23206

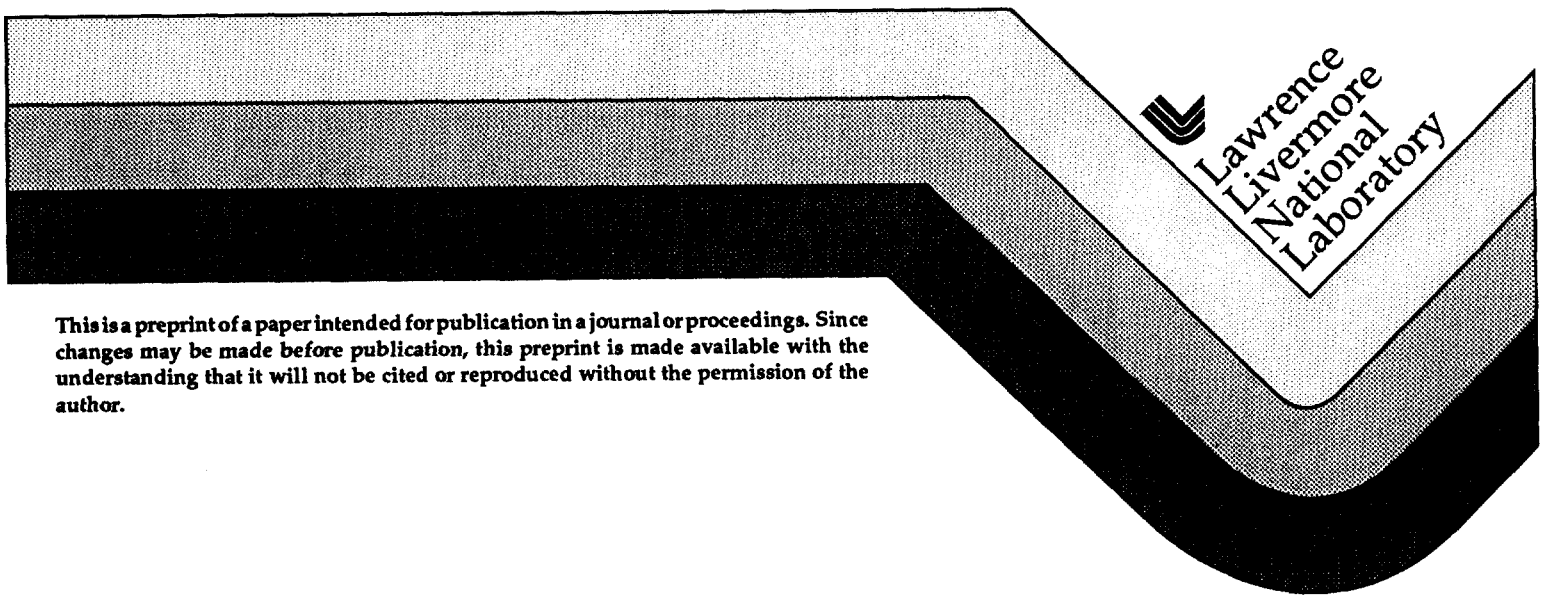
UCRL-JC-126231
PREPRINT

Designing for Success Software Evolution Process for the Department of Energy's Standard Security System

S. K. Allen

This paper was prepared for submittal to the
13th American Defense Preparedness Association Symposium & Exhibition
on Security Technology
Virginia Beach, Virginia
June 9-12, 1997

June 10, 1997



Designing for Success
Software Evolution Process for the Department of Energy's Standard Security System

Susan K. Allen
Lawrence Livermore National Laboratory,
P.O. Box 808, Livermore, CA 94550

Abstract

Computer software must evolve to support changing user needs, advances in computer technology, and continuous quality improvement. Control of the software development and evolution process becomes critical when the software is used in the protection of special nuclear material for the U.S. Government. The software evolution of the Argus security system is outlined as an example of a successful approach to this problem. Argus is a comprehensive, integrated security system incorporating intrusion detection, access control alarm assessment, command and control communications, event reporting and archiving. Special issues faced by the Argus team are outlined, and the methodology for software definition, delivery and control are described.

