather than long-term program goals.
- **Risk avoidance**. There is a tendency for developers to defer the implementation of more difficult (and more risky) features until later increments (deferring critical user involvement and feedback). This can result in late term design changes, scrap, and rework.
- **Supplier selection risk**. Contractors must to be selected not only for their development ability, but also on their willingness to work closely and cooperatively with the program office. This may be difficult to assess.
- **Exploitation risk**. Evolutionary acquisitions may reduce the Government's bargaining power, because the initial contract may not encompass the entire task, and subsequent contracts are unlikely to be competed. In negotiating later increments and phases, the supplier may exploit this situation by quoting unreasonably high prices.
- **Patchwork quilt effects**. Poorly controlled software changes can severely reduce the software quality. Similarly, the incremental requirements definition may result in a lack of requirements coherency. [HENDERSON97]

5.6.1.3 GCCS Evolutionary Life Cycle Process

The **Global Command and Control System (GCCS)** is an important cornerstone in the midterm implementation phase of the Command, Control, Computers, Communications and Intelligence for the Warrior (C4IFTW) concept. C4IFTW is an initiative to address joint C4I interoperability issues with a migration strategy to reduce the large number of C2 systems currently in use by the Services. GCCS will provide a fused C4I picture of the battlespace by providing core planning and assessment tools needed by joint force commanders. GCCS is a user-focused program to deliver joint C2 capabilities through a client/server architecture that uses commercial, open systems standards. [SHALIKASHVILI98]

Figure 5-20. GCCS Unified C4I Battlespace Concept [SHALIKASHVILI98]

The GCCS is using an evolutionary acquisition strategy that allows the planning and implementation flexibility needed to keep pace with changing requirements and evolving technologies. **DoD 5000.2R** requirements identification, validation, priority setting, technical and economic evaluation, risk assessment, development and operational testing, oversight, and management processes have been tailored to the program's evolutionary nature, as illustrated in Figure 5-21.

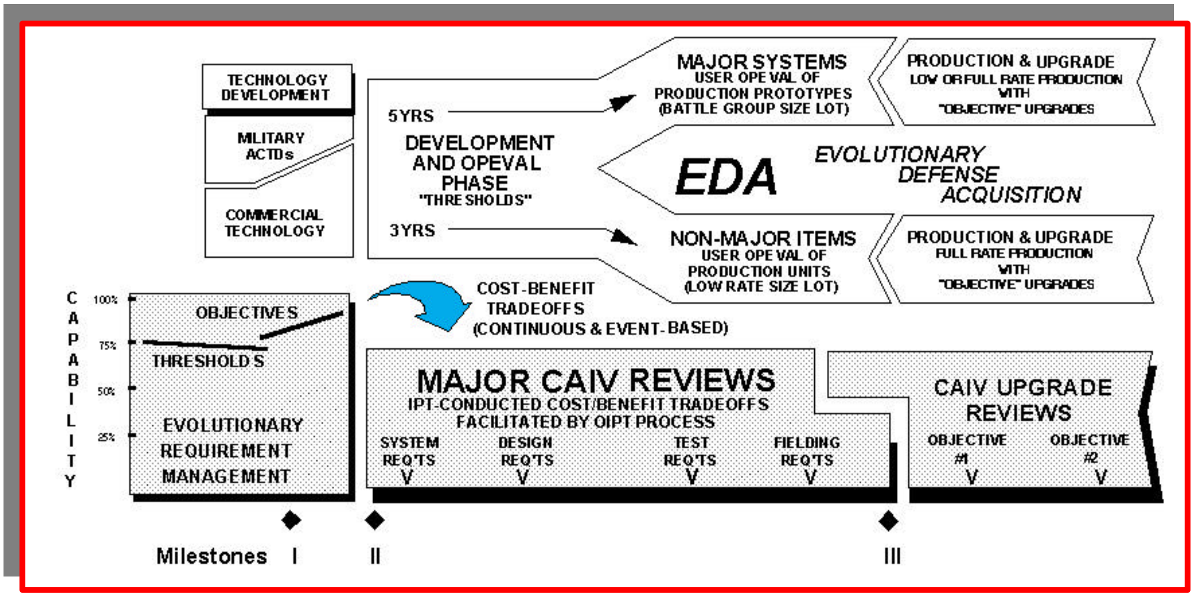

Figure 5-21. GCCS Tailored DoD 5000.2R Acquisition Process [CONDON98[1]]

The GCCS evolutionary acquisition strategy has coined a variety of new terms to describe the activities and procedures to implement flexible and responsive government oversight. Decision making and documentation processes have also been tailored to insure performance objectives and minimum acceptable requirements remain consistent with the GCCS MNS.

- **Evolutionary Acquisition Phases (EAPs).** EAPs are discrete time periods during which resources are used to fulfill mission needs.
- **Evolutionary Phase Implementation Plan (EPIP).** EPIPs formalize GCCS during the EAP, set forth cost, performance, schedule, test, economic, and budgetary issues, and identify deliverable C2 capabilities. EPIP planning and execution occurs concurrently.
- **Evolutionary Decision Review (EDR).** EDR reviews are conducted periodically to address multiple EPIP activities, as illustrated in Figure 5-22. Based on warfighter needs, technological opportunities, and available resources, EDRs are scheduled to assess ongoing EAP progress and approve the implementation of new EPIPs. Unlike traditional milestone reviews, EDRs are conducted at the lowest possible approval authority level commensurate with the risk inherent in the proposed EPIP, ongoing development success, and management activities. EDR decision making lies with empowered GCCS and IT OIPT members.

