9-1996

Development of a Standard Set of Indicators and Metrics for Artificial Intelligence (AI) and Expert System (ES) Software Development Efforts

Derek F. Cossey

AFIT/GSM/LAS/96S-3

DEVELOPMENT OF A STANDARD SET OF INDICATORS AND METRICS FOR ARTIFICIAL INTELLIGENCE (AI) AND EXPERT SYSTEM (ES) SOFTWARE DEVELOPMENT EFFORTS

THESIS

Derek F. Cossey, Captain, USAF

AFIT/GSM/LAS/96S-3

The views expressed in this thesis are those of the author and do not reflect the official policy or position of neither the Department of Defense nor the U.S. Government.

AFIT/GSM/LAS/96S-3

DEVELOPMENT OF A STANDARD SET OF INDICATORS AND METRICS FOR ARTIFICIAL INTELLIGENCE (AI) AND EXPERT SYSTEM (ES) SOFTWARE DEVELOPMENT EFFORTS

THESIS

Presented to the Faculty of the School of Logistics and

Acquisition Management

of the Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the Graduate Degree in

Systems Management

Derek F. Cossey, Captain, USAF

September 1996

Acknowledgments

I am deeply indebted to Professor Daniel Ferens for his guidance and technical support of my thesis. In addition, he was very understanding and supportive when my wife, Laura, and I found out that she had leukemia. The last six months have been very trying for me and my family, but Professor Ferens provided the encouragement needed to finish this effort. He always maintained his composure and exhibited tremendous patience and poise. In addition, I would like to thank Major Mark Kanko for his assistance and advice on this effort. Near the end, his quick responses aided greatly in completing this effort. His genuine concern for my wife and me during her illness was greatly appreciated.

I would also like to thank the staff at the National Software and Data Repository. They were always available and supportive of my requests. They also exhibited great patience with my typical inability to grasp anything involving software.

I will always remember and appreciate the support and prayers received from fellow students and the AFIT faculty and staff. Everyone at AFIT has been more than willing to help out in any way possible. It has been a great comfort to my wife and me to know that reliable people were always available.

I would finally like to thank my wife for her support during the past fifteen months. She provided the encouragement and motivation that I needed to complete this effort. Her courage and strength during her illness have been an inspiration to me.

Derek F. Cossey

Table of Contents

Page

Page

List of Tables

Abstract

The purpose of this research was to identify a standard set of indicators and metrics that can be used by program managers to improve their abilities to direct development efforts involving Artificial Intelligence (AI) and Expert Systems (ES). This research addresses two objectives. The first objective is to identify an appropriate set of software indicators and metrics to be used by government program offices for the management of development efforts involving software systems for AI and ES. The second objective is to demonstrate how the resources of the National Software Data and Information Repository (NSDIR) can be used in order to minimize the cost of the research endeavor and to demonstrate the value of the NSDIR as an information resource. A literature search identified a set of indicators and metrics that could be used by managers of AI and ES software development efforts. Data concerning AI and ES software development efforts were collected from the NSDIR. Unfortunately, substantiated conclusions regarding the value of the data in regards to AI and ES development efforts were unobtainable. The study did produce a recommended set of indicators and metrics that could serve as a feasible starting point for managers to use in the tailoring process for selecting indicators and metrics for AI and ES software development efforts.

DEVELOPMENT OF A STANDARD SET OF INDICATORS AND METRICS FOR ARTIFICIAL INTELLIGENCE (AI) AND EXPERT SYSTEM (ES) SOFTWARE DEVELOPMENT EFFORTS

1.0 Introduction

1.1 General Issue

The Department of Defense (DoD) has certainly benefited from many of the highly advanced software systems developed over the years. Many of these systems have become essential to the mission of ensuring the United States (US) is adequately defended. For these reasons, the software development efforts managed by DoD organizations play an important role in meeting the objectives of creating and maintaining a technological edge over potential adversaries. In order to maintain a credible and cost-effective global deterrence against having to employ weapons, United States Strategic Command (USSTRATCOM) depends on cutting-edge technological innovations (21:32).

The importance of software and computer related technologies in these new innovations is highlighted by the growing financial commitment to these areas. A study performed by the Electronics Industries Association indicated that the software costs for fiscal year 1994 were expected to be $33 billion, and the software costs for fiscal year 1995 were expected to be $35.7 billion (8:1-3,4).

As processor speeds and memory capabilities increase, the use of a new approach in software is becoming feasible. Artificial Intelligence (AI) represents an attempt for

computers to simulate “thinking.” By mimicking human thought processes, AI systems can perform activities such as teaching, researching, and analyzing in a more effective way than conventional computers. In addition, AI systems can use high speed processors and large knowledge bases to review vast amounts of data and refine information to the appropriate task at hand. In recognition of these capabilities, the DoD is actively involved with developing AI technology.

An important subset of AI is Expert Systems (ES). This particular area has seen large growth in use and development. As such, the DoD is currently involved with the development and application of different types of ES for many types of applications.

1.2 Background

Unfortunately, managers of software development programs, including AI, do not have the management tools required for the adequate assessment and control of the efforts. The lack of indicators to determine the health and progress of a development program prevents managers from making well-informed decisions and limits the insight of a program office. For example, indicators for conventional programs, such as Source Lines of Code (SLOC), computer resource utilization, memory utilization, and cost performance data, provide information for managers in the areas of software performance, testing, reliability, cost, and schedule. Because such indicators do not exist for AI development programs, poor decisions, based on lack of information or misinformation, lead to cost overruns and schedule slips. Also, because AI methods differ from

conventional software methods and techniques, existing indicators for conventional programs may not be effective for AI efforts.

Work is currently being conducted to develop appropriate indicators for AI systems; however, they have not matured to a state that is useful for program managers in charge of AI software development efforts. An effective set of indicators needs to be identified for DoD managers and engineers to use in the management of AI development efforts.

1.3 Problem Statement

The purpose of this research is to identify a standard set of metrics and indicators that can be used by program managers to improve their abilities to direct development efforts involving AI and ES.

1.4 Research Objectives

This research addresses two objectives. The first objective is to identify an appropriate set of software indicators and metrics to be used by government program offices for the management of development efforts involving software systems for AI and ES. Most of the focus for the first objective will center upon ES. The second objective is to demonstrate how the resources of the National Software Data and Information Repository (NSDIR) can be used in order to minimize the cost of the research endeavor and to demonstrate the value of the NSDIR as an information resource. Accomplishment

of the second objective will provide an example of how the NSDIR can be used to provide information and insight for current and future software development efforts of all types.

2.0 Literature Review

2.1 Software Effort Estimation as a Management Tool

The importance of software effort estimation has prompted a great deal of research in the past few years (20:126). As computers continue to pervade more facets of society, the complexity and size of software systems associated with computer technology has increased at a phenomenal rate. When a software program is developed and managed properly, the result of that development enhances the user's ability to be more productive. However, when the effort is not developed and/or managed properly, the result may lead to a waste of time and money. When considering software engineering efforts, accurate estimation of software development is critical. If a manager estimates too low, the software development team may be under pressure to complete the project quickly. The resultant product may contain errors that require corrections in the future. If the manager estimates too high, excessive resources may be committed to the software development project (20:126). Ultimately, poor management may lead to a number of undesirable situations that may include cost overruns and inefficient resource allocations. Some problems leading to the increased software costs have been technical; however, most have been managerial (8:1-3).

2.2 Advent of Artificial Intelligence and Expert Systems

AI is becoming a dominant force in the technological race towards the future. Telecommunications have grown to a point where equipment, technology, and procedures

change too quickly to keep pace with customer demands. As the usage of AI grows, ES are evolving as tools that will allow people to keep up with the ever increasing rate of change (12:11). The application of ES has been used effectively in a number of situations as replacements for human experts. ES have been developed to support many applications. These include engineering, medicine, chemistry, and business. These systems are able to identify elements of expertise and distribute them throughout an organization. ES also contribute to reduced cost, improved quality, reduced downtime, increased and consistent output, flexibility, and the ability to work with incomplete or uncertain data (16:245). AI is not a new idea or subject; however, it is only now beginning to branch out of the academic world and into the business community. The evolution of AI has been facilitated by the ability of computer technology to accommodate it through increased processor speed and expanded memory capability (13:71).

By conducting a review of the literature that focuses on AI and its potential uses in society, one finds that numerous sources discuss the current benefits and future potential of AI (1,3,4,5,11,12,13,16, and 19). Technologies involving AI will open new horizons of opportunity. AI programs will handle data and develop information in a way that has never been done before. The DoD has recognized this growth potential and is aggressively pursuing it through development of systems that utilize this new technology.

2.3 Definition of Expert Systems

ES can be defined by comparing them to conventional programming systems. Basically, ES manipulate knowledge while conventional programs manipulate data. In

order to be classified as an expert system, the system needs to demonstrate the characteristics of expertise, symbolic reasoning, depth, and self-knowledge. The components of ES consist of an induction capability, knowledge base, inference engine, and user interface. An ES must be able to apply its knowledge to produce solutions as efficiently and effectively as humans would (22:16).

Distinguishing between efficiency and effectiveness is an important key to understanding the difference between conventional computers and artificial intelligence. Most conventional computer systems justify their cost on the basis of efficiency. Most ES justify their cost only on the basis of effectiveness (19:56). In dealing with management issues, ES have demonstrated the advantages of improved productivity, increased personnel consistency, and improved competitiveness, and have reduced staff personnel as a result of providing automated decision making (5:112).

2.4 Lack of Management Tools for AI and ES Development Efforts

Management of AI and ES development efforts is a relatively new area of concern when compared to the years of development for more traditional software, based primarily upon data manipulation discussed in the paragraphs above. As hardware technology has advanced to a point where AI and ES technologies are more executable, management challenges arising from the development of these efforts have been realized. The new considerations relating to the these efforts should be evaluated with additional studies.

When considering the aggregate of software development activity, the current software crisis that involves problems with the software development process has had a

significant impact on the Air Force. Cost overruns and schedule slips have defined the crisis. In addition, software size estimates are growing at an almost uncontrollable rate (2:1-2). Jones supports this further by stating, "Although most companies could not survive or compete successfully without software and computers, senior executive management remains troubled by a set of chronic problems associated with software applications: long schedules, major cost overruns, low quality, and poor user satisfaction (15:1)." In Guidelines for Successful Acquisition and Management of Software Intensive Systems, the authors provide the following, "Unfortunately, ..., software-intensive systems management in the DoD and the Air Force has been plagued with similar problems. Software-intensive systems, especially software, have not always been properly managed. Software functionality ... has experienced performance shortfalls as well as cost and schedule overruns (8:1-1)."

As a result of some of these negative experiences, other DoD agencies have taken a closer look at how software systems are developed. Close examination led directorate software engineers within USSTRATCOM to discover that software processes were undefined and were contributing to unnecessary waste, duplication of effort, and unacceptable amounts of rework (21:32). This seems to suggest that opportunities for improvement in the area of software management exist.

In the specific area of ES, opportunities for improvement seem to exist, also. One aspect to consider involves determining whether or not resources expended on ES have had any useful benefit. The DoD has spent millions of dollars on decision support and

expert system technology with limited transfer of the systems to operational personnel (1:4).

