15

OPERATIONS AND SUPPORT

15.1 INSTALLING, MAINTAINING, AND UPGRADING THE SYSTEM

The operations and support phase of the system life cycle is the time during which the products of the system development and production phases perform the operational functions for which they were designed. In theory, the tasks of systems engineering have been completed. In practice, however, the operation of modern complex systems is never without incident. Such systems usually require substantial technical effort in their initial installation and can be expected to undergo significant testing and component replacement during periodic maintenance periods. Occasional operational glitches must also be expected due to operator error, operating stresses, or random equipment failures. In such cases, systems engineering principles must be applied by system operators, maintenance staff, or outside engineering support to identify the cause of the problem and to devise an effective remedy. Further, large complex systems, such as an air traffic control system, are too costly to replace in their entirety and therefore are subject to major upgrades as they age, which introduce new subsystems in place of obsolescent ones. All of these factors are sufficiently significant in the total role of systems engineering in the overall system life cycle to warrant a special place in the study of systems engineering.

The principal sections of this chapter summarize the typical activities that take place in the course of a system's operating life, beginning with the time it is delivered from the production or integration facility to the operational site until it is replaced by a newer system or otherwise rendered obsolete and disposed of. The section on *installation and test* deals with problems associated with integrating the system with its operating site and the successful interconnection of internal and external interfaces. The section covering *in-service support* concerns activities during the normal operations of the system; these include maintenance, field service support, logistics, and dealing with unexpected operational emergencies. The section on *major system upgrades* is concerned with periodic subsystem modifications that may be introduced to maintain system effectiveness in the face of changing user requirements and advancing technology. Such system upgrades require the same type of systems engineering expertise as did the original system development, and may also present new and unique challenges due to added constraints that may be imposed by the process of integrating new and old components. The last section on *operational factors in system development* describes the kinds of information that systems engineers should seek to acquire regarding the operational characteristics of the system being developed, together with the opportunities that they may have for obtaining such knowledge. Such knowledge is just as important to systems engineers who lead the system development as is a firm grounding in factors that affect system production processes and costs.

Place of the Operations and Support Phase in the System Life Cycle

Before discussing the systems engineering activities during the operations and support phase, it should be noted that this phase is the concluding step of the system life cycle. The functional flow diagram of Figure 15.1 shows the inputs from the production phase to be operational documentation and a delivered system, and the outputs to be an obsolete system and a plan for disposing it in an appropriate way.
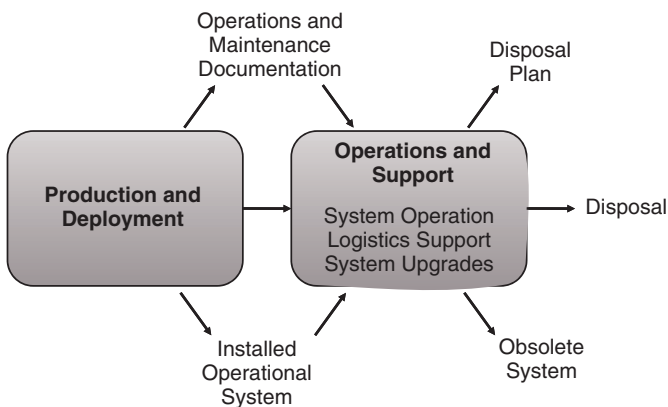
Figure 15.1. Operations and support phase in a system life cycle.
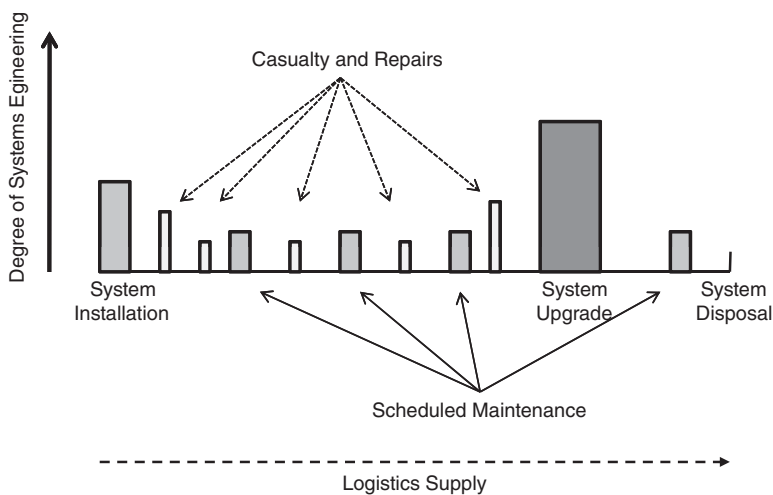
Figure 15.2. System operations history.

Systems Engineering in the Operations and Support Phase

During its operating life, a typical complex system encounters a number of different periods when its operation is interrupted. These incidents are represented in Figure 15.2. The abscissa is time, running from system delivery to its disposal. The ordinate represents the relative level of systems engineering involvement in the various events identified by the captions in the upper part of the figure. At the start, a column is seen symbolizing the installation and test period, which is shown to take substantial time (usually weeks or months) and a relatively large systems engineering effort. The four low, regularly spaced columns represent planned maintenance periods, which may require days of system downtime. The narrow spikes at irregular intervals are meant to correspond to random system breakdowns requiring emergency fault identification and repairs. These are usually fixed quickly but may take considerable systems engineering effort to find a solution that can be effected with minimum downtime. The large column on the right represents a major system upgrade, requiring a relatively long period (many months) and a high level of systems engineering. The latter may rival the effort involved in a new system development and may itself require a multiphase approach.