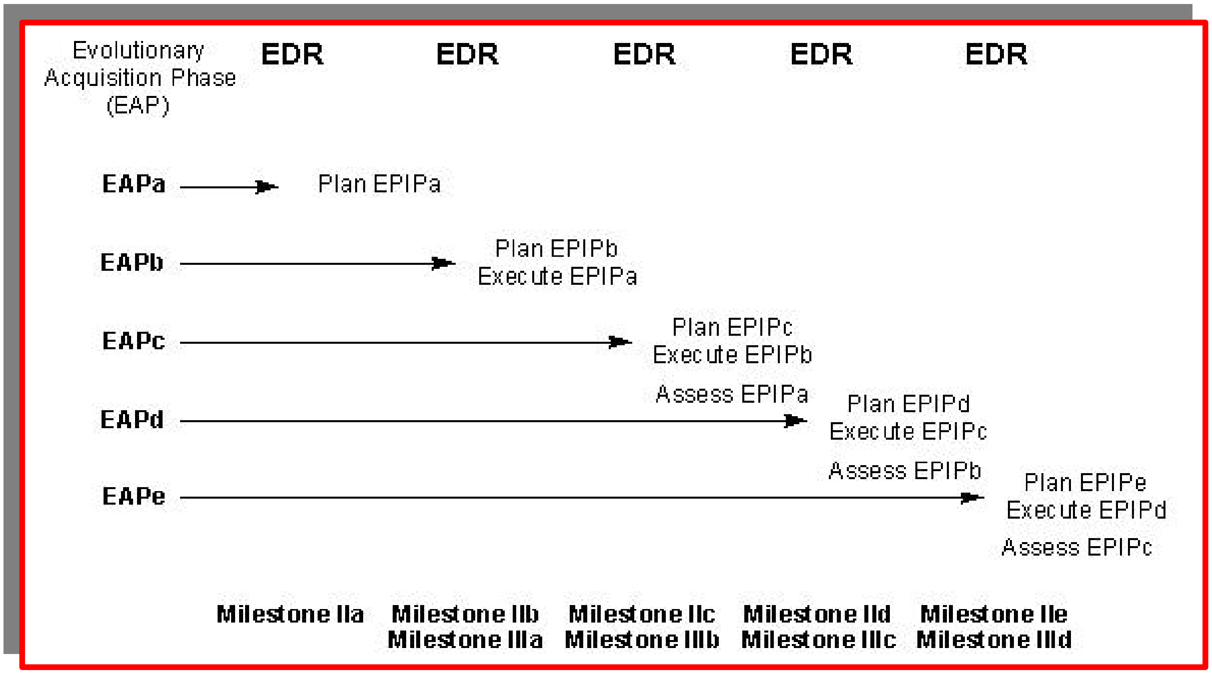

Figure 5-22. GCCS Milestones and Evolutionary Acquisition Phases [CONDON98[2]]

- **Segments.** Implementation activities that occur during an EAP are divided into manageable and responsive Segments (e.g., functional applications, technical or operational capabilities). Each Segment comprises a distinct development effort assigned to one or more GCCS stakeholder organizations and a set of deliverables. Segments (i.e., increments) provide a logical means for progressively translating broadly stated mission needs into well-defined system-specific requirements. Segments will ultimately evolve into operationally effective, suitable, and survivable systems.

The GCCS evolutionary acquisition strategy consists of a sequence of EAPs. For flexibility, Segments defined and initiated during one EAP may continue beyond that EAP, and in some cases, might span multiple EAPs. When this occurs, Segment progress is reviewed during EDRs when a decision is made to continue, modify, or discontinue the Segment. This allows the GCCS community to plan long-term activities and adjust the course of GCCS evolution over time in response to new technological opportunities and changing requirements, as illustrated in Figure 5-23. [CONDON98[2]]

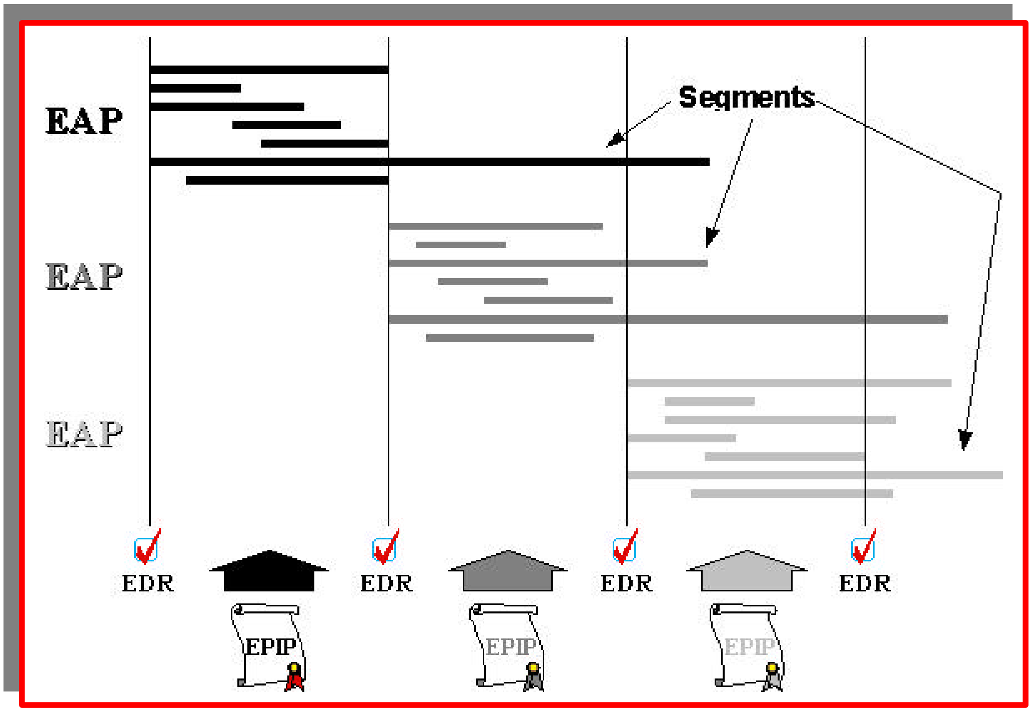

Figure 5-23. GCCS EAP, Segments, EDR, and EPIP Process **[CONDON98[2]]**

Core GCCS acquisition phases and milestones, tailored in accordance with DoD 5000.2R**,** minimize the time it takes to satisfy an identified need consistent with common sense and sound business practices. While GCCS acquisition phases and milestones are similar to those described in DoD 5000.2R, the time elapsed between milestones is much shorter to achieve the GCCS' planned 6-18 month implementation schedule. Figure 5-24 illustrates the GCCS evolutionary acquisition strategy mapped to the DoD 5000.2R process.