Despite the recent trend within the DoD acquisition community to remove military specifications and standards as requirements that are levied upon contractors of DoD acquisition efforts, the senior leadership within the DoD realizes that standards in some form will be needed for program managers to successfully manage acquisition efforts. In his Memorandum for Secretaries of the Military Departments, dated 29 June 1994, Dr. William J. Perry supports the use of "performance and commercial specifications and standards in lieu of military specifications and standards (18:1)." Special attention has been given to the software acquisition area by Ms. Darleen A. Druyun, Deputy Assistant Secretary of the Air Force (Acquisition), in her memorandum, dated 16 February 1994. She states,

> Software metrics shall cover cost, schedule, and quality. The metrics will be further subdivided to portray the following 'core' attributes: Size, Effort, Schedule, Software quality, and Rework. The collection, analysis, and use of metrics for the above core is mandated for software-intensive systems and is strongly encouraged for all others. (17:1)

These statements by senior DoD leaders indicate that even though military specifications and standards are not the preferred method for enforcing requirements during a developmental contract, some form of standards for software metrics will be needed for managers during the performance of their management functions.

Development efforts for ES still need further refinement. There are no industry standards for the infrastructure of ES. In addition, there is no widely accepted systems development life cycle for ES applications. Due to the lack of standards and development

methods, companies have incurred increased systems development costs and risks (11:54). Developers of one of the first ES developed found that the ability of the program to make a good decision did not guarantee that the system would be accepted by the users (4:23). Developers of ES technologies have not resolved key integration issues that include access to databases, interface issues with external systems, portability across hardware platforms, and operations within network-intensive computing environments (11:54).

As the evidence suggests, the creation of a standard set of indicators that can be used in the management of AI software development efforts that includes ES is needed. In consideration of the growing investment in ES technology, the impressive results organizations are achieving with ES, and the potential and versatility for ES, it is imperative that research on the use and on the management of this technology be expanded quickly (23:247). With the establishment of a standard, DoD program managers of AI software development efforts will be able to utilize resources in a more effective manner and thereby provide improved products to their customers. Once widely accepted software standards and new ES tools are available, a systems development life cycle can be followed to reduce development complexities, costs, and risks (11:55). Although ES are extremely difficult to develop, evaluation methods specifically tailored for decision support systems and ES can be used to increase the success rate of programs managing these efforts (1:1).

Currently, the majority of literature on ES does not address how the systems benefit the users. Most of the literature only focuses on the technological aspects of ES, how ES work, and the steps required to build ES. More work is needed to understand

and explain how the use of an expert system will lead to benefits experienced by users (3:99). Even with the widespread use and increasing importance of ES technology, little effort has been made to systematically identify and empirically test the determinants of ES success by users (23:227).

2.5 Review of Software Metrics

Many software metrics and almost as many opinions on how to apply the metrics exist. This review primarily addresses metrics typically associated with DoD program from a general perspective. Any specific discussion and application of software measurement metrics should usually be limited to a specific software development effort.

2.5.1 General Discussion of Software Metrics.

Managers of software development programs should ask many questions concerning the progress of their efforts. The use of management tools should be employed to help identify and quantify various aspects surrounding the software development effort. Once the important issues and data have been identified, a framework for evaluation needs to be established.

One example of a framework is provided by Practical Software Measurement: A Guide to Objective Program Insight. Software measurement principles provide the foundation for applying software measurement to the areas of program management, product engineering, and process improvement (14:12). As a flexible process, software measurement helps a program manager to succeed (14:9-10).

In Practical Software Measurement, software measurement provides objective software information to enable the program manager to communicate effectively throughout the program organization, identify and correct problems early, make the key tradeoffs, track specific program objectives, and defend and justify decisions (14:8-9). The six areas of common software issues covered by the guide are schedule and progress, resources and cost, growth and stability, product quality, development performance, and technical adequacy (14:14). Since all software development efforts are unique to some degree, the specific program issues and objectives need to drive the measurement requirements and appropriate software metrics for the development effort (14:13). The following table provides a listing of the indicators listed in Practical Software Measurement.

Table 2-1.
Software Metrics from Practical Software Measurement

Issue	Category	Measure
Schedule and Progress	Milestone Performance	Milestone Dates
	Work Unit Progress	Components Designed
		Components Implemented
		Components Integrated & Test
		Requirements Allocated
		Requirements Tested
		Test Cases Completed
		Paths Tested
		Problem Reports Resolved
		Reviews Completed
		Changes Implemented
	Schedule Performance	Schedule Variance
	Incremental Capability	Build Content - Content
		Build Content - Function
Resources and Cost	Effort Profile	Effort
	Staff Profile	Staff Level
		Staff Experience
		Staff Turnover
	Cost Performance	Cost Variance
		Cost Profile
	Environment Availability	Resource Availability Dates
		Resource Utilization
Growth and Stability	Product Size and Stability	Lines of Code
		Number of Components
		Words of Memory
		Database Size
	Functional Size and Stability	Requirements
		Function Points
	Target Computer Resource	CPU Utilization
	Utilization	CPU Throughput
		I/O Utilization
		I/O Throughput
		Memory Utilization
		Storage Utilization
		Response Time
Product Quality	Defect Profile	Problem Report Trends
		Problem Report Aging
		Defect Density
		Failure Interval
	Complexity	Cyclomatic Complexity
Development Performance	Process Maturity	Capability Maturity Model Level
	Productivity	Product Size/Effort Ratio
		Functional Size/Effort Ratio
	Rework	Rework Size
		Rework Effort
Technical Adequacy	Technology Impacts	Program Defined Measures

The above Table is an example of a comprehensive list of measures. Typically, a measurement program for a software development effort will be developed by tailoring a list of measures such as the list presented above. The tailoring effort should ensure adequate visibility into the effort while minimizing the costs of collecting the measures.

In his book Controlling Software Projects, DeMarco states that managers cannot control what they cannot measure. For most situations, the connection between measurement and control is taken for granted (7:3). DeMarco continues by saying:

> Staying in control means making sure that results match up to expectations. That requires two things: (1) You have to manage the project so that performance stays at or above some reasonable and accepted standard. (2) You have to make sure that original expectations are not allowed to exceed what's possible for a project performing at that standard. (7:5)

The above statement provides overall advice regarding the management of any endeavor. Proper planning and estimating are important for successfully controlling a project.

More specifically, the input required to improve planning estimates for software efforts is a useful set of metrics. The selected set of metrics should provide quantifiable indications of scope, quality, complexity, and performance (7:17). A useful metric will be measurable, independent, accountable, and precise (7:50). DeMarco continues by stating, "... a metric is a measurable indication of some quantitative aspect of a system. For a typical software endeavor, the quantitative aspects for which we most require metrics include scope, size, cost, risk, and elapsed time (7:49)."

DeMarco makes an interesting comment concerning situations where proper goals and metrics are not established. "Delivery in the shortest possible time" will become the default goal of the project team if no other goals are specified. Team members will

sacrifice all other goals to optimize the most important one (7:59). People involved with managing any effort know the potential pitfalls of allowing the schedule to drive a technical development project.

In his book Applied Software Measurement, Jones provides an additional perspective into the realm of software metrics. "The objective of applied software measurement is insight. We don't measure software just to watch for trends; we look for ways to improve and get better (15:185)." The practice of applied software measurement ensures that accurate and meaningful information is collected and that it is of practical value to software management professionals (15:1-2).

Jones suggests that three kinds of data are essential to software measurement. They are hard data, soft data, and normalized data (15:4-5). Hard data is quantifiable and involves little or no subjectivity. Soft data is used to evaluate human opinions. Normalized data is used to determine whether or not a project is above or below the normal in terms of productivity and quality (15:8). One area that Jones supports is the use of function points in the measuring of software development progress. Function-based metrics are becoming more prevalent in software measurement activities (15:9).

Jones continues by stating his vision of a software measurement program. "A fully applied software measurement system for an entire corporation or government agency is a multifaceted undertaking that will include both quality and productivity measures and produce both monthly and annual reports (15:23)." Jones provides another interesting commentary when he states, "Only when software engineering is placed on a base of firm metrical information can it take its place as a true engineering discipline rather than an

artistic activity, as it has been for much of its history. Measurement is the key to progress, and it is now time for software to learn that basic lesson (15:40-41)." Evidently, he believes that software development efforts have significant room available for improvement.

One reason that software metrics have not been accepted wholeheartedly by software professionals can be attributed to an underlying fear associated with metrics. A "cultural resistance" exists towards software measurement because software professionals fear that the measures will be used against them (15:2). "What causes the transition from apprehension to enthusiasm is that a well-designed applied measurement program is not used for punitive purposes and will quickly begin to surface chronic problems in a way that leads to problem solution (15:29)." When managers introduce a new measurement program, they must ensure that the program is accepted and understood by the employees involved with the effort.

Jones makes an interesting observation about people who become involved with software measurement efforts. "The managers and technical staff workers who embark on a successful measurement project are often surprised to find permanent changes in their careers. There is such a shortage of good numerical information about software projects and such enormous latent demand by corporate executives that, once a measurement program is started, the key players find themselves becoming career measurement specialists (15:32)." This statement supports the need for a resource where information and data about software development efforts can be obtained. It also implies that the software measurement field is still in its early stages.

2.5.2 Role of SEI's Capability Maturity Model (CMM).

In an attempt to improve the ability to assess the software development potential of a company, the DoD sponsored an effort with the Software Engineering Institute (SEI) to develop a methodology to evaluate the ability of a company to produce software. The eventual product of this effort was the Capability Maturity Model (CMM) (9:28). When using the CMM, firms that produce software can be classified into one of the five levels categorized as initial, repeatable, defined, managed, and optimizing with the last category being the most mature. In July 1994, the SEI reported that generalized results representing the software producing community indicated that less than one percent of the firms can be classified as "managed," which is the fourth level (10:5). The same results indicated that seventy-five percent of the existing software-producing firms remain at the "initial" level of the CMM. The generalized results are based on assessments conducted by the SEI and licensed vendors of SEI's Applied Software Measurement Course (10:5). These indications imply that software producing firms have the potential to improve their development capabilities, and the proper selection and application of metrics is one way to accomplish this goal.

2.5.3 Air Force Vision for Software Development.

The vision for software will be implemented along a two-fold approach. The first facet involves implementing sound software engineering practice throughout all software development programs sponsored by the Air Force. The second facet is the establishment of an Air Force software infrastructure (8:1-6). "The Air Force vision for software is to

continuously improve software quality through efficient application of software engineering discipline (8:1-6)." Successfully managed efforts are measured in terms of cost, schedule, performance, supportability, and quality. Quality software and process improvements cannot be realized without measurement (8:1-9).

In Guidelines for Successful Acquisition and Management of Software Intensive Systems, a distinction is made between indicators and metrics. "With indicators, the requirement for determining a relationship between the value of the indicator and the value of the software characteristic being measured is substantially relaxed. A reliability indicator, for example, will not describe an anticipated value of reliability (8:4-31)." Indicators are used to show trends in a software development effort. "Useful insights can be drawn from management indicators because they are derived from readily available data and do not require significant investment in resources or imposition on existing processes (8:4-31,32)."

Metrics are direct measures of a software product that is embedded in a hierarchy of relationships connecting the metrics with the software characteristics being measured (8:4-32). "Metrics are quantifiable indices used to compare software products, processes, or projects or to predict their outcomes (8:9-11)." This definition of a metric implies that metrics are given a more rigorous treatment than indicators.

Resourceful managers depend upon objective and subjective data to gain an accurate assessment of a software development effort (8:4-31). Objective data can be independently verified. Subjective data are based on the feelings and attitudes of people

involved with a software development effort (8:4-31). Many different forms of data must be used in the successful management of software development efforts.

2.5.4 Potential Software Metrics for AI and ES Development Efforts.

AI development efforts, specifically ES efforts, possess unique aspects that require special attention when compared to an inductive logic software applications. The aspects involved with developing the induction capability, knowledge base, inference engine, and the user interface require unique metrics to adequately assess the development of ES. In determining the appropriate metrics, a program manager must ensure that data about a system under development are gathered, filtered, and aggregated in order to test the hypothesis that all is going well. The purpose of this function is to determine whether or not the ES will do what decision makers and/or other members of the sponsoring team want it to do (1:15).