Introduction

The Argus security system is a comprehensive integrated security system incorporating intrusion detection, access control, alarm assessment, command and control communications, event reporting, and archiving. The Department of Energy (DOE), which funded the development of the Argus security system, has selected Argus as its standard security system for protecting DOE sites handling special nuclear materials. The Argus security system is operational at Lawrence Livermore National Laboratory (LLNL), Livermore, California; the Pantex Plant, Amarillo, Texas; Idaho National Engineering Laboratory, Idaho Falls, Idaho; and the Joint National Test Facility, a Department of Department facility, Falcon Air Force Base, Colorado Springs, Colorado. The evolution of the Argus security system is funded by DOE and Department of Department through operational sites as well as new sites implementing Argus (herein referred to as "sponsors").

The process of Argus software and hardware evolution originated in 1987 with the original software. The Security and Automation Technology Group (SAT) has implemented an effective process to support this evolution while minimizing operational costs and dependency on proprietary hardware and software. This process relies on interactions with Argus operational sites and communications with technical and operational users.

System and Process Design

The goal of the SAT staff is to deliver a system and products that meets security requirements of our sponsors, while maintaining a single system which is scaleable, flexible, configurable and responsive to the needs of each site..

Argus began as a design to meet LLNL's needs and has evolved to a site-configurable system. Argus has moved from a VAX-based VMS system to an Alpha-based VMS system. The staff has made conversions from third party specialized hardware and software to DOE developed hardware and software. Currently, the SAT staff is developing conversions for software encryption and replacements to current databases.

The Argus security system has an on-going change process that allows for continuous reassessment of the system and feed back from sponsors, which provides the opportunity to implement improvements.

The Argus management staff supports the concept of software evolution with continued maintenance key to design. SAT tasking is completed on a cost-incurred, best effort basis.

The following factors influence the change and success of the Argus software change process:

- Changes to DOE and DoD orders,
- Clear communication between development staff and sponsor sites,
- Advances in security technology - Argus technology must provide a reliable shield against adversaries using state of the art technology,
- Vendor dependencies - Argus must operate on hardware that is supported by software and hardware vendors for items such as operating system, databases, computers, displays, encryption and communication devices,
- Maintain a high level of security at a reasonable cost, and
- Having a well defined scope of requirements from the customer.

Argus proven practices in the current change process include:

- Regular meetings with sponsor site managers and end users to discuss and negotiate of needed functionality and performance of the system,
- Written commitment, to our sponsors, to deliver for during a system release,
- A well-defined tracking and change control process of system functionality and features,
- Continuous user feedback,
- Development completed in a secure limited area,
- A DOE "Q" Clearance development staff.

System Release Process

The software release process consists of four main phases: change control and prioritization, software development, product testing, and release and distribution. Strict levels of configuration management and quality assurance are maintained throughout this process.

The Technical Interchange Meeting (TIM) and client testing are crucial to the success of each release cycle.

- The TIM is an informational meeting for communicating upcoming release plans, discussing functionality, setting priorities, and soliciting feedback.
- Client testing is an opportunity for users to test the pre-released software and provide feedback to the developers before software is finalized.

To expedite and manage the product change, development and testing are done on isolated, stand alone systems of clustered computers, workstations and development field hardware. The LLNL production system is the stand alone operational system protecting LLNL assets.

Change Control and Prioritization

SAT uses a series of databases for tracking, maintaining, and reporting requests and deliverables to the process.

The Software Change Request (SCR) form is the foundation for the change process for system releases. This form allows users to submit requests for changes to the Argus product, which are tracked throughout the process from design, implementation and acceptance. A database for tracking SCRs is maintained with access and reports available to each operational site.

Each change cycle begins with the Technical Interchange Meeting (TIM), where representatives from each Argus sponsor site comes to LLNL to discuss their user needs and prioritization functionality and change requests for the next software release. Decisions for priority are generally based on site installation schedules, funding, and complexity of design which places a time constraint on completion. These meetings form the strong basis for communication with the SAT staff and Argus end users.

Selected SCRs are planned and committed upon completion of the TIM. The schedule for the next release is finalized and progress is tracked through the SCR database throughout the process.

Software Development

Argus software provides the foundation for complex levels of functionality (what it can do), performance (how well it can do it), adaptability (responsiveness to changing requirements), and configurability (ability to simultaneously support needs of different sites).

There are approximately 14 software developers, about five developers are of the original system design staff.

Argus consists of approximately 27 software products, which interface with each other to make Argus fully operational.