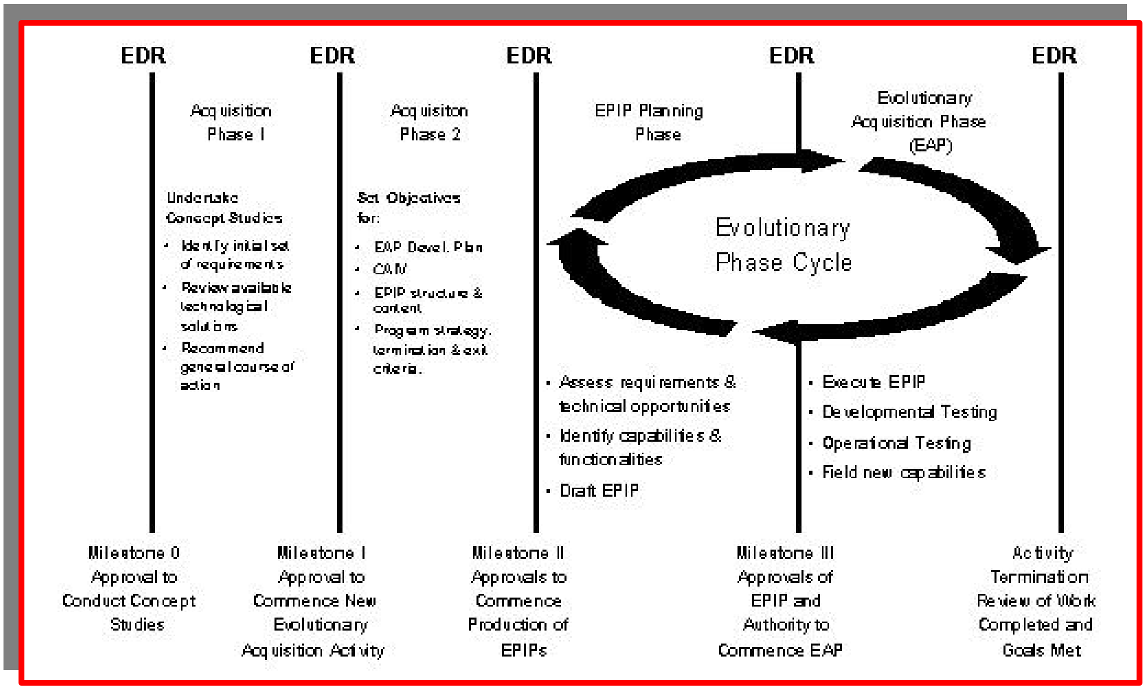

Figure 5-24. Overlay of GCCS Evolutionary Acquisition Strategy on DoD 5000.2R [CONDON98[1]]

5.6.2 Incremental Method

The incremental model is characterized by acquisition, development, and deployment of an operational capability through a series of clearly-defined, stand alone system increments. Using this strategy, user needs are determined, the architectural design is defined, and development occurs in a sequence of builds. The first build incorporates part of the planned capabilities. The next build adds more capabilities and so on until the system is complete, as illustrated in Figure 5-25. An example of this method is Pre-Planned Product Improvement (P3I) [discussed in Chapter 7, *Acquisition Planning*].

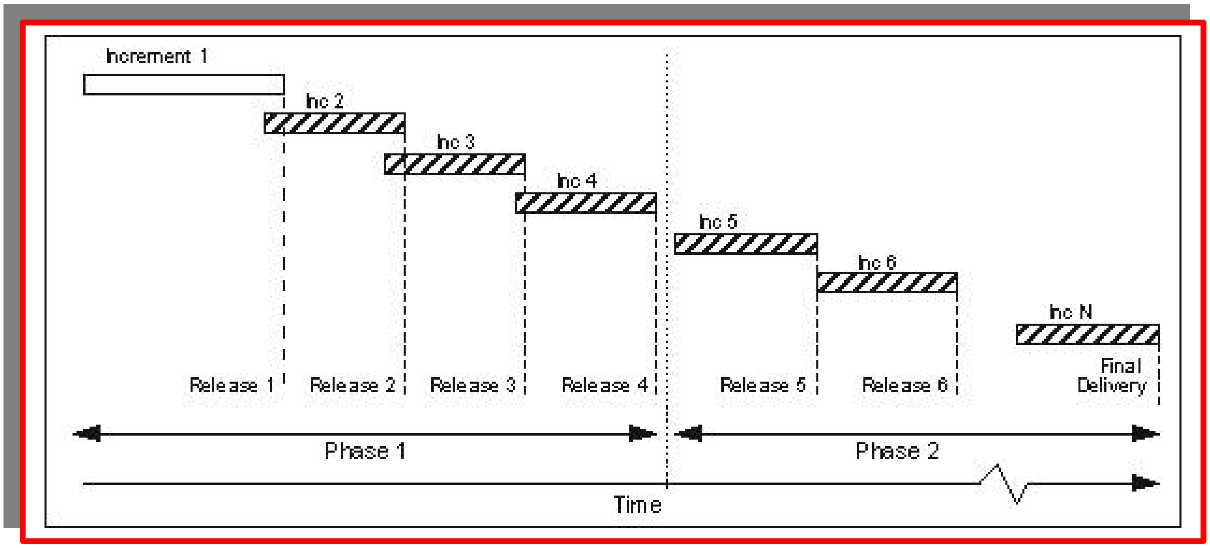

Figure 5-25. Basic Incremental Model

In deciding which activities apply to each increment, the requirements the increment will satisfy in the context of its hardware, software, and database components must be determined. To be useful, an increment should conform to the following rules:

- It satisfies a subset of the requirements to be met by the complete system.
- It constitutes an entity that can be used to demonstrate that a functional subset of the requirements has been met.
- It represents a logical design partition of one or more increments or builds.
- It provides a solid core for meeting the requirements assigned to the remaining builds.