15.2 INSTALLATION AND TEST

System Installation

The effort required for installing a delivered system at its operating site is strongly dependent on two factors: (1) the degree of physical and functional integration that had

been accomplished at the production facility and (2) the number and complexity of the interfaces between the system and the operating site (including other interacting systems). In the case of an aircraft, for example, virtually all significant system elements typically are assembled and integrated at the prime contractor's factory site so that when the aircraft leaves the production facility, it is ready for flight. The same is true for an automobile, a military truck, or almost any kind of vehicle.

The installation of many large-scale systems on a land or ship platform may be a major operation, especially if some of its subsystems are manufactured at separate locations by different contractors and are assembled only after delivery to the operating site. For example, an air terminal control system typically consists of several radars, a computer complex, and a control tower with an array of displays and communication equipment, all of which must be integrated to operate as a system and linked to the en route control system, runway landing control systems, and an array of associated equipment required to handle air traffic in and out of an airport. The installation and test of such a system is in itself a major systems engineering enterprise. Another such example is a ship navigation system, which consists of many subsystems that are manufactured at various sites, frequently by different contractors, and which have complex interfaces with ship elements. After the initial ship systems pass integration tests at a land site, subsequent production subsystems are often assembled and integrated only after they are delivered to the shipyard. The task of interfacing elements of the system with ship structure, power, controls, and communications is usually performed at the shipyard by experts in ship installations.

Internal System Interfaces. As discussed previously, the systems engineer has a responsibility to assure that system integrity is maintained throughout assembly and installation. Installation procedures must be carefully planned and agreed to by all involved organizations. The systems engineer must be a key participant in this planning effort and in seeing it properly carried out. However, regardless of the degree of planning, the proper integration of subsystem interfaces will always be a potential source of trouble and therefore deserves special effort.

As noted previously, interfacing of the subsystems at the operating site is especially complicated when major subsystems are designed and manufactured by different contractors. In shipboard systems, two common examples of such subsystems are propulsion and communications. These subsystems include interfacing elements that employ both low- and high-power digital and analog signals, together with numerous switching and routing processors. Some of these equipments will be new state-of-the-art elements designed specifically for this project, and some will be older, off-the-shelf items.

Under such circumstances, problems during installation and checkout are almost certain to be encountered. Moreover, some problems will be difficult to track down because the necessary resources, such as test specialists and troubleshooting equipment, may not be available at the installation site. In such instances, it is not unusual for the acquisition agency to bring in special "tiger teams" for assistance. If available, the people who developed the original system, particularly the systems engineers who will have the most system-level knowledge and management skills, are best qualified to work on these problems.

System Integration Site. For systems where integration of major subsystems is especially difficult to accomplish at the operational site, it becomes cost-effective to utilize a specially equipped and supported integration site where subsystems and components are assembled and various levels of checkout are performed prior to partial disassembly and shipment to the operational site. This may be the same integration site that is used during development to test and evaluate various elements of prototype equipment, or it may be a separate facility also used for training of operators and maintenance personnel. In either case, such a special site can also be extremely valuable in checking out fixes to problems encountered during initial operations, as well as in supporting the engineering of system upgrades.

External System Interfaces. In addition to numerous internal subsystem interfaces, complex systems have many critical external interfaces. Two examples are prime power, which is usually generated and distributed by an external system, and communication links, which interface through hardwired electronic circuits or by microwave links. Communication links not only have to be electronically compatible but must also have the appropriate set of message protocols, which are usually processed by software.

A further complicating factor is that large systems must often interface with systems that are procured from developers who are not under the control of either the prime system contractor or the system acquisition agent. This means that design changes, quality control, delivery schedules, and so on, can become major coordination issues. This also makes the resolution of problems more troublesome by raising the issue of who may be at fault and should therefore assume responsibility for correcting any resulting problems. Problems of this type emphasize the importance of having a well-planned and executed test program during system assembly and integration.

During system development, special pains must be taken to ensure that the details of external interactions are fully specified early in the design process. In many cases, the documentation of interacting systems is insufficiently detailed and sometimes so far out of date that their interface connections to the newly developed system are no longer valid. Systems engineers who have first-hand experience with system environments can often anticipate many such critical factors relating to external interfaces, thereby ensuring that their characteristics are defined early enough in the development process to avoid problems during system installation.

Communication links other than standard commercial communications are notoriously troublesome. They often employ special connections and message protocols whose detailed specifications are difficult to obtain ahead of time.