Evaluation is often a forgotten step in developing expert systems. As a result, managers lose the opportunity to gain valuable information about what potential users think about the system, how the code is written, and the extent of how the system performs once it is implemented (1:5). Evaluation is a critical link in the application of a requirements-driven development cycle because it provides the information that keeps the development process on track (1:5).

One ES evaluation approach suggested by Adelman focuses on the three categories of technical, empirical, and subjective evaluation (1:15). Adelman continues:

> The technical evaluation facet looks 'inside the black box.' The empirical evaluation facet assesses the system's effect on performance. The subjective evaluation facet obtains users' opinions regarding system strengths and weaknesses. (1:15)

Metrics associated with each facet should be selected and tailored for each specific development effort.

2.6 Summary

The importance of software in future applications for the DoD will continue to grow; however, current software management processes may not be adequate for successfully controlling the development of software for these applications. Along with the increased importance of all types of software used for defense applications, AI and ES technologies will play a larger role in performing mission essential duties. As the DoD learns to do more with less, the ability to properly manage the development efforts will become critical to achieve success.

A key to successful software management will involve using appropriate metrics and indicators to ensure that the final product meets the user's needs while avoiding significant cost and schedule overruns. Several methods, techniques, and indicators for managing software development efforts already exist for managers to employ in the management of software development efforts. Whether or not these resources and techniques are used effectively on a daily basis by managers in the field is debatable. The literature suggests that software development managers, commercial and government, have a great deal to learn concerning the proper management of software development efforts. The development of software needs to graduate from an art form into an

engineering discipline. Efforts are being made and progress has been seen; however, many opportunities to improve the management of software development efforts exist.

3.0 Methodology

3.1 Introduction

The development of a successful set of indicators and metrics will be attempted by identifying existing indicators and metrics used successfully for conventional software programs, reviewing current literature on AI and ES development to find potential indicators and metrics, and evaluating selected indicators and metrics from a database representing AI and ES development efforts. Previous work identifying indicators and information that managers considered important in the management of software development for conventional programs has been performed and presented in an Air Force Institute of Technology (AFIT) thesis by Ayers and Rock (2:1-3). The existing indicators for conventional software development efforts will be analyzed for their applicability towards AI programs.

3.2 Exploratory Phase

An evaluation of existing indicators specifically targeted at AI and ES programs will be conducted to identify what currently exists for both software areas. ES can be validated in terms of face validity, objectivity, reliability, and economics. Face validity compares the performance of the ES to a human expert. Objectivity reduces bias by comparing the developed ES to a group of independent human experts. Reliability of ES can be measured in terms of the stability of the system and its ability to generate identical solutions given identical input data. Economics is determined by evaluating the cost effectiveness of the system (11:60). One portion of the research will investigate whether

or not current indicators for AI and ES development efforts exist. Once determined through a literature review and/or survey analysis, the ability of the indicators to provide useful information to managers could be tested using data from the National Software Data and Information Repository (NSDIR).

3.3 NSDIR Background Information

The NSDIR evolved in response to a community-defined need. In August 1993, over 60 leaders from industry, academia, and government participated in the first Software Measurement

Workshop to discuss national-level software challenges. As a result, an agreement was established to develop a strategy to create an NSDIR and to develop a blueprint to create a national software council (6:3). The NSDIR is building a profile of the DoD software efforts that will help managers and executives within DoD and industry find timely and accurate answers to questions arising from software development projects. The primary goal of the NSDIR is to provide a tool to establish baselines and benchmarks for managers of software efforts to use in evaluating and assessing new developmental projects (6:2).

3.3.1 Basic Functions of NSDIR.

The NSDIR exists to support two basic functions: maintain a basic repository capability and provide information analysis capabilities (6:3). The repository capability includes collection, storage, management, and retrieval of software measurement data.

Information analysis capabilities are provided through tools available from NSDIR and the user support desk.

In addition, the National Software Data and Information Repository (NSDIR) has collected data on eighty-eight software development efforts sponsored by government agencies. NSDIR has collected information, which will be used for this research, on eight AI/ES software development efforts. The existence of the NSDIR provides a cost effective resource to test the identified indicators. An important objective of the research effort will attempt to demonstrate the value of using the NSDIR as a planning and development resource for DoD program managers. Capitalizing on the lessons learned from previous work will be critical to effectively developing weapons and associated support systems in affordable and effective ways.

3.3.2 Categories of Data Collected by NSDIR.

Currently, the top level categories of data collected by NSDIR include information concerning profile, cost, schedule, size, effort, and quality for the individual software programs maintained by NSDIR. Each of the top level categories contains several subcategories that provide more insight to the lower levels of detail. A primary limitation of this data is that NSDIR is primarily a passive player in the determination of the data that is collected and provided to its database. The individual program offices managing the software development efforts determine the data categories and amount to be collected.

3.4 Framework for Metrics

The thrust of the methodology is to assume the role of a project manager in charge of a software development effort for AI with primary emphasis on ES. A project manager needs to identify and define measurement metrics to be used during the development process. The NSDIR will provide the data required to learn more about ES development from ES programs that have already been developed. Also, the data obtained from the NSDIR will be used to verify the usefulness of the selected metrics.

3.4.1 Metric Selection for Analysis of AI/ES NSDIR Data.

The three sources, Practical Software Measurement (14:1), Evaluating Decision Support and Expert Systems (1:1), and Development of a Standard Set of Software Indicators for Aeronautical Systems Center (2:1), are the basis for selecting the indicators and metrics for this analysis. In Practical Software Measurement, the focus of the measurement metrics centers on answering management questions concerning software development while the measurement metrics in Evaluating Decision Support and Expert Systems focuses on specific AI and ES development issues. The third source, Development of a Standard Set of Software Indicators for Aeronautical Systems Center by Ayers and Rock, focuses on indicators that are considered valuable by government managers and engineers involved with software development efforts. Table 3-1 contains the indicators selected by Ayers and Rock in their thesis.

Table 3-1
Identified Indicators (2:5-6)

Area	Indicators
Requirements	CSCI Requirements Stability
	CSCI Design Stability
Software Performance	I/O Bus Throughput Capability
	Processor Throughput Utilization
Schedule	Requirements Allocation Status
	Preliminary Design Status
	Detailed Design Status
	Code and Unit Test Status
	Integration Status
Cost	Man Months of Effort
	Software Size

Table 3-2 was selected by the author based upon the literature review to provide a balance between information suitable for management and technical concerns. The two sources of Practical Software Measurement and Evaluating Decision Support and Expert Systems were used to develop a combined metrics table for use with this research. Also, an evaluation of the NSDIR database influenced the selection of the metrics. The selected metrics are listed in Table 3-2 below:

Table 3-2
Identified Metrics

Issue	Category	Measurement Metric
Schedule and Progress	Milestone Performance	Milestone Dates
	Schedule Performance	Estimated Duration/Size
		Actual Duration/Size
		Schedule Performance
Resources and Cost	Effort Profile	Estimated Effort/Duration
		Actual Effort/Duration
		Estimated Effort for S/W Type
	Cost Performance	Cost Variance
		Cost Profile
Growth and Stability	Product Size and Stability	Lines of Code (LOC)
		Knowledge Base Size
	Functional Size and Stability	Requirements
		Function Points
Product Quality	Defect Profile	Problem Report Trends
		Failure Interval
Development Performance	Productivity	Product Size/Effort Ratio
		Functional Size/Effort Ratio
	Rework	Rework Size
		Rework Effort
AI/ES to User Interface	Application Characteristics	User Interface
		Types of Data Files
		Expert Judgment Stored
		Ability to Modify Judgment
AI/ES to Organization Interface	Efficiency Factors	Speed
		Reliability
		Supportability
Org Environment Interface	Potential for Implementation	Decision Quality
		Technical Soundness
		Decision Process Quality

The last three issues in Table 3-2 relate specifically to AI and ES software development efforts. An important aspect of these last three issues involves the interface of the end product to the user. Special care should be taken to ensure that the user is able to interact with the system.

3.4.2 Method Design of Applying of Metrics to NSDIR Database.

Each database for the AI and ES programs stored at the NSDIR will be evaluated and compared to the metrics outlined in the above table. The goal of the procedure is to determine a suitable set of metrics that will serve as a useful program management function. Efforts will be made to ensure that the metrics and database items will be used to support decision-making activities that are typical to a program office involved with the development of a new AI or ES applications.

By reviewing what program offices currently use or have used in software development efforts for AI and ES efforts, the evaluation effort will determine what program offices consider important metrics and indicators in their measurement effort. In addition, the metrics and indicators that exist in the NSDIR database from past and on-going AI and ES development efforts will be evaluated against the recommended metrics, Table 3-2, from experts specifically involved with AI and ES development efforts. From this effort, the value of the NSDIR will be evaluated in regards to the availability of useful data for analyzing AI and ES development efforts.

3.5 AI/ES Programmatic Data Available from NSDIR Database

The organization of this section is designed to show what data were available concerning AI and ES software programs. After searching the database for all of the pertinent data, the information relating to AI and ES efforts were organized into the tables discussed below. Most of the data available relates to qualitative aspects. Quantitative data concerning the efforts was limited.

The NSDIR contains data on six programs relating to ES and two programs relating to AI. Because the NSDIR does not have any direct control over the data and format of the collected data, the categories and amount of collected data varies among the eight programs. PROJ_NO refers to the tracking number used internally by the NSDIR to track the program and its associated data. SW_TECH describes the software technologies linked to the program. PROJ_MLST identifies the most recent milestone of the program at the time of data submission. The security classification program is identified in the column labeled PROJ_SEC. The column labeled STD_CTRT indicates if standards were incorporated as part of the contract or as guidance accompanying the contract. An "A" designation indicates that standards were put on contract as binding requirements, and a "B" designation indicates that the standards were used only as guidance.

Table 3-3
Programmatic Data

PROJ_NO	SW_TECH	PROJ_MLST	PROJ_SEC	STD_CTRT
1000	ES	INSTALLATION	TOP SECRET	B
1007	ES	DEMONSTRATION	NONE	
1022	AI	SDR	NONE	A
1027	AI	IV	SPECIAL ACCESS	A
1058	ES	IV	NONE	A
1098	ES	SYSTEM FUNC RVW	NONE	A
1211	ES	CDR	CONFIDENTIAL	
1218	ES	NA	SPECIAL ACCESS	

In Table 3-4 below, additional top level data are provided about the eight programs under consideration. PROJ_NO was described in the previous paragraph. ORG_NAME describes the service or business sector of the organization. Of the eight programs, only one program is not managed by the AF; however, the NSDIR is currently

pursuing other organizations besides the AF. The next column, AF_ORG, identifies the AF organization responsible for the program. AETC represents the Air Education and Training Command, and AFMC represents the Air Force Material Command. ORG_NEW identifies the percentage of dollars spent for new organic (in-house) development, and ORG_MNT identifies the percentage of dollars spent for organic maintenance. CTR_NEW identifies the percentage of dollars spent by the contractor for new development, and CTR_MNT identifies the percentage of dollars spent by the contractor for maintenance. PM_CTRT and BB_CTRT identifies the percentage of dollars spent on the prime contractor and big business subcontractors, respectively. SB_CTRT refers to the percentage of dollars spent by a subcontractor categorized as a small disadvantaged business or minority owned. SB_NCTRT identifies the percentage of dollars spent by a subcontractor categorized only as a small business. NP_CTR refers to the percentage of dollars spent by a non-profit organization.

