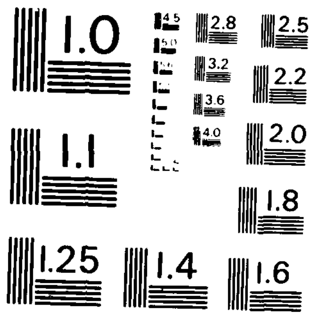
AD-A128 861 SOFTWARE TECHNOLOGY FOR ADAPTABLE RELIABLE SYSTEMS (STARS) IMPLEMENTATION APPROACH(U) OFFICE OF THE DEPUTY UNDER SECRETARY OF DEFENSE RESEARCH AND E.. 15 MAR 83 1/1

UNCLASSIFIED F/G 9/2 NL

END
DATE
FILMED
4-84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

UNCLASSIFIED

AD-A128861

REPORT DOCUMENTATION PAGE — READ INSTRUCTIONS BEFORE COMPLETING FORM

2. GOVT ACCESSION NO. A128861

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Software Technology For Adaptable Reliable Systems (STARS) Implementation Approach

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
The DoD Joint Service Task Force on the Software Iniative (STARS)

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Deputy Under Secretary of Defense
Research & Advanced Technology
Washington, D.C. 20301

12. REPORT DATE
March 15, 1983

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Unclassified

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Software applications, Implementation, automated tools, support environments, Ada, KIT, KAPSE, life cycle, methodologies, system interface standards, embedded computers, hardware, software engineering institute, mission critical, military systems, reliablilty, adaptability, STARS.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This STARS Implementation Approach defines a collection of activities which represent a basis for a more complete Implementation Plan. In developing the STARS Program, attention was first given to the technical feasibility of making significant progress in each of the task areas encompassed by the program. The intent was to identify high pay-off opportunities and a logical set of follow-on activities. The purpose of the document is to structure these activities into coherent streams that provide usable technology in the near, medium and long-term.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SOFTWARE TECHNOLOGY FOR

ADAPTABLE, RELIABLE SYSTEMS (STARS)

IMPLEMENTATION APPROACH

Department of Defense

15 March 1983

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1.0 INTRODUCTION

This STARS Implementation Approach defines a collection of activities which represent a basis for a more complete Implementation Plan. In developing the STARS program, attention was first given to the technical feasibility of making significant progress in each of the task areas encompassed by the program. The intent was to identify high pay-off opportunities and a logical set of follow-on activities. The purpose of this document is to structure these activities into coherent streams that provide usable technology in the near, medium and long-term.

The STARS planning approach is to start with this document and these functional task area strategies and produce an implementation plan through an interactive, DoD-wide process. The first step is to define a skeletal set of projects composed of those tasks which are on the critical path of the STARS program. The second step is to expand this skeletal set, selecting additional activities based on Component identified priorities in the context of ongoing and planned programs within DoD. The third step is to package the expanded set of activities into projects which, when integreated, make up the complete STARS program. The remaining steps are to plan, organize and carry out the projects over the life span of the STARS program.

The skeletal set of projects comprising the initial step in developing an implementation plan are described in this document. The rationale underlying the selection is explained in the next section in terms of their requirements for implementing the STARS program. The strategy underlying this planning approach is discussed in more detail. Finally, the projects within the skeletal set are described.

2.0 REQUIREMENTS FOR STARS IMPLEMENTATION

The requirements that serve to shape the initial, skeletal set of STARS projects are detailed in Figure 1. In the near term, the projects must lead to the delivery of effective technologies that support DoD mission critical system software over its complete life cycle. These technologies should include usable, modern, production-quality, automated support environments which will provide for the bulk of the transfer of software development and in-service support technology into practice. It is also important that these technologies include modern system building material (e.g., system parts as Ada packages, modern support software, etc.) in addition to aids for producing systems using this material. Finally, there should be quantitative demonstration of the benefit of both the automated support environments and the system building material and this demonstration should occur early, prior to extensive use on actual projects.

Just delivering the technology products is not sufficient--it is necessary that their delivery be the first step toward a significant, long-lasting improvement to the state of practice. This longer-term improvement process should start rapidly by building and capitalizing upon, existing and currently planned projects. In particular it should build on and extend the momentum of the Ada and APSE work. The STARS program will not be the only program, even within DoD, that attempts to provide automated support environments and system building material. It is critical that the efforts under the STARS program be strongly and compatibly linked with parallel efforts. Finally, it is important that the STARS program achieve a long-lasting effect both by priming the research-development-utilization pipelines and by establishing the practices and organizations that serve to keep the technology pipeline full.

/

DELIVERY OF EFFECTIVE SOFTWARE LIFE CYCLE TECHNOLOGIES

- o usable, modern, production-quality, automated support environments
- o modern system building material

SIGNIFICANT, LONG-LASTING IMPROVEMENT TO STATE OF PRACTICE

- o BUILD ON EXISTING AND CURRENTLY PLANNED EFFORTS
  - emphasize Ada and APSE's
  - expand on current projects
  - coordinate with existing plans
- o MAINTAIN COMPATIBILITY WITH PARALLEL EFFORTS
  - use DoD needs as driving force
  - compatibility among STARS supported automated support environments
  - seek common environments, compatible with others that are developed outside the STARS program
- o PROVIDE LONG-LASTING IMPROVEMENT
  - fill the technology pipeline
  - establish lasting organizations and practices that facilitate technology insertion

COST EFFECTIVE

- o leverage resources
- o promote duplication of effort only to minimize risk or enhance quality
- o early, quanitative demonstration of benefit

FIGURE 1: STARS Implementation Requirements

The third and final requirement for the STARS program implementation plan is that it's projects be cost effective approaches to meeting the other two requirements. The program does not have the resources to completely fund the production of effective products and establish the necessary technology transfer practices and organizations. STARS funds must be leveraged by seeding the appropriate activities and must be carefully used to avoid duplication unless essential to assure that high-quality products are realized quickly and effectively.

3.0 MAJOR PROJECT AREAS

As explained in the STARS Program Strategy, the STARS program will focus on a window into the technology pipeline and consist of three major phases: consolidation, enhancement and transition. The projects to be supported through the STARS program can be categorized as paths, depicted in Figure 2. Each of these project streams will be discussed below.

3.1 Building Automated Support Environment

The STARS program will facilitate the transition of technology through the construction of automated support environments. Each Service must maintain support environments for their systems. The more commonality that can be introduced among the Services, the greater will be the leverage for DoD to accelerate technology improvements. On the other hand there are different approaches which must be investigated, demonstrated and evaluated to ensure appropriate technology infusion. This seemingly contradictory situation leads to two different approaches to environment construction which can be effectively coordinated to yield an effective basis for improved embedded computer software.

3.1.1 Construction of a Common Automated Support Environment

It is essential that STARS build a common automated support environment. The Ada Program has defined the concept of a Kernal Automated Programming Support Environment (KAPSE) into which additional tools may be integrated. Two such initial systems are under development with DoD support (AIE & ALS), and others are being constructed independently in industry. The long term goal is to have a standard automated support environment for DoD use, but that goal is neither technically feasible nor realistic in the short term.

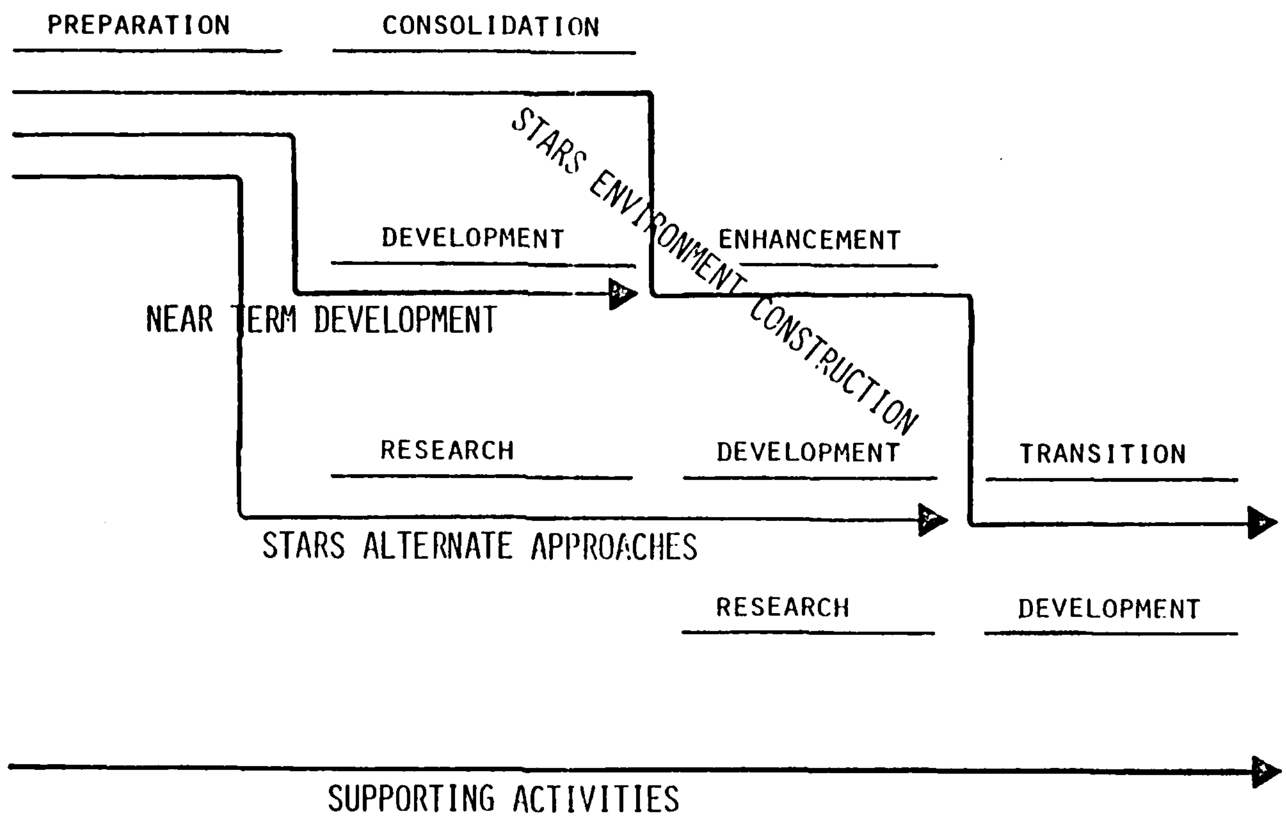


FIGURE 2: STARS Project Areas

Any common DoD support system must be hosted on a variety of computer and operating systems and must provide tools to cover the entire life-cycle. In rehosting the support system, differences in implementation will naturally result. Likewise, the state-of-the-art does not offer the basis for definition of a single life-cycle methodology upon which to base a complete environment. Further, the need for a mixed programming language environment must be considered for the foreseeable future with the added complexity that important languages are Service dependent. These factors do not, however, preclude DoD from continuing on a program aimed at reducing the level of duplication and increasing the development of standards.

The first step along this path has been taken in the Ada Program. Based on a memorandum of agreement among the Service Assistant Secretaries for Research and Development, a joint Service KAPSE Interface Team (KIT) and complementary industry associates (KITIA) have developed a draft System Interface Standard. Once refined and adopted, this standard will define the interface requirements between a KAPSE and additional tools. This standard will provide the foundation from which to evolve toward greater commonality among the Services and enable the consistent construction of sharable tools. This strategy offers the opportunity for a common core system of interfaces and generic tools but does not promise a single standard environment. A complete set of life-cycle tools must support a methodology or set of methodologies, including both management and technical methodologies. Different application areas may require different tools and techniques. While a substantial number of tools may support more than one methodology and therefore be common, our current understanding does not permit the specification of a standard without seriously impeding progress through experimentation.

The development of commonality in the support system is already a stated goal of the DoD within the context of the Ada Program. The

STARS program will aggressively pursue that goal by sponsoring the development of tools, techniques and an evaluation capability to ensure conformance to evolving standards. Projects to support this direction will be a responsibility of the Software Engineering Institute which will evolve the common automated support environment from a KAPSE, ensuring consistent development and implementation of the Systems Interface Standard. As described in the STARS Program Strategy, it will incorporate new tools and techniques developed under the auspices of DoD laboratory management both through Service supported efforts and those under the STARS program, as well as from technology independently obtained from industry and universities.

From the resulting state-of-the-art environment, the Services may derive more specific environments to support their programs. From the collection of tools in the Institute's environment, the Services will be able to reconfigure their environments, adding Service-specific capabilities such as tools to support specific management techniques, linkages to previously used language systems, and code generators for specific machines.

3.1.2 Parallel Industry Environment Projects

Many of the major defense contractors have undertaken, or are in the process of undertaking, the construction of life-cycle automated support environments to gain individual competitive advantages. These efforts are at varying levels of sophistication, often fragmented and not always used on defense systems. The DoD has an opportunity to realize substantial gains by encouraging this activity, seeding the process of adopting the evolving Systems Interface Standards, reaping the benefits of early application of these environments on major defense systems, and evaluating differing techniques.

The approach is to offer industry the opportunity of partial DoD subsidy to accelerate these developments, to participate in and conform to evolving standards and to use the environments on defense applications. Each of these projects will involve:

- building, within a three-year period, an initial version of an automated support environment,
- building subsequent versions that 1) incorporate enhancements reflecting suggestions for improvement stemming from actual use and 2) serve to introduce new tools of demonstrable value,
- rigorously demonstrating each version in one or more DoD application areas,
- providing guidance, based on experiences in building, demonstrating and using the automated support environments, for both enhancing the automated support environments and producing new technology for tooling and inclusion in subsequent versions.

The timing and inter-relationships of these components are shown in Figure 3. The building of the initial version will be broken down into several phases:

- a six-month DEFINITION phase
- a nine-month DESIGN phase
- a one-and-three-quarter-year IMPLEMENTATION phase.

Following the construction of this initial version, its enhancement will begin and the production of enhanced versions will be a continuing activity for at least four years. These subsequent versions must incorporate new capabilities that are selected from the pool of new, partially-demonstrated technologies developed outside the project and consciously identified both as compatible with the automated support environment's philosophy and concepts and of demonstrable value. All versions must be rigorously tested through their use in developing significant portions of defense application systems software on a DoD brassboard system. After demonstration (and possible modification) the automated support environment could be used either on an existing

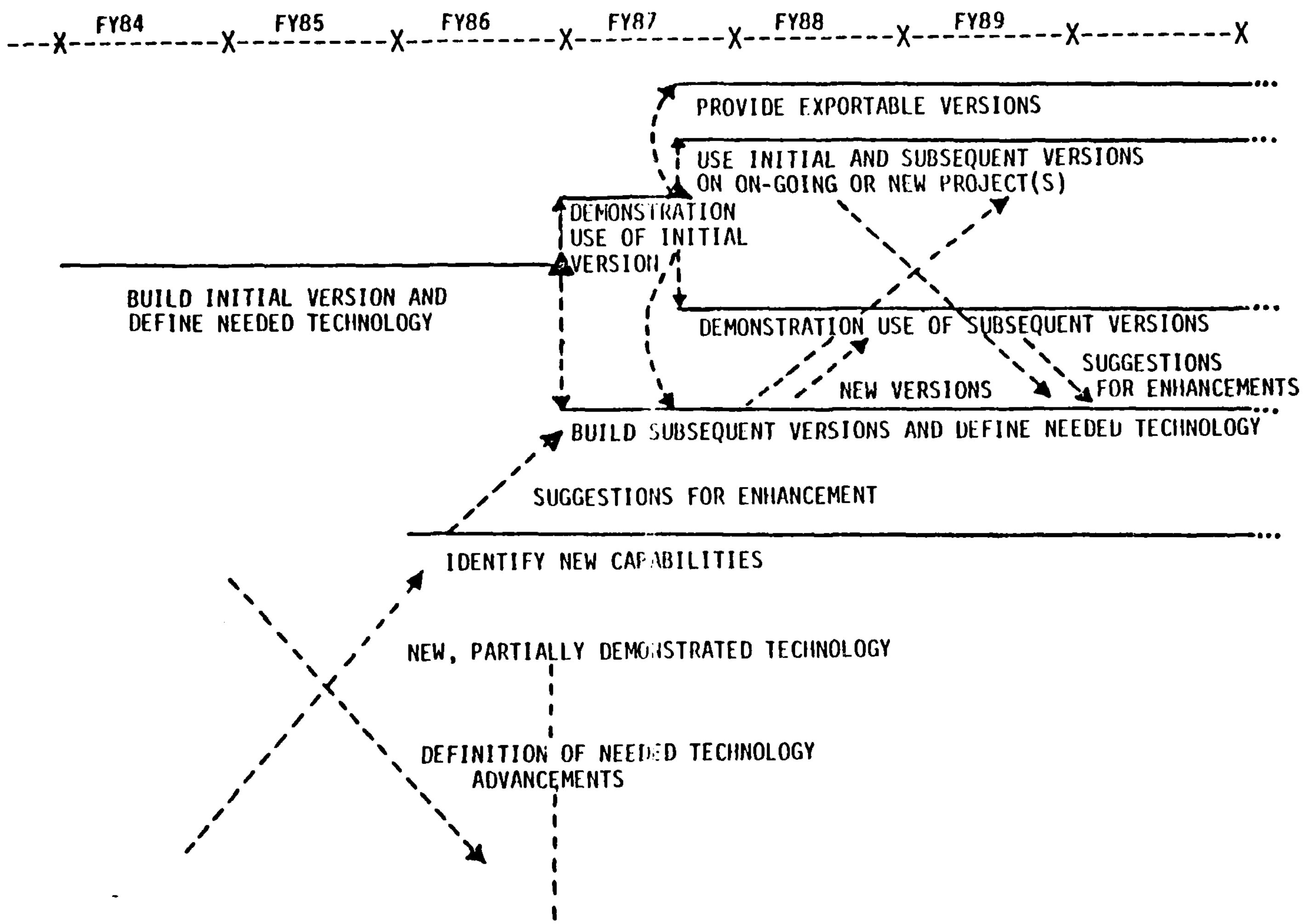


FIGURE 3: STARS Environment Construction Project

project (after migration of the project to the new automated support environment) or on a new project. In addition, exportable components of the automated support environment must be prepared for transfer to other organizations. All of the demonstration, use and export activities will result in suggestions for improvement that will be fed back into the development of subsequent versions.

Each project will be focused on the early production of usable, well-defined automated support environments by requiring that the automated support environments:

- o be oriented toward two or more defense application areas
- o support well-defined methods for
  - project management
  - full life-cycle software development and in-service support

The possible application areas will be defined by a late FY83 project that will survey and categorize defense application areas. The development and in-service support method definition will be part of the project and the Methodman categorization scheme will be used to put the method definitions on a common basis.

Additionally, the automated support environments should exhibit the following characteristics:

- o they should incorporate available technology
- o they should be more than trivially integrated
- o they should be well human-engineered
- o they should be rehostable to other host systems
- o they should be retargetable to other application areas
- o they should be based on KAPSE.

Several different approaches might be used in the construction of these environments and in the way they interface to a KAPSE and evolving Systems Interface Standards. The remainder of this section describes several possible scenarios to help generate an understanding of what is intended and to emphasize the range of possible approaches. These are neither the best nor the only approaches and are included solely to help understand the nature and intent of these projects.

One scenario for the construction parts of a project as a whole is to overlap the construction of the successive versions. In the abstract, the project would involve the activities depicted in Figure 4. New "generations" or "system releases" would be produced at roughly one-year intervals and each would be motivated by the desire to capitalize on technology that is not quite mature enough to include in the previous version.

The second through fourth generation construction efforts are similar. To structure them, the automated support environment is considered to be organized into multiple layers as pictured in Figure 5. Using this organization, the steady-state, second-through-fourth generation construction efforts could be structured as charted in Figure 6. At the core of each effort would be a traditional life-cycle of define, design, and implement. The definition phase would be divided so that each layer is considered separately and would end with the consolidation of the definitions of the separate layers. At the end of design, the prototype tools could be made individually available as well as used as the basis for implementing the environment.

Construction of the first generation would be done differently, as depicted in Figure 7, so as to both get a broader attack on definition and capitalize on existing efforts. Definition of the entire automated support environment would be done by one team with respect

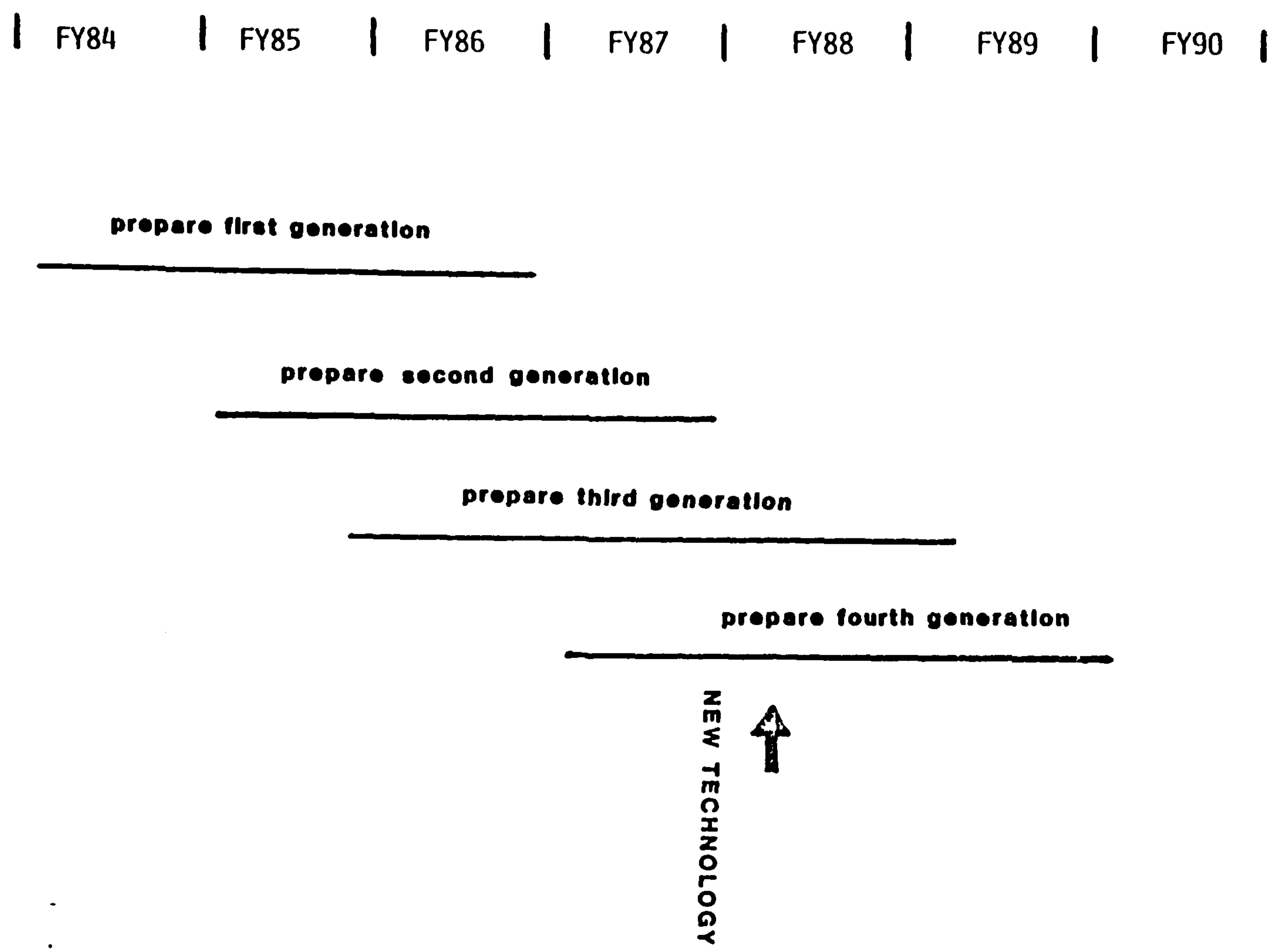


FIGURE 4: A Sample Project Scenario

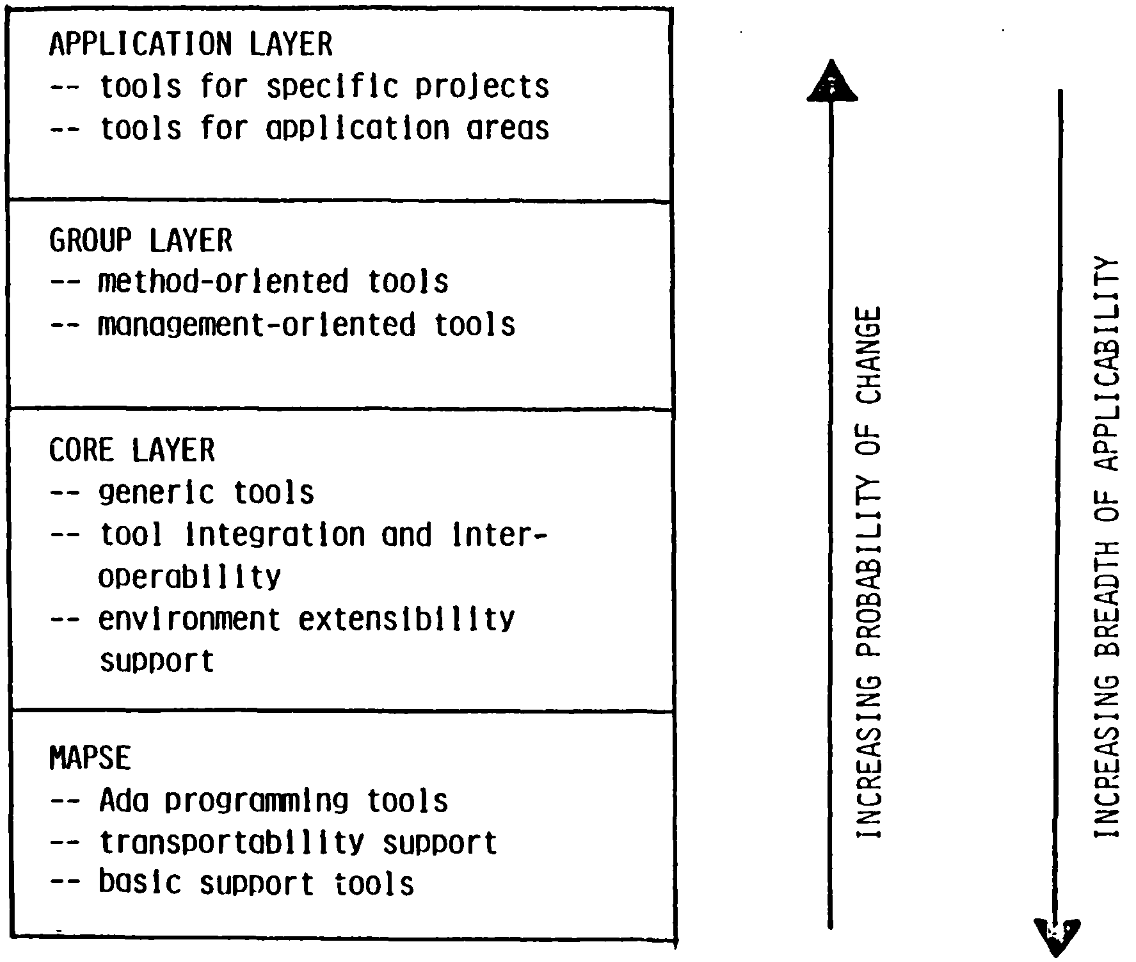


FIGURE 5: Possible Environment Organization

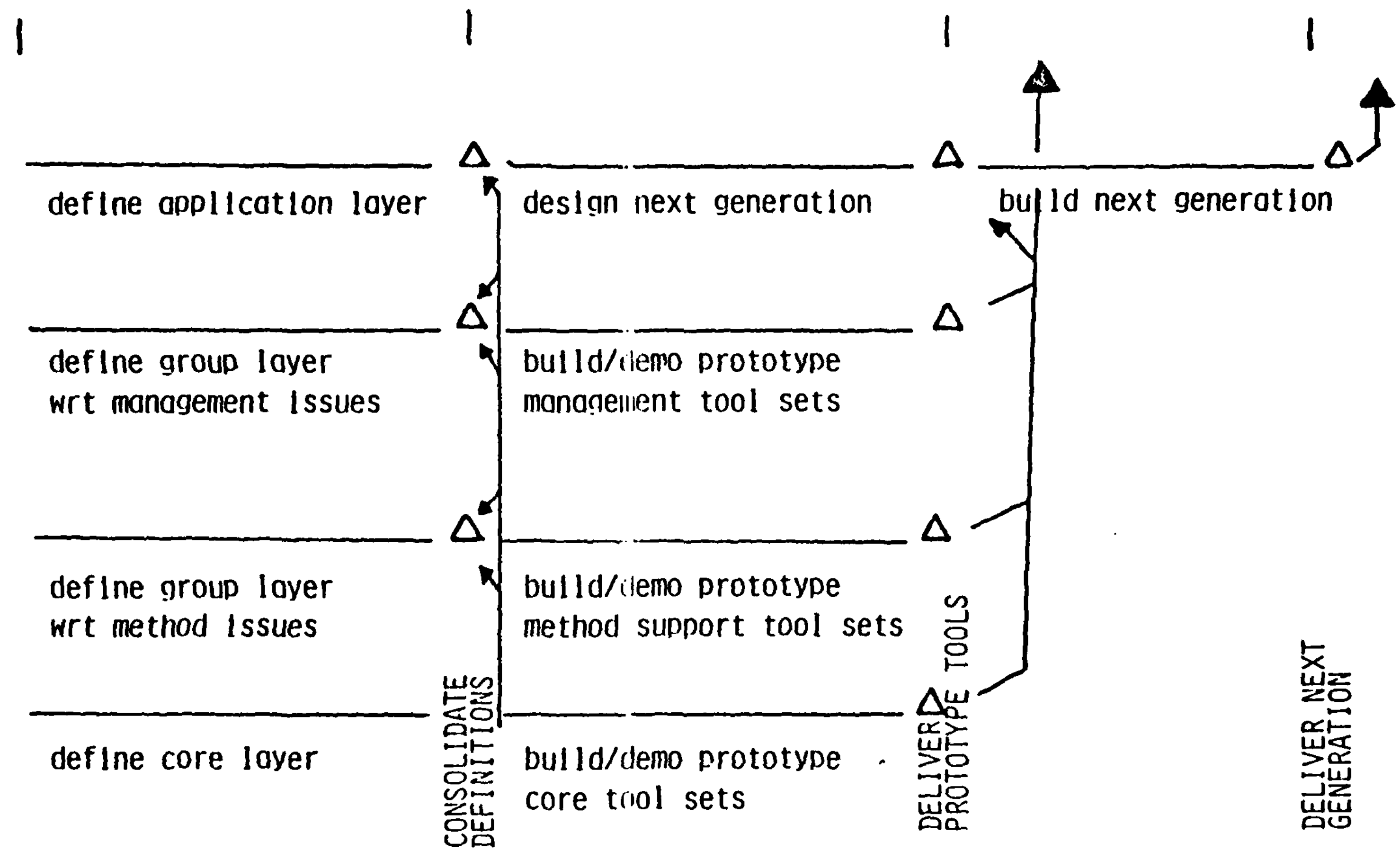


FIGURE 6: Constructing Second -- Fourth Generations

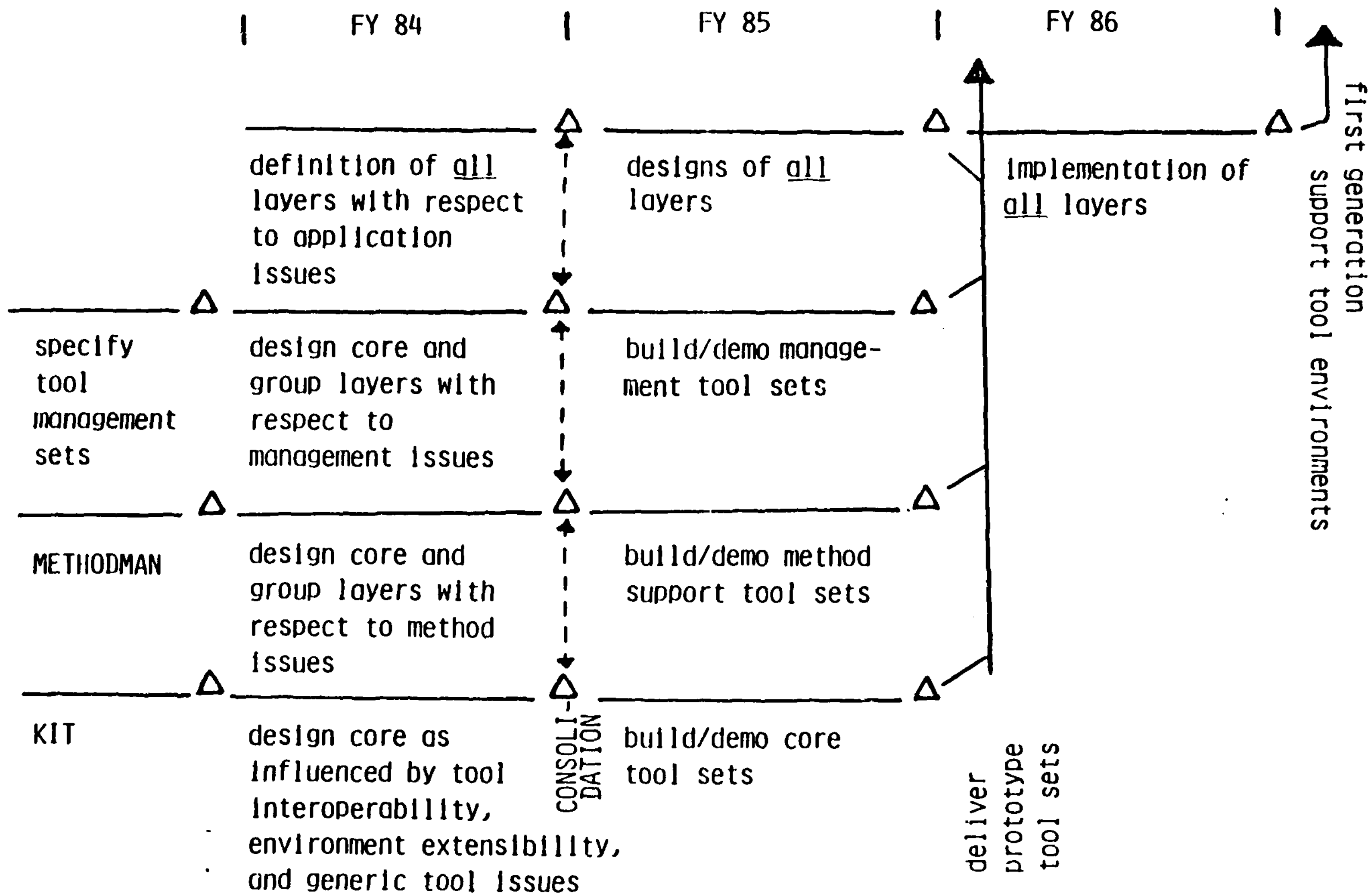


FIGURE 7: Constructing First Generation

to application layer issues. In parallel, other teams would capitalize on the previous definition efforts oriented toward the issues that need to be addressed in considering the other layers. These teams would carry out trial designs, extending the previous work. After consolidation, the first generation automated support environment would be designed and built. As before, prototype tools would be built and demonstrated in parallel with design of the entire automated support environment.

Other scenarios are possible. One major difference would be how the production of successive versions is staged. One could, for example, build an initial version that has been designed to be extensible so that the absorption of new tools is done by inserting them into the existing version, and the redesign and redefinition of the automated support environment is a rare event.

Another major difference in scenarios could concern how and when a version of the environment is made to run on a KAPSE. In the previous scenario a version is built directly on top of a KAPSE. Alternatively, one could start with VAX Unix TM and use existing tools, import or build new tools, and build the software to integrate the tools into a coherent method-oriented and application-oriented collection. Preliminary demonstration could then be done on this Unix-based system. In parallel, the ALS (which will run on VAX/VMS) could be imported and the collection of tools could be migrated to the ALS to provide a version for final demonstration.

Another alternative approach would be to build the preliminary version around some other host and its operating system, then rehost one of the available KAPSEs and migrate the tools to Ada and the rehosted KAPSE. A third approach might be to build the preliminary version on VM/CMS. The AIE (which will run on VM/CMS) could be imported when it becomes available and the environment migrated to Ada.

Exactly what scenario is reasonable depends on the expertise and experience and interests of the proposing organization proposing it and what they choose to capitalize on to get the project started. These projects will be directed by a statement of work that constrains some, but not all, of the details of the automated support environments and the approaches to building them. Whenever possible, the inclination will be to <u>not</u> specify a detail so as to permit innovative proposals.

However, there must be some commonality among the automated support environments that are built. One constraint to help assure some commonality is to require that each automated support environment be oriented toward <u>two</u> or more application areas. Several other constraints are needed, however, to assure a higher degree of commonality.

An additional constraint will be to require that the automated support environments reflect the guidelines and Systems Interface Standards initially specified by the KIT and evolved by the Software Engineering Institute. These will appear during the period that the environments are being built and will evolve over a period of time. Thus, the constraint cannot be to require conformance to the guidelines and standards except for those parts constructed after a specific guideline or standard has been officially adopted. It will obviously be desirable to develop a construction approach in which the KIT effort is carefully tracked and it is possible to quickly conform to the guidelines and standards.

Another constraint to enhance commonality is to require that all the automated support environments have a similar, high-level organization. Thus, it will be required that each automated support environment be organized (at least logically) according to the structure pictured in Figure 5. With this minimal structure it will be easier to compare various automated support environments and there will be greater opportunity for a higher degree of tool transportability and interoperability. The MAPSE and Core layers are also prime targets for early expansion of the System Interface Standards.

Finally, the competetive construction projects will be reviewed at the end of the definition, design and implementation phases, as well as after demonstration, with the intent of reducing, at each review, the total number of parallel efforts that will be allowed to proceed to the next phase. A primary selection criterion at each phase review point will be the degree to which an automated support environment provides a common base for application areas, development and in-service support methods, and project management methods other than those which it implements.

There are several advantages to these multiple automated support environment construction projects. Several major defense contractors will substantially improve their ability to deliver better quality software, much in the same way that the VHSIC program seeded the development of microelectronic design and fabrication facilities. The DoD will be able to evaluate different approaches. The Defense industry is more likely to participate in development of the System Interface Standards and adopt the results especially if the winning contractors can expect a long term payoff for their efforts on future system contracts. DoD will benefit from industry investment and will get the results of that part of the development which it supported.

This approach is not inconsistent with the evolution of Systems Interface Standards and the goals of common support systems. The Software Engineering Institute will be able to evaluate different approaches and derive common characteristics. In addition, the competing activities will produce individual tools and techniques which can be incorporated into the baseline. Finally, the defense industry will have the incentive to use the evolving System Interface Standards. If DoD is prepared to pay in the form of licenses and royalties for results from private investment as understood and negotiated prior to contract awards.

While the individual automated support environments will include different tools to implement different techniques and methods, adherence to the evolving Systems Interface Standard will offer the flexibility to require the use of standard tools. For instance, if the software is to be maintained by the DoD, the responsible Service may wish to require that specific tools supporting configuration management and documentation control be used. They may also require that other tools used by the contractor be available to the government, perhaps under some license arrangement.

Cost estimates of these parallel developments are not yet available. The costs will depend on the number of contractors chosen and the amount of industry investment. The definition and design stage would require approximately $1.5-2M level of effort seeding per contractor, spread over FY84 and FY85.

These contracts should be handled by a single contracting office and managed by a Joint Service Team under direction of the STARS Joint Program Office.

3.2 Alternative Approaches Project Area

The STARS construction projects are intentionally constrained to force emphasis on the consolidation of well-developed technology into demonstrably usable and effective production-quality automated support environments. Alternative approaches must be investigated to complement the construction projects by stressing the development of alternative approaches to software development and in-service support. These may be alternative approaches to organizing an environment, or alternative approaches to tooling technology for delivery to practitioners.

A project in the alternative approaches area will involve building, again within three years, a prototype automated support environment, followed by the demonstration and testing of its utility and effectiveness. After demonstration, a production version could be built or perhaps the new technology could be absorbed into the production-quality automated support environments being produced as a result of the STARS environment construction projects. In either case, the requirement will be to successfully transition into practice the demonstrably effective technology that emerges.

Projects in this area need not be formulated under as many constraints as in the STARS construction project area. Specifically:

- the automated support environment produced within three years will be prototypical rather than production-quality

- the automated support environment must be oriented toward producing DoD systems but need not be oriented toward any specific application area

- the automated support environment could be independent of a particular method for software development and in-service support

- the automated support environment could reflect a non-traditional approach to software development and in-service support; it could, for example, be based on a rapid prototyping, verification, or knowledge-based approaches.

It is again useful to provide several possible project scenarios to illustrate what is desired for an alternative approach project. One possibility is charted in Figures 8 and 9. In this example, the focus is on prototyping a core environment that incorporates a KAPSE but is not necessarily organized as a layer which runs on a MAPSE. Figure 8 indicates the major activities and the development of capabilities for the core is detailed in Figure 9.

Another possibility, which focuses on developing a prototype of a knowledge-based support environment, is charted in Figure 10. The

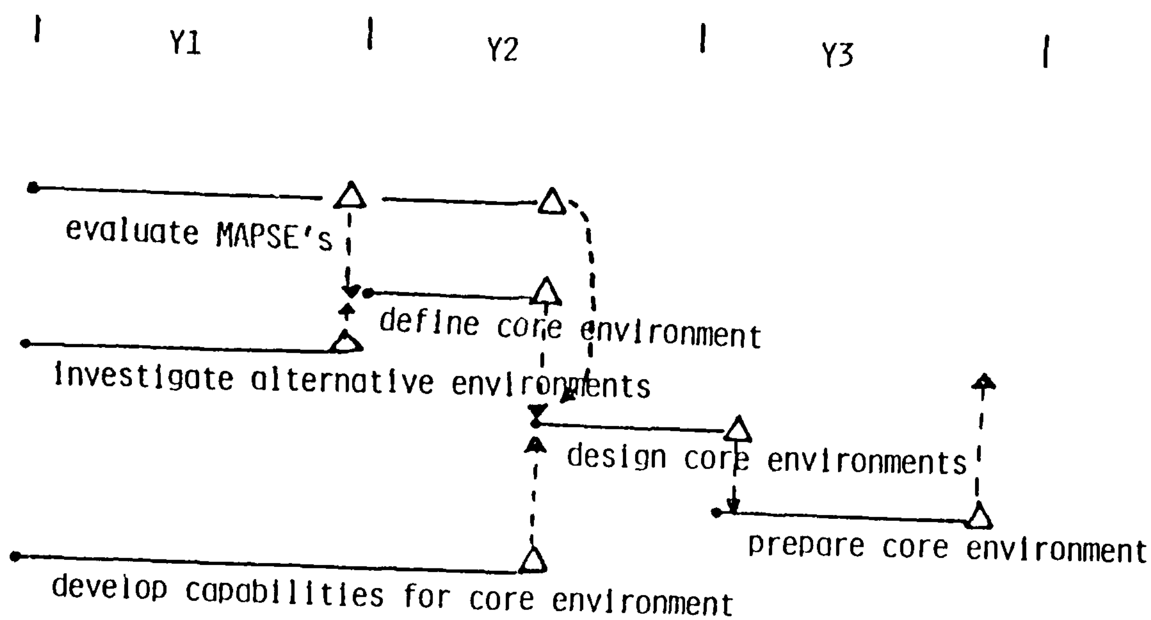


FIGURE 8: Building a Core Environment

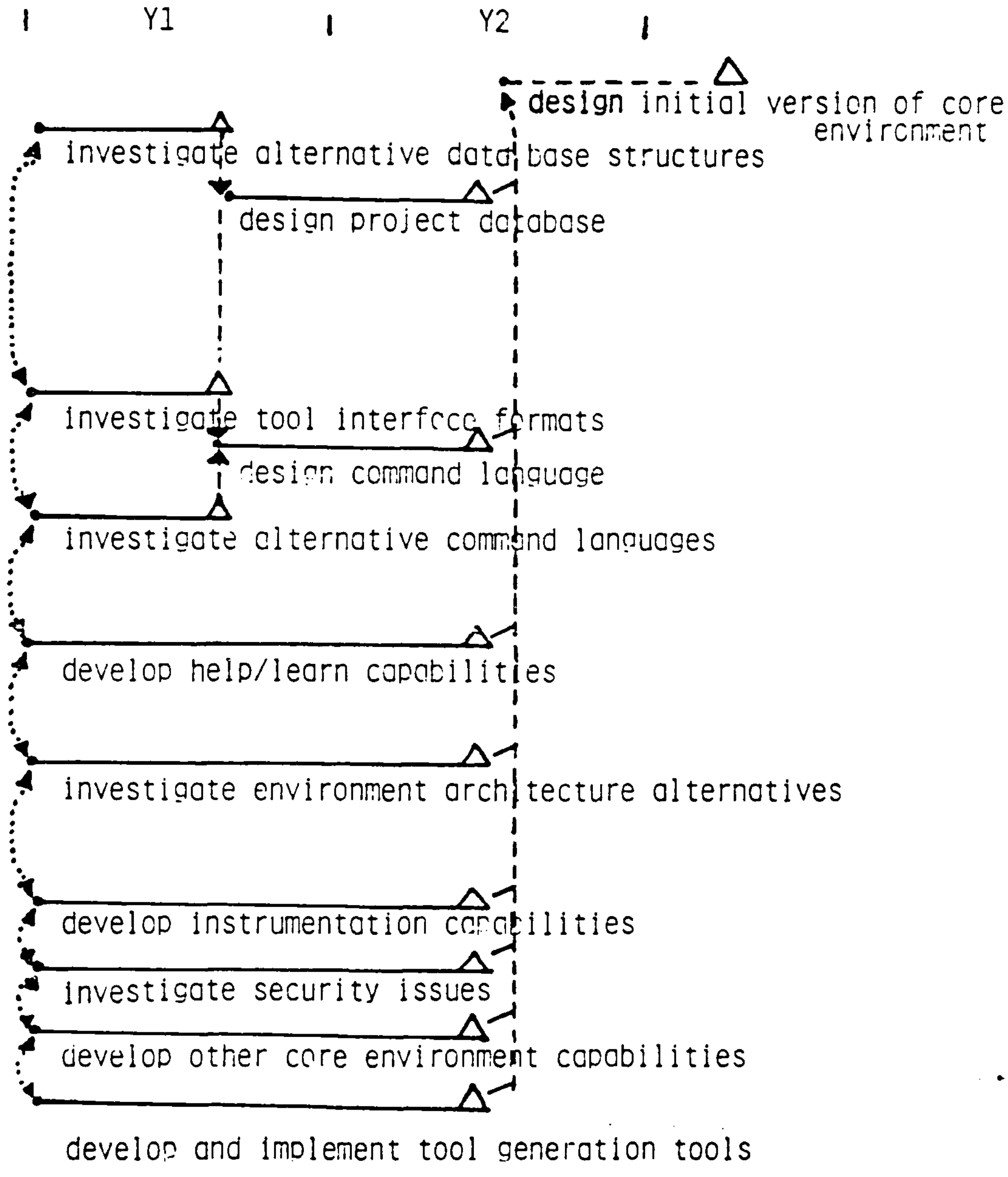


Figure 9: Detail for Developing Core Environment

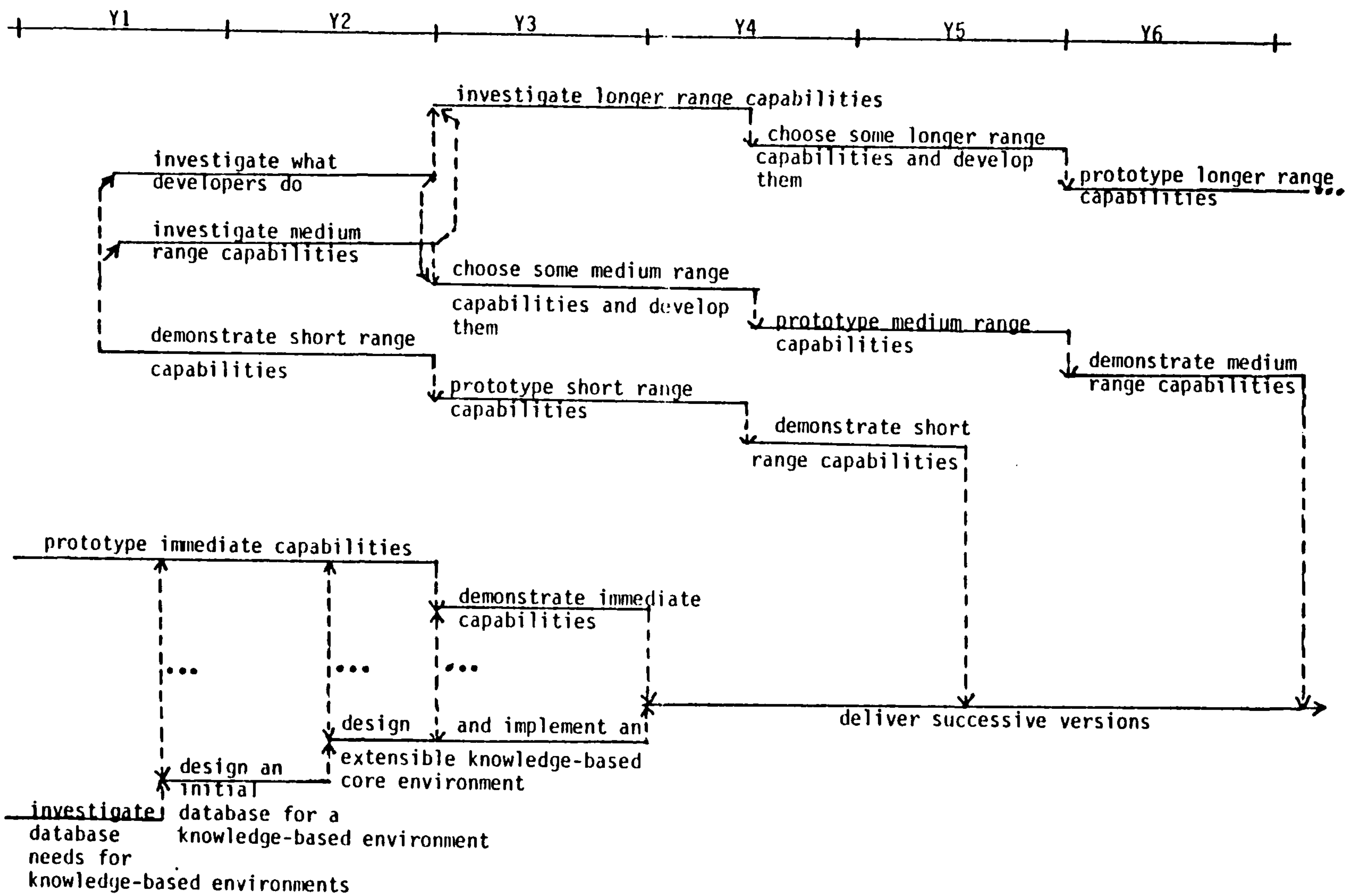


FIGURE 10: An Approach to Knowledge-Based Environments

major thrust, at the bottom of Figure 10, is to prepare an initial prototype that incorporates currently available knowledge-based tools. In addition, future enhancement is handled by the staged development of new capabilities and their periodic insertion into the prototype automated support environment.

A third possible scenario is depicted in Figure 11. In this approach, the building of the automated support environment parallels its use on an actual application system development project that goes through two application life-cycle cycle passes. The environment is built by developing the necessary description notations during the first life-cycle pass and then developing the necessary description analysis tools during the second life-cycle pass. In addition, the details for a scenario such as this one could specify that the methodology underlying the automated support environment would be specified iteratively, influenced by the trial use of the notational and analysis tools.

These scenarios indicate the breadth of possibilities for projects in the less-constrained alternative project area. Specific projects will be formulated by the Services.

3.3 Near-Term Development Projects will be Selected by Need

The automated support environment construction projects will quickly consolidate existing technology and produce some new tools and techniques. However, the functional task area strategies have identified many other opportunities. Selection of projects to realize these other opportunities will depend on the priorities established by the Services. Each Service will propose development projects from the functional area strategies to support the STARS objectives for which that Service is prepared to take the lead. From this set of proposed projects, a program plan will be derived. Identification and selection of development projects by the Services will

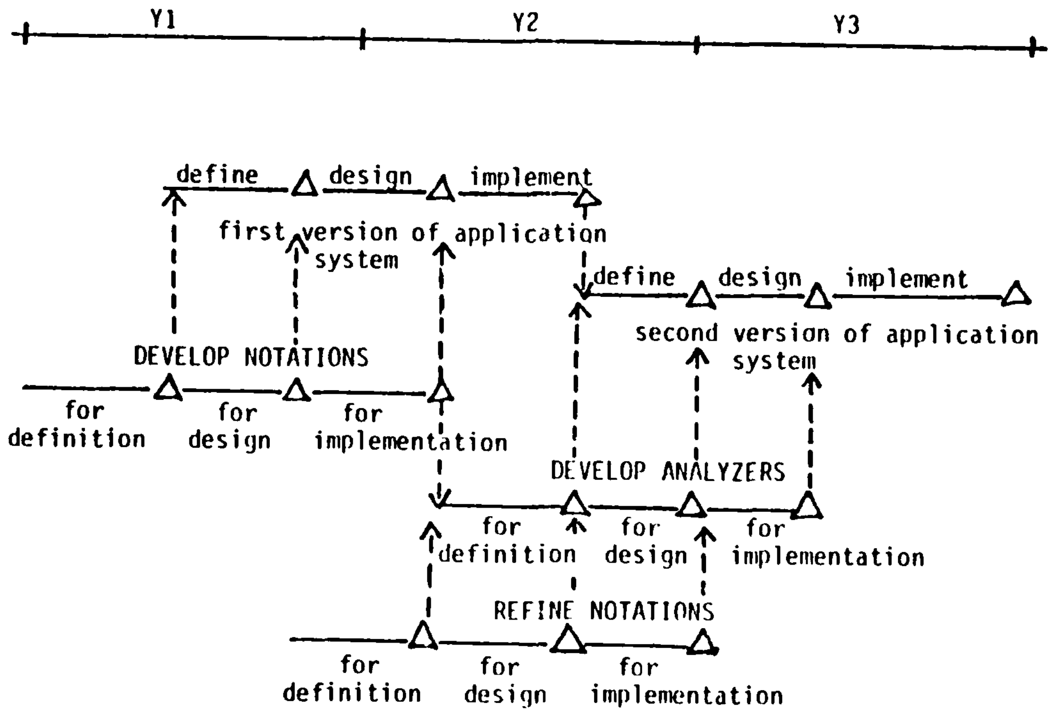


FIGURE 11: A Use-Driven, Method-Independent Scenario

ensure that techniques are developed to support specific needs and that maximum benefit is derived from existing projects.

However, several projects identified in the functional task area strategies are on the critical path of the STARS program. If these projects are not supported early, later developments will be hampered. These projects are:

- o <u>Development of application-oriented Ada package sets</u>: The immediate task requires identification of the important application areas. An initial list of six to twelve application areas that are well suited for initiating a software reusable parts technology is to be composed—for example, digital avionics, communications, command and control, tactical missiles, smart monitors, ground-based air defense systems, and artillery fire control. While it is believed that almost all areas will eventually be suitable for this technology, some are presently more suitable than others. Areas of early interest to Defense systems will be selected for technology demonstration. It is important to establish a window between the users community and the STARS Program, so that reflected need of the users can drive the STARS Program. The Application Specific Environment is expected to provide the coordinated product of all task areas of the STARS Program and to contain the benefits of STARS technologies particularly the important benefit of re-usable software. Libraries of application-oriented Ada package sets are the first type of reusable target software to be pursued.

- o <u>Develop evaluation criteria for modern systems software</u>: Two tasks should be accelerated.

  (1) Evaluation criteria should be developed for Ada and computer systems architectures. The DoD contractor-builders of Ada compilers and Ada programming support environments are also developing evaluation criteria including assessments of the suitability of use of Ada by available computer systems architectures. This task proposes to leverage the on-going activity by establishing pilot projects to evaluate and demonstrate use of Ada/Ada environments in a real world application areas.

  (2) Ada access should be provided to target run-time systems. A means would be provided to start using Ada for

systems development as early as possible without requiring that the underlying systems software for the target be written in Ada. Ada access to systems software in the target system environment would be provided by such means as using a cross compiler with interface and representation specifications and/or Ada access to device drivers would be provided along with the means for developing new device drivers.

- Developing techniques and tools for assessing and enhancing reliability: Early tasks should include a review of Ada reliability notions and identification of related system reliability issues. Parallel tasks will be established to develop methods and techniques including testing methods and techniques, and a handbook developed to support their use. This work would be coordinated with the pilot projects to evaluate Ada/APSE in a real world application. This task will provide an early framework for later work as well as early usable results.

- Developing techniques and tools supporting tool integration: The underlying premise is that methods, languages and tools must together form a coherent framework, held together by a realistic, modern view of the system life-cycle. These issues include but go beyond the strictly technical issues of integrating tools and techniques. One key question to be answered is how different tools and methods can be (re)configured to suit the needs of a particular mission critical area.

- Developing techniques and tools for environment instrumentation and environment usage data analysis: This measurement task would support the development of instrumentation tools required for collecting the data required to drive the models and metrics. The instrumentation tools would implement both manual and automated data collection during the software life-cycle. Deliverables include a standardized description of the data needed to drive a selected set of models and measures for establishing and maintaining the baselines, a set of procedures for collecting data for the baselines and tools for data collection. These tools and techniques are important to assist the acquisition, project mandagement an technical personnel in performing their own measurements to assess progress, cost and quality of their products.

3.4 STARS Management Support Projects

There are several projects that are critical to the management of the STARS program, the smooth flow of new technology into the environment, or the propagation of the environment into use. While many of these projects will most naturally arise as adjuncts to existing and already planned activities within DoD, there are several that must be initiated immediately to assure coherency of the STARS program. These projects are detailed in the STARS Functional Task Area Strategies.

Program Management Projects

- Establish baseline data against which to compare future development and in-service support activities in order to assess progress: This task involves collecting, consolidating and analyzing measurement data on selected projects through developing, refining and maintaining a set of baselines, these baselines would provide life-cycle information on the cost, quality and resources for a representative sample of software projects.

  Measurement data baselines are important to two types of communities: software and systems managers would be aided who are currently experiencing great difficulty in estimating cost and resources required for achieving acceptable software quality on new projects; STARS Program managers would be aided in assessing STARS progress. By conducting a thorough and exemplary implementation of measurement activities in a few software projects, it would be possible to demonstrate how measurement enables one to understand and hence improve software engineering during all phases of the software life cycle.

- Determine program-success measures: This task focuses on identifying success measures for the STARS Program itself. The STARS candidate tasks could be, in part, evaluated by using these measures. The importance of this tasks is that it provides the basis for quantitative assessment of the program and might help assure that the STARS Program will contain high payoff efforts whose value can be defended and proven, within a relatively short period of time.

- Establish criteria and measures for evaluating automated support environment definitions, designs and implementations: Criteria and an evaluation method for comparing definitions and designs will be developed. Quantitative assessments of the value of automated support environment implementations will be supported by developing experiment based demonstrations and evaluations of methods, tools, and environments. This activity includes designing the experiments, instrumenting the environments so that the necessary measurements can be made, assessing conduct of the experiments and interpreting the results.

Technology Identification and Selection Projects

- Conduct Methodman experiment: The experiment is needed to answer the question: "what is the effect of various software design methods on the maintainability of systems..." The experiment will be conducted through an approach described in the Methodman document: "Comparing Software Design Methods for Ada: a Study Plan." The four broad activity areas involve (1) creation of architectural designs for a specific problem (2) Implementation in Ada of the design and checkout, by several teams (3) modification of each implementation by several maintenance teams and (4) evaluation and reporting on the impact of the architectural design method on the maintainability of the resulting Ada-coded systems. An advisory board will also be established to review experience and results.

  This experiment is considered to be an important step in the objective investigation of software productivity. These investigations of the ways in which software is developed must be carried out to permit us to make more rational decisions regarding the way we organize and carry out system development.

- Develop measurement criteria, metrics and experimental techniques: The measurement task area would develop measures of the software product development and support processes and resources. Measures define the criterion which is measured. Techniques include the definition of models to predict desired attributes or factors of interest.

  There is a grave need for measurement in early stages of software development. The activities to establish measurement criteria and experimental techniques early in software projects' life-cycle stand to provide a basis for correction of over 60% of all errors prior to software/system construction.

- Conduct an analysis of the project management function: Projects will be analyzed in order to describe generic project activities and their relation to one another in both sequence and required coordination. Activity elements, and policies and procedures covering thier relationships will form the foundation for systematic improvement. This foundation will be used for increased visibility, better reporting, and as a basis for the development of automated project management tools. Important benefits are foreseen from establishing methodologies for project management based on improved understanding of the process of managing software.

Propagation Projects

- Coalesce current acquisition practices: This task should include two types of activities

  (1) Review impediments in current acquisition practices

  This task would include the definition of the software activities which are conducted as part of the system acquisition process followed by identification of the problems, deficiencies, impediments and restrictions associated with the process. Contractual vehicles, policies, regulations and standards, and their implementation, management and contracting acquisition tools and databasbes that could be expanded to improve the software acquisition process would be identified and assessed.

  This task is important in that it would establish the basis for action to remove the identified impediments.

  (2) Establish approach to protection of software including proprietary, classification, and foreign export issues: The STARS Program should stimulate industry investment in improving software engineering capabilities. To encourage industry, economic considerations must be given to major prime contractors, subcontractors, and entrepreneurial firms. The software acquisition process must complement profitability and protect trade secrets whenever possible. DoD must initiate creative approaches to resolving software protection issues while providing access for the Defense community to the

developed technologies, methodologies and tools. Areas to be addressed include: software data rights, revisions of the Defense Acquisition Regulations and Federal Acquisition Regulations (FARs).

Resolving this issue may constitute one of the key contributions of the STARS Program to solving current problems with embedded computer software. Worthwhile software methodologies and tools must be widely propagated in the best interest of the entire defense software industry's productivity and reliability.

- Establish Acquisition Panel: A panel would be established by the Under Secretary of Defense for Research and Engineering to serve as the DoD focal point for implementing measures to improve and unify policy practices and procedures related to the acquisition of Defense systems employing software in Mission Critical Computer Resources. The panel would be composed of members representing various DoD elements significantly impacted by or having significant impact upon Defense software acquisition.

  It is intended that the Software Acquisition Panel facilitate sound and logical business and contract practices associated with all facets of Defense software acquisition; and to provide appropriate incentives to encourage enhanced contractor participation, productivity, software quality and software reliability.

- Assess human resource and management skill needs: This task would assess software related skills and their utilization within the DoD community. Both the types and quality of software related skills of DoD personnel and the utilization of the DoD personnel would be assessed. Products would include skill requirement descriptions and numbers of personnel required for each position.

  There is concern within the DoD community that the existing personnel system does not adequately address software personnel. This task is the important first step in increasing the level of expertise in DoD's embedded computer professional population. The skill assessment provides the front-end work for the several human resources improvements which are envisaged such as course development, education, training and career structures.

- Establish mechanism to evaluate and prioritize human engineering research, methodology, and tools: The identification of specific high priority activities and tools, and of specific research projects requires special planning and continual feedback and re-assessment since the human engineering of computer systems is currently an interdisciplinary endeavor. A specific mechanism must be put into place to insure a focused, effective effort. An Advisory Panel or other mechanism would be established for this purpose. An Advisory Panel might consist of a small group (7 or 8) of the leading practitioner or researcher from each of the relevant disciplines.

  It is important to consolidate the dispersed human engineering methodologies and tools in order to facilitate their insertion and achieve their benefits in embedded computer software developments and use. This mechanism should be established early in the STARS Program for evaluating and focusing on human engineering benefits.

- Form user groups within application area and system development and support communities: Two type of user groups would be established:

  (1) Following identification of initial application areas for a software parts technology, active user groups will be created to guide the establishment and demonstration of reusable parts inventories. Organizations and individuals who have the technical and leadership qualities to form such groups will be identified.

  The establishment of application specific user groups is important to transitioning the application specific technologies. A major sub-objective of user groups is to gain leverage from cooperative support both of the military program manager and, through Government/industry cooperative efforts, of the private sector. The application specific users groups will be utilized in such a manner as to foster such cooperation by building on existing DoD/industry structures.

  (2) In order to assure the utility of the software engineering environment to be established at the Software Engineering Institute, a user group comprised of software engineering practitioners will be established. Personnel will be sought who produce software as their preponderant responsibility, either in Government or in

industry. The group will aid in evaluating new methodologies and tools, establishing requirements, providing nucleus of informed users of the new capabilities, and providing a path for feedback.

The Software Engineering Institute user's group should provide important motivation and leverage for spreading the use of the new software methodologies and tools.

3.5 The STARS Program Links

Figure 12 portrays how all of the STARS program projects discussed above directly support existing Service projects related to mission critical system software development and support. It also shows that all STARS activities are designed to improve the DoD's future technical lead in software engineering. This constitutes the fundamental conceptual framework for STARS program implementation.

Figure 12 shows two major streams of projects, those related to on-going Service activities and those to be sponsored under STARS. The Service activities include software dependent mission critical system development life-cycles (one shown), and the evolutionary improvement of existing Service specific software environments (at least three).

STARS has three main streams of activities directly related in the near term to the Service project streams. These are 1) the development of a STARS common software environment (long term goal with work beginning now), 2) the construction of improved mission critical system automated support environments (mid-term goal with work beginning now), and 3) research aimed at solving known critical problems whose solutions are necessary to specific mission critical software environment development projects.

The remaining STARS project stream involves research aimed at making breakthroughs and quantum jumps in state of the art alternative software environments. This work is not tied directly to the

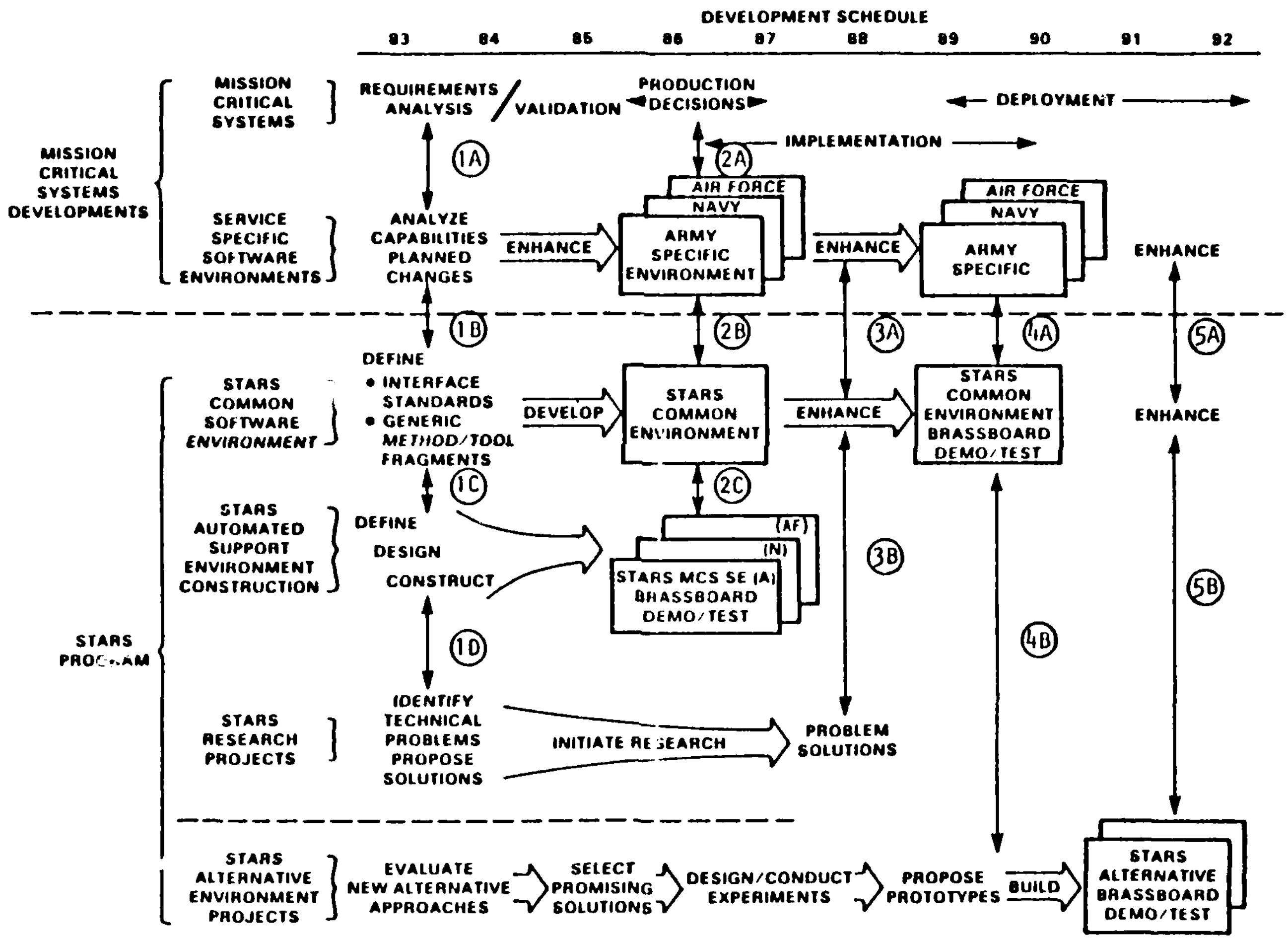


FIGURE 12: STARS Program Implementation Concept

other five project streams until near the end of the STARS seven year life.

The five sets of linkages (labeled 1A through5B) between the six project streams are designed to aid one or more of the following three technology improvement objectives:

a. A real improvement in the state of the art.

b. The reduction to practice of an improvement in the state of the art.

c. The sharing of the current state of practice among different organizations (e.g., between the Services and other DoD components)

A very brief description of the objective of some of these linkages indicates how the underlying rationale for the implementation concept is formulated.

Linkage 1A—comparison of results of a current system requirements specification with existing software environment capabilities should promote improvements in the state of the art and their reduction to practice (a and b above) to upgrade the Service environment before the production decision.

Linkage 1B--compares planned changes to existing Service environments due to 1A with what standards and generally accepted "best" generic methods and tools exist that should be used. This leads to both Service system enhancement and improvement of the standards.

Linkage 1C—depicts a flow down of information from 1A and 1B to enable a contractor to define, design and construct an improved application specific environment which will provide useful new methods and tools for the STARS common environment through linkage 2C after a realistic demonstration on a Service-owned system brassboard

(avionics hot bench, flight simulator, communications test bed, "plastic tank", research and development ship, etc.)

Linkage 1D—funnels problems identified through linkages 1A, 1B and 1C for which no ready solution is apparent. This generates applied research projects whose resulting solutions are eventually fed back to the Service specific environments by means of linkages 3B and 3A.

Linkage 4B--the fruits of the fundamental research stream that is seeking a truly "better way" to engineer software eventually reach a stage of proposed revolutionary change. This linkage makes the comparison to determine if an alternative should in fact be built. If so, a demonstrated better environment is linked back to the service specific world by means of 5B and 5A.

The remaining linkages should be self explanatory.

Thus, all of the parts of the very large and complex STARS program logically fit together in the dimensions of time, technology evolution and technology revolution. This provides a coherent STARS Program Implementation Concept.

3.6 Summary

A structure has been presented for the STARS program that selects critically important activities from the Functional Task Area Strategies and packages them into projects that can be executed by the technical community with a fairly high degree of leverage of the STARS program resources. A complete set of projects has not yet been proposed. The Components will develop implementation plans to execute and augment this set in a way that capitalizes on their existing and already planned activities. The project definitions are not yet ready for publication as Requests for Proposals. The step of developing detailed requests for proposals will be taken based on advice and suggestions from the technical community on the general strategic issues discussed.

It is important to progress quickly in the STARS environment construction, alternative approaches, and supporting activities project area. The plan is charted in Figure 13. It has been specifically designed to allow accommodation of comments received on the STARS Implementation Approach discussed here. It has also been designed so that proposals in the alternative approaches area can be made with knowledge of what will be done in the automated environment construction projects.

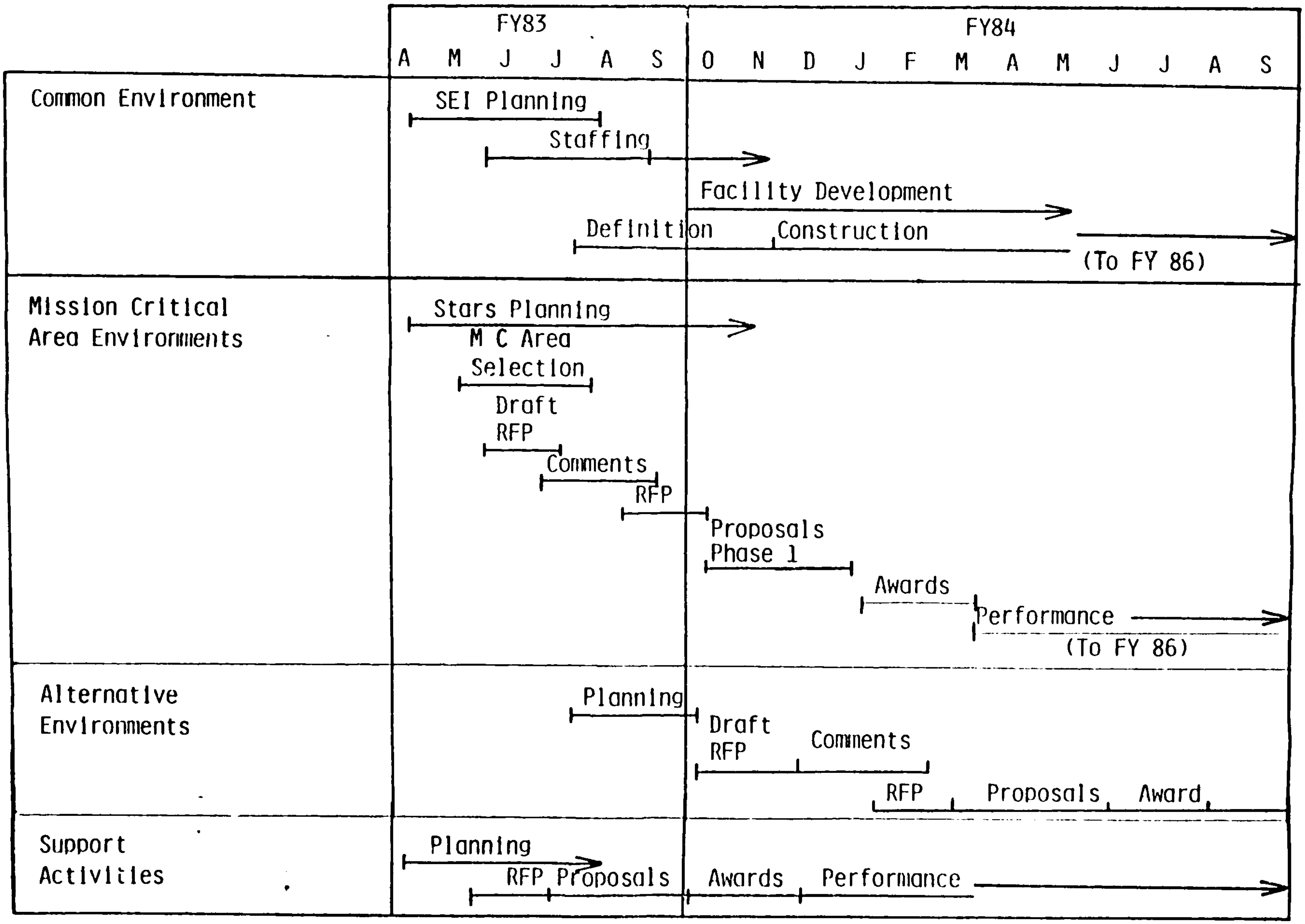


FIGURE 13: Stars Schedule for FY83-84

DAT
FILM