The result can lead to surprises during system installation and initial operation, with no clear evidence as to which organization is responsible for the incompatibility and capable of resolving it. In such circumstances, it is usually advisable for the development contractor to take the initiative to at least identify the specific technical problem and to propose the means for a solution. Otherwise, the blame for the lack of system interoperability is commonly placed arbitrarily on the new system and its developer.
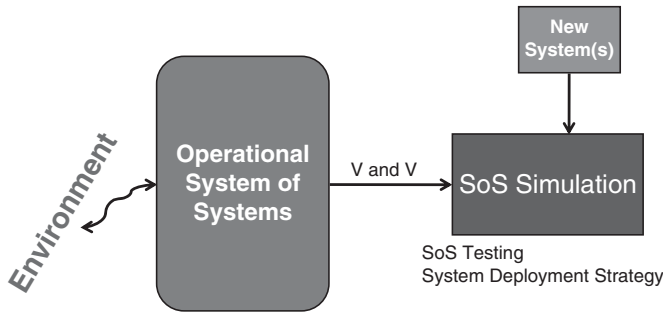
Figure 15.3. Non-disruptive installation via simulation.
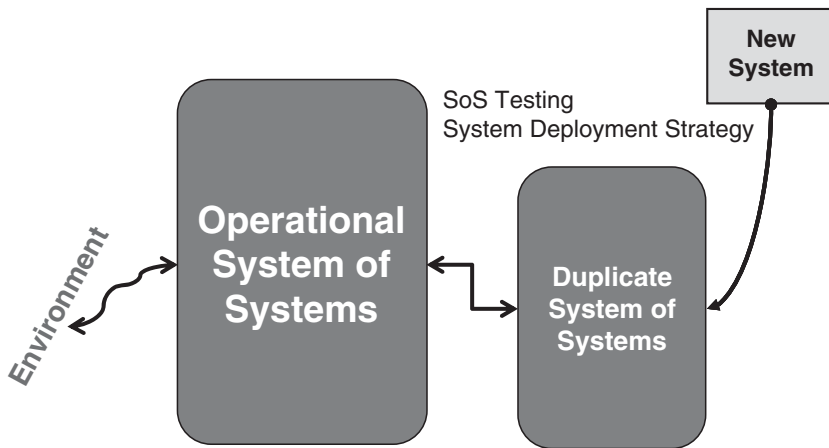
Nondisruptive Installation

Some critical systems require continuous operations and cannot be stopped or paused during system installation or upgrades. This tends to be the case when installing a system into a large system of systems. The installation of a new or upgraded system into the system of systems cannot disrupt current operations. Examples include system installation into a city power grid, a complex industrial wide area network, a national communications network, a major defense system of systems, and the national air traffic control system of systems. All of these examples require 24-hour operations without significant disruption.

Installing major systems into a system of systems without disruption requires careful planning and attention to detail. In the recent past, two general approaches have emerged to assist in this area: maintaining a system of systems simulation and maintaining a system of systems test bed. Figure 15.3 depicts the first option.

With this strategy, a system of systems simulation with hardware-in-the-loop is created. This simulation is typically user-in-the-loop as well, as opposed to stand alone. This simulation facility is verified and validated against actual data collected from the operational system of systems, which interacts with the environment. Typically, the simulation would not interact with the environment (although there are exceptions to this).

The system of systems simulation is used as a test bed to determine (1) the impact the new system will have on the system of systems before it is actually installed and (2) an installation strategy that will keep operations at an acceptable level. Once a strategy has been developed and verified using the system of systems simulation facility, knowledge and confidence is gained on how to install the system into the actual system of systems.

The advantage of this nondisruptive installation mode is the cost savings and the ability to model installation procedures and techniques before installing the system into the actual system of systems. The system of systems simulation facility, while expensive and complex, is only a representation of the actual system of systems and can be scoped to desired budget and tolerance levels. Obviously, if the system of systems in

Figure 15.4. Non-disruptive installation via a duplicate system.

question is a defense network responsibility for the survival of a nation, extremely high tolerances would be required. However, if the system of systems is a business information technology (IT) network, tolerance may be relaxed to a comfortable risk level.

The second concept used within nondisruptive installations involves the development of a duplicate system of systems, scaled down from the operational one, and is depicted in Figure 15.4. The concept is similar to the first concept in that the system is installed into the scaled-down version of the system of systems, and testing occurs. During this process, the duplicate system of systems is typically disconnected from the operational system of systems to avoid any interference or disruption. An installation strategy is developed from the experience to apply for installation into the full-scale system of systems.

Once confidence in the risk of disruption is acceptable, the system is installed into the operational system of systems. Many times, the operational system of systems is disconnected from the environment—the duplicate system of systems is used as a surrogate during the installation. This is typically performed during a low demand situation or time frame to ensure the limited capacity of the duplicated system of systems is sufficient.

Although this strategy for nondisruptive installation is expensive (you are basically building a scaled-down version of the operation system of systems), it has two major benefits: (1) the duplicate system of systems is an architecture copy of the operational system of systems and is the closest representation that is possible without duplicating both the architecture and scale of the original; and (2) during peak demand, the duplicate, scaled-down system of systems can be used to augment the operational system of systems. National communications systems use this technique to keep its networks operational continuously, and to allow for unexpected peak demand periods.

Facilities and Personnel Limitations

Neither the facilities nor the personnel assigned to the task of system installation and test are normally equipped to deal with significant difficulties. Funds are inevitably budgeted on the assumption of success. And, while the installation staff may be experienced with the installation and test of similar equipment, they are seldom knowledgeable about the particular system being installed until they have gained experience during the installation of several production units. Moreover, the development contractor staff consists of field test engineers, while systems engineers are seldom assigned until trouble is encountered, and when it is, the time required to select and assign this additional support can be costly.