Table 3-4
Top Level Data

PROJ NO	ORG NAME	AF ORG	ORG NEW	ORG MNT	CTR NEW	CTR MNT	PM CTRT	BB CTRT	SB CTRT	SB NCTRT	NP CTR
1218	AF	AETC	66	22	10	2	10		90		
1211	AF	AETC	66	22	10	2	10		90		
1098	AF	AFMC	0	0	100	0	26	75			
1058	AF	AFMC	0	30	100	70	100				
1027	AF	AFMC		15	60	25	88			4	8
1022	AF	AFMC			100		100				
1007	AF	AFMC		100							
1000	NON GOV'T										

Table 3-5 provides more descriptive data about the AI and ES programs that have contributed information to the NSDIR. PROJECTDESC describes the development program, and PROJECTYPE identifies the type of software development project.

DEVAPPRCH provides the developmental approach used to manage the software development effort. ORG identifies the organization that managed the effort. TOT_PER_YRS identifies the number of person years required for the project, and TOTSLOC identifies the total source lines of code (SLOC) developed under the program.

Table 3-5
Descriptive Data

PROJ NO	PROJECTDESC	PROJECTTYPE	DEVAPPRCH	ORG	TOT_PER YRS	TOTSLOC
1218	NEW SYSTEM	MIS		AF		
1211	NEW SYSTEM	SYSTEMS S/W	INCREMENTAL	AF		
1098	ENHANCEMENT	AVIONICS	WATERFALL	AF	16	2770
1058	RE-ENGINEERING	AVIONICS	PROTOTYPING	AF		
1027	OTHER	WEAPONS SYS		AF		
1022	NEW SYSTEM	TRAINING	PROTOTYPING	AF	9	50000
1007	ENHANCEMENT	OTHER	PROTOTYPING	AF		
1000	ENHANCEMENT	COMM	INCREMENTAL	NON GOV	2	

The data in Tables 3-3, 3-4, and 3-5 describe top level characteristics of the programs. These data show that AI and ES are developed for a variety of reasons with different approaches to the development technique. The varied mixture of development efforts highlight the need to develop a good baseline of indicators and metrics that can be tailored accordingly for AI and ES efforts.

3.5.1 Qualitative Data.

Table 3-6 provides even further detail concerning the content, tools, and methods used in the development efforts. Reading horizontally, the top row of the table identifies the project number and its associated technology. In Table 3-6, the data represent project

1000 which is an expert system. The remaining data is identified by columns with appropriate headings such as SW_TECH and SW_MTDS. SW_TECH describes the software technologies associated with the development effort. A program may have several software technologies involved with it. SW_MTDS describes the software development methods used to develop this particular expert system. For this project, several development and testing methods were used. SW_TOOLS describes the software tools used in the development effort. DOC_TYPE describes the types of documentation that were prepared in support of this effort.

Table 3-6
Project 1000

PROJ_NO	1000	ES	
SW_TECH	SW_MTDS	SW_TOOLS	DOC_TYPE
ES	BLACK/WHITE BOX TESTING	WORD PROCESSING	DESIGN DOCUMENT
SECURITY	REQUIREMENTS TRACING	SCHEDULING	ACCEPT TEST PLANS
CLIENT/ SERVER	PROTOTYPING	CONFIG MGT	INTEGRATION TEST PLANS
	REGRESSION TESTING	PROJECT MANAGEMENT	PROGRAMMER MANUALS
	RAPID APPLICATION DEVELOPMENT		PROJ MGT PLANS
	INTEGRATION TESTING		REQMENTS DOCUMENTS

Table 3-7 describes project number 1007, which is another expert system. This particular program did not seem to involve as many elements as the program described above. However, the information provided does provide a view of the programmatic effort.

Table 3-7
Project 1007

PROJ_NO	1007	ES
SW_TECH	SW_MTDS	SW_TOOLS
INTERACTIVE	REQUIREMENTS TRACING	DATABASE
CLIENT/SERVER	PROTOTYPING	
ES		
SECURITY		
DATABASE		

In Table 3-8, a new category is provided. SW_TOOLS_OTR describes tools that are not included in the pre-defined values for SW_TOOLS. With this designator, the program office can identify software tools used in the development of the system that are unique to the effort. For this program, certain commercial software tools were used.

Table 3-8
Project 1022

PROJ_NO	1022	AI		
SW_TECH	SW_MTDS	SW_TOOLS	SW_TOOLS_OTR	DOC_TYPE
INTERACTIVE	Process Modeling	Code Generators		Requirements Documents
MULTI-MEDIA	Object-Oriented Design	Word Processing		DESIGN DOCUMENT
AI	PROTOTYPING	SCHEDULING		UNIT TEST PLANS
GRAPHICS		OTHER	OBJECT WORKS	ACCEPT TEST PLANS
		OTHER	ENVY DEVELOPER	Project Management Plan
		OTHER	RATIONALE BASE	SEMS
		OTHER	NEOPUS	Data Accession List
				Student Workbook
				Administrators Guide
				CWBS
				Presentation Material

Tables 3-9, 3-10, 3-11, 3-12, and 3-13 provide more of the same data for the remaining programs under consideration for this research effort. The column names of all of the categories have been described in previous paragraphs.

Table 3-9
Project 1027

PROJ_NO	1027	AI	
SW_TECH	SW_MTDS	SW_TOOLS	DOC_TYPE
BATCH PROCESS	PROCESS MODELING	CONFIGURATION MANAGEMENT	ACCEPTANCE TEST PLANS
INTERACTIVE	OBJECT-ORIENTED ANALYSIS	CODE GENERATORS	DESIGN DOCUMENT
CLIENT/ SERVER	DATA MODELING	COMPLEXITY	INTEGRATION TEST PLANS
AI	OBJECT-ORIENTED DESIGN	COST ESTIMATING	PROG MANUALS
REAL-TIME	INTEGRATION TESTING	DATABASE	PROJECT MGT PLANS
HIGH RELIABILITY	BLACK/WHITE BOX TESTING	DEBUGGERS	REQMENTS DOCUMENTS
SECURITY	REQUIREMENTS TRACING	DESIGN/CASE	UNIT TEST PLANS
DATABASE	PROTOTYPING	FLOWCHARTING	TECH ORDERS
OTHER	STRUCTURED ANALYSIS	METRICS	USER MANUAL
	EVENT MODELING	PROJECT MANAGEMENT	
	STRUCTURED DESIGN	SCHEDULING	
	REGRESSION TESTING	TEST GENERATORS	
	OTHER	WORD PROCESSING	

Table 3-10
Project 1058

PROJ_NO	1058	ES	
SW_TECH	SW_MTDS	SW_TOOLS	SW_TOOLS_OTR
INTERACTIVE	INTEGRATION TESTING	PROJ MANAGEMENT	
ES	BLACK/WHITE BOX TESTING	DATABASE	
REAL-TIME	STRUCTURED ANALYSIS	WORD PROCESSING	
HIGH RELIABILITY	STRUCTURED DESIGN	FLOWCHARTING	
	REGRESSION TESTING	METRICS	
		SCHEDULING	
		DEBUGGERS	
		CONFIGURATION MGT	
		OTHER	AVIONICS SIM TEST SET

Table 3-11
Project 1098

PROJ_NO	1098	ES
SW_TECH	SW_MTDS	SW_TOOLS
INTERACTIVE	INTEGRATION TESTING	PROJ MANAGEMENT
ES	REQUIREMENTS TRACING	DATABASE
REAL-TIME	STRUCTURED ANALYSIS	WORD PROCESSING
DATABASE	STRUCTURED DESIGN	FLOWCHARTING
	REGRESSION TESTING	SCHEDULING
	MULTI-FUNCTIONAL TEAMS	TEST GENERATORS
		DEBUGGERS
		COST ESTIMATING

Table 3-12
Project 1211

PROJ_NO	1211	ES	
SW_TECH	SW_MTDS	SW_TOOLS	DOC_TYPE
BATCH PROCESS	BUSINESS AREA MODELING	PROJECT MANAGEMENT	ACCEPT TEST PLANS
INTERACTIVE	PROCESS MODELING	CODE GENERATORS	DESIGN DOCUMENT
ES	DATA MODELING	DATABASE	INT TEST PLANS
REAL-TIME	REGRESSION TESTING	DESIGN/CASE	PROGRAMMER MANUALS
DATABASE	BLACK/WHITE BOX TESTING	WORD PROCESSING	PROJECT MGT PLANS
HIGH RELIABILITY	REQUIREMENTS TRACING	FLOWCHARTING	REQMENTS DOCUMENTS
	STRUCTURED ANALYSIS	COST ESTIMATING	UNIT TEST PLANS
	STRUCTURED DESIGN	TEST GENERATORS	
	INTEGRATION TESTING	COMPLEXITY	
	PROTOTYPING	DEBUGGERS	
		CONFIGURATION MANAGEMENT	
		SCHEDULING	

Table 3-13
Project 1218

PROJ_NO	1218	ES
SW_TECH	SW_MTDS	DOC_TYPE
CLIENT/SERVER	BUSINESS AREA MODELING	ACCEPT TEST PLANS
ES	PROCESS MODELING	DESIGN DOCUMENT
DATABASE	EVENT MODELING	INTEGRATION TEST PLANS
	DATA MODELING	PROGRAMMER MANUALS
	REGRESSION TESTING	PROJECT MGT PLANS
	INTEGRATION TESTING	REQUIREMENTS DOCUMENTS
	BLACK/WHITE BOX TESTING	UNIT TEST PLANS
	REQUIREMENTS TRACING	
	PROTOTYPING	

A review of the data above shows that the efforts cover a wide range of topics and programmatic concerns. The table demonstrates the variety of purposes being addressed by AI and ES technologies. It also demonstrates the variety of management techniques and tools used to develop the efforts.

3.5.2 Detailed Project Data.

This section describes whether or not the software Capability Maturity Model (CMM) criteria were used as part of the selection process. GSC_MAT_SEL indicates if the government evaluated the prime contractor based upon a software maturity level criteria model developed by the government. GSC_MAT_DET describes the method used by the government agency to determine the maturity level of the prime contractor. GSSC_MAT_SEL indicates if the government evaluated any subcontractors based upon a software CMM level criteria developed by the government. SW_APPR describes the development approach used by the developer. PROJ_PHS identifies the current phase of the program at the time of data submission to the NSDIR. Additional descriptions of the categories are provided in Appendix A.

Table 3-14
Capability Maturity Model (CMM) Data

PROJ_NO	ORG_NO	PROJ_DSCN	GSC_ MAT_SEL	GSC_ MAT_DET	GSSC_ MAT_SEL	SW_APPR	PROJ_PHS
1000	1000	Enhancement	NO		NO	INCREMENTAL	Prototype
1007	1011	Enhancement	NO			PROTOTYPING	
1022	1027	New System	NO		NO	PROTOTYPING	DEM/VAL
1027	1031	OTHER	YES	INTERNAL	UNKN		SUS-OPS
1058	1047	RE-ENG	NO		NO	PROTOTYPING	PROD
1098	1068	ENHANCE MENT	YES	EXTERNA L		WATERFALL	S/W REQT ANL
1211	1103	NEW SYSTEM				INCREMENTAL	FULL DEV
1218	1103	NEW SYSTEM	UNKN				NA

Table 3-15 indicates whether or not International Standards Organization (ISO) certification was obtained in the conduct of the program. EFF_REP describes whether or not metrics were reported to the government. COTS_PERC describes the percentage of a system's software in terms of either functionality or size for commercial-off-the-shelf (COTS) products. COTS_MSMT describes the basis for measurement of percentage given for COTS products. The value "E" means a non-measured based estimate was used versus an actual measurement.

