

Part III.

MANAGEMENT AND OVERALL APPROACH

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Part III.

MANAGEMENT AND OVERALL APPROACH

To summarize the lessons learned from ABM R&D is challenging, because the period of time was long, the breadth of research wide, and the follow-on development program immense. This part of the report discusses the most important lessons learned about the management and overall approaches used in a project such as this. The discussion is discursive and makes no attempt to present detailed supporting data or arguments. Every member of Bell Laboratories Divisions 65 and 66 management contributed to this report by suggesting items, writing portions, and/or criticizing early drafts. There is a reasonable consensus on the views expressed here, but probably not on their relative importance.

Because this was a very big project, particularly the SAFEGUARD portion, the lessons are most directly applicable to other very large projects and many will probably also apply to smaller programs. There has been no effort to analyze the effect of project size or to modify the conclusions for different conditions.

Under the prime contract with Western Electric, Bell Laboratories had overall responsibility for essentially all R&D work. Since Bell Laboratories has its own special organizational characteristics, many of the statements made here have probably been affected by those characteristics. There has been no attempt to

determine just how these characteristics have affected the conclusions or how the conclusions might be modified by different organizations. However, the close relationship between Bell Laboratories and Western Electric was exploited continuously during development and production.

OVERVIEW

Feasibility

Perhaps the most important lesson of this experience, and particularly of the SAFEGUARD part, is that development of a large weapons system can be completed on schedule to prescribed performance specifications, with effectively controlled costs. It is true that changes during development drastically cut down the overall deployment. Also, the overall system design was modified because of changes in objectives and because of test results obtained during development. However, the development, including integration of the first installed site, proceeded on schedule and the system met the prescribed performance specifications. Cost performance is harder to determine because of inflation during the period, deployment changes, and lack of a design-to-cost objective. But it seems clear that costs were controlled effectively.

Given the experience with some other important weapons system developments (most of them smaller or less complex than this one) in failing to meet schedules, performance specifications, or cost objectives, the ABM R&D effort is valuable as an example that such developments can be successful. This is particularly important because one major argument against developing SAFEGUARD was that it could not be done successfully.

Overall Approach

One principle followed during the entire research and development period was to continuously evolve the system concept; that is, to repeatedly synthesize and analyze systems that met prescribed objectives against prescribed threats. Although most of the effort was spent on research and development of specific subsystems (radars, missiles, data processors, etc.), there were continuous studies of how subsystems could be used within a system. The sensitivity of system performance to subsystem specifications and to changes in objectives and threat was repeatedly studied. Designers concentrated on aspects of subsystem performance most important to overall system performance.

The overall approach might be summarized as plan, simulate, test — plan, simulate, test — plan, simulate, test. Every part of the development benefited from detailed planning. While there was little or no attempt to specify a planning format or technique, every subproject manager was strongly encouraged to plan in detail the design, implementation, and testing of his piece of the system. Although these plans had to be revised many times, it was very beneficial to have them. To keep track of progress, several methods of reporting were tried, as discussed later.

Simulations were used on all levels of detail and for all parts of the design. In every case, simulations more than paid for themselves in either improved performance or reduced cost.

When trouble was found, it was also discovered that more or better simulation would have prevented, or at least reduced, the difficulties. Because the simulation results were important, there were extensive efforts to validate them with data from real tests.

Perhaps the most important part of the overall approach was testing. Effective testing depends strongly on detailed test specifications. Most of the development time and cost went into testing, and it always happened that even more tests were possible and, in fact, desirable. Consequently, test planning almost always consisted of selecting only the most essential tests from among those desirable. After each test, data analysis continued until any failures or anomalies were understood. These comprehensive analyses, particularly of the Meck Island tests at Kwajalein Atoll, frequently resulted in significant system improvements.

Constructing the prototype Missile Site Radar (MSR) at the Kwajalein test site made way for the Kwajalein field experiments, which were undoubtedly the single most important step in achieving success. The project would have been even more successful if a more complete system prototype had been built. Building a prototype Perimeter Acquisition Radar (PAR) at a northern site early in the R&D program would have definitely produced a better system design. There were major problems with developing the PAR software, mostly because no complete, consistent set of requirements could be checked out on a prototype PAR system. Without a prototype, the problems with requirements were not identified clearly by the developers until the final development phase. No large project should be undertaken without allowing time and resources for constructing and extensively testing a complete prototype.

The Kwajalein tests are perhaps better termed "field experiments," and they were almost unbelievably successful. Successful tests meant not just meeting normal test objectives; they

produced data invaluable for completing system design. For instance, the data obtained on tank breakup were vital to the success of the project. Experiments were carried out in a carefully controlled environment (which helped both the conduct and analyses of tests) rather than in the design threat environment only. It would have been impossible to obtain data for the final design if the tests had only been trials against the design threat.

In many other projects, separate organizations within a company carry out their own field operations. A principal reason for the successful ABM field test program was the rotational assignment of systems, hardware, and software designers to the field sites. The reward was not only the transfer of design intent to the field tests but also the experience fed back to the design work when the designers returned to their laboratories.

Probably second in importance only to the Kwajalein tests were the tests at the Tactical Software Control Site (TSCS) in Madison, N. J. Not only were these tests vital to software development, they were also an important source of data on the hardware and helped validate the system simulations. Later sections of this discussion expand on the importance of testing and the TSCS.

The one constant factor throughout the project was the inevitability of change. At no time were the objectives, threat, design, or deployment completely stable, and it was unrealistic to assume that any of them would remain fixed. Therefore, it was vital to be prepared for changes and to have effective procedures for controlling them. Making changes in hardware was a familiar problem from Bell Laboratories' experience on Bell System and other military projects, and procedures for handling them were reasonably well established. But the complexity and high performance of the PAR and MSR systems necessarily produced a greater number of changes during

testing and checkout than ever experienced before with smaller systems of lesser performance.

By careful planning, many changes were introduced into major subsystems — the MSR at Meck, the TSCS at Madison, the tactical MSR in North Dakota, and the PAR in North Dakota — during tests without materially affecting schedules. By scheduling test time among the many equipment users, changes could be introduced if the necessary drafting, shop, and quality assurance teams were available when required. A mutual understanding between designers of hardware and designers of software testing routines, plus careful scheduling, normally allowed enough time to physically disrupt hardware to introduce a change. Changes necessarily must be designed in small packages, but large packages of changes could be introduced because they were scheduled for periods when the system was down for a day or more and because workcrew schedules were kept flexible. After the changes, enough time was allowed for regression testing of areas affected by the changes.

By keeping the main technology fixed during the program, it was possible to produce, on schedule, equipment with proven reliable designs. On many other projects, hardware designs are continually changed to keep abreast of changing technology, schedules continually slip, and hardware is unreliable.

While controlling hardware design changes was a big problem, controlling system and software changes was even more difficult because procedures were not as well established. It took a great deal of effort to develop techniques for controlling such changes. These techniques, plus strict discipline, resulted in a control procedure without which the software development would have failed. (See Chapter 4 of Part II for a discussion of lessons learned in controlling software changes.)

Overall system requirements were specified early in the development, and formal procedures for controlling changes in these requirements

were immediately instituted. Software design was tied closely to these controlled system requirements. Even though there were many, many changes, this procedure was crucial to successful software development.

Organization

One of the most important characteristics of this project was the interface between Bell Laboratories and the Army agency responsible for ABM development. Relationships were open, frank, and informal. Although formal contractual requirements were carefully negotiated and carried out, most of the information exchanges between the Army and the contractor were informal. Consequently, problems could be discussed as they came up and proper action taken immediately. Also, the significance of problems could be determined and the right priorities established. Little effort was wasted on insignificant problems.

It was very important that the Army side of the interface be a single, responsible Program Manager. As with any very large project, there were many interfaces at various levels with the Army's SAFEGUARD organization both in Washington and in Huntsville, Alabama. In addition, liaison was necessary with many Army organizations; e.g., nuclear effects, facilities requirements, communications. Despite the many interfaces, the SAFEGUARD System Office and the SAFEGUARD System Command became an effective single agency responsible for resolving differences and adjudicating problems whenever they occurred.

Overall system changes were negotiated between the System Requirements Department at Bell Laboratories and the corresponding organization in the SAFEGUARD Project Office at Huntsville. Questions about system performance were discussed in proper context, and appropriate formal action taken to change the system requirements. Subsystem development was thereby protected from many "what if" studies which could have delayed progress.

This project, like others its size, took several companies to carry out. However, it was made clear that Bell Laboratories had overall responsibility, and top managers in Bell Laboratories used their authority to make major decisions quickly. If responsibility had been split among several companies, or if other red tape had impeded Bell Laboratories managers, it is improbable that schedules could have been met and very probable that costs would have increased. Because Bell Laboratories could make decisions quickly, technical authority over subcontractors could be delegated to lower levels when cost was not involved. This speeded up decision-making; it also increased motivation and inbred a sense of responsibility in the people working on the project.

Interfaces between Bell Laboratories and the subcontractors received special attention. In very large and technically complex development projects, the importance of this cannot be overemphasized. Technical developments and problems in each area affect other areas. Whether the related developmental areas are within a single company or split among contractor and subcontractor, almost continuous interchange of information is essential. Monthly or quarterly technical reviews are inadequate unless informal exchanges on the engineering level go on, by telephone and personal visits, at least once every week or two. With major subcontractors, it is often necessary to keep an engineering group at the subcontractor's plant. Without such frequent and informal engineering level exchanges, the technical development would not have been as quick or successful as it was.

In hardware subcontracting, enough time should be spent, before development starts, to work out a detailed design specification that both contractor and subcontractor fully understand. Requirements will change, so an early base for defining them is very important. This is particularly true for a cost/schedule/performance incentive contract.

Attention to contractor/subcontractor relationships was particularly important with software development. Because it is impossible to state software requirements precisely until after system design is completed, it is very difficult to employ subcontractors to develop complex software. If subcontractors are used, the interface must be carefully specified and monitored. The large subcontractor effort used to develop SAFEGUARD software was divided into smaller, well-defined tasks that interfaced directly with the Bell Laboratories first-line supervision. Tasks were rated monthly by the technical supervisor, and the rating results determined the final fee on the Cost-Plus-Award-Fee (CPAF) contract. This interface ensured responsive subcontractor performance at a manageable level on a very large project. These methods worked well, and when problems developed it was usually where the customary interface procedures had not been used.

Perhaps the most important principle followed in organizing the project was to install a top-quality manager over each subproject and give him full responsibility for schedule, cost, and performance. The managers had broad and deep technical competence as well as managerial competence. Furthermore, they were expected to be aggressive in not just passively accepting a job definition and carrying it out. Instead, they actively explored interfaces between their subprojects and all others to anticipate problems. Although they used a variety of techniques to carry out their responsibilities, time after time their ability to recognize technical problems was fundamental in finding solutions to them. While it was certainly important to appoint the best general manager, it was absolutely vital that he have a high technical competence as well.

SPECIFIC SUGGESTIONS

This section contains comments less basic, but more specific, than those in the preceding sections. While the suggestions are grouped

into general areas, each is largely independent of the others.

Requirements

System

To design and develop a large, complex system in a few years, a large number of people is obviously required (several thousand at the peak for SAFEGUARD). To properly coordinate their efforts, a good set of requirements and/or specifications is essential. For SAFEGUARD, the major specifications were:

- Brief overall SAFEGUARD System specifications
- Missile Site Equipment Subsystem specification
- Perimeter Acquisition Site Equipment Subsystem specification
- SPARTAN Subsystem specification
- SPRINT Subsystem specification
- Communication Equipment Subsystem specification
- Ballistic Missile Defense Center specification
- Data Processing System Performance Requirements (DPSPR).

The DPSPRs controlled software implementation and actually specified the detailed system requirements. The following comments are based on experience with all of the specifications, but mostly with the DPSPRs.

A decision about the level of detail and the completeness of requirements must be carefully weighed. The decision depends on the relationship between the organization preparing the requirements and the organization that implements them, the resources available, the time available, and various psychological factors. Obviously, the closer the organizational relationship, the less need for completeness. On the other hand, if the organizations belong to different companies, requirements must be complete and detailed. It is probably better to err on the side of too much detail rather than too little. On SAFEGUARD, too few resources (about 7 percent) were allocated to preparing the requirements.

Designers naturally object to overly detailed requirements because it limits them, and this important, legitimate criticism should be carefully guarded against. However, problems resulting from lack of detail can be even more serious, and therefore must be guarded against even more carefully.

It is vital that requirements be available as early in the program as possible. Those preparing requirements must make decisions quickly, on the basis of incomplete analyses, and publish them in the shortest possible time. Management should soften criticism of defects in the early issues of the requirements, because the defects inevitably result from pressure for early availability.

Because objectives change, because problems become better understood, and because requirements must be written quickly, changes are inevitable. It is absolutely necessary to institute a way to control changes relatively early in the process. This change control must keep proposed changes from getting lost and ensure that every designer is working with a consistent and updated set of requirements.

Requirements serve two purposes. They tell the customer what he is buying and the designer what he is designing. If these purposes conflict, the priority is to give the designer clear and unambiguous requirements. If necessary, give the customer explanatory material.

To summarize:

- Publish the first issue of requirements as early and as completely as possible. The first issue then forms a baseline for changes.
- Controlling changes in requirements is essential. The control system should become formal before the project is half completed.
- Carefully plan the degree of detail and completeness in the requirements. Do not allow unanalyzed resource restrictions to determine these answers.

Availability/Reliability

In a system like SAFEGUARD, a complete and easily understood availability and reliability budget is essential for each subsystem. This budget reflects the system effectiveness requirements. In the real world of fixed resources and flexible system objectives, these requirements should be considered as objectives. As such, they should be reviewed carefully at intervals to ensure that the system stays in balance and that the cost of achieving the original objectives is clearly understood. In SAFEGUARD, for example, if system effectiveness is measured by the number of Minuteman ICBMs that survive an attack, a wide range in availability and reliability will produce acceptable results. The original objectives were set for the SENTINEL Area Defense System, which required more stringent requirements than the Minuteman defense. More thorough consideration of possible reductions in objectives might have lowered some requirements for components and subsystems.

Man-Machine Subsystem

The requirements originally specified for the CRT console, the lightpen, the teletypewriter, and other units in the Command and Control Display System (CCDS) were far more extensive than necessary to support the man-machine relationship. Furthermore, it was found later that to supply many of these special capabilities required a software effort greater than the project could support. The man-machine interface was therefore greatly modified, resulting in two out of three available lightpen modes not being used. In addition, many CRT console capabilities, such as the A-scope mode and lower case letters, are unused throughout the CCDS. At the beginning, a hard-nosed and realistic examination of what man really had to do to control an ABM system, plus the assurance that requirements were compatible with the capability of supporting software, would have drastically reduced requirements on the Command and Control System. The equipment would have been more economical to design, produce, and maintain.

Similarly, the user directed that many engagement control actions be included in the requirements because they were useful in anti-aircraft systems, even though they were ineffective for ABM systems. These actions included Hold Fire, Force Intercept, and Command Destruct. If the requirements for ABM control actions had been agreed on sooner, the costs of designing, developing, and integrating display and control software would have been lower. While these problems did not significantly reduce capability of the man-machine subsystem, they definitely reduced the quality and "elegance" of the design.

Nuclear Surety and Safety

There should have been earlier, more effective communication among design organizations and the government agencies responsible for nuclear surety and safety: the National Security Agency (NSA) and the Nuclear Weapons System Safety Committee (NWSSC). Prompt liaison could have avoided misunderstandings, mistakes, and costly last-minute changes to the Nuclear Enable Authority and Launch Enable systems.

The agencies began intensive final reviews after designs, which incorporated changes suggested during their preliminary reviews, had been released for manufacture. Under these circumstances, designers were understandably reluctant to accept the agencies' comments and suggestions. After an additional year of review, NWSSC suggestions became a mandatory requirement for a new Launch Enable design concept. The designers regarded the new design and other practices imposed by the agencies as "ex post facto" requirements. Much cost, delay, and contention could have been avoided if:

- All the organizations involved had better understood the roles of NSA and NWSSC and the strength of their requirements and recommendations
- Before design work started, the agencies had documented their requirements, preferred concepts, and standards developed and imposed on other projects (e.g., use

of nonsymmetrical connectors, omission of continuity loops, etc.)

- The above had been used as the basis for agency review with minimum application of criteria developed during review from "state of mind" guidelines
- The reviews had been more in parallel with design, to supply incremental concurrences from design through field test.

Intelligence Data

Gathering intelligence through U. S. observations of Soviet ICBM tests should have been much more responsive to the ABM development. The sensors on the U. S. instrumentation ships should have been modified so they could collect the required data. Instead, large sums were used for devising questionable frequency and radar cross-section scaling relationships, for extensive target modeling, and for costly simulations which could not always be validated.

Vast quantities of intelligence data were collected. There was enough information on the composition and levels of Soviet forces, and an adequate design threat was eventually defined. However, data essential for design, development, and evaluation of critical target selection and tracking algorithms were lacking. They could have been obtained by observing Soviet operational tests. During the years 1967 through 1972, data requirements were sent repeatedly to the Research, Development, Test and Evaluation (RDT&E) Directorate at Huntsville, who in turn, forwarded the requests to the intelligence agencies. With the advent of SENTINEL in 1967, UHF and S-band data were badly needed, yet UHF data remained scarce and incomplete and S-band data never were collected.

Radar cross-section and trajectory data for supporting the ABM development were needed at the operating frequencies, polarizations, and resolutions of the MSR and PAR. There should have been a deliberate effort to collect data on RV wake characteristics at S-band, on tank breakup at S-band, and on all exoatmospheric objects at UHF. Wake and tank breakup data were collected later by the Meck MSR during

live U. S. tests at Kwajalein, but these data were not totally adequate in the exoatmospheric regime.

Hardware Design Implementation

Hardware Design Release

When SAFEGUARD was committed to production, schedules were quickly established for releasing hardware designs for manufacture. These designs were developed during the R&D program, and had to be baselined and transferred from the R&D contract to the production contract. Hardware could then be manufactured according to production schedules, but subsequent design changes had to be made under formal production control procedures. These procedures proved to be cumbersome in making changes to R&D designs not yet stabilized for the production system. It would have been more cost effective to start manufacture with a less rigid configuration control program, and to control changes under somewhat less stringent procedures. Then, after a transition period to allow for design stabilization, formal production procedures could have been introduced.

A related problem was the Critical Design Review (CDR) required before the design could be released to production. The CDR was intended to allow the Army and Bell Laboratories to assure themselves that the design met system requirements and was ready for release to production. Since the equipment components had different lead times and were ready for release at different times, CDRs were held at various levels, on hardware ranging from magnetic apparatus to power supplies to power racks to subsystems. The CDRs should have been held only for large, very important items such as the MSR receiver, while lesser items should have been released prior to a CDR if required by production schedules. Army engineers could have been kept informed of the design status through the same type of regular contact with the design engineers that existed in the development program.

Design Drawing Reviews

In normal practice, the drafting organization and the responsible engineer spot-check completed drawings for accuracy and then release them for manufacture. If errors are detected later, the responsible engineer is notified and a change order for the correction is prepared.

In SAFEGUARD, an outside contractor reviewed selected drawings submitted for release to manufacture. The intent was to avoid subsequent change orders by assuring that the drawings contained no errors. In the early submittals, which came from reputable subcontractors, many errors were detected. More often than not, the errors did not affect manufacturing the product. Instead, they involved incomplete compliance with various drawing practices and format requirements.

The result was numerous drawing rejections and led to much effort being expended on additional checks of drawings before they were released. Subcontractors completely checked all their drawings before releasing them. Then a prime contractor representative visited the subcontractor plant and conducted a further review. In spite of these repetitive reviews, drawing changes were still required when shops used the drawings for procurement and manufacture, because the shops found real problems associated with product manufacture. The expensive, time-consuming drawing quality audit, more often than not, uncovered only superficial problems.

Drawing Designation

An excessive number of drawings were generated because of the initial requirement that each separate detail be on a different 11-million-numbered drawing. This volume of drawings is not required in commercial manufacture, nor is it used in the Bell System. It is expensive both in dollars and in the time required for drafting.

After much discussion of the cost of the program, the contracting agency gave some relief

by allowing contractors to include families of similar parts on a single drawing, with designation of a particular item by dash number. Certain drawing details were also permitted on top assembly drawings only when the detail did not have a separate requirement, e.g., as a spare part. If this relief had been extended to all subassemblies and been present from the start, considerable expense could have been avoided.

Organization of Work

System Responsibility

Major responsibility for SAFEGUARD was concentrated in two Bell Laboratories groups — the SAFEGUARD System Design Department and the SAFEGUARD System Evaluation Department. The System Design Department generated requirements for the various subsystems based on the overall design. Implementing these requirements was delegated to various project managers. The System Evaluation Department concentrated, at least initially, on evaluating the design, i.e., the requirements, rather than the implemented system. Neither of these departments was responsible for coordinating all implementation activities and no other organization was assigned that responsibility. Therefore, coordination had to be supplied by the individual project managers and higher levels of management.

The results were incompatibilities in system interfaces; significant delays in correcting system requirements; delayed awareness of difficulties; and lack of adequate system evaluation of process design, program execution times, missile and radar interfaces, and the manual display and control subsystem. All of these areas were corrected by individual project managers but only at the cost of added effort and delay.

In future projects, a system coordination group should be established within the systems area. The group should be responsible for (1) coordinating all implementation, (2) interface

compatibility, (3) coordinating system evaluation, (4) establishing and achieving project schedules, and (5) technical reviews of system design, system evaluation, implementation, and test programs. In addition, the group should ensure that lessons and techniques learned from the prototype system are transferred to the tactical design. The group should act as a technical contributor to all of the above activities, not just as an administrative staff responsible only for standards, schedules, and reporting.

System Maintenance Organization

Most SAFEGUARD designers recognized the importance of maintenance, and considerable effort was devoted to maintenance by individual design groups. However, there was no centralized guidance for maintenance design, and results were less than optimum. For example, coordination of status reports and error responses from the various subsystems could have been improved, and documentation and testing of the overall maintenance system could have been better.

To get a better maintenance design, an organization with the following charter should have been established:

- Develop and maintain system and subsystem maintenance requirements consistent with the system availability goals
- Guide selection of maintenance techniques and tradeoffs in each hardware and software design area
- Guide development of operations and maintenance procedures and the associated documentation
- Develop plans for system-level tests of maintenance tools (especially for real-time operations).

Like any other element in a system, maintenance requirements will continually evolve as system design evolves. These requirements and their implementation are highly dependent on system Availability/Reliability goals (which themselves may change with time). Thus, it is important to keep system maintenance requirements up to date and appropriate to the system's primary mission.

Unified Responsibility for Design and Development

The success of a complex system, including its ability to operate on demand, depends on the availability and reliability of every essential element in it. Thus, "system" cannot mean portions of an overall complex, partitioned to fall within predetermined administrative jurisdictions; no element is separable if its failure would prevent the system from fulfilling its operational task. In SAFEGUARD, the critical complex includes support facilities, such as power, environmental cooling, and communications, as well as the technical equipment. A diesel engine, a cooling system fan, or a data transmission circuit may be just as crucial to battle readiness as a data processor unit or a missile guidance set.

An important reason for the success of the SAFEGUARD project was that responsibility for the design and development of essentially the entire system was concentrated in one organization. The only exceptions, the Tactical Support Equipment (TSE) and the communication system, caused significant problems. Also, there were difficulties because responsibility for training, supply, and maintenance was separated from the development responsibility.

Because the communication system's technical direction and funding were under the SAFEGUARD Communications Agency (SAFCA), not the SAFEGUARD System Command (SAFSCOM), issues which could not be settled mutually had to be resolved by the SAFEGUARD System Manager. As a result, some problems were ignored and others took more time and effort to resolve than necessary. There was also a tendency to insulate communications suppliers from Bell Laboratories designers, probably to minimize the number of design changes. But this impeded liaison on the working level, leading to poor understanding of how the communications interfaces worked and ultimately to interface problems. To try to coordinate communications system design, frequent meetings were held (typically monthly).

These meetings used up time and manpower and required attendance by many agencies and companies, but they were no substitute for direct working-level contact among designers.

With the TSE, the prototype test site's outstanding success in meeting objectives would have been impossible without unified authority for operating and maintaining both technical and support facilities. Many problems surmounted only with difficulty (e.g., the problems with the Meck power plant) could have been lessened or eliminated if such authority had started earlier during design, procurement, and installation. At the tactical site, placing more of the administrative responsibility for TSE with the Weapon System contractor would have allowed more efficient operation with fewer problems and delays. During a demanding test and installation program, it is impossible to attain either adequate speed of response or a common view of priorities with divided authority. Hence, unified responsibility for the total system, including both technical and support facilities, is extremely important.

Management

Subcontractor Arrangements

The size of the development required that design work be assigned to subcontractors and responsibilities be shared with government agencies. The production phases required decisions as to which potential suppliers should receive contracts. The success of the project attests that almost all of the arrangements with subcontractors and agencies were quite successful; however, here too, there were also some difficulties.

Arrangements involved a wide range of subcontractors and items. For example, Martin Marietta for development and manufacture of the SPRINT missile, IBM for software development, a manufacturer other than the design agency for production of missile motor cases, the Atomic Energy Commission (AEC) and

its subcontractors for design and manufacture of the warheads.

Subcontracting work successfully requires attention to things which are, for the most part, obvious. However, these matters are so vital that elaboration is in order.

Consider first the design responsibility that the prime contractor gives to a subcontractor. Requirements must be consistent, controlled by the prime contractor, and understood well by each party. Note that requirements need not be perfect and that during the development they will change.

Ways to evaluate progress should be carefully determined before the subcontract goes into effect and then followed tenaciously so that real status is known by the technical project management on each side of the interface. Because requirements do change, progress will necessarily be assessed against a moving base.

Contracts and other formal arrangements must not impede the flow of information or inhibit changes. They should encourage progress, not stifle it.

The method for subcontracting the software development was somewhat unique. Work was broken into a collection of well-defined relatively small tasks that could be handled between prime and subcontractor at a first-line supervisory level. Progress was rated monthly by the prime contractor's supervisors, and the rating determined the award to the subcontractor on the cost-plus-award-fee contract.

Extreme care should be taken when it seems desirable to award manufacturing contracts directly from the government to other than the subcontractor who did the original design. If this is done with items that are intricate or use new technology, the expertise of the original designer is lost. Also, the expertise of the prime contractor in making system tradeoffs with respect to the item is lost. This kind of "breakout" should be used only where items are quite stable.

When an item involves multiple government and contractor agencies, such as development of a missile warhead, it is obvious that the responsibility and requirements assigned to each agency must be considered with care for all phases of the program. With warhead development, the designs and interfaces could have been simplified if complete responsibility, including the adaption kit, had been delegated to a single agency. Again, communication, flexibility, and constant monitoring are required if development is to be effective.

In summary, on a big project, assigning work to subcontractors is absolutely necessary, can be handled successfully, and requires constant management attention.

Cost-Plus-Award-Fee Contracting for Software Development

On the SAFEGUARD project, a large part of the software development had to be subcontracted. This posed a difficult problem, because the software had to be of high quality and be delivered on time, in spite of many requirement changes and complex interfaces. With all these constraints, development problems had to be sensed rapidly and accurately, and acted on promptly and effectively, or else development would get out of control. The key need was for close and continued attention to the subcontracts.

To attract and hold the attention of subcontractors, the profit from the job was determined by job performance. In principle, the Cost-Plus-Incentive Fee (CPIF) contract does this by adding a fee determined by applying a previously agreed-upon formula to objective measurements of cost and/or performance and schedule events, on completion of the work. Incentive fee contracts were used effectively on many parts of the SAFEGUARD development. However, CPIF contracts are difficult to use for developing complex software, because it is impracticable to set up predetermined performance goals to measure against the final product.

The Cost-Plus-Award Fee (CPAF) contract overcomes this handicap by providing that the fee be determined in real time and based on the subcontractor's performance as judged subjectively by the customer. The CPAF contract is a cost-reimbursable, level-of-effort arrangement in which the fee to be paid for each (pre-determined) period is based on the customer's unilateral judgment of the supplier's performance, measured against previously agreed-upon subjective performance criteria. The fee is not subject to change. This type of contract had been used to some limited extent by NASA and the Navy, but was not in general use.

The SAFEGUARD CPAF subcontract for software development was in operation for more than four years, involved up to about 800 people, and as of October 1974, covered over \$130 million. It has been a good vehicle for dealing with a large, complex, dynamic problem, where the customer needs as good a job as he can get, and on time. This type of contract requires good faith between customer and supplier, and substantial monitoring and evaluation. The format encourages good customer-supplier communications and the management involvement that is necessary for successful performance. The improved visibility of problems makes it possible to solve them quickly.

Incentive Contracts

Traditionally, during large, long-term development projects, principal events are used to demonstrate the achievement of significant milestones. Most military contracts now include incentives — contractually established monetary awards (positive or negative) — for achieving certain events. Many such incentive events were scheduled and successfully achieved during the SAFEGUARD development. The event system was valuable in demonstrating that performance requirements were being met and in encouraging project-wide planning. Furthermore, the events were interim goals, each clearly supporting and contributing to ultimate

success, so that morale was boosted by a sense of achievement when goals were reached. The monetary awards associated with incentives can provide these same important advantages. However, requiring the completion of certain events in the contract ensures that they are scheduled, while the monetary awards ensure proper management attention.

Incentive contracting has potential disadvantages as well as advantages. Based on the SAFEGUARD experience, the following cautions and recommendations are given.

- Incentive contracting of performance, schedules, and costs is beneficial only if firm performance requirements, schedules, and costs can be negotiated. The effectiveness of incentives in state-of-the-art development projects is severely limited, because milestones cannot be easily identified and costs are difficult to negotiate. Incentive contracting loses many of its advantages if the contract has to be reopened because of major changes in requirements and schedules. If this happens, a contractor fully aware of his problems will make every effort to "get well" as part of the change negotiations.
- Incentive events must be significant, well-defined, and carefully chosen in-line milestones that must be reached to complete the project. They should be major performance or schedule achievements that require the success of many lesser milestones. Consequently, they should not be ends in themselves, and achieving them should not divert a substantial portion of the project's resources. Furthermore, if conditions change and an incentive event no longer serves its original purpose, the military and contractor should immediately negotiate to remove or modify it. To help avoid excessive diversion of resources, limit the time allowed for demonstrating an event.
- Establish incentive monetary awards only for results that are useful to the project. For example, an award should not be given for delivering a unit before it can be used. Similarly, an award should not be established for greater power output than other elements can handle or the system can use. Sound technical and business management skills are needed to work out the most challenging incentive events for a contractor and to assure overall success.
- Schedule incentive events infrequently to keep them significant and to allow preparations to be integrated with normal work.

For projects several years long, schedule incentive events no more often than every three to six months for each major subsystem. Events should be scheduled approximately eight to twelve months ahead of their demonstration date.

- Well in advance of performance incentive events, agree upon a reasonable time period or "window" for making the performance measurement. Failure to reach the event within that time period should result in a maximum penalty.
- Well in advance of a specific incentive performance, prepare a "catalog" to cover how the test is to be carried out, what instrumentation is required, what constitutes a success or failure, and how to score a failure from causes outside the contractor's responsibility. Murphy's Law states that if there is a chance for misunderstanding, it will occur. Therefore, careful attention should be given to preparing the catalog to make sure all factors are considered.

Reporting Systems

In large programs such as SAFEGUARD, reporting of technical performance, schedules, and costs is required for internal control and for informing the customer of program status. While these program aspects are interdependent, they are handled by separate reports and will be discussed separately. Each reporting system evolved over the years to meet changing program needs.

The goal of the reporting system is to convey the right level of timely information while conserving the effort required to prepare the report. Early in the program, technical progress reports were submitted every six months. They were difficult to prepare and, because of the interval covered, contained much outdated information which, while of historical interest, was not of immediate concern to program managers. Since then, and throughout most of the SAFEGUARD program, Bell Laboratories prepared monthly technical progress reports for the Army, submitting them by the 20th of the following month.

These reports had enough detail to present the current technical status for program management and to serve as a permanent record of job progress. Additional details were transmitted less formally by individuals, at meetings,

and via memoranda. The monthly technical report also gave Bell Laboratories managers reasonably detailed information, not only about their own activities, but also about other phases of the program — a necessary part of the program coordination.

Several weekly TWX reports were sent between different project locations. These were useful because they were timely and concise.

For reporting schedules, Bell Laboratories evolved the SAFEGUARD Program Schedule Charts, which showed program milestone events. They were submitted to the Army at monthly intervals and were also distributed within Bell Laboratories. These significant benchmarks, plotted against a calendar grid, supplied the information needed to assess program status and to coordinate various program activities with other Army units, such as the Corps of Engineers.

These schedule charts tracked a limited number of easily recognizable milestones. Each represented substantial program effort and was essential to program completion. The charts were backed up by detailed schedules kept by each manager of a major subsystem. If the dissemination of the schedule charts raised questions, the manager could supply the answers from his more detailed schedules. The schedule charts retained all milestone dates once they were established. Where changes were required and new targets agreed on, the old dates were retained so that the slippages were easily visible. The dates when events were accomplished, either early or late, were also shown.

The level of detail in these SAFEGUARD Program Schedule Charts proved to be entirely adequate for the overall program management. They were much preferred to schedule systems that showed thousands of events, each involving a small amount of effort covering a short time interval. Had such a system been used, administrative costs alone would have been excessive.

Cost reporting used a system that summarized detailed cost information to facilitate management review and decisions. The reporting structure showed costs for each subsystem with a further breakdown to some half dozen separate accounts. In each account, costs were shown in four categories: Engineering, Drafting, Subcontracts, and Other Direct Charges. At the beginning of each year, a spending plan was submitted with this same structure. The Financial Management Report compared actual monthly costs against the yearly plan. When necessary, the responsible technical manager submitted a revised estimate of the costs for each remaining month of the year, with a concise explanation of the reasons for the revision.

Although every effort was made to see that the Financial Management Report clearly set forth program status, an analysis of this report, together with any recommendations for redistributing funds, was furnished to the Army each quarter. The analysis and recommendations were essential to keeping top-level management informed. This reporting system and its level of detail were adequate for the successful financial control of the program.

Managing the massive, highly complex software subsystem for SAFEGUARD presented new challenges, and new systems evolved to control it. In general, standardized reporting systems met internal project needs fairly successfully. They enforced a basic level of planning and stimulated periodic surveillance of status. However, different levels of management required different levels of detail, and their requirements often could not be satisfied by one report. The useful life of status information to managers who had to take direct action was much shorter than the shortest time it took to gather data for a project-wide report, prepare the report, and distribute it. Also, it was frequently difficult for managers to evaluate the accuracy of information in written standard reports. Most managers relied on direct one-to-one discussions with their subordinates to obtain information that required immediate action.

One of the reporting systems employed was the computerized Management Reporting System (MRS). Its data base, which had some 3000 items, contained scheduled start and completion dates of various significant activities. Dependencies between the various items were indicated. Information was updated each month and published as a standard report. The time needed to gather, process, and publish information was approximately three weeks.

In addition to the general positive and negative attributes of these standard reports, two aspects of MRS stand out:

1. Schedule problems between different project areas were highlighted.
2. Special weekly reports could be hand-tailored for one person. In some cases, they were more useful than the standard monthly report.

In software development, numerical measures of progress, such as the number of routines coded and the number of trouble reports corrected, were usually found to be of limited value. The quantities counted or measured were not uniform with respect to the amount of work involved or its importance. Consequently, it was very difficult to interpret their significance.

Principal Events Reports

The most successful report prepared for agencies outside Bell Laboratories was the Principal Events Report used for software development and system integration. It identified a number of important concrete milestones and defined them in detail. Many of these milestones marked the completion of tests on various functional capabilities. Reports on such events were normally made by TWX within 48 hours after their scheduled completion. The TWXs gave the dates for completion of late items, while follow-up TWXs were then sent on the rescheduled dates. The Principal Events Report, which was issued quarterly, described each item, gave its status, and accounted for previously completed events. This system of

reporting was extremely valuable to higher management, with the TWX reports particularly useful.

Miscellaneous

Alternative Problem Solutions

There are often many alternative approaches to solving management or technical problems. In many cases it was more effective to choose a good alternative and then carefully implement it rather than spend a lot of time trying to optimize the choice among alternatives. Often, the initial analysis of a problem leads to a complex solution which may not really be essential to implementation, and further review may lead to solutions which are "elegant" in their simplicity. Project managers should repeatedly challenge their engineers to seek simple solutions to design problems.

In several instances, limited budgets led to modified requirements and less complex and costly designs. Constraints on dollar resources can establish hard priorities and stimulate simpler solutions. This was well expressed by Robert Townsend:*

"A tight budget brings out the best creative instincts in man. Give him unlimited funds and he won't come up with the best way to a result. Man is a complicating animal. He only simplifies under pressure. Put him under some financial pressure. He'll scream in anguish. Then he'll come up with a plan which, to his own private amazement, is not only less expensive, but also faster and better than his original proposal, which you sent back."

Independent Testing Operations

Inevitably, the time allocated to hardware installation and testing in any big system will be compressed as much as possible. This obviously makes it desirable to parallel operations. However, in complex systems the required testing is also unusually complex. When the system

is controlled by a centralized data processor, as in SAFEGUARD, the tests must usually include the data processor, so the amount of parallel testing possible is limited. Using relatively inexpensive minicomputers for much of the early testing can alleviate this problem. The savings from reduced installation and test time may be much greater than the costs of using minicomputers. One example of this approach concerned the Remote Launch Equipment (RLE).

A one-farm, two-missile prototype of the RLE had been tested at Kwajalein, and the tactical hardware had been given limited string testing in North Carolina. The first attempt to operate in a full four-farm configuration, over actual data links with a full complement of active launch cells, took place during installation and testing at Grand Forks, North Dakota. Therefore, rather complete tests were necessary.

During planning, it was predicted that when unit tests of RLE were complete, the ensuing system tests would require full-shift operation on most working days for several months. Schedule conflicts precluded timely completion of RLE system testing under control of the SAFEGUARD data processor. So a minicomputer, with a hardware interface unit and appropriate software, was temporarily installed at the tactical site to simulate the data processor. A similar installation had been used successfully at Meck Island during the prototype remote launch tests and at North Carolina locations for string tests. With the minicomputer, the tests were completed on schedule. The minicomputer also served as a diagnostic tool for difficult and complex design troubles, and surpassed the power of the Maintenance and Diagnostic (M&D) System and the built-in self-checks.

Communications for System Tests

Testing a system in a building as large and complex as the Missile Site Control Building (MSCB) requires a lot of internal communication, and running tests in several buildings at sites

*Up the Organization (New York, Knopf, 1970).

many miles apart is even more difficult. Careful planning is required for the communication system used during this testing. Instead of planning separate communication systems for testing and tactical operations, one should use the tactical communication system for testing wherever possible. However, nontactical circuits are required to support testing, particularly in the installation and early test phases. These circuits should be constructed around the basic framework of tactical communications. Such an approach allows an early evaluation of the tactical communication system, avoids the disruptions of switching from the testing to the tactical communications, and minimizes the cost of circuits required only for testing. This approach is best carried out by a single organization responsible for planning both the tactical communications and the testing communications.

Importance of Test Planning

It is universally accepted that good test planning is important to development of any system. For large real-time systems it may well be the most important part. Even when its importance is recognized, it is very difficult to make the best use of testing at each development stage. Although test planning in SAFEGUARD was probably more extensive than in any other single development, additional effort would have been worthwhile.

Many aspects of system development are strongly affected by the way tests are planned. The following are some of the most important effects recognized during the SAFEGUARD development.

- One of the first issues settled was the time allocated to system testing. Immediately after the deployment decision, the design, production, site construction, and installation had to be scheduled and coordinated. Inevitably, the planners wanted to achieve the earliest possible completion date. Because it had a direct impact on the completion date, one of the most important decisions was how much time to set aside for final system integration and test. It was impossible to list the problems which would be faced, and so it was very difficult to specify the tests

and, therefore, a definite time interval. For a system as complex as SAFEGUARD, the test interval will be between one and two years. Hence, the question is an important one.

With any system, the amount of testing (and therefore the testing time) is related directly, although in a very complicated fashion, to system quality. Reducing the amount of testing reduces quality, but increasing the amount of testing does not increase quality at a fixed rate. It is not at all unreasonable to limit the time and resources used for testing if it is recognized that doing so will have definite implications on quality.

For a system like SAFEGUARD, a limited-quality product can be accepted at the outset of system life. Quality can then be constantly improved as testing continues at site installations and at the prototype development site. This additional testing does not have the same incremental cost in resources as testing during the development cycle, because most of the resources must be available for routine operation. For a multisite deployment, the first installation need not have the ultimately desired quality, because tests can continue while other sites are installed. Consequently, the system test program can extend well beyond the first installation and perhaps even well beyond the final installation in a multisite deployment. The planners must understand the implications for system quality before the entire test plan is finished.

Another issue which has a major impact on schedule is the number of software tests. For a complex system, it is always possible to conceive of an infinite number of tests, so test planners must first establish a finite set. Next, project managers are rarely willing to allocate the time required for all the tests, so the problem is to select some subset of tests which is in some sense optimal. In SAFEGUARD, as a first step, the major functional capabilities to be used in implementing the system were listed. Then this set of possible tests was analyzed to see how it exercised these functional capabilities. A choice was then made as to whether each capability was to be exercised at all, nominally, or stressed to some extent. Then, a set of tests which implemented these decisions was chosen, and software test planning was pointed at that specific set. This approach seems to have produced an adequate testing program.

- Detailed test planning for system integration, based on experience with the prototype MSR, began in 1970, four years before Equipment Readiness Date (ERD). Planning considered test facilities (system

exerciser, threat tape generation) and the scope of testing (threat scenarios, traffic levels, system configurations and operating modes, threat excursions, confidence levels). Planning went on continually from 1970 until ERD, but even then, late deliveries of threat tapes caused problems in system test and integration at the TSCS. Although it is probably impossible to avoid this, careful monitoring is essential to minimize such problems because their impact on schedules can be major. In test planning, it is important to consider the reliability of test facilities. It is tempting to reduce the reliability requirements on test equipment not required for tactical operation (e.g., tape drives), but lower reliability will inevitably lengthen test time.

- As test planning proceeds, there will probably be important interactions with system design. The earlier these interactions are recognized, the lower their impact on cost, schedule, and system performance. Consequently, specification of system requirements should include requirements on the subsystems needed for testing.

To illustrate such interaction, when design of the system exerciser was well along, a serious short-time peak developed in the data processing requirements to test the system at the desired traffic level. Providing this capacity by increasing the data processing hardware in the system exerciser would have been costly. By modifying the tactical system design, the requirements on the system exerciser were reduced and the necessity for the additional hardware was thus avoided. This design modification did restrict actual system performance, although only very slightly. If the situation had been recognized later, it could have been resolved only by much greater financial costs, disrupting the schedule, or limiting performance.

- Another major question is how much testing to do at each installation of the system. For SAFEGUARD, it was decided that each installation would be tested as thoroughly in all respects, including traffic level, as the system was during development. Hence, each installation was given a complete system exerciser facility, even though this involved substantial cost. Such a decision hinges on issues specific to a particular system. In the case of SAFEGUARD, the major issues were:

1. Because the system might never be battle-tested prior to an actual engagement, it would be very difficult to determine whether its capability was deteriorating unless regular exercises could stress it to nearly its full extent.

2. Because crews must be ready to respond in minutes, if not seconds, it was important to exercise their responses.
3. It was important to thoroughly test each site, because each was assembled from tens of thousands of sub-assemblies, drawers, and racks; there were millions of interconnections and contacts and countless critical signal and timing interrelations. If a system could not be tested, it could not be made to work, or be kept working.

- Defining the requirements for data reduction of test results is important but difficult. In a system as complex as SAFEGUARD, every test produces a tremendous amount of data. For testing to proceed at a reasonable pace, the tester must be able to look at just the portion of the data he's interested in and to obtain it in the most easily understood way. However, early in the design he is unable to specify in detail what data he needs or how he wants to see it. Thus, it is difficult to build a data reduction facility by the time it is needed. In SAFEGUARD, this problem was approached by developing a series of data reduction systems. While this may be the only solution, it did require a very substantial effort. By applying more pressure earlier in the design cycle, designers would be forced to agree on a set of requirements to reduce this total effort.

Site for Prototype Testing

The TSCS, a partial prototype of the PAR and MSR, was built for developing software for the deployed system. Earlier in the program a prototype data processing system had been provided for developing the R&D software. Software could be verified with the TSCS only to a certain limit; because actual sensor and missile ground/interface equipment was lacking, the software development had to be completed at the field site. These R&D software deficiencies were remedied in the TSCS by including sensor parts that had direct or critical timing interfaces with the data processing system and selected parts of the missile ground and communications subsystem. Simulating responses for the sensor and other equipment is not a complete substitute for the equipment itself. Responses have to be limited, and the simulation can introduce artificial timing problems. It is difficult to

anticipate and design a simulation that has all the non-nominal and unexpected responses which can have serious impact on software operation. Simulation can only model the hardware, and the actual hardware may not conform to design intent.

The choice of equipment for the TSCS proved to be effective, and software developed and tested at the TSCS was brought up rapidly at the installation site. The software used to install the hardware at the site and later to maintain it was complete and of good quality. Not only did the high quality of this software allow the site installation to proceed quite rapidly, it also established a stable base for the more orderly development of the applications software at the TSCS.

Software designed to control hardware not included in the TSCS could only be partially debugged and tested at the TSCS; testing was completed at the tactical site. There was not a great deal of this software, but it was clearly not as well checked out when it arrived at the site.

As software was developed at the TSCS, many hardware design and manufacturing problems were uncovered. Many of the problems were corrected before hardware was installed at the site and, in almost all cases, solutions developed at the TSCS were available for site installation before the hardware was needed for the system test program.

The TSCS made site installation very efficient, so that the major efforts could be directed to solving problems unique to the site. Besides the efficiency gained in installation and check-out, the essential completeness of the prototype system-level problems could be identified and resolved. These problems, which included operational situations, were found long before they would have been detected at the tactical site. Hence, system integration progressed rapidly.

Dissemination of Information

It was hard for people to keep up with overall progress and problems of a job as complex as SAFEGUARD. To acquaint people with basic decisions and problems was essential, but it had to be balanced with the cost in time. For higher management, the need was amply met by monthly half-day sessions covering technical status and project-wide problems. Such meetings keep subsystem project heads technically sharp about problems in their areas of responsibility — particularly when the top managers have the technical acumen to probe sensitive areas. Clerical and support personnel were kept informed by quarterly filmed progress reports, and the inevitable time lag was acceptable for this audience. For lower management, there were occasional briefings on problems closely related to their work, typically on an organizational basis. Little in the way of information dissemination was done for nonsupervisory technical people.

More time should have been devoted to communicating with lower management and technical nonsupervisory personnel. The reports described earlier were good, but they were not always very current, could not be questioned, and frequently went unread. A better system might have been a half-day of several individual talks every month or six weeks. Information such as the following might have been communicated more effectively:

- Problems already solved and reflected in decisions. This area is important because the decisions can affect people's work directly, and the problem-solving techniques may be applicable elsewhere.
- Selected tough problems not yet solved. The payoff here is to encourage ideas from people who may be able to supply them. Also, a tough problem can sometimes be alleviated by energetic action in another area. A difficulty is that groups are reluctant to discuss a problem not yet completely solved, lest it suggest incompetence. Management should encourage the airing of such problems when necessary.

- Problems which have been formulated, but where solutions are not at all apparent, or are likely to be put off by the pressure of events. By raising these problems, they can be considered explicitly where they would otherwise have to be bypassed because of time pressure if left to upper management.

Transfer of Personnel

On a large development project, information must be exchanged between groups working on prototype systems and those working on production systems, and between groups preparing system requirements and those implementing them. The only feasible way to transmit most of this information, of course, is in writing or verbally. Some of this information exchange should occur via transfers of people between organizations so that ideas and points of view can be incorporated into the work as fully as practicable.

In the short term, transfers reduce the ability of the organization losing the personnel. However, this short-term cost is more than made up for by the long-term benefits of understanding gained by the receiving organization. On SAFEGUARD, substantial numbers of people were transferred from prototype to production development organizations and between other groups during production development. Even more transfer of personnel would probably have been desirable.

Timeliness of Problem Recognition

It is important that problems be recognized quickly, that solutions be carefully planned, and that proper resources be applied. Some problems which should be addressed early simply do not get timely attention. Others are addressed too early and inappropriately "solved," only to reappear later.

For example, initial attempts to design SAFEGUARD data reduction packages foundered because of inadequate attention. They were aimed at testing and integration activities then several years away, while operating system design, language processors, algorithm

development, and guidance simulation were commanding prime attention. This tendency to discount* problems perceived as remote in space or time also hindered communication subsystem design.

The impact of discounting should be minimized; it probably cannot be eliminated. Management should guard against it by periodic reviews of plans, forecasts, and current design activities. A comprehensive development plan is a decided help. Management should also restructure groups or departments as necessary to start early attacks on late-maturing design problems. Both of these schemes were tried and found useful.

Staff Assistants for Managers

Staff assistants, called Management Report Analysts (MRAs), were assigned to many second-level managers with project management responsibilities. The MRAs were, in general, college graduates with backgrounds in planning, scheduling, and budgeting. They were trained in the objectives, procedures, formats, and use of the SAFEGUARD Managing Reporting System (MRS) and could input up-to-date information into the MRS through their close association with their project manager.

They also assisted the project managers by preparing and controlling budgets, administering contracts, planning, scheduling, allocating space, arranging personnel moves, recording and following up on action items generated at project meetings, and performing numerous other administrative details.

Almost all of the managers who were assigned MRAs were enthusiastic about their usefulness. By relieving managers of many purely administrative functions, the MRAs made it possible for managers to concentrate on the more critical technical aspects of the job.

*See H. A. Linstone, IEEE Spectrum, April 1974, for a discussion of discounting and forecasting.