The lesson to be learned is that the installation and test part of the life cycle should be given adequate priority to avoid major program impact. This means that particular attention to systems engineering leadership in the planning and execution of this process is a necessity. This should include the preparation and review of technical manuals describing procedures to be followed during installation and operation.

Early System Operational Difficulties

Like many newly developed pieces of equipment, new systems are composed of a combination of new and modified components and are therefore subject to an excessive rate of component failure or other operational problems during the initial period of operation, a problem that is sometimes referred to as "infant mortality." This is simply the result of the difficulty of finding all system faults prior to total system operation. Problems of this type are especially common at external system interfaces and in operator control functions that can be fully tested only when the system is completely assembled in an operational setting. During this system shakedown period, it is highly desirable that a special team, led by the user and supported by developer engineers, be assigned to rapidly identify and resolve problems as soon as they appear. Systems engineering leadership is necessary to expedite such efforts, as well as to decide what fixes should be incorporated into the system design and production, when this can best be done, and what to do about other units that may have been already shipped or installed. The need for rapid problem resolution is essential in order to effect necessary changes in time to resolve uncertainties regarding the integrity of the production design. Continuing unresolved problems can lead to stoppages in production and installation, resulting in costly and destructive impact on the program.

15.3   IN-SERVICE SUPPORT

Operational Readiness Testing

Systems that do not operate continuously but that must be ready at all times to perform when called upon are usually subjected to periodic checks during their standby periods to ensure that they will operate at their full capability when required. An aircraft that has been idle for days or weeks is put through a series of test procedures before being

released to fly. Most complex systems are subjected to such periodic readiness tests to ensure their availability. Usually, readiness tests are designed to exercise but not to fully stress all functions that are vital to the basic operation of the system or to operational safety.

All systems, sooner or later, will experience unexpected problems during operational use. This can occur when they encounter environmental conditions that were not known or planned for during development. Periodic system tests provide information that helps assess and resolve such problems quickly when they occur.

Periodic operational readiness tests also provide an opportunity to collect data on the history of the system operating status throughout its life. When unexpected problems occur, such data are immediately available for troubleshooting and error correction. System readiness tests have to be designed and instrumented with great skill to serve their purpose effectively and economically—a true systems engineering task.

Readiness tests often must be modified after system installation to conform more fully to the needs and capabilities of system operators and maintenance personnel. Development systems engineers can effectively contribute to such an activity. Location of data collection test points and the characteristics of the data to be collected, for example, data rate, accuracy, recording period, and so on, also represent systems engineering decisions.

Commonly Encountered Operational Problems

Software Faults. Faults in complex software-intensive systems are notoriously difficult to eliminate and tend to persist well past the initial system shakedown period. The difficulties stem from such inherent features as the abstractness and lack of visibility of software functionality, sparseness of documentation, multiplicity of interactions among software modules, obscure naming conventions, changes during fault resolutions, and a host of other factors. This is especially true of embedded real-time software commonly found in dispersed automated systems.

The variety of computer languages and programming methodology further complicates system software support. While most analog circuitry has been replaced by digital circuits in signal processing and many other applications, computer code written in older languages, such as COBOL, FORTRAN, and JOVIAL, is still in widespread use. This "legacy" code, mixed with more recent and modern code (e.g., C++, Java), makes it that much more difficult to maintain and modify operational computer programs.

Remedies for software faults are correspondingly complicated and troublesome. A corrective patch in a particular program module is likely to affect the behavior of several interacting modules. The difficulty of tracing all paths in a program and the mathematical impossibility of testing all possible conditions make it virtually impossible to ensure the validity of changes made to correct faults in operational software.

The relative ease of making software changes often leads to situations where these changes are made too quickly, and without significant analysis and testing. In such cases, documentation of the system changes is likely to be incomplete, causing difficulties in system maintenance.

The only way to prevent serious deterioration of system software quality is to continue to subject all software changes to strict configuration control procedures and formal review and validation as practiced during the engineering design and production phases. As noted elsewhere, proving-in changes at a test facility by experienced software engineers prior to installing these in the operational system is an excellent practice; this procedure will pay for itself by minimizing the inadvertent introduction of additional faults in the course of system repair. Chapter 11 is devoted to a discussion of all of the special aspects of software engineering.

   Complex Interfaces. In the section on system installation and test (Section 15.2), it was stated that external system interfaces were always a potential source of problems. During installation, there is always a strong push for accomplishing the process as quickly as possible so that operational schedules are maintained. So, while documented installation procedures are generally followed, insufficient time is often allocated to exercise thoroughly the necessary checkout procedures. As noted earlier, examples of areas where operational problems typically show up in a shipboard system are displays, navigation, and communication subsystems. The control panels for these subsystems are usually distributed among various locations and therefore have a strong functional as well as physical interaction. In such cases, the operational crews should be alerted to the potential problems and should be provided with explicit information on the locations and interfaces of all interacting system elements.

Field Service Support

It is common for deployed complex systems to require field support during the lifetime of the system. In the case of military systems, this is often provided by an engineering support unit within a branch of the service. It is also common for that unit to contract with civilian agencies to provide general engineering support to keep the system operating as intended.