Table 3-15
ISO Certification and Metrics Reporting

PROJ_NO	SW_TECH	ISO_CERT	EFF_REP	COTS_PERC	COTS_MSMT
1000	ES	UNKNOWN		20	E
1007	ES	NO	YES	35	E
1022	AI	NO			
1027	AI	UNKNOWN	YES	5	E
1058	ES	NO	NO		
1098	ES	NO	YES	0	
1211	ES	NO		15	E
1218	ES	NO	NO		

The following table describes the percentage of a system's software in terms of either functionality or size for developmental software. DEV_PERC describes the percentage of system software that is considered new development. DEV_MSMT indicates the basis of measurement for the DEV_PERC value. An "E" designator represents an estimate while a "M" value represents a measure based estimate. INT_DEV_PERC refers to the percentage of system's software in terms of either functionality or size for internally developed reusable packages.

Table 3-16
Percentage of New Software Development

PROJ_NO	DEV_PERC	DEV_MSMT	INT_DEV_PERC
1000	60	E	
1022	100	E	
1027	95	E	5
1058			100
1098	1	M	
1211	60	E	2

In Table 3-17, INT_DEV_MSMT refers to data in the last column of Table 3-16. It represents the basis of measurement for the INT_DEV_PERC value. The “E” and “M” designators represent the same indications as before. GOTS_PERC refers to the percentage of software used in the software development effort that was government-off-the-shelf (GOTS). GOTS_MSMT represents the basis of measurement for GOTS_PERC.

Table 3-17
Percentage of Software Development from GOTS

PROJ_NO	SW_TECH	INT_DEV_MSMT	GOTS_PERC	GOTS_MSMT
1000	ES		20	E
1007	ES		65	E
1027	AI	E		
1058	ES			
1098	ES		99	M
1211	ES	E		

Table 3-18 provides more top-level data concerning the programs. SW_TYPE describes the end application of the developed software. SW_TOOLS_OTR describes tools that are not included in the pre-defined values for SW_TOOLS. SLOC_CAL indicates whether or not comments were included or excluded in the calculation of source

lines of code (SLOC) for the program. SLOC_CAL_OTR indicates other methods used to calculate SLOC for the program.

Table 3-18
Software Type

PROJ_NO	SW_TECH	SW_TYPE	SW_TYPE_OTR	SLOC_CAL	SLOC_CAL_OTR
1000	ES	COMM			
1007	ES	OTHER	INFO WARFARE	INCLUDING	
1022	AI	TRAINING		EXCLUDING	
1027	AI	WEAPONS SYS			
1058	ES	AVIONICS		EXCLUDING	
1098	ES	AVIONICS		INCLUDING	
1211	ES	SYS S/W		OTHER	%DEVEL EFFORT
1218	ES	MIS			

Table 3-19 describes whether or not measurement efforts were used in the performance of the development effort. MSMT_PLAN indicates whether or not a documented measurement plan was used or not. METR_DB indicates whether or not a metrics database was used or not. METR_PRSN and MSMT_GRP indicate whether or not a metrics coordinator and/or metrics group existed during the course of the program. METR_REP describes whether or not the metrics were regularly reported. The final category, QA_FUNC, describes whether or not an independent test and quality assurance (QA) function was employed on the program.

Table 3-19
Metrics and Measurement Plans

PROJ_NO	SW_TECH	MSMT_PLAN	METR_DB	METR_PRSN	MSMT_GRP	METR_REP	QA_FUNC
1000	ES	NO	NO	NO	NO	NO	NO
1007	ES	UNKNOWN	YES	YES	NO	NO	NO
1022	AI	YES	YES	YES	YES	NO	NO
1027	AI	YES	NO	YES	YES	YES	YES
1058	ES	YES	YES	YES	NO	NO	YES
1098	ES	YES	YES	YES	NO	YES	YES
1211	ES	NO	NO	NO	NO	NO	NO
1218	ES	NO	NO	NO	NO	NO	NO

3.5.3 Schedule Data.

Table 3-20 provides a cursory view of the schedule data associated with the program efforts. It is evident that all of the programs did not provide this information. PROJ_START_EST indicates the estimated start date of the program, while PROJ_START_ACT represents the actual starting date of the program. PROJ_END_EST represents the estimated end date of the program, while PROJ_END_ACT represents the actual end date for the program.

Table 3-20
Schedule Data

PROJ_NO	PROJ_START_EST	PROJ_START_ACT	PROJ_END_EST	PROJ_END_ACT
1211	11/20/89	11/20/89	12/31/95	
1098	2/1/95	3/1/95	8/31/98	
1027	1/1/69	1/1/82	12/31/20	
1022	8/1/93	8/1/93	8/31/94	2/28/95
1000	10/15/94	10/30/94	3/1/95	2/15/95

3.5.4 Source Lines of Code per Project.

The next table provides a limited amount of SLOC data associated with two of the programs. Once again, most programs did not provide this data. Most efforts seem willing to provide descriptive qualitative data, but they are not willing to provide descriptive quantitative data. In this table, TOT_SLOC_EST represents the original estimate of SLOC for the program,
TOT_SLOC_REV represents the revised estimate, and TOT_SLOC_REM represents the remaining SLOC to be developed. Project 1098 was just beginning when the data was submitted, and project 1022 appears to be in a growth phase.

Table 3-21
SLOC Data

PROJ NO	TOT SLOC EST	TOT SLOC REV	TOT SLOC REM
1098	315153	315153	315153
1022		56000	11200

The following table provides additional data relating to SLOC. NEW_SLOC_EST describes the estimated amount of new and modified SLOC which will be needed for the program. NEW_SLOC_REV provides the revised estimate of the previous value. NEW_SLOC_REM describes the amount of new and modified SLOC development remaining for completion of the program. TOT_FP_EST indicates the total number of function points for the program. TOT_FP_REM represents the total number of function points needing development before completion of the program. As indicated in the scheduling data, project 1022 has been completed.

Table 3-22
New SLOC Developed

PROJ_NO	NEW_SLOC_EST	NEW_SLOC_REV	NEW_SLOC_REM	TOT_FP_EST	TOT_FP_REM
1098	2770	2770	2770		
1022		50000	0	2950	0

3.5.5 Documentation.

Table 3-23 indicates that some data were retrieved in tracking the documentation pages needed for the efforts. DOC_PGS_EST describes the total number of estimated documentation pages required for the effort, DOC_PGS_REM describes the total number of pages remaining for the project, and DOC_PGS_REV describes the revised estimate of documentation pages needed for the project.

Table 3-23
Documentation

PROJ NO	DOC PGS EST	DOC PGS REM	DOC PGS REV
1211	5000	2700	4800
1022			630
1000	400		2500

3.5.6 Effort Represented in Person Years.

The next table provides some limited insight into the effort required for the development programs. TOT_PRSYS_EST describes the estimated number of person years required for the project, TOT_PRSYS_REV describes the revised estimate for the project, and TOT_PRSYS_REM describes the number of person years remaining to complete the project.

Table 3-24
Effort in Person Years

PROJ_NO	TOT_PRSYRS_EST	TOT_PRSYRS_REV	TOT_PRSYRS_REM	PERC_RWK	RWK_MSMT
1098	16	16	16		
1022	6	9	0		
1000	0	2	0	5	E

3.5.7 Training Data.

The table below shows that two of the programs considered training when the development efforts were in progress. TRNG_SW_MGMT identifies the percent of participation in software management training by people involved with the development effort. TRNG_SW_MGMT identifies the percent of participation in software engineering methods training by people involved with the development effort. TRNG_QA identifies the percent of participation in quality assurance training.

Table 3-25
Training

PROJ_NO	TRNG_SW_MGMT	TRNG_SW_MTDS	TRNG_QA
1211	1	5	20
1000	15		

3.6 Summary

The objective of this thesis is to establish a set of indicators and metrics that can be used by program managers who will manage software development efforts involving AI and ES technologies. Three steps have been taken with the methodology phase of this thesis to accomplish this goal. The first step involved the review of current literature concerning indicators and metrics used for conventional software development efforts.

The second step involved the review of current literature concerning indicators and metrics used in the management of AI and ES software development efforts. The third step retrieved the existing AI and ES management data maintained at the NSDIR.

Accomplishment of the three steps led to the selection of Tables 3-1 and 3-2. Table 3-2 combines elements relating to both conventional and AI/ES software development efforts. The retrieval of AI and ES data from the NSDIR produced Tables 3-3 through 3-25. The data in Tables 3-3 through 3-25 provide a large qualitative picture of how the efforts were managed and developed. By comparing the recommended indicators and metrics in the literature to the indicators and metrics collected by completed and on-going software development efforts, a final set of recommended indicators and metrics will be established during the analysis phase of this thesis.

4.0 Analysis & Findings

4.1 Introduction

The focus of this effort is to establish a useful set of indicators that can be used by program managers in charge of software development efforts involving AI or ES technologies. A two-pronged approach has been attempted. The first portion involved the review of literature to establish an academic structure for the proposed indicator framework. Secondly, information and data from current or completed programs involving AI and ES technologies were reviewed to determine what indicators and metrics that managers of these efforts consider important. Ultimately, a useful set of indicators will be developed by analyzing the academic structure and the "front-line" data and combining the best elements of both into a single framework.

4.2 Suitability of Selected Indicators and Metrics

The selection of indicators and metrics identified in Tables 3-1 and 3-2 represent a comprehensive list that future program managers of AI and ES software development efforts can use as a baseline. The two tables are based upon sources representing competent research and expert opinion. For each unique software development effort, the indicators and metrics listed in the tables should be tailored to provide the needed insight on the progress of the development effort while minimizing costs. Proper selection and application of the chosen indicators and metrics should help reduce costs while ensuring the development of a quality product. The indicators and metrics in Table 3-1 and 3-2

provide a suitable basis for evaluating the NSDIR data concerning AI and ES software development efforts.

4.3 Analysis of NSDIR Database

The data resident in the NSDIR database concerning AI and ES software development efforts proved to be more qualitative than quantitative. Most the data was descriptive and more representative of top level information. Unfortunately, none of the projects supplied recurring data to NSDIR. Recurring data in this sense refers to information such as size over time, problem reports over time, effort over time, etc. Without this type of data, an assessment of the development success of a project cannot be made.

4.4 Adequacy of NSDIR Resources

The staff and available resources at the NSDIR were sufficient for the purposes of this research effort. To access the NSDIR database, two software tools provided by the personnel at the NSDIR were installed on a personal computer at home and used to retrieve the data. The two software tools were SQL-Retriever and the NSDIR Information Analysis Tool (IAT). Proficiency with the tools was needed to retrieve the data, but the personnel at the NSDIR were readily available to facilitate this process. The people providing assistance were very helpful and patient. Overall, the NSDIR demonstrated that it had the resources to manage and retrieve the data that it maintained. Unfortunately, the available data on AI and ES software development efforts was limited.

4.5 Usefulness of NSDIR Resources for AI/ES Software Development Efforts

The NSDIR possesses the potential to serve as a useful resource for managers in charge of all types of software development efforts. Currently, the NSDIR does not maintain a large database on AI and ES efforts. However, the data that the NSDIR does have relating to AI and ES efforts provides some insight into what program managers of those efforts consider as important indicators and metrics. These items are discussed in the paragraphs below.

4.5.1 Source Lines of Code (SLOC).

Projects 1098 and 1022 indicated that data reflecting the amount of SLOC developed for their individual efforts were collected. The other six projects may have collected SLOC data; however, the other six projects may have decided not to provide that information to the NSDIR. Only two of the eight projects reported information relating to SLOC. Whether or not SLOC data is considered important by managers of these efforts cannot be determined. Despite the limited data, this research effort will recommend the inclusion of SLOC data as a useful metric, which is supported by the findings contained in the literature review.