Argus software consists of approximately 700,000 lines of source code. Ada is the primary language used and was selected for its software engineering and maintenance features. The operating system selected was VMS using VAX host computers and converting to Alphas in the last two years. VMS provides features as redundancy of disks, processor and networks valued in the security environment. Version control is maintained using the Digital Equipment Corporation (DEC) Code Management System (CMS). Ada is also used in the field processors.

The development process consists of defining requirements, design and implementing code changes, unit testing, updating any associated documentation and performing a system build procedure.

Defining requirements for enhancements may be done by review of functional requirements documents from the site, site visits, and developer-SCR requester discussions of the functionality requested.

Software problems are handled by the developer recreating the problem on the development system and/or by information provided by the requester identifying and logging the problem.

Software design and changes are facilitated by a number of techniques used by the development staff; these include:

- Using open protocols
- Using modular design based on the client/server model
- Isolating external dependencies through Ada packaging
- Allowing site specific selection of preferences through the use of configurable data tables
- Designing Ada modules that allow extensive reuse within Argus Subsystems
- Using recognized industry standards, such as Standard Query Language (SQL) and Open Software Foundation (OSF) Motif. (1)

A major emphasis of the development staff during the design process is to maintain the configurabiltiy of the system and compatibility with existing hardware and operational features.

The coding process includes informal peer reviews of code changes. These reviews check implementation standards and coding style. Additional review examines the impact of the changes on the existing portions of the product. (1)

Unit testing is completed on the development system by the software programmer and a software development staff member using test programs.

Once coding and unit testing are completed the Argus software is built from source code including all new code changes and renumbered for release to the Operational staff for regression and quality assurance testing for system level releases. Incrementals for quick fixes of software problems are built into kits containing the executable files of the software changed.

Product Testing

After the kits or source builds are completed, the Operational staff maintains configuration management of the Argus product.

A separate, isolated test facility is maintained with each operational and development site's hardware configuration, with maps and databases created to simulate operational sites for testing.

System level regression testing takes place through a formal process of test procedures, tracking of testing with test logs, tracking and resolution of testing incidents and reporting. Testing is performed to certify that software changes have not altered basic Argus functionality and new features function as designed.

Before final release of code the sponsors are invited to LLNL for client testing. Client testing is valuable for three reasons:

- Gives Argus software an additional set of independent testing by independent testers
- To validate that functionality has been delivered as requested
- Training of end users of new functionality prior to delivery

There are approximately 500 individual test procedures that may be performed for any given release. A test summary of the testing results is delivered to the sites with each release.

Release and Delivery

Each set of product changes is formally released as a package.

A complete package is prepared containing release notes, technical notes, installation instructions, a test summary, and updated documentation. Each product is numbered for release and each package of products is numbered for full release and delivered to Argus operational sites with source code as well as current release executables.

The delivery of released software is tracked through a formal release process requiring management and Quality Assurance approval.

All Argus sites have isolated test systems for the initial installation and check out of software.

LLNL has agreed to be the test bed for the initial installation of the software on production. Once successfully running on the LLNL production system other sponsor sites will install on their production systems.

Software distributed from LLNL is protected as proprietary software of the DOE and LLNL, controlled distribution through LLNL's Automatic Data Processing Review and Inventory Office and limited to sites who have signed a non-disclosure or use of government developed property agreement for the control, distribution, and use of Argus source code.

The process is complete once each sponsor site has successfully, installed, tested, and accepted the release. Requesters sign off their SCRs after installation, testing, and verification at their site of their request. Only after a requester has signed off can an SCR be closed out.

Acknowledgments

I wish to thank Erv Behrin of Erv Behrin Security Consulting (2), Denis Schrader and Rick Wilson for their contributions to the content of this paper.

References

1. Eric F. Wilson, Software Evolution in Practice for the U.S. Department of Energy, Experience Report, IEEE Computer Society proceedings, September 19-23, 1994.

2. Erv Behrin, An Overview of the Argus Software Change Process prepared for Mason & Hanger - Silas Mason Co. Inc. under Contract #LHC000108 and provided to Lawrence Livermore National Laboratory, Security and Automation Technology group, March 25, 1996.

The work was performed under the auspices of the U. S. DOE by LLNL under contract no. W-7405-Eng-48

Technical Information Department • **Lawrence Livermore National Laboratory**
University of California • Livermore, California 94551