   When system operating problems are detected, it is necessary first to determine whether the problem is due to a fault in the operational system or is a result of improper functioning of a built-in fault indicator. For example, the device may be erroneously signaling a failure (false alarm) or may be ascribing it to the wrong function. Therefore, the field engineer who is called upon to troubleshoot a problem should be knowledgeable in system operation, including especially the functioning of built-in test devices.

   When any fault is encountered during system operation, the required remedial actions are more difficult to implement than they would have been during development or even during installation and test. This is because (1) user personnel are not technical specialists; (2) special checkout and calibration equipment used during installation will have been removed; (3) most analysis and troubleshooting tools (e.g., simulations) are not available at the operational site; and (4) most knowledgeable people originally assigned to the development project are likely to have been reassigned, to have changed jobs, or to have retired. Because of these factors, for operational fixes to be done reliably, they often have to be developed remotely; that is, data will have to be collected at the operational site and transmitted back to the appropriate development site for

analysis; corrective action will have to be formulated; and finally, the required changes will have to be implemented at the operational site by a special engineering team.

As noted previously, facilities at the developer's test site are excellent locations for follow-on system work because of the availability of knowledgeable people, configuration flexibility, extensive data collection and analysis equipment, and the opportunity to carry out disciplined and well-documented tests and analyses.

Scheduled Maintenance and Field Changes

Most complex systems undergo periods of scheduled maintenance, testing, and often revalidation. Nonemergency field changes are best accomplished during such scheduled maintenance periods, where they can be carried out under controlled conditions by expert personnel and can be properly tested and documented. Fortunately, this usually accommodates the majority of significant changes. In most cases, as in that of commercial aircraft, such operations utilize special facilities with a full complement of checkout equipment, have a substantial parts stockpile and an automated inventory system, and are conducted by specially trained personnel.

Any changes, large or small, to an operational system require careful planning. As noted earlier, changes should be made under configuration control and should conform to documentation requirements that specifically state how they will be carried out. All changes should be viewed from a system perspective so that a change in one area does not cause new problems in other areas. Any technical change to an operational system will usually also require changes in hardware–software system documentation, repair manuals, spare parts lists, and operating procedure manuals. In this process, systems engineering is required to see that all issues are properly handled and to communicate these issues to those responsible for the overall operation.

Severe Operational Casualties

The previous paragraphs dealt with operational problems that could be corrected during operations or short periods of scheduled maintenance. It must be assumed that a complex system built to operate for a dozen years or more may accidentally suffer a failure of such magnitude that it is effectively put out of commission until corrected, such as by a fire, a collision, or through other major damage. Such a situation normally calls for the system to be taken out of service for the time necessary to repair and reevaluate it. However, before undertaking the drastic step of an extended interruption of service, a systems engineering team should be assembled to explore all available alternatives and to recommend the most cost-effective course of action for restoring operation. The severe casualty poses a classical system problem where all factors must be carefully weighed and a recovery plan developed that suitably balances operational requirements, cost, and schedule.

Logistics Support

The materials and processes involved in the logistics support of a major operational system constitute a complex system themselves. The logistics for a major fielded system

may consist of a chain of stations, extending from the factory to the operational sites, which supplies a flow of spare parts, repair kits, documentation, and, when necessary, expert assistance as required to maintain the operating system in a state of readiness at all times. Technical manuals and training materials should be considered part of system support. The effort of developing, producing, and supporting effective logistics support for a major operating system can represent a substantial fraction of the total system development, production, and operating cost.

A basic problem in logistics support is that it must be planned and implemented on the basis of estimates of which system components (not yet designed) will need the most spare parts, what the optimum replacement levels will be for the different subsystems (not yet completely defined), what means of transportation, and hence time to resupply, will be available in potential (hypothetical) theaters of operation, and many other assumptions. These estimates can benefit from strong systems engineering participation and must be periodically readjusted on the basis of knowledge gained during development and operating experience. This means that logistic plans will need continual review and revision, as will the location and stocking level of depots and transport facilities.

There are also direct connections between the logistics support system and system design and production. The sources of most spare parts are usually the production facilities that manufacture the corresponding components and may include the system production contractor and the producers of system components. Moreover, subcomponents and parts commonly include commercial elements and hence are subject to obsolescence design changes or discontinued availability.

System field changes also directly affect the logistic supply of the affected components and other spare parts. Since the process of reflecting such changes in the logistics inventory cannot be instantaneous, it is essential to expedite it, as well as to maintain complete records of the status of each affected part wherever it is stored.

It can be seen from the above that the quality and timeliness of overall support provided by the logistic system will have direct effects on operability. This is particularly true for systems operating in the field, where the timely delivery of spare parts can be crucial to survival. In the case of commercial airlines, timely delivery of needed parts is also critical to maintaining schedules. Managing such a logistics enterprise is itself an enormous task of vital importance to the successful operational capability of the system.

15.4   MAJOR SYSTEM UPGRADES: MODERNIZATION

In the chapters dealing with the origin of new systems, it was noted that systems are usually developed in response to the forces of advancing technology and competition, which combine to create technical opportunities and generate new needs. Similarly, during the development and operational life of a system, the dynamic influence of these same factors continues, thereby leading to a gradual decrease in the system's effective operational value relative to advances made by its potential competitors or adversaries.

Advances in technology are far from uniform across the many components that constitute a modern complex system. The fastest growth has been in semiconductor technology and electro-optics, with the resultant dramatic impact on computer speed and memory and on sensors. Mechanical technology has also advanced, but mainly in relatively limited areas, such as special materials and computer-aided design and manufacture. For example, in a guided missile system, the guidance components may become outdated, while the missile structure and launcher remain effective.

Thus, obsolescence of a large complex system often tends to be localized to a limited number of components or subsystems rather than affecting the system as a whole. This presents the opportunity of restoring its relative overall effectiveness by replacing a limited number of critical components in a few subsystems at a fraction of the cost of replacing the total system. Such a modification is usually referred to as a system upgrade. Aircraft generally undergo several such upgrades during their operating life, which, among other modifications, incorporate the most advanced computers, sensors, displays, and other devices into their avionics suites. A complication often encountered is discontinued production by manufacturing sources, which requires adjusting system interfaces to fit the replacements.

System Upgrade Life Cycle

The development, production, and installation of a major system upgrade can be considered to have a mini life cycle of its own, with phases that are similar to those of the main life cycle. Active participation by systems engineering is therefore a vital part of any upgrade program.

Conceptual Development Stage. Like the beginning of a new system development, the upgrade life cycle begins with the recognition through a needs analysis process of a need for a major improvement in mission effectiveness because of growth in the mission needs and deficiencies in the current system's response to these needs.

There follows a process of concept exploration, which compares several options of upgrading a portion of the current system with its total replacement by a new and superior system, as well as with options for achieving the objective by different means. If the comparison shows a convincing preference for the strategy of a limited system modification or upgrade, and is feasible both technically and economically, then a decision to inaugurate such a program is appropriate.

The equivalent of the concept definition phase for a system upgrade is similar to that for a new system, except that the scope of system architecture and functional allocation is limited to designated portions of the system and to those components that contain the parts of the system to be replaced. Proportionally greater effort is required to achieve compatibility with the unmodified parts of the system, keeping the original functional and physical architecture unaltered. The above constraints require a high order of systems engineering to accommodate successfully the variety of interfaces and interactions between the retained elements of the system and the new components, and to accomplish this with a minimum of rework while assuming that performance and reliability have not been compromised.

Engineering Development Stage. The advanced development phase of the upgrade program, and most of the engineering design phase, is limited to the new components that are to be introduced. Here again, special effort must be directed toward interfacing the new components with the retained portions of the system.

The integration of the upgraded system faces difficulties well beyond those normally associated with the integration of a new system. This is caused by at least the following two factors.

First, the system being modified will likely have been subjected to numerous repair and maintenance actions over a period of years. During this time, changes may not always have been rigorously controlled and documented, as would have been the case if strict configuration management procedures had then been in force. Accordingly, over time, the deployed systems are likely to become increasingly different from each other. This situation is especially troublesome in the case of software changes, which themselves are often patched to repair coding errors. The above uncertainty in the detailed configuration of each fielded system requires extensive diagnostic testing and adaptation during the integration process.

Second, while vehicles and other portable systems are normally brought to a special integration facility for the installation of the upgrade components, many large land- and ship-based systems must be upgraded at their operating sites, thereby complicating the integration process. The upgrading of the navigation systems on a fleet of cargo vessels with new displays and added automation requires effecting these changes on board ship, using a combination of contractor field engineers and shipyard installation technicians. Installation and integration plans should provide special management oversight, extra support when needed, and generous scheduled time to ensure a successful completion of the task.

System Test and Evaluation. The level and scope of system test and evaluation required after a major system upgrade can range all the way from evaluating only the new capabilities provided by the upgrade to a repeat of the original system evaluation effort. The choice usually rests on the degree to which the modifications affect a distinct and limited part of the system capabilities that can be tested separately. Accordingly, when the upgrade alters the central functions of the system, it is customary to perform a comprehensive reevaluation of the total system.

Operations and Support. Major system upgrades always require correspondingly large changes in the logistics support system, especially in the inventory of spare parts. Operation training, with accompanying manuals and system documentation, must also be provided.

These phases require the same expert systems engineering guidance as did the development of the basic system. While the scope of the effort is less, the criticality of design decisions is no less important.

Software Upgrades

As described in Chapter 11, software is much easier to change than hardware. Such changes usually do not require an extensive system stand-down or special facilities.

With increasing system functionality being controlled by software, the pressure for software upgrades tends to make them considerably more frequent than major hardware upgrades.

However, to ensure that such operations are successful, special systems engineering and project management oversight is required to manage the difficulties inherent in system software changes:

1. It is essential that the proposed changes be thoroughly checked out at the developer's site before being installed into the operational software.
2. The changes must be entered into the configuration management database to document the changed system configuration.
3. An analysis must be performed to determine the degree of regression testing necessary to demonstrate the absence of unintended consequences.
4. Operation and maintenance documentation must be suitably updated.

The above actions are required for any system change but are often neglected for apparently small software changes. It must be remembered that in a complex system, no changes are "small."

Obsolescent legacy programs suffer from two disadvantages. First, the number of software support personnel willing to work on legacy software is diminishing and becoming inadequate. Second, modern high-performance digital processors do not have compilers that handle the legacy languages. On the other hand, the task of rewriting the programs in a modern language is comparable to the task of its original development and is generally prohibitively costly. This presents a difficult system problem for systems in the above position. Some programs have successfully used a software language translation to greatly reduce the cost of converting legacy programs to a modern language.