4.5.2 Effort.

Projects 1098, 1022, and 1000 collected data on the amount of effort required for the software development project in total person years. Measuring effort in some fashion is seen as a useful metric in managing a software development program. This statement is

supported by the sources cited in the literature review section of this effort. With the measurement of effort in some form, a program manager can measure progress and compare the results to the original estimate to obtain some indication of the success or lack of success of the effort.

4.5.3 Percentage of New Development.

Projects 1000, 1022, 1027, 1058, 1098, and 1211 collected data that describe the percentage of the software that was developed or taken off-the-shelf. In addition, the data reflected the percentage of software development that was attributable to a contractor or the government. Also, this category identified the percentage of software developed internally by government resources and the percentage of software developed externally by other firms or organizations. Identifying the percentage of work performed in the various areas is seen as a useful metric. Inclusion of this type of data will be recommended for consideration by program managers. This will identified by the metric "percentage of software effort developed".

4.5.4 Scheduling.

A review of the data shows that projects 1000, 1022, 1027, 1098, and 1211 collected data on scheduling. The collection of scheduling data provides a useful way of evaluating the schedule status of AI and ES software development efforts. This assertion is supported by some of the findings discussed in the literature review. This research will recommend the consideration of schedule indicators and metrics.

4.5.5 Document Pages.

Projects 1000, 1022, and 1211 collected data concerning the number of document pages required for the effort. This was an interesting observation because none of the sources cited in the literature review included this as a useful indicator or metric. This research effort will not recommend the use of document pages required for a development effort as a useful indicator or metric.

4.6 Results of NSDIR Database Analysis

The analysis of the NSDIR database found that the available information provides a top level qualitative view of eight programs involving AI or ES technologies. A review of the existing data on AI and ES software development efforts may provide a manager of future software development efforts with different ideas on how to setup various development strategies. The data in the NSDIR did provide insight into how the various program offices developed the software.

However, the data provided was not sufficient to evaluate how successful these projects performed. Surprisingly, no cost data was available. This may be due to the fact that the projects wanted to ensure their anonymity. Effort data was provided on a limited basis, but program offices did not provide this information on a recurring basis reflecting effort over time. In general, the database lacked recurring data to indicate progress, in some form, over time. Without this type of data, it is difficult to assess the success of a project.

4.7 Limitations of NSDIR

The primary limitation of the NSDIR is the passive collecting operation used to obtain the information and data from the projects. The program office personnel working with the software development efforts only provide the data that they deem important. None of the projects are required to provide data to the NSDIR. The NSDIR must rely on the program offices to supply the data, and the personnel at the NSDIR cannot actively pursue the data. If the program office chooses not to provide certain items, the NSDIR staff has no ability or recourse to retrieve the omitted data. Until organizations are required to provide data to the repository, the NSDIR will be limited by the current passive collection operation.

Another limitation of the NSDIR is that the people who maintain the database are not the people who collected the data. Therefore, insight into what the data actually represents may be limited. This is further hampered by the fact that the projects may collect data in varying formats and levels of detail.

4.8 Limitations of the Research

The research is limited by the lack of actual communication with the program managers of the AI and ES software development efforts. Because the available data concerning AI and ES development efforts were limited, the ability to determine the indicators and metrics that the managers consider as important for effectively controlling the development efforts is questionable. Despite this limitation, one aspect of the research was demonstrated by using the NSDIR at its face value. A goal of the research effort was

to evaluate the value of the NSDIR to a program manager in search of information and data involving AI and ES technologies. In the areas involving AI and ES technologies, the data were limited. However, other software technologies may be well represented in the NSDIR database. Compared to other software technologies, AI and ES are still relatively new. This may be the reason for the limited data that exists concerning them.

Another limitation of the research is due to the qualitative approach that was used to select the proposed indicators and metrics and evaluate the data from the NSDIR. A more sound approach could have surveyed experts in the fields of AI and ES technologies to provide a better assessment of the indicators and metrics and the NSDIR data. The qualitative approach was chosen because the data from the NSDIR was more representative of qualitative issues.

4.9 Recommended Indicators and Metrics

The recommended indicators and metrics will remain essentially the same as presented in Tables 3-1 and 3-2. One addition relating to percentage of software development will be made to Table 3-2. This change is reflected in the Table 4-1 below.

Table 4-1
Selected Metrics

Issue	Category	Measurement Metric
Schedule and Progress	Milestone Performance	Milestone Dates
	Schedule Performance	Estimated Duration/Size
		Actual Duration/Size
		Schedule Performance
Resources and Cost	Effort Profile	Estimated Effort/Duration
		Actual Effort/Duration
		Estimated Effort for S/W Type
		Percentage of S/W Effort Developed
	Cost Performance	Cost Variance
		Cost Profile
Growth and Stability	Product Size and Stability	Lines of Code (LOC)
		Knowledge Base Size
	Functional Size and Stability	Requirements
		Function Points
Product Quality	Defect Profile	Problem Report Trends
		Failure Interval
Development Performance	Productivity	Product Size/Effort Ratio
		Functional Size/Effort Ratio
	Rework	Rework Size
		Rework Effort
AI/ES to User Interface	Application Characteristics	User Interface
		Types of Data Files
		Expert Judgment Stored
		Ability to Modify Judgment
AI/ES to Organization I/F	Efficiency Factors	Speed
		Reliability
		Supportability
Org Environment Interface	Potential for Implementation	Decision Quality
		Technical Soundness
		Decision Process Quality

Table 3-1 will remain the same as the recommended indicators that program managers should consider when establishing an indicator framework for the new development effort. Program managers may use these tables as a starting point in the tailoring process when determining the important indicators and metrics to collect during the course of the software development effort involving AI and ES technologies. Use of the recommended indicators and metrics should provide the insight required to develop a successful product.

4.10 Summary

From the research obtained from the literature review in Chapter II, the researcher selected two tables with one representing indicators and the other representing metrics that could be used by managers of future software development efforts involving AI and ES technologies. In addition, information and data, maintained at the NSDIR, pertaining to AI and ES software development efforts were analyzed to determine which indicators and metrics were considered important by the management personnel of these programs. By analyzing information and data from completed or on-going software development programs, an attempt was made by the researcher to gain insight into what managers, who were actually involved with the day-to-day activities of the development programs, have used in the areas of indicators and metrics. Unfortunately, the researcher was unable to identify which indicators and metrics were considered the most useful by managers of completed or on-going AI and ES development efforts from the data maintained at the NSDIR. Once the work related to the literature review and the analysis of NSDIR data was completed, a recommendation was made regarding the indicators and metrics that managers of AI and ES software development efforts may find useful in the performance of their duties.

5.0 Conclusions and Recommendations

5.1 Overview of the Research

As the DoD continues to procure weapon, support, and training systems, the importance of software involved with these systems should continue to play a larger role in these efforts. Because of the significant role that software may fulfill in future systems, managers responsible for the successful development of these efforts will need management tools such as software indicators and metrics in order to understand the increased complexity and manage the growing size of these software systems. This research has attempted to show that methods and measures, which will enable managers to control their programs more effectively, do exist. Whether or not these program management tools for software development are being used is an important question. The literature seems to indicate that most managers of software development are not using the suggested methods and measures designed to aid in the effective management of these efforts.

As the DoD continues to develop and acquire more technically advanced systems, the use and importance of AI and ES software will become more prevalent in many of these systems. Through proper selection of indicators and metrics, program managers of these systems dealing with AI and ES software have the opportunity to reduce some of the programmatic risks associated with the development efforts. This research has tried to provide a set of indicators and metrics that managers may use as a starting point during the initial indicator and metric selection process.

5.1.1 Objectives of the Research.

Two primary objectives were sought with this research effort. The first objective was to establish a baseline of software indicators and metrics that could be used by managers during the course of software development efforts that involve AI and ES technologies. The second objective of this research was to demonstrate the usefulness of the NSDIR as a resource that provides information and data to managers of software efforts working with AI and ES technologies. Ultimately, the work associated with the two objectives was combined to provide a recommended set of indicators and metrics that could be used by managers of AI and ES software development efforts.

5.1.2 Research Methodology.

The indicators and metrics that were selected as a result of the literature review and the ones that were retrieved from the NSDIR are presented in Chapter III. In Chapter IV, the information and data obtained from the NSDIR were analyzed to determine which indicators and metrics were considered important by managers of completed or on-going efforts involving AI and ES technologies. As mentioned before, the ability to determine what managers considered important in terms of indicators and metrics was very limited. The information and data from the NSDIR was compared to the indicators and metrics selected from the literature review sources. Upon completing the work associated with Chapters III and IV, a set of indicators and metrics, based primarily upon the literature review effort, was selected as the recommended set that managers could use during the development of AI and ES software systems.

5.2 Results of the Research

Accomplishment of the first objective was attempted by performing the literature review, which found a need for an increased use of indicators and metrics for the successful development of software systems. Through the review, the existence of material recommending indicators and metrics was shown. By reviewing this material, a recommended set of indicators and metrics relating to AI and ES software development efforts were selected and presented in Chapter III.

The results of meeting the second objective were not very successful. A review of the NSDIR data pertaining to AI and ES technologies discovered the data to be mostly qualitative. The data only provided a limited amount of insight concerning the indicators and metrics managers considered important. After comparing the set of indicators and metrics from the literature review to the ones obtained from the NSDIR, a final set of indicators and metrics was recommended for use by managers of software development efforts utilizing AI and ES technologies. The recommendation is based primarily on the information presented in the literature review.

5.3 Recommendations for NSDIR Developed from the Research

The NSDIR has the potential to provide valuable information to managers of software development efforts. It also has the potential to serve as an important central resource that can be used to evaluate the management of software development projects. In order to achieve this potential, NSDIR could emphasize three areas that may possibly improve its value as a resource. The first area deals with the continued emphasis on the

need for all DoD organizations involved with software development to provide the data associated with the indicators and metrics used on their programs. Based upon the direction, discussed in Chapter II, from senior DoD and Air Force leaders, organizations will be required to provide the type and quantity of data that the NSDIR requires.

The second area involves a transition from the passive retrieval system to more of an active data retrieval system. With the use of more active measures to retrieve the data, the NSDIR will be in a better position to influence the standardized format and content of the data provided by the program offices. Consistent and uniform data input is the primary thrust of this recommendation. Implementation of a more active system may incur more costs and run into problems of management politics; however, a combined effort by everyone working with software development will be required if successful management of software development is to become the rule rather than the exception. Hopefully, the current support of senior DoD leadership will make this a reality.

The third area involves the continued effort of NSDIR to ensure that all DoD agencies managing software development efforts are aware its efforts. As more organizations become aware of the potential that the NSDIR has to offer, the value of the NSDIR will grow. Instead of re-inventing software effort that has already been accomplished in one form or another, organizations can use the NSDIR to improve new software development efforts. Improvements from these efforts should reflect lower cost, better quality, and mission accomplishment.

5.4 Recommended Indicators and Metrics for AI/ES Development Efforts

The following tables represent the recommended indicators and metrics that managers may use during the course of developing AI and ES software systems. Table 5-1 represents the indicators, which are used in a less rigorous manner that metrics. These indicators can be used to indicate trends in a software development effort. Table 5-2 represents the metrics that can be used during the management of software development efforts.

Tables 5-1 and 5-2 are provided at this point to provide a consolidated view of the suggested indicators and metrics. Table 5-1 is the same as Table 3-1 because no recommended changes were identified in Chapter IV. Table 5-2 is the same as Table 4-1 with the one addition involving "percentage of software effort developed."