The incremental life cycle management method lets the user implement a part of the final product. It is characterized by a *build-a-little*, *test-a-little* approach to deliver an initial functional subset of the final capability, as illustrated in Figure 5-26. This subset is subsequently upgraded or augmented until the total scope of the stated user requirement is satisfied. *An incremental methodology is most appropriate for low to medium-risk programs*. It is employed when user requirements can be *fully defined*, or the assessment of other considerations (e.g., risks, funding, schedule, program size, early benefit realization, etc.) indicate that a phased approach is the most auspicious.

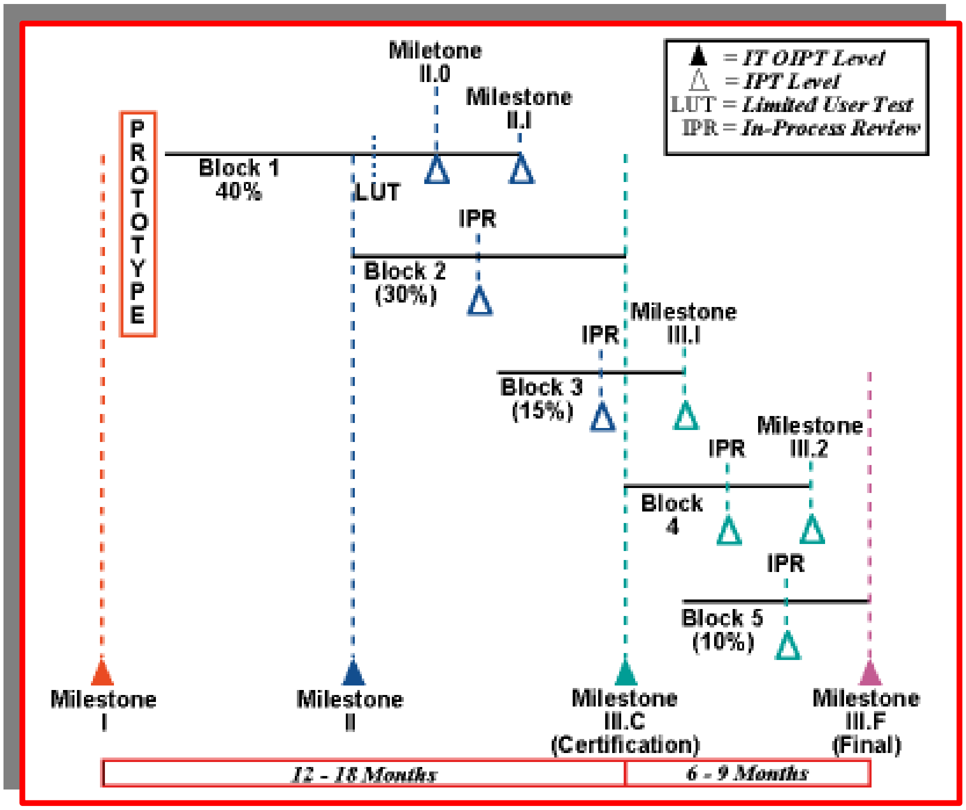

Figure 5-26. Example Incremental Life Cycle Method

Figure 5-27 provides an example of how the incremental method might be related to the IPT and review process. Allowing the user to employ the partially completed system before the entire system is integrated and tested promotes early discovery of problems and facilitates corrections.

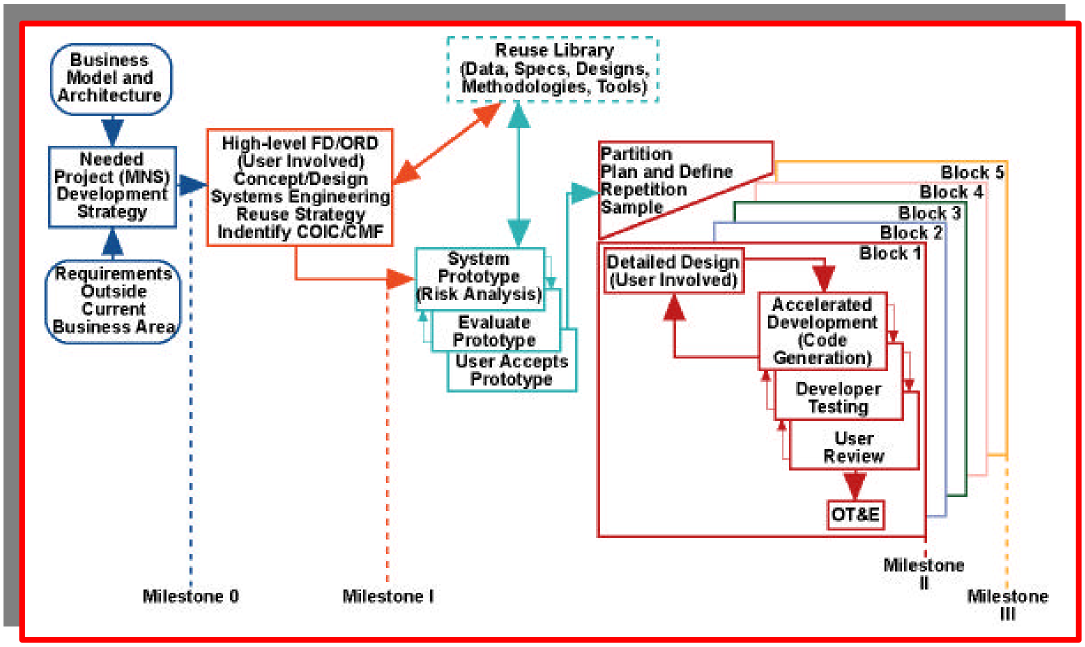

Figure 5-27. Example of Nominal Incremental Method

B aware, while the program as a whole may be large enough to merit IT OIPT (or DAB) oversight, *incremental development and fielding decisions only address small subsets of the system*. For example, a $240 million program with IT OIPT oversight may have a Milestone II Decision that impacts $15 million of program funds. Thus, it may be inappropriate for the MDA to make a $15 million milestone decision, although the decision to develop the subset of system functionality is, in fact, a Milestone II Decision (for a small effort). **DoD 5000.2-R** describes the elements required for a development decision (e.g., adequate cost estimate and funds, firm documentation of operational requirements, mature acquisition strategy, etc.).

5.6.2.1 Incremental Method Benefits

The following benefits may be realized through proper use of the Incremental Method:

- **Reduced risk**. Risk is spread across several smaller increments instead of concentrating in one large development;
- **Stable requirements**. Requirements are stabilized (through user buy-in) during the production of a given increment by deferring nonessential changes until later increments; and
- **Understandability**. Understanding of the requirements for later increments becomes clearer based on the user's ability to gain a working knowledge of earlier increments.

5.6.2.2 Cautions About the Incremental Method

Due to their nature, evolutionary/incremental acquisitions often encounter complications. Questions arise because each incremental build block provides only a small part of the system capability to be acquired. In addition to normal development decision criteria, additional questions must be answered, which include:

- Is the decision to develop this functionality for this amount of money a good idea?
- Is this the time to address the functionality question (user priorities, dictates of the evolution itself)?
- Is this a reasonable price for the functionality being added (or are we *gold plating* one functional area before developing all required capabilities)?
- Will we run out of money before we complete the required system?

ATTENTION! Critical to evolutionary or incremental methods is a sound *architecture,* which permits the addition of capability, core enlargement, or added increments.

5.6.3 Spiral Method

Spiral method (implemented by the Spiral Model), developed by Barry Boehm, provides a risk reducing approach to the software life cycle. As illustrated in Figure 5-28, radial distance in the Spiral Model is a measure of effort expended, while angular distance measures progress. It combines the basic waterfall *[discussed next]* building block and evolutionary/incremental prototype approaches to software development. The building block activities of architectural (preliminary) design, Preliminary Design Review (PDR), detailed design, Critical Design Review (CDR), code, unit test, integration and test, and qualification test are sequentially followed to deliver an *initial operational capability (IOC)*. After IOC, the product is reviewed to determine how its operational capability can be enhanced. Support and maintenance issues are revisited through risk analysis. The product is updated and an operational prototype(s) is demonstrated and validated. The system then goes through an updated waterfall development process with final delivery of a *full operational capability (FOC)* product.

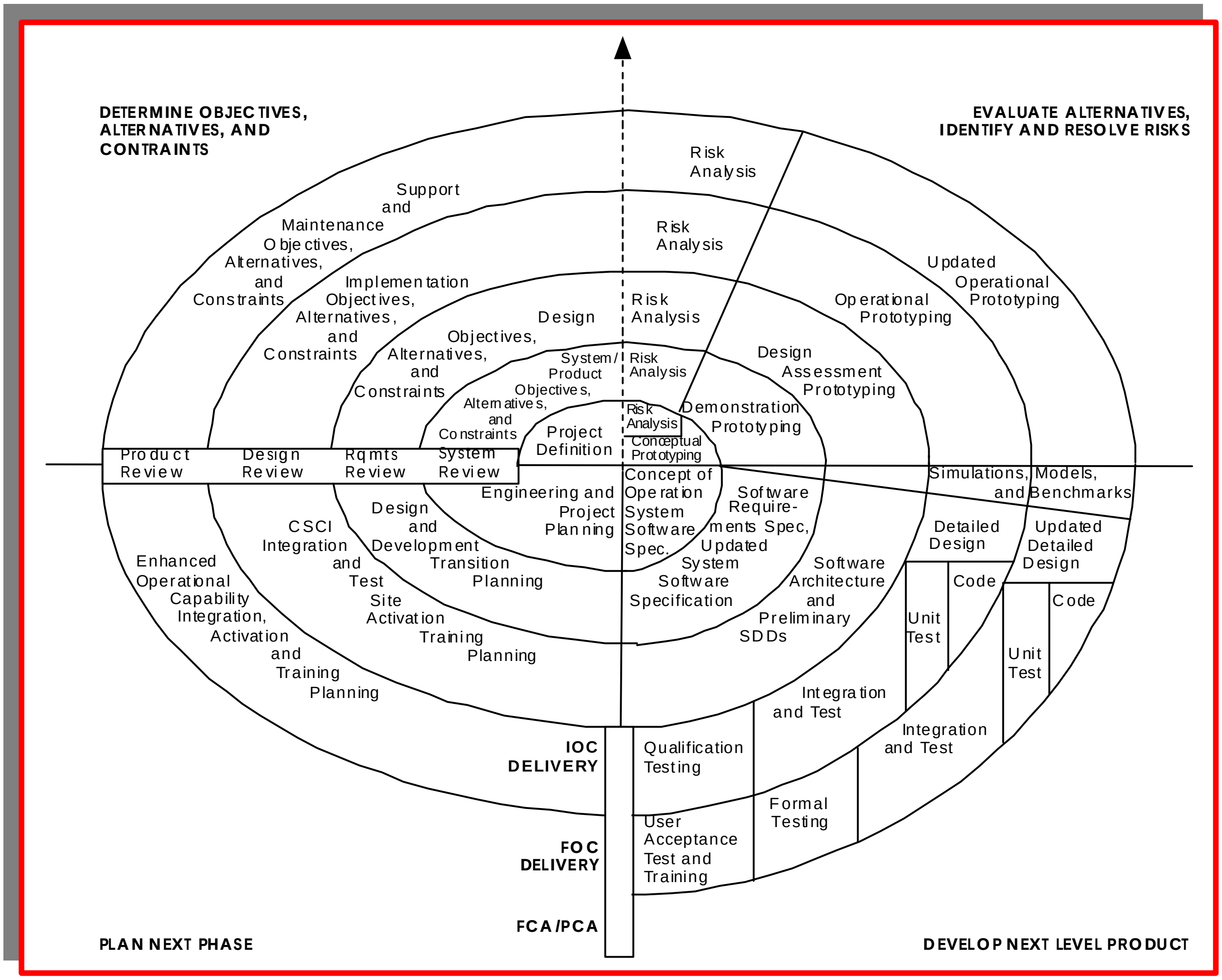

Figure 5-28. Spiral Model

The spiral method emphasizes the evaluation of alternatives and risk assessments, which are addressed more thoroughly than with other strategies. A review at the end of each phase ensures commitment to the next phase, or if necessary, identifies the need to repeat a phase. The advantages of the Spiral Model are its emphasis on procedures (such as risk management), and its adaptability to different life cycle approaches. If the Spiral Model is employed with demonstrations, baselining, and configuration management, continuous user buy-in and a disciplined process is achieved. [BOEHM88]

5.6.3.1 Ada Spiral Model Environment

The Ada Spiral Model Environment is an adaptation of Boehm's Spiral Model (Figure 5-28 above), which combines a model and tool environment. It uses Ada as a life cycle language and offers the ability to have continuous *touch-and-feel* of the software product (as opposed to paper reports and descriptions).

The Ada Spiral Model Environment is a performance-based process that employs a top-down incremental approach, early continuous design, and implementation validation. The advantage of this environment is that it builds from the top down, where each level of development is be demonstrated. Because Ada supports partial implementation, the structure is automated, realistic, and easily evolved. Each build and subsequent demonstrations validate the process and structure. Each level may serve as a formal baseline that is controlled. [WOODWARD89]

5.6.4 Choosing Among Evolutionary, Incremental, and Spiral Models

Most software-intensive acquisition programs use a combination of the evolutionary, incremental, and spiral methods. The Spiral Model is an overlay of either incremental or evolutionary with the addition of risk management. In the past, the Spiral Model has been difficult to implement in the DoD acquisition environment. Predefined deliverables and schedules do not easily accommodate repeating phases, changing deliverables, or requirement tradeoffs without difficult contract modifications. In a commercial environment, this approach can be described as *market-driven*, where *time-to-market* and competitive forces determine when a product must be delivered and which features are included (which may change rapidly in light of fierce competition). In reality, all software evolves. Commercial products are always evolving and Defense systems are continuously enhanced, modified, or evolved to new capabilities during *operations and support* (or *maintenance)*. In selecting the appropriate strategy, the unique circumstances of individual programs should be considered and the strategy chosen must remain consistent with DoD acquisition policy. The following factors should be considered when defining a life cycle management method:.

NOTE: The order of importance and weighting of factors will vary from program to program and among commercial and military applications.

FACTORS FOR SELECTING LIFE CYCLE METHOD
1. Time to market/deployment
2. Requirements stability and understanding
3. Technology obsolescence
4. Priority of user needs
5. Expected system useful life
6. Complexity
7. Parallel hardware development
8. Interfaces to existing and future systems
9. Effort size and magnitude

Table 5-12. Factors to Consider When Selecting a Life Cycle Management Methodology [QUANN95]

ATTENTION! Incorporating the principles of performance-based and M&S methods into the evolutionary and/or incremental methods is highly recommended, as they are proven techniques for delivering quality products and acquisition risk reduction.

5.6.5 Waterfall Model Method

The waterfall model was first identified in 1970 as a formal alternative to the *code-and-fix* software development method prevalent at the time. [ROYCE70] The waterfall model was the first to formalize a framework for software development phases, and placed emphasis on up front requirements and design activities and on producing documentation during early phases. This strategy is essentially a *once-through, do-each-step-once* strategy. Simplistically, waterfall steps include the following, as illustrated in Figure 5-29.

- Determine user needs,
- Define requirements,
- Design the system,
- Implement the system,
- Test, fix, and deliver.

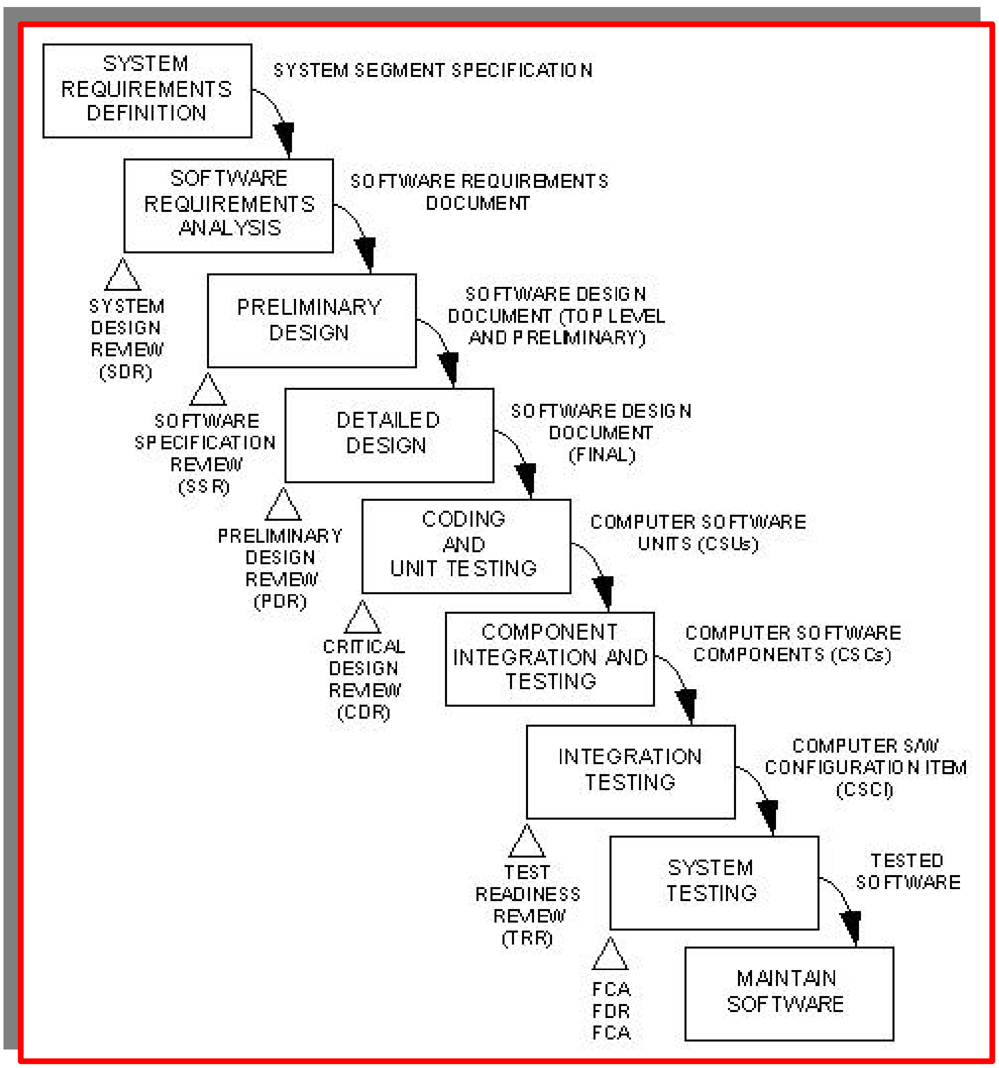

Figure 5-29. Waterfall Model [BRUNDICK95]

The major drawback to this model is its inherent sequential nature — any attempt to go back two or more phases to correct a problem or deficiency results in major increases in cost and schedule. While the Waterfall Model (also referred to as "*Grand Design*") provided an early structured method for the software life cycle, *it is not suited for modern development techniques*.

In the traditional Waterfall Model, each stage is a prerequisite for succeeding activities, making this method a risky choice for unprecedented systems because it inhibits flexibility. With a single pass through the process, integration problems usually surface too late. Also, a completed product is not available until the end of the process, discouraging user involvement. Taking these factors into account, *the other life cycle methods discussed in this chapter are recommended over the Waterfall!*

CAUTION! The waterfall method is NOT recommended for major software-intensive acquisition programs! If it must be used (due to the integration of added requirements into the system's overall system engineering methodology), the software management and engineering techniques described throughout these Guidelines must be used to reduce program risk.

5.6.6 Fast Track Methods

Although the focus of these Guidelines is on *"major"* software-intensive systems, a distinction between major and non-major programs should be understood. Software-intensive programs not considered major acquisitions and using a fast-track life cycle methodology require greater life cycle tailoring. This may be based on a time criticality arising from a variety of reasons, such as a national threat, technological obsolesce, or changed mission needs.

These programs are less formal and on a shortened life cycle to benefit the Government where the primary focus is on a fast track schedule (e.g., *proof-of-concept* programs). *Fast track or abbreviated software-intensive programs always assume short maintenance phases where system support is usually performed by the development contractor.*

Process tailoring is necessary to meet shortened life cycle requirements. While some acquisition and development steps must be maintained, the formality or scope of others are greatly reduced. Innovative methods may be appropriate if agreed upon by both the Government and the contractor before program start.

Organizations considering fast track or abbreviated programs should have demonstrated successful experience with similar technologies and have a mature, defined software development process to minimize risk. Fast track programs must also have a clearly defined, stable set of requirements. Abbreviated life cycle strategies may be appropriate when a program can proceed as an Engineering Change Proposal (ECP) to an existing contract, or where substantial familiarity exists and/or risk at a minimum.

ATTENTION! The objective of a fast track program is not to solve world hunger, but rather to feed a few starving beggars!

5.6.6.1 Concurrency

Concurrency is a fast track method where development and operational testing are combined with a concurrent follow-on development and initial production phase. With this method, government involvement is often limited. Lessons learned from the General Accounting Office (GAO) about employing fast track methods include the following recommendations.

- **Technical risk assessment**. An initial, detailed assessment of the technical risks involved in individual subsystems, as well as in the integration of those subsystems into a workable product must be performed. Whether the technology being attempted is compatible with an accelerated acquisition strategy must also be determined.
- **Flexible acquisition strategy**. Provisions must be built into the Acquisition Strategy for adjusting schedules and other program activities if technical difficulties occur.
- **Technology-based solution adjustments**. Technological progress must be periodically assessed to see if it is still compatible with the planned acceleration. If technical progress is no longer keeping pace with the acceleration, the strategy must be adjusted to bring it in line with the technology.
- **Testing for requirement satisfaction**. The Acquisition Strategy must include sufficient test and evaluation for requirement satisfaction so decision-makers can identify any major shortcomings in the system's operational suitability and effectiveness. Any issues identified during testing must be resolved before initial production is approved. [GAO86]

ATTENTION! See Chapter 10, *Software Development Maturity*, [for an in-depth discussion on the importance of the contractor's software development process, methodologies, tools, and capabilities. The software measurement life cycle is described in Chapter 13, *Software Estimation, Measurement, and Metrics*. Words to include in your Request for Proposal (RFP) requiring that offerors provide you with adequate information for proposal evaluation are found in Chapter 8, *Contracting for Success*. How to evaluate offerors' proposals so you select the contractor with the *best process* as well as product is also found in Chapter 8.

5.7 System Life Cycle and Your Program

When accused of making snap decisions, General George S. Patton, Jr. firmly proclaimed,

> "*I've been studying the art of war for forty-odd years. When a surgeon decides in the course of an operation to change its objective...he is not making a snap decision but one based on knowledge, experience, and training. So am I.*" [PATTON47]

The DoD acquisition life cycle process was designed so Program Managers are not forced into making snap decisions. Acquisition life cycle management is a structured process, replete with controls, reviews, and milestone decision points. The life cycle process is a logical flow of activity representing an orderly progression from the identification of a mission need to final operational deployment and support. It provides the basis for sound-decision making based on knowledge, experience, and training.

As a Program Manager, you must be prepared to develop a tailored management approach that will achieve an operational capability with the most effective use of resources and time available. Choosing the right life cycle management methodology depends on the nature of your program's operational environment, stability of requirements and technology, your software domain, the methods and tools used, and the controls and deliverables required. Whichever acquisition life cycle method you choose, remember the goals of acquisition reform:

- Reduced cycle times,
- Program stability,
- Cost savings, and
- Technology insertion.

If appropriate, life cycle phases can be combined or omitted. Some programs [especially in the case of AIS where technology is well-developed, purchased as COTS, or GOTS, or NDI] may enter the life cycle midstream. Each life cycle phase must be designed to instill a level of confidence in the solution(s) offered and to reduce the risks involved in making a decision to proceed to the next phase. Outputs of each phase must constitute a definitive, documented baseline for entry into subsequent phases.

5.8 References

[ANDERSON98] Anderson, Dave, "General Acquisition Process," Defense Acquisition Deskbook, U.S. Department of Defense, The Pentagon, Washington, DC, 18 December 1998.

[BOEHM88] Boehm, Barry W., "A Spiral Model of Software Development and Enhancement," *IEEE Computer*, May 1988.

[BOWSHER96] Bowsher, Charles A., Executive Guide: Effectively Implementing the Government Performance and Results Act, GAO/GGD-96-118, Comptroller General of the United States, United States General Accounting Office, Washington, DC, June 1996.

[BRUNDICK95] Brundick, Bill, Editor, *Parametric Cost Estimating Handbook*, Naval Sea Systems Command, Arlington, Virginia, Fall 1995.

[CONDON98[1]] Condon, Chris, "Mapping Evolutionary Acquisition to DoD-5000.2R," OSD/C3IA, May 1998; Defense Acquisition Deskbook, U.S. Department of Defense, The Pentagon, Washington, DC, 18 December 1998.

[CONDON98[2]] Condon, Chris, "GCCS Evolutionary Acquisition Process," OSD/C3IA, May 1998; Defense Acquisition Deskbook, U.S. Department of Defense, The Pentagon, Washington, DC, 18 December 1998.

[DRAYER98] Drayer, Lt. Col. Dennis (USAF), "Performanced-Based Business Environment: PBBE — A Business Vision We Can Live With," *Program Manager*, Defense Systems Management College, Fort Belvoir, Virginia, January-February 1998.

[DSMC90] Caro, Lt Col Isreal, et al., Mission Critical Computer Resources Management Guide, Defense Systems Management College, Fort Belvoir, Virginia, 1990

[GAO86] General Accounting Office, "Sergeant York: Concerns About the Army's Accelerated Acquisition Strategy," Report to the Chairman, Committee on Government Affairs, United States, Senate, GAO/NSIAD-86-89, May 1986.

[HENDERSON97] Henderson, Derek E., and Andrew P. Gabb, "Using Evolutionary Acquisition for the Procurement of Complex Systems," DSTO Electronics and Surveillance Research Laboratory, Salisbury, South Australia, March 1997; Defense Acquisition Deskbook, U.S. Department of Defense, The Pentagon, Washington, DC, 18 December 1998.

[HINTON98] Hinton, Jr., Henry L. *Best Practices: Successful Application to Weapon Acquisitions Requires, Changes in DOD's Environment*, GAO/NSIAD-98-56, National Security and International Affairs Division, United States General Accounting Office, Washington, D.C., 24 March 1998.

[HUMPHREY95] Humphrey, Watts S., personal communication to Capt. Joseph Stanko (USAF), 15 September 1995.

[KAMINSKI97] Kaminski, Paul G., John J. Hamre, and Emmet Page, Jr., OSD Memorandum: Requirements for Compliance with Reform Legislation for Information Technology (IT) Acquisitions, (Including National Security Systems), Office of the Secretary of Defense, The Pentagon, Washington, DC, 1 May 1997

[MONTGOMERY58] Montgomery of Alamein, Field Marshall Viscount, Bernard Law editor, The Memoirs of Field Marshall Montgomery, The World Publishing Company, Cleveland, Ohio, 1958.

[PASSMORE94] Passmore, John M., "Life Cycle Methodology Offers Software Starting Point," *Signal*, March 1994.

[PATTON47] Patton, GEN George S. (USA), War As I Knew It, Houghton Mifflin Company, Boston, 1947.

[PIPLANI94] Piplani, COL Lalit K. (USA), Lt. Col Joseph G. Mercer (USAF), and Lt. Col. Richard O. Roop (USAF), Systems Acquisition Manager's Guide for the Use of Models and Simulations, Report of the DSMC 1993-1994 Military Research Fellows, Defense Systems Management College Press, Fort Belvoir, Virginia, September 1994.

[PRESSMAN92] Pressman, Roger S., Software Engineering: A Practitioner's Approach, Third Edition, McGraw-Hill, Inc., New York, 1992.

[PRESSMAN93] Pressman, Roger S., "Understanding Software Engineering Practices: Required at SEI Level 2 Process Maturity," Software Engineering Training Series, Software Engineering Process Group, 30 July 1993.

[PRESTON94] Preston, Colleen, Testimony to Before the United States Congress, March 1994.

[QUANN95] Quann, Eileen., personal communication to Lloyd K. Mosemann, II, September, 1995.

[ROYCE70] Royce, Winston W., "Managing the Development of Large Software Systems," *IEEE*, WESCON, 1970.

[SHALIKASHVILI98] Shalikashvili, GEN. John M., C4I for the Warrior Brochure, Office of the Joint Chiefs of Staff, The Pentagon, Washington, DC, 1998.

[WOODWARD89] Woodward, Herbert P., "Ada: A Better Mousetrap," *Defense Science*, November 1989.