Preplanned Product Improvement (P³I)

For systems that are likely to require one or more major upgrades, a strategy referred to as P³I is often employed. This strategy calls for the definition during system development of a planned program of future upgrades that will incorporate a specified set of advanced features, thereby increasing system capabilities in particular ways.

The advantage of P³I is that changes are anticipated in advance so that, when needed, the planning is already in place; the design can accommodate the projected changes with minimum reconfiguration; and the upgrade process can proceed smoothly with minimum disruption to system operations. These preplanned changes will vary in magnitude and complexity depending on the need and availability of appropriate technology. Commercial airlines, for example, will often plan for a stretched version of an existing aircraft that will carry more passengers and incorporate larger engines and new control systems. By modifying an existing aircraft instead of developing a new one, the problems of government recertification can often be alleviated. In the military, the planned upgrade process has the advantage of prior mission justification. Since the

current system is operational and performing a needed function, the proposed system changes will not affect already approved mission and system objectives.

In the case of future improvements defined during initial system development, the contract for implementing them is usually awarded to the development contractor. This is the most straightforward contractual arrangement for carrying out a major system upgrade. It is also most likely to secure the services of engineers familiar with the current system characteristics to participate in the planning and execution of system changes. While even in this case the original development team may have largely dispersed, that part that remains provides a major advantage by its knowledge of the system. However, as can sometimes occur in government-sponsored programs, the pressure for competition can become especially severe and can even lead to the selection of a different contractor team for the upgrade contract. In such cases, an intensive education program will be required for the new team to learn the finer points of the system environment and detailed operation.

15.5   OPERATIONAL FACTORS IN SYSTEM DEVELOPMENT

In Chapter 14, Production, it was pointed out that systems engineers who guide the development of a new system must have significant first-hand knowledge of relevant production processes, limitations, and typical problems in order to coordinate the introduction of producibility considerations into the system design process. It is likewise important that systems engineers be knowledgeable about the system's operational functions and environment, including its interaction with the user(s), in order to be aware of how the system design can best meet the user's needs and accommodate the full range of conditions under which the system is to be used.

Unfortunately, the kinds of opportunities described in Chapter 14 that exist for systems engineers in a development organization to learn about manufacturing processes frequently are not available for learning about the system's operational environment. The latter is seldom accessible to development contractor personnel, except for those who provide technical support services, and these are more likely to be technicians or equipment specialists rather than systems engineers. Another inhibiting factor is that the operational environment is usually so system specific that acquaintance with the environment of an existing operational system does not necessarily provide insight into the conditions under which the particular system under development will operate.

The type of operational knowledge that systems engineers must acquire can be illustrated by the example of developing a new display for an air traffic control terminal. In this case, it is essential that the systems engineers have an intimate knowledge of how the controllers do their job, such as the data they need, its relative importance in sending messages to aircraft, the expected fluctuations in air traffic, the traffic conditions that are deemed critical, and a host of other data that impact the controller's functions. Engineers developing a control console for a civil air terminal can usually observe the operations at first hand and interview controllers and pilots.

However difficult, it is essential that engineers responsible for system design acquire a solid understanding of the conditions under which the system being developed

will operate. Without such knowledge, they cannot interpret the formal requirements that are provided to guide the development since these are seldom complete and fully representative of user needs. As a result, it is possible that deficiencies due to faulty operational interfaces will be discovered only during system operation, when they will be very costly or even impractical to remedy.

The term "operational environment" as used here includes not only the external physical conditions under which a system operates but also other factors such as the characteristics of all systems interfaces, procedures for achieving various levels of system operational readiness, factors affecting human–machine operations, maintenance and logistic issues, and so on. Figure 3.4 illustrates the complex environment in which a passenger airliner routinely operates.

Operational environments can vary radically depending on the type of system under consideration. For example, an information system (e.g., a telephone exchange or airline reservation system) operates in a controlled climate inside a building. In contrast, most military systems (airplanes, tanks, and ships) operate in harsh physical, electronic, and climatic conditions that can severely stress the systems they carry. Systems engineers must understand the key characteristics and effects of these environments, including how they are specified in the system requirements and measured during operations.

Sources of Operational Knowledge

A number of potential sources of operational knowledge may be available in certain situations. These include operational tests of similar systems, integration testing during system installation, system readiness tests, and maintenance operations. These activities all address the problems associated with successfully integrating the system's external interfaces with the site and with associated external systems. These can often expose serious problems that are not adequately revealed by the interface specifications provided to the developer.

To gain the necessary operational background, the systems engineer should endeavor to witness the operation of as many systems of the type under consideration as possible. Serving as an active participant in system test operations, or even by simply acting as an observer, is a good opportunity for learning. When present at such tests, the systems engineer should make the most of the opportunity by asking questions of system operators at appropriate times. Of special importance is information regarding what parts of the system are the sources of most problems and why. Learning about operational human–machine interfaces is particularly valuable because of the difficulty of realistically representing them during development.

System Readiness Tests. A useful source of operational knowledge is observing procedures used to determine the level of system readiness. All complex systems go through some form of checklist or fast test sequence prior to operation, often using automatic test equipment under operator control. A commercial airliner goes through an extensive checklist prior to each takeoff and a much more thorough series of checks prior to and during scheduled maintenance. It is instructive to observe how operators

react to fault indications, what remedial action is taken, what level of training these operators have been given, and what type of documentation has been provided.

Operating Modes. Most complex systems include a number of operating modes in order to respond effectively to differences in their environment or operating status. Some systems that must operate in a variety of external conditions, such as a military system, usually have several levels of operational readiness, for example, "threats possible," "threats likely," "full-scale hostilities," as well as periods of scheduled maintenance or standby. There may also be backup modes in case of degraded system operation or power failure. The systems engineer should observe the conditions under which each mode is induced and how the system responds to each mode change.

Assistance from Operational Personnel

In view of the limited opportunities for the developer's systems engineers to acquire an adequate level of operational expertise, it is often advisable to obtain the active participation of experienced operational personnel during system development. A particularly effective arrangement is when the user stations a team designated to be system operators at the development contractor's facility during the period of systems engineering, integration, and test. These individuals bring knowledge of the special circumstances of the system's interaction with the intended operational site, as well as represent the system operator's viewpoint.

Another source of operational expertise comes from system maintenance personnel who are experienced in the problems of servicing similar systems at their operating sites and in their logistics support. Systems engineers can gain considerable knowledge by well-planned interviews with such individuals. As noted earlier, complex systems often have maintenance support facilities that may be excellent sources of operational knowledge.

15.6   SUMMARY

Installing, Maintaining, and Upgrading the System

The application of systems engineering principles and expertise continue to be required throughout the operational life of the system. The operations and support phase includes installation and test, in-service support, and implementation of major system upgrades.

Interface integration and test can be challenging due to a mix of various organizational units, complex external interfaces, and incomplete or poorly defined interfaces.

Installation and Test

Installation and test problems can be difficult to solve because installation staff have a limited system knowledge. Systems engineers are seldom assigned until trouble is encountered. However, periodic operational readiness testing is necessary for systems

that do not operate continuously. This can help minimize unexpected system problems.

Where nondisruptive installation is required, care to plan the installation procedures, via a hybrid simulation or a duplicate system operating in parallel, is absolutely essential.

In-Service Support

System software must be subject to strict configuration control to prevent serious deterioration of software quality. In this vein, built-in fault indicators are very valuable for detecting internal faults, although they sometimes produce false alarms. Therefore, field engineers should be knowledgeable about built-in test devices.

Remedial actions to correct operational problems are difficult to implement: operational personnel are not technical specialists. Furthermore, troubleshooting tools are limited. And materials and processes involved in logistics support themselves constitute a complex system.

Major System Upgrades: Modernization

Logistics cost is a large part of system cost. Therefore, P^{3}I facilitates improvement of systems during major upgrades. Advanced features are defined during system development, and advanced planning permits minimum disruption to system operation.

Operational Factors in System Development

Possible sources of operational knowledge include operational and installation tests—by observing system operations within its environment. Of course, assistance from operational and maintenance personnel is invaluable.

PROBLEMS

15.1 Identify and discuss four potential problems associated with the installation and test of a complex navigation and communication system aboard a transoceanic cargo vessel. Assume that some of the subsystems have been integrated at land sites prior to shipment. Assume that a number of contractors are involved, as well as the shipping company and government inspectors.

15.2 Interface problems are usually difficult to diagnose and to correct during final system integration. Why is this so? What measures should be taken to minimize the impact of such problems?

15.3 Operational readiness testing is an important function for deployed systems. As a systems engineer who is familiar with the design and operation of a large complex system, describe how you would advise operational personnel to define and conduct this type of testing.

15.4 Many complex systems incorporate a built-in fault indicator subsystem. This subsystem can itself be complex, costly, and require specialized training and maintenance. List and discuss the key requirements and issues that must be considered in the overall design of a built-in test subsystem. What are the principal trade-offs that must be addressed?

15.5 An effective logistics support system is an essential part of successful system operational performance. While the support system is "outside" of the delivered system, discuss why the systems engineer should be involved in the design and definition of the support system. Discuss the functions of some of the characteristics that must be considered, such as the supply chain, spare parts, replaceable part level, training, and documentation.

15.6 Discuss the types of systems that are best suited for applying P^3I during the design phase. Describe the key elements in justifying the additional cost of a P^3I approach.

15.7 In maintaining an operational system, hardware faults are usually corrected by replacing the offending subcomponent by a spare. Software faults are typically coding errors and must be eliminated by correcting the code. In complex systems, software changes must be made with extreme care and must be validated. Discuss ways in which software faults can be handled in a controlled manner where the operating system is remote from the development organization.

FURTHER READING

B. Blanchard and W. Fabrycky. *System Engineering and Analysis*, Fourth Edition. Prentice Hall, 2006, Chapter 15.

Performance Based Logistics: A Program Manager's Product Support Guide. Defense Acquisition University (DAU Press), 2005.

N. B. Reilly. *Successful Systems for Engineers and Managers*. Van Nostrand Reinhold, 1993, Chapter 11.

Systems Engineering Fundamentals. Defense Acquisition University (DAU Press), 2001, Chapter 8.