Table 5-1
Recommended Indicators

Area	Indicators
Requirements	CSCI Requirements Stability
	CSCI Design Stability
Software Performance	I/O Bus Throughput Capability
	Processor Throughput Utilization
Schedule	Requirements Allocation Status
	Preliminary Design Status
	Detailed Design Status
	Code and Unit Test Status
	Integration Status
Cost	Man Months of Effort
	Software Size

Table 5-2
Recommended Metrics

Issue	Category	Measurement Metric
Schedule and Progress	Milestone Performance	Milestone Dates
	Schedule Performance	Estimated Duration/Size
		Actual Duration/Size
		Schedule Performance
Resources and Cost	Effort Profile	Estimated Effort/Duration
		Actual Effort/Duration
		Estimated Effort for S/W Type
		Percentage of S/W Effort Developed
	Cost Performance	Cost Variance
		Cost Profile
Growth and Stability	Product Size and Stability	Lines of Code (LOC)
		Knowledge Base Size
	Functional Size and Stability	Requirements
		Function Points
Product Quality	Defect Profile	Problem Report Trends
		Failure Interval
Development Performance	Productivity	Product Size/Effort Ratio
		Functional Size/Effort Ratio
	Rework	Rework Size
		Rework Effort
AI/ES to User Interface	Application Characteristics	User Interface
		Types of Data Files
		Expert Judgment Stored
		Ability to Modify Judgment
AI/ES to Organization I/F	Efficiency Factors	Speed
		Reliability
		Supportability
Org Environment Interface	Potential for Implementation	Decision Quality
		Technical Soundness
		Decision Process Quality

Tables 5-1 and 5-2 provide a feasible starting point for managers of new AI and ES software development efforts. From the indicators and metrics listed in the tables, managers can begin the tailoring process to select the appropriate indicators and metrics for their particular software development effort.

5.5 Recommendations for Future Research Efforts

Future research could apply the recommended set of indicators and metrics to a specific AI or ES development effort. This could also be done for multiple efforts. Another research endeavor could survey managerial and technical personnel associated with AI and ES software development efforts on the value and suitability of the recommended indicators and metrics.

Other areas of research could explore the NSDIR information and data that pertains to other software technologies. Another research effort could survey the software development community to determine the general awareness that exists concerning the NSDIR and what it has to offer. An additional survey could query people who have utilized the NSDIR. The survey could evaluate the usefulness and value of the NSDIR perceived by the people who have used it. Many possibilities of research dealing with the value of the NSDIR and management of AI and ES software development exist. Research in the these areas and many more will be required if the goal of improved software development management is going to be achieved.

5.6 Summary

A primary aim of this research effort was to identify the software indicators and metrics that can be used to develop AI and ES software systems. Hopefully, the use of the indicators and metrics will support the development of AI and ES applications that function correctly and satisfy the needs of the users. Government support for the use of indicators and metrics is continuing to grow. It the belief of the author that the use of

indicators and metrics will be required in order to reduce software development costs while ensuring the delivery of a quality product.

A key to the methodology used with this research effort involved the researcher assuming the role of a typical government program manager interested in the development of an AI or ES software system. The aim of this approach was to evaluate the NSDIR at face value. This approach did provide some insight which led to the selected indicators and metrics. Hopefully, this suggested set of indicators and metrics will provide program managers with a useful start to the tailoring process of selecting the appropriate indicators and metrics for future AI and ES software development efforts.

Appendix A

NSDIR Repository Information Request:
Metric Collection Guide

NSDIR
National Software &Data Repository

Initial _____

Repository Information Request
Metnc Collection Guide, Update

Update _____

Section A - Identification

Project _____ Report _____

POC Name:_____
Title/Rank:_____
Company Name/Service:_

Organization Designation:_____ Of fice Symbol/Code:_____

•

Street:_____ State:_____

City:_____ Country:_____

Zip Code: _____

Daytime (non-DSN)

Phone Number: _____ Ext:_____

Fax Number: _____

Project Full Name:
_____

Project: Report Date: _

Section B - Organization

Service or Sector (Check One):

O Army
O Navy
O Air Force
O Marine Corps
O Coast Guard
O Defense Agency
O Non-DoD Government
O Non-Government

(org_name:pg 17)

Air Force Organization (Check *One)(af org:pgl7)*

O ACC
O ABTC
O AFMC
O AFSPC
O AFSOC
O AMC
O PACAF
O USAFE

Field Organization
Agency:
Direct Reporting Unit:

Other: *(ax org_dtt:pg18)*

Software Development Expenditure (Total = 100%)

organic/m -house		contract	
new development % of total budget	maintenance % of total budget	new development % of total budget	maintenance % of total budget
(org_new:pg 18)	*(org_mnt pg 18)*	*(ctr_new:pg 18)*	*(ctr_mnt:pg 18)*
_____	_____	_____	_____

Contracted Software Development Labor Expenditure (Total = 100%)

Prime contractor (less sub-contractors) contract	*(pm_ctrt:pg 18)*	% of total labor
Small business (8A) sub-contractor	*(sb_ctrt:pg19)*	% of total labor
Small business (non-8A) sub-contractor	*(sb_nctrt:pg 19)*	% of total labor
Non-profit organization	*(np_ctrt pg 19)*	% of total labor
Other sub-contractor	*(bb_ctrt:pg 18)*	% of total labor

Project: Report Date:

Project Description (Check the One Most Applicable) *(proj_dscn:pg 33)*

- New system
- Enhancement to current system (new capability)
- Maintenance (defect repair for existing systems)

Re-engineering of existing system (*re_eng_dtl:pg 33)*

- Same language, same platform
- New language, same platform
- Same language, move to new platform or open systems specification with new documentation
- New language, move to new platform or open systems specification with new documentation
- Other:_____*(proj dscn otr: pg 33)*

Software Type (Check the one most applicable) *(sw_type:pg 37)*

- Systems software
- Communications or telecommunications
- Command and control
- Process control
- Management information system
- Scientific data processing
- Robotics
- AVIONICS
- Intelligence Automated test equipment Weapons system
- Logistics system
- Financial
- Training
- Simulation

Other (*sw_type_otr:pg 37)*

System Technologies (Check all that apply to the above software type) *(sw_tech:pg 42)*

- Batc_ processing
- Client/Server
- Expert system
- Neural net
- Real-Time
- Security
- Interactive processing
- Multi-Media
- Artificial intelligence
- Grapics/Animation/lmage processing
- High reliability/dependability
- Database

Other (*sw_tech_otr:pg 43)*

Dcwlopment Life-Cycle Phase (proj_phs:pg 36)

- Requirements
- Design
- Code and Test
- Integration and Test

Other:

Last Milestone (proj_mlst pg 36)

- Project Start
- System Design
- Implementation Complete
- Build, Release, or Version Delivery
- Project Complete
- Other: Design Document Delivered

Section D - Process Maturity

Government Contract—Contractor Selection Criteria (Check all that apply)

O Minimum CMM Maturity Level of: _____ *gsc_mat sel:pg 33*

O IS09000 certification: _____ *gsc_iso:pg 34*

Government Contract—Contractor CMM Maturity Level and Method

The contractor CMM Maturity level (1-5) was: The CMM Maturity level was determined through*:gsc_mat_det:pg 34*

gsc_mat_lvl:pg 34

O Formal internal assessment
O Independent external assessment
O Government Software Capability Evaluation
O Other:

Government Contract—Sub-Contractor Selection Criteria (Check all that apply)

O Minimum CMM Maturity Level of: *gssc_mat_sel:pg 34*

O IS09000 certification *gssc_iso:pg 34*

Government Contract—Sub-Contractor CMM Maturity Level and Method

The contractor CMM Maturity level (1-5) was: *gssc_mat_lvl:pg 35*

The CMM Maturity level was determined through gssc_mat_det pg *34*

Formal internal assessment
Independent external assessment
Government Software Capability Evaluation
Other:

Non-Government Contract—Contractor Selection Criteria (Check all that apply)

O Minimum CMM Maturity Level of: *ngsc_mat_sel:pg 35*

O IS09000 certification *ngsc_iso:pg 35*

Non-Government Contract—Contractor CMM Maturity Level and Method

The contractor CMM Maturity level (1-5) was: The CMM Maturity level was determined through*:ngsc_mat_det:pg 35*

ngsc_mat_lvl:pg 35

Formal internal assessment
Independent external assessment
Other

Section E - Development Constraints

Development Approach. (Check the one most applicable) *(sw_appr:pg 35)*

O Waterfall / Grand Design

O Incremental

O Evolutionary

O Other

Software Development Standards (Check all that apply)

Military *(mlt_std:pg 14)*

O DoD-STD-2167A

O DoD-STD-2168

O DoD-STD-7935A

O DoD-STD-499A

O Mil-STD-498

O Other:

O Government (gov_std:pg *14)*

O Commercial (com_std*:pg 11)*

O Developed internally *(proj_std:pg 31)*

Contract Standards (government contract a only) *(std_ctrt:pg 36)*

O Placed on contract

O Provided as guidance

Security Clearance (Check highest required) *(proj_sec:pg 36)*

O None

O Confidential

O Secret

O Top Secret

O Special Access

Project: Report Date:

Section F - Methods, Tools, and Training

System Development Methods: (Check all that apply *(sw_mtds pg 41)*

O Joint Application (Development)	Business Area Modeling (BAM)
O Structured Analysis (SA)	Process Modeling (PM)
O Event Modeling (EM)	Object Oriented Analysis (OOA)
O Structured Design (SD)	Data Modeling (DM)
O Regression Testing (FOT)	Object Oriented Design (OOD)
O Rapid Application Development (RAD)	Integration Testing
Black Box (functional) / White Box (structural) testing (BBWB)	Multifunctional Teams (MT)
Requirements Tracing (ROT)	Prototyping (P)
Other *(sw_mtds_otr pg 42)*	

Tool blip": (Check all that apply) *(sw_tools pg 44)*

Project management (planning/scheduling, earned value)	
Database	Cost estimating tool
Process Management	Complexity measuring
Environment Management	Debuggers
Document and Presentation Production	Configuration Management
	Quality Assurance
Metrics collection and analysis tools	Requirements Traceability
Upper CASE tools	Lower CASE tools
Other *(sw_tools_otr pg 44)*	

Training Areas (Check all available and enter percent participation during last year)

Training Availability	% participation
Software project management	*(trng_sw_mgmt pg 31)*
Software engineering methods	*(trng_sw_mtds pg 31)*
Software engineering tools	*(trng_sw_tools pg 31)*
Language skills	
Quality assurance	*(trng_qa pg 32)*
Configuration management	*(trng_conf man pg32)*
Measurement	*(trng_msmt pg 32)*
Testing	*(trng_tst pg 32)*
Other	*(trng_otr pg 32)*

Section G - Software Source and Language

Software Source

Category		*(cots_msmt::pg 38)*
COTS	*(cots_perc:pg 38)*	
GOTS	*(gots_perc:pg 38)*	
Developmental Software	*(dev_perc:pg38)*	*(int_dev_msmt:pg 39)*
Internally developed reusable packages	*(int dev_perc:pg 39)*	

Programming Languages

Language *(language: pg 24)*

Actual SLOC *(act_sloc. p24)*

O physical SLOC, incl comments
O physical SLOC, excl comments
O Other _____

Approximate Fraction of Total System(lang_apprx:pg 24)

Less than 1/3	between 1/3 and 2/3	More than 2/3
Ada		
Atlas		
Jovial		
FORTRAN		
C		
C++		
CMS-2M		
CMS-2Y		
Basic		
Pascal		
SPL/1		
Assembler		
COBOL		
4GL		
GUI development tools		
LISP		
Other(s)		

Project: Report Date:

Section H - Measurement Practices

Software Measurement Resources	**Yes**	**No**	**Don't Know**
A documented measurement plan *(asset_plan:pg 39)*			
A metrics database *(metr_db:pg 39)*			
A metrics coordinator or responsible person *(metr_prsn:pg 39)*			
A separate measurement group/function *(msmt_grp:pg 39)*			
Regular reporting of metrics data *(metr_rep:pg* 40)			

Metrics Usage (Check all that apply)

Type of **metric**	**individual level**	**team wide**	**system wide**	**organization wide**
Effort (staff hours) *mtrs_eff:pg 15)*				
Software Quality (errors/discrepancies)(mtrs_qual:pg *16)*				
Software Size (SLOC, function points)(mtrs_size:pg 17)				
Schedule (milestone completion vs. milestone commitments) *(mtrs_sch:pg 16)*				
Rework (change orders, staff hours) mtrs_rwk:pg *16)*				
Other *(mtrs_ort:pg 15)* *(mtrs_otr_dtl:pg 15)*				

Reported to the Government (Government Contract Only)
(eff_rep_dtl:pg 13)

Yes

No

Some

Section I - Software Metrics Profile

Schedule	*Original Estimate*	*Actual*
Start Date	*(proj_start_est:pg 24)*	*(proj_start act:pg 25)*
Completion Date	*(proj_end_est:pg 25)*	*(proj_end act:pg25)*

	Original	Current/revised estimate	Remaining to completion
total new and modified (this project)	*(tot_sloc_est:pg 25)*	*(tot_sloc_rev:pg 25)*	*(tot_sloc_rem:pg 25)*
	(new_sloc_est:pg 25)	*(new_sloc_rev:pg 25)*	*(new_sloc_rem:pg 25)*
Function Points			
total	(tot_fp_est:pg 26)	(tot_fp_rev:pg 26)	(tot_fp_rem pg 26)
new and modified (this project)	(tot_fp_est:pg 26)	(tot_fp_rev:pg 26)	(tot_fp_rem:pg 26)
Number of documentation pages			
total new and modified (this	(doc_pgs_est:pg 26)	(doc_pgs_rev:pg 26)	(doc_pgs_rem:pg 27)
project) Check all that apply	(doc_pgs_est:pg 26)	(doc_pgs_rev:pg 26)	(doc_pgs_rem.pg 27)
	(doc_type:pg 13)		

O Requirements documents
O Design document
O Unit test plans
O Acceptance Test plans
O Integration test plans
O Programmer/maintenance manuals
O Project management development plans
O O her

Effort

Person-years	(tot prsyrs_est:pg 27)	(tot_prsyrs_rev:pg 27)	(tot_prsyrs_rem:pg 27)

Rework

		Estimate	*Measured based estimate*	*Actual*	
Percentage of effort	(perc_rwk:pg 27)	*O*	*O*	*O*	(rwk_msmt:pg 28)

NSDIR
National Software Data & Information Repository
Recurring Data Form
Metric collection Guide. Update

Submission Date:

Schedule

Suggested Milestone	Your Alternative Milestone	Planned Date *(proj_start_plan:pg 20)*	Actual Date *(proj_start_act:pg20)*
Start (REQUIRED)		*(proj_end Dlan:pg20)*	*(proj_end_act.pg20)*
Completion (REQUIRED)	*(mlst_name:pg 29)*	*(mlst_plan:pg 29)*	*(mlst_act:pg 29)*
System Design			
CSCI Development			

Effort

Suggested Activity	Your Alternative Activity	Planned Hours *(tot_stfhrs_plan:pg 20)*	Actual Hours *(tot_stfhrs_act:pg 21)*	Rework Hours *(tot_stfhrs_rwk:pg 21)*
All (REQUIRED)	*(stfhrs_name:pg 23)*	*(stfhrs_plan:pg 23)*	*(stfhrs_act:pg 23)*	*(stfhrs_rwk:pg 23)*
Management				
Environment				
System				
Requirements				

Quality

Suggested Priority	Your Alternative Priority	# Open Reports *(tot_open:pg 21)*	# CIosed Reports *(tot_closed:pg 21)*
All (REQUIRED)	*(prty_name:pg 28)*	*(open:pg 28)*	*(closed:pg 28)*
ONE			
TWO			
THREE			

Suggested Measurement	Your Alternative Measurement	Planned Total	Actual Total
KSLOC (REQUIRED)		(tot_ksloc_plan:pg 21)	(tot_ksloc_act:pg 21)
FP (REQUIRED)		*(totfp_Dlan:pg 21)*	*(totfp_act:pg 21)*
Doc. Pages (REQUIRED)		*(doc_pgs_plan:pg 22)*	*(doe ngs_act pg22)*
# CSCIs	*(msmt_name:pg 30)*	*(tot_size_act:pg 30)*	*(tot_sue_plan:pg 30)*

Bibliography

1. Adelman, Leonard. Evaluating Decision Support and Expert Systems. New York: John Wiley & Sons, 1992.

2. Ayers, Bradley J. and William M. Rock. Development of a Standard Set of Software Indicators for Aeronautical Systems Center. MS thesis, AFIT/GSS/ENC/92D-1. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1992 (AD-A258424).

3. Basden, Andrew. "Three Levels of Benefits in Expert Systems," Expert Systems: The International Journal of Knowledge Engineering and Neural Networks, 11: 99-107 (May 1994).

4. Berry, Diane C. "Involving Users in Expert System Development," Expert Systems: The International Journal of Knowledge Engineering and Neural Networks, 11: 23-28 (Feb 1994).

5. Byun, Dae-Ho and Eui-Ho Suh. "Human Resource Management Expert Systems Technology," Expert Systems: The International Journal of Knowledge Engineering and Neural Networks, 11: 109-119 (May 1994).

6. Card, David, Scott Hissam, and Renee T. Rosemeier. "National Software Data and Information Repository," Crosstalk: The Journal of Defense Software Engineering, 9: 4-8 (Feb 1996).

7. DeMarco, Tom. Controlling Software Projects. Englewood Cliffs NJ: Prentice-Hall Inc., 1982.

8. Department of the Air Force. Guidelines for Successful Acquisition and Management of Software Intensive Systems: Weapon Systems, Command and Control Systems, and Management Information Systems, Volume I. Software Technology Support Center, February 1995.

9. Dickerhoff, William G. and William J. Sommers. The Adaptation of the SEI's Capability Maturity Model to the Air Force Software Acquisition Management Process. MS thesis, AFIT/GSS/LSY/92D-3. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1992 (AD-A258419).

10. Fad, Bruce E. "Adapting Computer Aided Parametric Estimating (CAPE) to Process Maturity Levels," Report to Martin Marietta PRICE Systems, Los Angeles CA, December 1994.

11. Goel, Asish. “The Reality and Future of Expert System: A Manager’s View of AI Research Issues,” Information Systems Management, 16: 53-61 (Winter 1994).

12. Hochron, Gary. "Capture That Information on an Expert System," The Journal of Business Strategy: 11-15 (Jan/Feb 1990).

13. Holloway, Clark and Herbert H. Hand. "Who's Running the Store, Anyway? Artificial Intelligence!!!," Business Horizons: 70-76 (Mar/Apr 1988).

14. Joint Logistics Commanders, Joint Group on Systems Engineering. Practical Software Measurement: A Guide to Objective Program Insight: Version 2.1. Naval Undersea Warfare Center Division. Newport RI, 1996.

15. Jones, Caper. Applied Software Measurement. New York NY:McGraw-Hill Co., 1991.

16. Nikolopoulos, Chris and Paul Fellrath. "A Hybrid System for Investment Advising," Expert Systems: The International Journal of Knowledge Engineering and Neural Networks, 11: 245-250 (November 1994).

17. Office of the Assistant Secretary, Department of the Air Force. Memorandum for Distribution, Software Metrics Policy--ACTION MEMORANDUM. Pentagon, Washington DC 16 February 94.

18. Office of the Secretary of Defense, Department of Defense. Memorandum for Secretaries of the Military Departments, Specifications and Standards--A New Way of Doing Business. Pentagon, Washington DC 29 June 1994.

19. Rhines, Wally. "Artificial Intelligence: Out of the Lab and Into Business," Journal of Business Strategy: 50-57 (Summer 1985).

20. Srinivasan, Krishnamoorthy and Douglas Fisher. “Machine Learning Approaches to Estimating Software Development Effort,” IEEE Transactions on Software Engineering, 21: 126-136 (February 1995).

21. Umphress, David A., Victor M. Helbling, John R. Russell, and Charles Keene. “Software Process Maturation,” Information Systems Management, 11: 32-42 (Spring 1995).

22. Waterman, Donald A. "How Do Expert Systems Differ From Conventional Programs?," Expert Systems: The International Journal of Knowledge Engineering and Neural Networks, 3: 16-19 (January 1986).

23. Yoon, Youngohc. and Tor Guimaraes. “Assessing Expert Systems Impact on Users’ Jobs,” Journal of Management Information Systems, 12: 225-249 (Summer 1995).

Vita

Captain Derek F. Cossey is from Waco, Tx. He graduated from Texas A&M University with a degree in Aerospace Engineering in December 1988. After receiving his commission, he served as a project manager at the Phillips Laboratory, Edwards AFB, CA. His responsibilities included the management of in-house test facilities and in-house test plans, and contracts dealing with large space structure research. In November 1992, Capt Cossey was assigned to the Space Test Program at the Space and Missile Systems Center, Los Angeles AFB, CA. There he served as the Spacecraft Integration and Test Engineer and the Chief Engineer, Space Experiments Satellite Division. Capt Cossey entered the School of Logistics and Acquisition Management in May of 1995 in pursuit of a Master of Science in Systems Management.

Permanent Address: [redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1996	Master's Thesis

4. TITLE AND SUBTITLE
DEVELOPMENT OF A STANDARD SET OF INDICATORS AND METRICS FOR ARTIFICIAL INTELLIGENCE (AI) AND EXPERT SYSTEM (ES) SOFTWARE DEVELOPMENT EFFORTS

5. FUNDING NUMBERS

6. AUTHOR(S)
Derek F. Cossey, Captain, USAF

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
2750 P Street
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GSM/LAS/96S-3

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

SAF/AQRE
1060 Air Force, Room 4c283
Pentagon
Washington, DC 20330-1060

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 Words)*

The purpose of this research was to identify a standard set of indicators and metrics that can be used by program managers to improve their abilities to direct development efforts involving Artificial Intelligence (AI) and Expert Systems (ES). This research addresses two objectives. The first objective is to identify an appropriate set of software indicators and metrics to be used by government program offices for the management of development efforts involving software systems for AI and ES. The second objective is to demonstrate how the resources of the National Software Data and Information Repository (NSDIR) can be used in order to minimize the cost of the research endeavor and to demonstrate the value of the NSDIR as an information resource. A literature search identified a set of indicators and metrics that could be used by managers of AI and ES software development efforts. Data concerning AI and ES software development efforts were collected from the NSDIR. Unfortunately, substantiated conclusions regarding the value of the data in regards to AI and ES development efforts were unobtainable. The study did produce a recommended set of indicators and metrics that could serve as a feasible starting point for managers to use in the tailoring process for selecting indicators and metrics for AI and ES software development efforts.

14. SUBJECT TERMS
Artificial Intelligence, Expert Systems, Program Management, Contract Management, Defense Acquisition, Department of Defense

15. NUMBER OF PAGES
90

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL