TASK: US40
CDRL: 03070
30 July 1991

US40 – Risk-Reduction Reasoning-Based Development Paradigm Tailored to Navy C^2 Systems

Informal Technical Data

STARS-SC-03070/001/00

30 July 1991

TASK: US40
CDRL: 03070
30 July 1991

INFORMAL TECHNICAL REPORT

For The

SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS)

RISK-REDUCTION REASONING-BASED DEVELOPMENT PARADIGM TAILORED TO NAVY C^2 SYSTEMS

STARS-SC-03070/001/00
Publication No. GR-7670-1219(NP)
30 July 1991

Data Type: A005, Informal Technical Data

CONTRACT NO. F19628-88-D-0031
Delivery Order 0003

Prepared for:

Electronic Systems Division
Air Force Systems Command, USAF
Hanscom AFB, MA 01731-5000

Prepared by:

TRW Systems Division
under contract to
Unisys Defense Systems, Inc.
Tactical Systems Division
12010 Sunrise Valley Drive
Reston, VA 22091

TASK: US40
CDRL: C3070
30 July 1991

Data ID: STARS-SC-03070/001/00

Distribution Statement "C"
per DoD Directive 5230.24
Distribution Authorized to U.S. Government Agencies and their Contractors:
Administrative (30 July 1991)

Developed by: TRW Systems Division under contract to
Unisys Defense Systems, Inc.

This document, developed under the Software Technology for Adaptable, Reliable Systems (STARS) program, is approved for release under Distribution "C" of the Scientific and Technical Information Program Classification Scheme (DoD Directive 5230.24) unless otherwise indicated. Sponsored by the U.S. Defense Advanced Research Projects Agency (DARPA) under contract F19628-88-D-0031, the STARS program is supported by the military services, SEI, and MITRE, with the U.S. Air Force as the executive contracting agent.

TASK: US40
CDRL: 03070
30 July 1991

INFORMAL TECHNICAL REPORT
RISK-REDUCTION REASONING-BASED DEVELOPMENT PARADIGM TAILORED TO NAVY C² SYSTEMS

Approvals:

Richard E. Creps 7/29/91

Task Manager *Richard E. Creps* *Date*

(Signatures on File)

TASK: US40
CDRL: 03070
30 July 1991

INFORMAL TECHNICAL REPORT

For The

SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS)

RISK-REDUCTION REASONING-BASED DEVELOPMENT PARADIGM TAILORED TO NAVY C^2 SYSTEMS

STARS-SC-03070/001/00
Publication No. GR-7670-1219(NP)
30 July 1991

Data Type: A005, Informal Technical Data

CONTRACT NO. F19628-88-D-0031
Delivery Order 0003

Prepared for:

Electronic Systems Division
Air Force Systems Command, USAF
Hanscom AFB, MA 01731-5000

Prepared by:

TRW Systems Division
under contract to
Unisys Defense Systems, Inc.
Tactical Systems Division
12010 Sunrise Valley Drive
Reston, VA 22091

Contents

List of Figures

1 INTRODUCTION

This report defines a STARS trusted, reuse-oriented Navy Command and Control (C^2) Process Model (NCCPM) for system development. The NCCPM describes the entire system development lifecycle from early concept through contract award, design, development and operations and maintenance with an emphasis on software development. The NCCPM description combines the STARS Composite Process Model (SCPM) documented in [21] and preliminary Navy C^2 domain analysis work contained in the Appendix to this report and in the Spiral 0 descriptions of Subsection 3.2..

This work integrates and adapts previous DARPA, STARS, SEI and industry process modeling work, as appropriate. The work incorporates the process model concepts and issues of risk–based activities; high performance, trusted system development; software reuse; library support for reusable assets; and domain considerations within the Navy C^2 application domain. These results directly address the STARS goals for a technology for building adaptable, highly reliable and cost effective software systems. Specifically, they provide a framework for the development of reuse–driven, trusted systems within the Navy C^2 application domain.

This is the second of two reports developed during STARS Task US40. The previous report was the *Draft Composite Paradigm Report*, defining the STARS Composite Process Model from which the NCCPM was derived. In these reports the words "process model" and "paradigm" are used interchangeably.

1.1 Background

The Phase I Process Model results of the DARPA/ISTO funded Advanced Computing Systems (ACS) Project at TRW provides a basis for the SCPM and the NCCPM. In particular, the development of systems requiring trust and high performance requires an increased, early emphasis on clear identification of risks, risk mitigation activities and development process controls. For a specific application, this emphasis includes the risks and characteristics native to the application domain. The domain aspects for tailoring to a Navy C^2 system generate important activities in the development process. The process model documented in this final report incorporates the domain analysis activities and precontract effort essential to the development of a reuse–based Navy C^2 system. These activities are defined within the precontract discussions of Spiral 0 in subsection 3.2.

STARS planning includes work to establish reuse process building blocks, reuse libraries and domain specific environments with a goal of instantiation of a domain–specific Software Engineering Environment (SEE) for reuse. The existence of a Navy C^2 reuse infrastructure will be a fundamental requirement for practical reuse–based system development.

The risk–driven characteristics of the SCPM are rooted in the Boehm Spiral Model [1]. Starting with the Spiral Model as a foundation, key elements of the DARPA ACS trusted system Process Model were identified. As described in [7], the key elements of the DARPA ACS Process Model are the following:

- The domination of the development process by risk management;
- The integration of engineering for trust and performance;
- The specialization for Ada across multiple activities of the lifecycle;
- The integration of other software engineering techniques (analysis, assurance and configuration control).

The DARPA ACS Process Model was defined to integrate security, broad trust and performance engineering with a modern risk-driven system development paradigm for Ada. The traditional waterfall development process has often been ineffective as a model for large scale, complex systems, particularly those with stringent trust and performance requirements. The DARPA ACS Process Model is intended to guide and support the project process to increase the productivity of the development team and the quality of the resulting system while reducing the inherent project risks for that particular domain.

The SCPM focused on reuse-driven activities that were needed to expand the DARPA ACS Process Model to the STARS environment. Its scope included the life-cycle development process once a contract award had begun and after a certain amount of domain analysis work was already accomplished.

1.2 Focus of the Current Work

This task addresses the inadequacy of current software development paradigms, especially for trusted systems, and focuses on the adaptation of a STARS-relevant process model based on previous work. In this task the following results were integrated.

- The current results of the SCPM work
- The results from preliminary domain analysis work in the Navy C^2 application domain
- The definition of precontract activities that are essential precursors to system development
- The identified domain risks applied to the spiral process
- The determination of Government-specific activities

The DARPA ACS Process Model foundation for high performance trusted systems in Ada provides an opportunity for software improvement within the STARS environment. The current subtask leverages the TRW DARPA/ISTO process model work and incorporates specific reuse and application domain considerations. The application domain tailoring efforts to trusted Navy C^2 systems as part of this subtask were documented as a separate report which is included as an appendix to this document.

Reuse analysis is integrated into all aspects of the SCPM foundation and the resulting process model. Process control and well-defined transitioning criteria in high-risk, early spirals of activity remain a primary consideration within the process model.

1.3 Subtask Approach

Top level functions for the NCCPM approach are illustrated in Figure 1, US40.2 Subtask Approach. The basic inputs to and outputs from the next level subtasks and the relationships of the activities are represented. Major results are the SCPM and the NCCPM. The synergistic relationship of process model and Navy C^2 domain analysis results, working group and other activities and exchanges is also illustrated by Figure 1.

In this task, TRW adapted, tailored and integrated the TRW DARPA/ISTO trusted system process model results and the current results from the STARS reuse process paradigm, the results from process model application efforts and results of the SEI process research and the STARS prime contractor initiatives relevant to the process model definition. This resulted in a composite paradigm which provided a trusted system development process model for STARS.

TRW initially reviewed STARS reuse information and reuse research documentation and worked with the relevant STARS subcontractors and primes to obtain information and insight on aspects of STARS reuse goals and reuse software development approaches. In particular, the reuse activities and the conceptual framework of the Unisys Reusability Library Framework (RLF) provided information for this task. TRW discussed the process model within the ongoing Process Model Working Groups. SEI process research and other relevant process model efforts were analyzed and integrated as appropriate into the resulting composite process model.

The risk-driven, Spiral Model basis provided a foundation for a high integrity, high performance system development process that focuses on reuse principles. Specific risk mitigation approaches such as modeling and prototyping may provide candidate reuse components for high risk software development. A general definition of the basic spirals of activity that includes reuse considerations may provide reusable, tailorable objectives and transitioning criteria within the paradigm.

Each key element of the process model based on the TRW DARPA/ISTO work was analyzed with respect to the reuse paradigm and other process model work as required. The key process model elements, primary motivation and primary constraints are illustrated in Figure 2. Reuse analysis goes beyond Ada considerations, and reuse was integrated into all aspects of the process model foundation. Process control remains a primary consideration within the process model description, and the importance of well-defined transitioning criteria in high-risk, early spirals of activity is emphasized.

The process model analysis resulted in the documentation of the composite formulation, the SCPM. TRW analyzed the SCPM results, refined the paradigm and formulated a composite

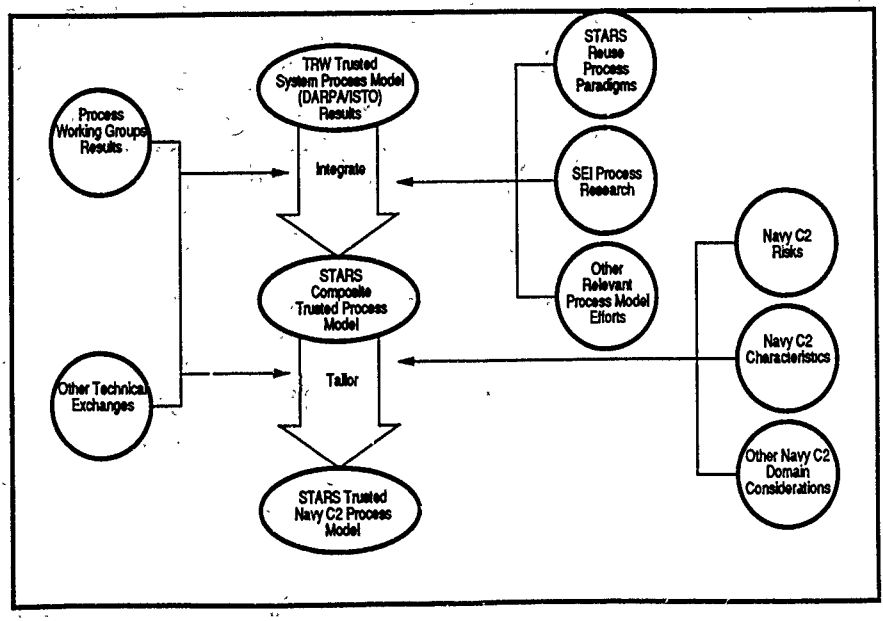


Figure 1: US40.2 Subtask Approach for the Process Model Adaptation

paradigm that represents a trusted development, reuse process model for STARS.

To perform the domain-tailoring function, TRW identified the domain-specific characteristics and risks for Navy C^2 Systems. Navy C^2 domain experts within TRW provided the primary inputs for this subtask.

Through technical exchanges and analyses of real world projects, TRW determined and incorporated specific characteristics and risk drivers for the development of Navy C^2 systems. TRW then analyzed the applicability of these characteristics to the NCCPM definition.

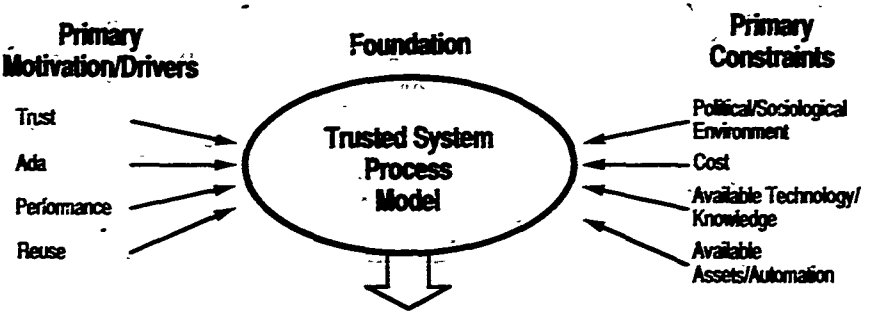
Based on the identified Navy C^2 characteristics and risks and process model guidance, TRW defined criteria for assessing these risks and determined activities and approaches for risk mitigation. These risk drivers and mitigation approaches provided some of the specific spirals of activity appropriate for a Navy C^2 system development process. Identification of specific Navy C^2 domain characteristics will ultimately help to provide candidate components for reuse within the reuse paradigm.

The composite process model results and the Navy C^2 domain analysis work were combined to tailor the SCPM to Navy C^2 domain-specific developments. TRW defined a composite and domain-specific paradigm for the Navy C^2 trusted system development process. The resulting model represents the integration of information from the two reports, from domain expert reviews and from working group technical exchanges. The resulting model provides a prototype process description applicable to trusted Navy C^2 developments and provides a basis for the goals of machine representation in the Navy C^2 domain environment.

2 TRUSTED NAVY C^2 PROCESS MODEL BACKGROUND

The development of trusted Navy C^2 systems remains a high risk endeavor today. To define a process for system development that identifies and mitigates major project risks is one way to address the development challenges. Such a process description is itself a challenge, particularly when the process scope includes the entire lifecycle of Navy C^2 systems from early concept through maintenance. The NCCPM description lists and describes basic process activities within major project stages that are spiral based. The Navy C^2 domain analysis and precontract activities are defined as part of the process and described in an early set of spirals, denoted Spiral 0, Concept through Contract Award. Many issues and considerations are addressed including the current DoD-STD-2167A standard that guides most defense system developments and the Trusted Computer System Evaluation Criteria (TCSEC) [19] that helps define top level computer security requirements.

This section introduces spiral process model concepts and the trust, reuse and Navy C^2 domain adaptations presented in this report. The subsections provide an overview of the application domain risks and characteristics and summarize risk mitigation activities for reuse-based, trusted Navy C^2 system developments. This section also provides correspondences from the DoD-STD-2167A, from DoD 5200.28-STD (TCSEC) and from reuse and human interface products to the major spirals of activity in the process model.



Key Elements of the STARS Composite Process Model

Risk Management
- Formal risk management techniques
- Modeling
- Planning for reuse
- Prototyping and demonstrations
- Analysis of reuse candidates
- Incremental development

Ada
- Homogeneous representation
- Language support for reuse
- Consistent metrics

Engineering for Trust, Performance and Reuse
- Architecture assessment (modeling; prototyping)
- Critical mechanisms prototyping
- Integration of critical reusable assets

Control and Assurance
- Reasoning-based analysis/assurance
- Reuse of assurance results
- Configuration management and control
- Control of reuse library

Figure 2: Derivation of the STARS Composite Process Model

2.1 Spiral Process Model Concepts

The key Spiral Model features are risk management, robustness and flexibility. The Spiral Model was developed at TRW [1] as an alternative to the more conventional, primarily linear–based waterfall process model in use today. The Spiral Model attempts to provide a disciplined and flexible framework for software development that accommodates activities such as prototyping, reuse and automatic coding as part of the process. A consequence of the Spiral Model flexibility is that managers and developers are faced with choices at many stages of the process, and with choice comes risk. This overview of the basic spiral concept is taken from Appendix 1 of [7].

The Spiral Model views the development process in polar coordinates. The r coordinate represents cumulative project cost, the w coordinate represents progress to date. A cycle of the model is an increase of 360 degrees in w. The plane is divided into four quadrants that represent different kinds of activities.

- Quadrant 1: Determination of objectives, alternatives and constraints; a time to review plans and translate them to specific activities for the spiral
- Quadrant 2: Evaluation of alternatives, identification and resolution of risks; activities such as analysis, evaluation, modeling and prototyping are conducted
- Quadrant 3: Development activities; actual products, i.e., study results, documents and code are are produced
- Quadrant 4: Review and planning for future cycles; planning and management activities including formal reviews and planning documents are some of the possible activities in this quadrant.

The boundary between Quadrant 1 (clock position of 9:00) and Quadrant 4 (9:45) represents a commitment to carry the project through another cycle. In this conceptual representation, the w (progress) coordinate does not move evenly with time. Some spirals may require months to complete while others are of very short duration. Similarly, while increasing w denotes progress within a spiral, it does not necessarily denote progress toward project completion.

As a framework for development, the model emphasizes early planning, software engineering and development activities. These activities require the support of a wide variety of tools. There is heavy reliance on frequent and extensive reviews to ensure the project stays on track.

The DARPA ACS Process Model is described in detail in [7]. TRW produced this spiral-based paradigm for high–performance, trusted systems in Ada by tailoring and enhancing the Spiral Model to incorporate the following characteristics:

- The impact of trust and performance are pervasive;

- Trust and performance decisions made at the beginning are irrevocable;
- Implications of trust principles are poorly understood;
- The conceptual foundations of trust are fragile and incomplete; and
- Significantly greater emphasis is placed on analysis and assurance.

Common crucial risks for high performance trusted system developments were used to define a general pattern for early development activities in the DARPA ACS Process Model. Figure 3 illustrates the conceptual view of this model.

In Figure 3, the additional sectors that appear in Quadrants 2, 3 and 4 are used to represent the continuation of certain risk mitigation activities over different spirals. For example, trust assessments may occur throughout spirals 2, 3, 4 and even into maintenance within the risk mitigation quadrant. Sectors that represent modeling and prototyping activities occur in both Quadrant 2 and Quadrant 3 since continuing product results are sometimes appropriate for these risk mitigation activities

The SCPM description incorporates reuse activities into the DARPA ACS Process Model foundation and provides lists of activities for each of five major cycles within a perceived STARS reuse framework. Details of the SCPM may be obtained in [21]. The conceptual view of the SCPM in that report is presented in five separate spiral diagrams to reduce the volume and complexity of the graphic representation for the reader.

The conceptual view of the NCCPM described in the current report is also partitioned for a more manageable presentation. Within the Navy C^2 domain, the NCCPM expands the SCPM scope to include domain analyses and precontract activities and provides an explicit identification of Government activities. Precontract activities are defined in Section 3 and viewed in a separate Spiral 0 which consists of explicitly defined subspirals of activities. The post-contract spirals of activity for NCCPM are partitioned in Section 4 into development contractor activities and Government activities.

2.2 Trusted Navy C^2 Application Domain Overview

For this subtask, TRW identified domain-specific characteristics and risks for Navy C^2 systems with an emphasis on trust and reuse considerations. This preliminary domain analysis was accomplished through technical exchanges, analyses of real world projects and meetings with Navy C^2 domain experts. The scope, approach and results of this work are documented in the Appendix to this report.

As a result of the domain analysis, a set of Navy C^2 characteristics has been identified. The identified characteristics include:

1. Secure/trusted system

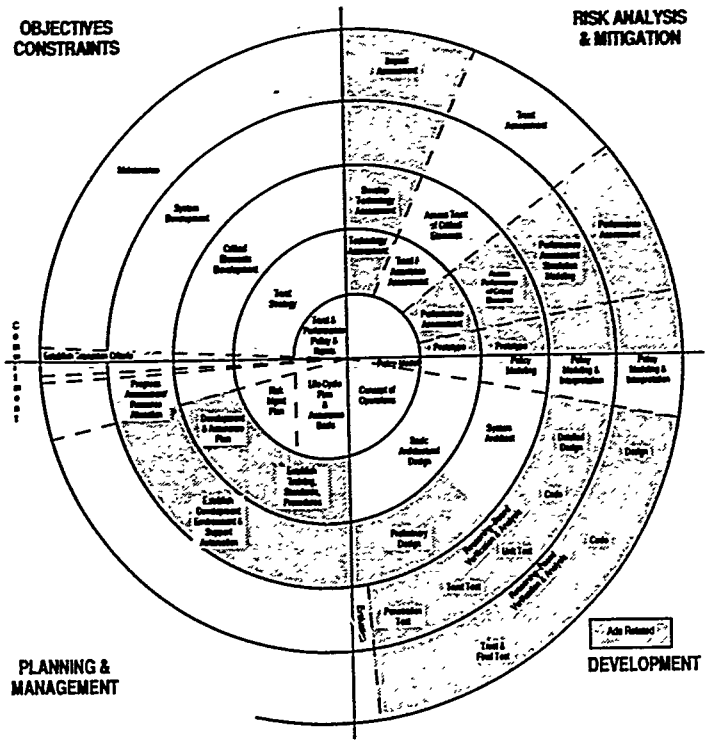


Figure 3: Initial Foundation for the STARS Composite Paradigm

2. Man-machine interface
3. Communications
4. Message handling
5. Open architecture
6. Adherence to hardware standards
7. Supportable by Navy logistics
8. Reliability, maintainability, and availability
9. Data fusion
10. Decision aids and automated support functions
11. Man-in-the-loop
12. Distributed architecture
13. Flexible architecture
14. Near real-time system operation.

Each characteristic is discussed in the Appendix.

TRW identified three categories of risk for a trusted Navy C^2 system development: technic programmatic and both technical and programmatic. The latter category was used to defi risks that were not cleanly partitioned into either of the first two categories since some ris contain strong elements of both categories.

The most crucial risk for any system development is the potential for misunderstanding misinterpreting the system requirements. This risk area has programmatic elements; howe it is categorized here as a technical risk. TRW identified the following Navy C^2 domain ri for system development:

- Both technical and programmatic risks
  1. Reuse
  2. Trust Policy
  3. Evaluations, Certifications and Accreditation
- Technical risks
  1. Understanding and Communicating Requirements

RISKS	MITIGATION	SPIRAL
Reuse		
- Inadequate management support	- Joint Government/contractor education, communication, and committment	- Spiral 0 - 5
- Lack of understanding of reuse requirements	- Domain analysis for all potential systems to be developed - Use of domain experts - Experimentation with assets	- Spiral 0,1 - Spiral 0,1 - Spiral 0,1
- Inadequate planning for adaptability	- Coordination between program personnel within the application domain - Preliminary reuse plan	- Spiral 0 - Spiral 0
- Lack of reuse asset library and support tools	- Research, analysis of automation support - Analysis for library definition and creation - Set up and maintain library	- Spiral 0 - Spiral 0,1,2 - Spiral 1,2,3,4,5
- Obsolete assets	- Design for open architecture, evolvability	- Spiral 0,1,2,3
Trust Policy		
- Inadequate and/or erroneous formulation of trust policy - Inaccurate trust policy model	- Detailed trust policy analysis, prototypes, security concept of operations, and integration of policy mandates from all authorities - Experienced modeler with understanding of mission and trust requirements	- Spiral 0,1,2 - Spiral 1,2
Evaluations, Certifications, System Accreditation		
- Lack of well defined accreditation requirements	- Identification of accrediting authority, roles, responsibilities, preliminary accreditation plan, security environment definition, and requirements - Determine software certification requirements - Maintain close relationship with Designated Approval Authority throughout life cycle	- Spiral 0,1 - Spiral 0,1 - Spiral 0 - 5
- Impact of system/software modifications on reaccreditation	- Analysis of changes to trusted elements of system and software; trust assessments	- Spiral 0 - 4 for reused systems - Spiral 5
Understanding and Communicating Requirements		
- Lack of understanding of technical requirements	- Models and prototypes of security elements and reusable components	- Spiral 0,1,2,3,5
- Misinterpretation or miscommunication of requirements	- Requirements analysis - Requirements traceability - Joint Government/User/Contractor meetings Simulation of external interfaces - Concept of Operations	- Spiral 0,1 - Spiral 0 - 5 - Spiral 0 - 5 - Spiral 2,3,4,5 - Spiral 0,1

Figure 4: Risk Summary

RISKS	MITIGATION	SPIRAL
- Lack of understanding of how operator uses system	- Site visits - Critical task analysis - Models and prototypes of MMI commands and displays - Demos of prototypes to users - User (actual people at operational sites, not just people from Program Office) meetings with development contractor - Verify operator interface with user - Concept of Operations	- Spiral 0 - 5 - Spiral 0,1,2 - Spiral 0,1,2,3 - Spiral 0,1,2,3 - Spiral 0 - 5 - Spiral 1,2,3 - Spiral 0,1
Frequently Changing Requirements		
- Mission changes due to world events	- Design for maximum flexibility and evolvability	- Spiral 0,1,2,3
- Rescoping of requirements due to budget constraints	- Design for maximum flexibility and evolvability	- Spiral 0,1,2,3
Assurance		
- Technology limitations	- Research and analysis	- Spiral 0,1,2
- Pervasiveness of trust functions	- Analysis, models, formal methods, prototypes, extensive documentation, testing	- Spiral 0 - 5
- Cost of assurance	- Trade-off studies - Employ assurance techniques early	- Spiral 0,1,2 - Spiral 0,1,2
Trust Skill Specialization		
- Limited number of trust analysts	- Training	- Spiral 1,2
- Isolation of trust team	- Integration of trust engineering, system engineering, software development team and activities	- Spiral 0 - 5
Architecture		
- Formulation of generic reuse architecture	- Domain analysis - "Levels and Views" architecture	- Spiral 0,1 - Spiral 0,1
- Satisfy trust policy	- Security architecture	- Spiral 0,1,2
Technology		
- Lack of understanding of integrity and assured service	-Research and trade-offs	- Spiral 0,1,2
- Inexperience at domain analysis	- Training - Analysis - Modeling	- Spiral 0,1 - Spiral 0,1 - Spiral 0,1
- Limited availability of products: trusted, reusable and SEE	- Assessment of available products - Testing - Prototyping	- Spiral 0,1 - Spiral 0,1,2 - Spiral 0,1,2
- Immaturity of Ada language	- Analysis and trade-offs	- Spiral 0,1
Performance		
- System trust added functionality	- Analysis and trade-offs - Performance modeling and benchmarking - Prototyping	- Spiral 0,1,2,3 - Spiral 0,1 2,3,4 - Spiral 0,1,2,3

RISKS	MITIGATION	SPIRAL
- Integration of reused software assets	- Analysis and trade-offs - Performance modeling and benchmarking - Incorporation of actual reuse software benchmarks into performance model (from previous use) - Prototyping	- Spiral 0,1,2,3 - Spiral 0,1,2,3,4 - Spiral 0,1,2,3 - Spiral 0,1,2,3
- Ada language	- Analysis and trade-offs - Selection of mature, performance-tested Ada compiler - Coding standards (SDP)	- Spiral 0,1,2,3 - Spiral 0,1 - Spiral 0,1,2
- Inability to meet Navy C^2 near real-time mission requirements	- Analysis and trade-offs - Simulation of external interfaces - Performance modeling/benchmarking - Use of actual site data for development and testing	- Spiral 0,1,2,3 - Spiral 2,3,4,5 - Spiral 0,1,2,3,4 - Spiral 2,3,4,5
Ada-related		
- Limited number Ada-experienced developers	- Training	- Spiral 0,1
- Immature support tools	- Analysis and trade-offs	- Spiral 0,1
- Integration of Ada and non-Ada code	- Encapsulation of non-Ada code interfaces in separate packages	- Spiral 2,3
- Conversion of non-Ada code to Ada	- Experienced designers to convert the code	- Spiral 2,3
Documentation		
- Reusable software documentation	- As-built documentation - Guide reuse - Asset certification guidance - Clear, concise documentation standards	- Spiral 4,5 - Spiral 2,3,4,5 - Spiral 2,3,4,5 - Spiral 2,3,4,5
- Lack of integration of trust into software documents	- Integration of trust engineering, system engineering, software development team and activities	- Spiral 0 - 5
Standards		
- Evolving open system and trust standards	- Research, analyze,and incorporate current standards	- Spiral 0,1,2,3,4
- Misunderstanding of coding standards	- Clear, concise PDL, coding, language, comment, naming and data description standards - Enforce coding standards (SDP)	- Spiral 0,1,2,3,4 - Spiral 0,1,2,3,4
Trust Assurances During Maintenance		
- Invalidation of original assurance	- Trust and impact analysis, modeling, reverification, and recertification	- Spiral 5
- New personnel for software maintenance	- Adequate documentation, training and communication	- Spiral 2,3,4,5
Programmatic/Political/Sociological		
- Poor communication	- Technical exchanges - Management meetings - Documented meeting follow-up	- Spiral 0 - 5 - Spiral 0 - 5 - Spiral 0 - 5

RISKS	MITIGATION	SPIRAL
- Different cultures	- Technical exchanges - Management meetings - Documented meeting follow-up	- Spiral 0 - 5 - Spiral 0 - 5 - Spiral 0 - 5
- Staffing instability	- Good people management	- Spiral 0 - 5
Opposing Interests		
- Multiple organizations	- Identify who is responsible for what document when planning CDRL items - Track responsibility	- Spiral 0,1,2 - Spiral 1 - 5
Cost Constraints		
- Technical risks	- Evolve system engineering effort to site implementation - Continue systems engineering throughout program - Additional systems engineering resources up front	- Spiral 3,4 - Spiral 0,1,2,3,4 - Spiral 0,1,2
- Budget cuts	- Flexible to schedule and requirements changes - Trade-offs	- Spiral 1 - 5 - Spiral 1 - 5
Schedule Constraints		
- Tighten or lengthen schedule	- Flexible to schedule and requirements changes - Trade-offs	- Spiral 1 - 4 - Spiral 1 - 4
- Poor planning and insufficient tracking of progress	- Automated tracking mechanism - Risk Management Planning	- Spiral 0 - 5 - Spiral 0 - 5
Program Coordination, Management and Assurance		
- Complexity of trusted Navy C^2 system development	- Automated tracking mechanism - Management reviews, engineering and WBS - Risk Management Planning	- Spiral 0 - 5 - Spiral 0 - 5 - Spiral 0 - 5
- Unrealistic budget	- Proper initial planning	- Spiral 0,1

2.4 NCCPM Correspondences to DoD-STD-2167A and TCSEC

As the NCCPM is intended to be applicable to relevant software development standards, it is useful to give examples of correspondence with prevailing standard documents, deliverables, etc. This is done by providing tables of various activities and deliverables and indicating which of the spirals is most probable to initiate or encompass the item.

Current Navy C^2 systems are developed under the DoD-STD-2167A for software development. Therefore, any process model for Navy C^2 system development in the near term will need to address the requirements of the 2167A standard. This subsection provides a first-cu mapping of the 2167A activities and products to the major spirals of activity defined for the NCCPM.

The general deliverables and activities given in DoD-STD-2167A software development standard are listed in Figure 5 and Figure 6. The software development phase given in 2167A is shown in the left hand side of the figure, the specific deliverable or activity within that phase in the center, and the spiral(s) expected to correspond is to the right.

Similarly, the TCSEC is used to guide trust requirements in the development of Government systems, and some correspondence between the NCCPM and the TCSEC would be useful. The information in [20] was reviewed and interpreted to support the TCSEC and application system trust to NCCPM mapping included in this subsection. The TCSEC guidance requires interpretation before it can be applied to a system development. Additional documents to assess trust risks and requirements for the specific environment are necessary.

Other less-well-defined requirements exist for reuse, and human engineering. Some examples have, however, been included along with the TCSEC mapping in Figure 7, Figure 8, and Figure 9, with again, the expected spiral initiating or encompassing the activity or deliverable on the right hand portion of the figures, opposite the item as given.

3 THE NCCPM PART I: PRE-CONTRACT ACTIVITIES

The NCCPM Part I describes pre-contract process model activities defined for a major spiral denoted Spiral 0: Concept through Contract Award. These process model activities present the domain analysis for reuse and the analyses, system engineering and products associated with early planning by the Government and potential development contractor(s). Spiral 0 is defined in terms of 5 subspirals.

3.1 The Early Process

Before a software development can be defined with reuse as a primary driver, a domain analysis and reuse planning must be done and a reuse methodology and support environment must be available. Thus, a defined methodology, domain analysis and the definition and development of reusable assets, generic architectures and support tools are all necessary

Phase	2167A Deliverables: Documents & Reviews	Spiral
System Requirements Analysis	• Preliminary System Specification: - System/Segment Specification (SSS) - Prime Item Development Specification (PIDS), and/or - Critical Item Development Specification (CIDS) • System Requirements Review (SRR)	0,1 1
System (Requirements) Design	• System Specification (SSS, PIDS, and/or CIDS) • System/Segment Design Document (SSDD) • Preliminary Software Requirements Specification(s) (SRS) • Preliminary Interface Requirements Specification (IRS) • Software Development Plan (SDP) • System Design Review (SDR)	1 1 1 1 1 1
Software Requirements Analysis	• Software Requirements Specification(s) (SRS) • Interface Requirements Specification (IRS) • Software Specification Review (SSR)	2,3 2,3 2
Preliminary Design	• Software Design Document(s) (Preliminary Design) (SDD) • Software Test Plan (Test ID's) (STP) • Preliminary Interface Design Document (IDD) • Preliminary Design Review (PDR)	3 3 3 3
Detailed Design	• Software Design Document(s) (Detailed Design) (SDD) • Software Development Files (SDF) • Software Test Description(s) (Cases) (STD) • Interface Design Document (IDD) • Critical Design Review (CDR)	4 4 4 4 4
Coding and CSU Testing	• Source Code Listings • Source Code • (No Mandated Review)	4 4 -
CSC Integration and Testing	• Software Test Descriptions(s) (Procedures) (STD) • Test Readiness Review (TRR)	4 4
CSCI Testing	• Updated Source Code • Software Test Reports(s) (STR) • Computer Resources Integrated Support Document (CRISD) • Computer System Operator's Manual (CSOM) • Software User's Manual (SUM) • Software Programmer's Manual (SPM) • Firmware Support Manual (FSM) • Version Description Document(s) (VDD) • Software Product Specification(s) (SPS) • CSCI Functional and Physical Configuration Audits	4 4 4 4 4 4 4 4 4 4

Figure 5: DoD-STD-2167A Deliverables – NCCPM Correspondence

Phase	2167A Activities	Spiral
Software Development Management	• Software Development Process	All
	• Formal Reviews/Audits	All
	• Software Development Planning	1
	• Risk Management	All
	• Security	All
	• Subcontractor Management	All
	• Interface with Software IV&V Agent(s)	All
	• Software Development Library	2,3,4,5
	• Corrective Action Process	3,4
	• Problem/Change Report	3,4
Software Engineering	• Software Development Methods	All
	• Software Engineering Environment (SEE)	All
	• Safety Analysis	All
	• Non-Developmental Software (NDS)	All
	• Computer Software Organization	1,2,3
	• Traceability of Requirements to Design	1,2,3
	• High Order Language (HOL)	All
	• Design and Coding Standards	1-5
	• Software Development Files (SDF's)	3,4,5
	• Processing Resource and Reserve Capability	1,2,3,4
Formal Qualification Testing (FQT)	• Formal Quality Test Planning	4,5
	• Software Test Environment	4,5
	• Independence in FQT Activities	1,4,5
	• Traceability of Requirements to Test Cases	4,5
Software Product Evaluations	• Independence in Product Evaluation Activities	1,4,5
	• Final Evaluations	4,5
	• Software Evaluation Records	4,5
	• Evaluation Criteria	4,5
Software Configuration Management	• Configuration Identification	1-5
	• Configuration Control	2,3,4,5
	• Configuration Status Accounting	3,4,5
	• Storage, Handling, and Delivery of Project Media	1-5
	• Engineering Change Proposals	1-5
	• Specification Change Notice	1-5
Transitioning to Software Support	• Regenerable and Maintainable Code	3,4,5
	• Transition Planning	4,5
	• Software Transition and Continuing Support	4,5
	• Software Support and Operational Documentation	4,5

Figure 6: DoD-STD-2167A Activities – NCCPM Correspondence

Trust Documents and Activities	Spiral
• Security Concept of Operations	1
• Trust, Risk and Vulnerability Analysis	1-5
• Philosophy of Protection	2
• Security Policy Model	2
• Descriptive Top-Level Specifications (DTLS)	3
• Formal Top-Level Specifications (FTLS)	3
• Security Policy Model to FTLS Correspondence	3,4
• DTLS and FTLS Correspondence to Trusted Computing Base (TCB)	4
• Covert Channel Analysis	3,4
• Functional Testing	4
• Security Testing	4
• Security Specific Documentation	4
- Trusted Facility Manual	
- Security Features User's Guide	
- Configuration Management Plan	

Figure 7: Trust – NCCPM Correspondence

Reuse Documents and Activities	Spiral
• Preliminary Reuse Plan	1
• Reuse Plan	2
• Reusable Assets Documentation	4
• Configuration Management (CM) Tracking	4
• Reuse Impact Assessment	5

Figure 8: Reuse – NCCPM Correspondence

Human Engineering Documents	Spiral
• Human Engineering Plan	1
• User Interface Document (UID)	3

Figure 9: Human Engineering – NCCPM Correspondence

elements for a reuse-based Navy C^2 development.

The reuse process should begin very early, before the system concept phase. Planning for reuse actually needs to start at the Government policy stage and should be motivated by management goals and directions. To implement a reuse strategy, much technical analyses and political compromise will be required within the Navy C^2 domains of interest.

There is considerable current research in domain analysis. Some of the work is described in [3]. The NCCPM incorporates the results of the TRW preliminary Navy C^2 domain analysis to identify domain characteristics and risks. Domain analysis process activities are identified and described in this report in Spiral 0.

3.2 Spiral 0: Concept Through Contract Award

The specific focus for Spiral 0 is the development of trusted, high performance, reusable systems in the Navy C^2 domain. In the current exercise, the SCPM is expanded by Spiral 0 to include essential domain analyses and precontract activities not addressed in the earlier model.

Figure 10 illustrates activities performed by the Government and potential development contractor(s) that may be included in Spiral 0 for a Navy C^2 system acquisition involving software and hardware reuse, trust, and high performance. The Government activities are "bolded" in the figure and marked with a "(G)" in the text. Government activities incorporate the analyses, studies and special tasking provided by support (e.g., SETA) contractors. The development contractor(s) activities are totally independent of Government activities, and the development contractor(s) can in no way influence the Government activities. The Spiral 0 figure provides a conceptual view of all activities from beginning of the Government Concept Exploration phase through contract award and negotiation. Some of the activities may occur in parallel or may overlap which is not obvious in the conceptual figure. Spiral 0 starts after Government Milestone 0 (Mission Need Determination) when the Mission Element Needs Statement (MENS) has been signed. In actual practice, some of these activities may be combined or may not be required depending on project size and complexity (program acquisition category) and specific requirements. Many of the activities are as appropriate for the current emphasis on system integration of interoperable components, GOTS and COTS as they are for large scale system developments, the more traditional approach to Navy C^2 system acquisition. Spiral 0 is described below by five sub-spirals, moving clockwise around the spiral, beginning with Objectives and Constraints and ending with Planning and Management for each sub-spiral.

3.2.1 Initial Planning

Spiral 0_1 is the first sub-spiral in the risk-driven acquisition process and includes early planning, requirements definition, trust and reuse analyses and the initial identification of risks and constraints. The objective for the Government during this sub-spiral is to begin

funding approval while defining requirements, developing and enhancing the Navy C^2 generic architecture and identifying potential areas for software and hardware reuse for the specific application. The objective for the potential development contractor(s) during this sub-spiral is to identify available domain knowledge and use this knowledge to develop an early architecture. In summary, Spiral 0_1 may include:

- (G) Identification and control by the Government of political and funding constraints.
- Identification and evaluation of cost and budget, schedule, political, and technical and performance constraints and customer and user preferences.
- (G) and contractor (separately). Identification of broad risk categories, focusing primarily on reuse, trust and high performance.
- Development of early architecture. A "Levels and Views" methodology may be used to document the domain knowledge and develop an early architecture. The "Levels and Views" methodology is described below.
- (G) Analysis of reuse feasibility. This activity includes estimation of percentage of software that can be reused concentrating on inserting portions of software with (mostly parameter) changes and minor modifications rather than newly created software and the analysis of the feasibility and role of COTS and GOTS products and NDIs, hardware and software. The analysis of reusable assets in addition to software includes the analysis of elements such as high level designs, architectures, data base models, domain analysis results, certification assurance and other documents to support reusability considerations. This effort also includes an analysis and evaluation of the automated support and process methodology (i.e., the software engineering environment (SEE)) available and analysis of what is needed for reuse process activities within the Navy C^2 domain.
- (G) Poll of user group(s) for operational requirements inputs. Include definition of reuse requirements at same time as operational requirements, and make determination of trust and performance requirements compatibility.
- (G) Initiation of pre-accreditation analyses and activities to identify responsibilities and top level security policies and requirements.
- (G) Description and enhancement of a generic architecture for Navy C^2 systems on which the current application can be based.
- (G) Development of an Operational Requirement (OR) document (or other initial document to define the procurement and begin funding approval in support of the acquisition process) by the Government customer. The procurement documents to support the acquisition process will vary depending on the category of the acquisition. Approval of the OR constitutes the completion of the Government Concept Exploration phase and generation of the Program Element (PE) number and Program Element Description (PED).

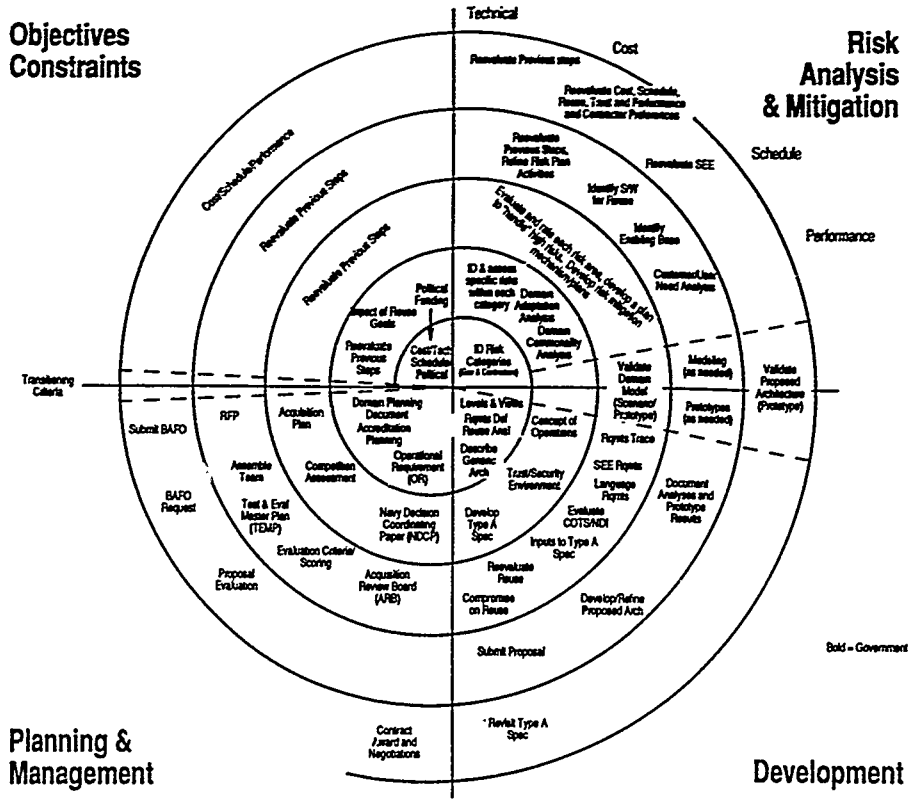


Figure 10: A Conceptual View of Navy Command and Control Spiral 0

- (G) Planning and identifying potential requirements for certifications and accreditation.
- Development of a Domain Planning Document that bounds the domain, scopes and plans the domain analysis activities, establishes guidelines and standards, SEE support, and assesses the costs, risks and benefits of the effort. This effort uses planning documents and knowledge of previous systems. The planning must be within programmatic goals and constraints. The risks identified here initiate the risk analysis and mitigation activities of later (development) subspiral/spirals.
- (G) and contractor (separately). Establishment of transitioning criteria for next project spiral. Both contractor and Government elements within their individual, spiral-based processes must determine when a subspiral/spiral is complete.

"Levels and Views" Methodology. The "Levels and Views" methodology, cited above, disciplines the process of defining a system architecture, requiring engineering attention to all facets, or views, of the system early. The purpose of the levels and views approach is to develop a top-down comprehensive system architecture with emphasis on system issues, risks, and "too hards." The steps of this approach are as follows:

- Identify the architectural framework of the system using the levels and views methodology
  - View: a perspective on the architecture of the system (e.g., topology, functions, interfaces)
  - Level: the varying degrees of detail to which a perspective may be defined
- Develop the system's architectural elements emphasizing the issues and risks. Here, reuse is a primary issue.
- Automate the architectural description

Three layers are used in the development of the architecture views: Mission layer, Implementation layer, and Administration layer.

The Mission layer describes the system's objectives. The objectives are the goals in functionality, reuse, trust, performance, interfaces, and topology that the system is to meet regardless of the "how" of making that happen. The Mission layer includes views with definitions as follows:

- Functional: the mission-related and support activities to be performed by the system It represents the analysis of the functional requirements to be supported by the subsystems and delivered to the sites. For this effort it includes the analysis of the common functionality of multiple systems within the Navy C^2 domain for reuse considerations and analysis of any Government-defined generic domain functions.

- Interface: the transition points and the methodology for sharing information and/or control among or within segment components. This exchange may be a data exchange, where the information to be developed is the data content, format, and rates, or it may be a processing exchange, in the form of initiation, interaction, or control.
- Topology: the set of sites and components where the segment functionality will be performed and the combinations of the system's building blocks which will support that functionality in its variety of locations. The data for the topology view also provides the basis for site installation surveys and planning.
- Trust/Security: Analysis to determine required trust levels and cost and technology constraints allows a top level view of the security safeguards and assurance required and their feasibility. This analysis must be coupled with reuse requirements and criticality issues.
- Performance: the description of the behavior of the end items that make up the system. This documents performance requirements and drivers or, at the very least, the assumptions used by the designers. Performance requirements analysis must incorporate potential impacts of the trust and reuse requirements.

The Implementation layer describes how the system is to fulfill its mission and achieve its objectives. The Implementation layer includes views with definitions as follows:

- Software: the computer executable program modules used to provide the segment functionality.
- Hardware: the equipment used to process, store, display, and communicate system data and software.
- Data: the representation of information pertinent to the mission and support functions used to monitor, control, evaluate, or perform the activities of the system.
- Man-Machine Interface (MMI): the means by which the system is presented to its users, and the mechanism for the users to interact with it. The MMI plays a crucial role in determining the effectiveness and the user acceptability of a system, and is best developed using a specialized engineering discipline.

The Administration layer describes the things that must be done to make the system accessible and effective, and the operating parameters within which it must be managed The administration layer includes views with definitions as follows:

- Procedures: describes those human-oriented activities relevant to the control of the system to enable it to achieve its mission objectives.
- Management. describes the activities which have been established to control development, operations, and maintenance of the system to achieve program objectives

3.2.2 Acquisition Strategy and Funding

Spiral 0_2 is the second sub-spiral in the acquisition process and includes development of the Acquisition Plan, Type A System Specification and Concept of Operations, as well as software commonality analysis and competition assessment. The prototyping spans both the risk analysis and mitigation quadrant and the development quadrant. In summary Spiral 0_2 may include:

- (G) Control of political and funding constraints by the Government.
- Re-evaluation of cost and budget, schedule, political, and technical and performance constraints and preferences and the impact of reuse-driven goals.
- Identification of specific risks and initial assessment. A real world example of Navy C^2 system acquisition risks is: absence of a Memorandum Of Understanding (MOU) between two related programs where program A is relieved of a requirement that program B is depending upon for reuse.
- Domain commonality analysis – the goal is to model likeness between systems in the domain in support of reuse goals with an output of a domain dictionary. The domain dictionary includes terms and definitions on the language of the application domain, including the relationship of terms.
- Domain adaptation analysis – the goal is to model differences between systems in the domain and determine adaptation requirements imposed by the domain. Anticipated areas of adaptation may include: flexibility in operation, mission, environment, site, platform, user, and technology.
- Development of Concept of Operations. Initial draft should include mission statement, physical and performance characteristics, operational and trust constraints and manning, operations requirements, goals and desires and support required from logistics, training and personnel. In some cases (e.g., should there be a multilevel secure operational requirement), a separate Security Concept of Operations may be desirable.
- (G) Development of Type A System Specification. The Type A Specification should identify clearly reuse, trust and performance requirements. The Type A System Specification is written during the Government Demonstration and Validation phase.
- (G) Development of Navy Decision Coordinating Paper (NDCP) or other documentation to support the acquisition process. The NDCP would include program description, goals and thresholds, threat considerations, reuse issues, acquisition strategy, schedule and funding. Approval of the NDCP provides the funding profile for the Program Objective Memorandum (POM) submission.
- (G) Development of Acquisition Plan (AP). The AP would include objectives, strategy, and planning requirements. The AP must be developed for Acquisition Review Board approval during Government Milestone II.

- Assessment of the competition to include identification, strengths, weaknesses, and strategies.
- (G) and contractor (separately). Establishment of transitioning criteria for next project spiral. Both contractor and Government elements within their individual, spiral-based processes must determine when a subspiral or spiral is complete.

3.2.3 Acquisition Review and Request for Proposal (RFP)

Spiral 0_3 is the third sub-spiral in the acquisition process and includes prototyping for validation of domain model, assembly of team for competition and RFP release. In summary, Spiral 0_3 may include:

- (G) Control of political and funding constraints.
- Re-evaluation of cost and budget, schedule, political, technical, performance, trust and reuse constraints and preferences. Includes assessments of potential for use of COTS and GOTS products and NDIs in system.
- Evaluation and rating of risks within each area. This activity includes development of a draft risk management plan for handling high risks and defining risk mitigation mechanisms and plans. An example of "handling" accepted known risks may be conducting user interface meetings to achieve acceptance of proposed product.
- (G) Determination that all Government participants are on the "reuse bandwagon," all of the requirements are covered and well understood in-house and by the user community.
- Initial prototyping based on operational scenarios.
- Early validation of domain model using prototype(s), simulations and analysis as feasible.
- Requirements traceability - This activity provides opportunity to reassess known ambiguous requirements.
- Determination of language requirements - Consider constraints imposed due to reuse requirements.
- Determination of requirements for SEE support and evaluation of candidate tools.
- Prioritization of requirements - Anticipate compromise by the Government.
- Provide inputs to Type A System Specification - Requires coordination with Government. The inputs are only provided by the development contractor if the Government asks for industry comments on the Type A System Specification.

- (G) Evaluation of candidate COTS, GOTS and NDIs.
- (G) Re-evaluation of reuse – Is the current need compatible with past reuse efforts and future reuse goals; what are the specific risks associated with reuse?
- (G) Meeting of the Acquisition Review Board (ARB) for approval of the Acquisition Plan. This meeting constitutes completion of Milestone II and approval for program go ahead to a full scale engineering development contract procurement.
- (G)Development of Test and Evaluation Master Plan (TEMP) for approval by Operational Test and Evaluation Force (OPTEVFOR). The TEMP must be completed during Milestone II.
- (G) Generation and release to industry of RFP package.
- Research and assessment of anticipated proposal evaluation criteria and scoring based on previous RFPs from customer and similar procurements.
- Assembly of proposal team (subcontractors) and signing of teaming agreements
- (G) and contractor (separately). Establishment of transitioning criteria for next project spiral. Both contractor and Government elements within their individual, spiral-based processes must determine when a subspiral or spiral is complete

3.2.4 Proposal and Best and Final Offer (BAFO)

Spiral 0_4 is the fourth sub-spiral in the precontract acquisition process and includes identification of reuse software and other assets, development and refinement of proposed architecture, proposal submittal and evaluation, and BAFO. Re-evaluation of risks and constraints are continuing. In summary, Spiral 0_4 may include:

- (G) Control of political and funding constraints.
- Re-evaluation of cost and budget, schedule, political, and technical and performance constraints and preferences.
- Evaluation and rating of risks within each area. This activity includes development and refinement of a risk management plan for handling high risks as well as definition of risk mitigation mechanisms and plans.
- (G) Confirmation that all Government participants are on the "reuse bandwagon," all of the requirements are covered and well understood in-house and by the user community.
- Identification of lower levels of software reusability. Includes identification of enabling component base. This activity will help identify the lower levels of reusability (e.g., math operations, user interfaces, operating system, data structures and manipulations, information management subsystems, and communications).

- Conduct of customer and user need analysis. Factored into this process are a preliminary need analysis, trust and accreditation requirements, cost, schedule and political constraints, technical limits, desirable COTS and GOTS products and NDIs, availability, feasibility and adequacy of support tools (SEE support) and operational concept all which support a verified customer need.
- Development and refinement of a proposed architecture – Review architecture specifications, trust impacts to architecture, traceability to domain model, perform trade–offs (provide rationale), and develop guidelines for using generic architecture.
- (G) Compromise on reuse requirements – contributing factors are cost and schedule restrictions, user community acceptance, technology limits, accreditation needs and other systems constraints.
- Generation and submittal of technical, management, and cost proposal to customer
- (G) Proposal evaluation with concentration on system reuse.
- (G) Request for BAFO to contractors with responses submitted to customer.
- (G) and contractor (separately). Establishment of transitioning criteria for next project spiral. Both contractor and Government elements within their individual, spiral–based processes must determine when a subspiral or spiral is complete.

3.2.5 Contract Award

Spiral 0_5 is the final sub–spiral in the precontract acquisition process and includes final validation of proposed architecture, any required revision of Type A System Specification prior to contract award, and finally contract award and negotiation. In summary Spiral 0_5 may include:

- Re-evaluation of cost, schedule, reuse, trust and performance constraints and preferences.
- Evaluation and rating of risks within each area. This activity includes development of a plan for handling high risks as well as development of risk mitigation mechanisms and plans.
- Re-evaluation of candidate tool support and overall SEE applicability.
- (G) Reconfirmation that all Government participants are on the "reuse bandwagon," all requirements are covered and well understood in–house and by the user community
- Revisit documents and other product assets that tangibly support reuse
- Validation of proposed generic, reuse–based architecture using prototypes and simulations.

- (G) Revisit Type A System Specification – refine requirements based on inputs from previous sub–spirals.
- (G) Award of the Full Scale Engineering Development (FSED) contract for a trusted, high performance Navy C^2 system incorporating reuse.

4 THE NCCPM PART II: POST-CONTRACT AWARD ACTIVITIES

The post–contract award NCCPM provides guidance for the early identification of trusted Navy C^2 project risks and for the determination of activities to address those risks. Under the process paradigm, reuse, trust and performance engineering are integrated with modern software engineering practices and supported by the tools of a flexible SEE that satisfies the needs of the Navy C^2 application domain.

The NCCPM emphasizes the integration of various engineering practices, the use of Ada throughout multiple phases of development, and the inclusion of a spectrum of risk reduction development, analysis, and reasoning–based assurance techniques and tools. Configuration management is an extremely important mechanism for coordination and status accounting within the NCCPM having the process dynamic activity sequencing and reuse emphasis.

Many kinds of personnel, activities and products are required for the development of high–performance, trusted Navy C^2 systems in Ada, and the process descriptions for the lifecycle from contract award through maintenance are necessarily voluminous. This section provides a high level description of the overall process with an emphasis on the activities of the development contractor.

The activities of the Government, which include Government support contractors, are described separately here and are listed outside of the process conceptual spirals to avoid excessive complexity and to support ease of understanding. Government activities for each major spiral are important to the overall process. These activities provide management, control and technical oversight to the complex Navy C^2 system development. Government participants include Navy military and civilian personnel, other DoD, intelligence and various agency personnel as required and support contractors who perform special consulting, IV&V and SETA functions as needed.

4.1 Overview

Like the SCPM [21], the NCCPM is viewed conceptually with 5 major project stages that are defined within a risk–driven spiral paradigm. Each major spiral stage consists of multiple risk–driven activities that may themselves be modeled as subspirals within the bounds of the larger spiral that contains them. Additionally, there may be subspirals of activity that overlap major spirals and/or extend across several spirals.

The concurrency and overlapping potential of the spiral–based model activities makes con

ceptual, graphical visualization difficult. The conceptual spirals used in this section to illustrate the development contractor activities should be interpreted without assuming time duration, complexity or exact sequencing of activities.

During each spiral, four *generic* classes of activities are carried out in sequence Each class is represented as an activity quadrant transversed clockwise during each cycle. In the first quadrant, beginning at 9 o'clock, objectives, alternatives for achieving those objectives and constraints on possible alternatives are identified. This may result in the more precise determination of activities to be conducted and any products to be developed within the spiral. In the second quadrant, alternatives are evaluated in terms of probability and cost of failure, and potential magnitude of payoff. This is primarily a task of information gathering and analysis, involving prototyping, analytic modeling, interviews and surveys, literature searches and/or other techniques. In the third quadrant, one or more of the favorable alternatives are selected and pursued. In the early spirals, pursuit may mean making and documenting strategic technical decisions. In later spirals, it may mean further refinement of prototypes, formal analysis and modeling or undertaking such product development steps as producing plans, specifications, designs or even a completed system. Reasoning-based techniques have a role in both the second and third quadrants as the attendant modeling, specification and analysis activities can support either risk mitigation by providing alternatives or product development for such products as a performance specification or a formal top level specification for trust.

The spiral illustrations include activity sectors within quadrants for types of activities that may extend beyond a single spiral or may sequentially occur throughout a number of major spiral stages.

The process activities defined for the NCCPM are at a mid to high level of description for this full life-cycle process model. While the granularity of process description varies, TRW attempted to cover the full range of possible activities at a consistent level. The activity lists provide a base for the goals of automated process management. Each activity can be broken down further, and the dependencies among activities can be more explicitly defined to provide detailed process building blocks for process automation.

4.2 NCCPM Spiral 1 Through Spiral 5

The domain-specific considerations for reuse-based, trusted Navy C^2 system developments are reflected throughout the development process in each major spiral of activity of the NCCPM. This process description is an adaptation of the SCPM, and five major development spirals are defined for the NCCPM as in the SCPM. The major differences are that activities here are more specific to the Navy and DoD environment, and explicit Government activities are added to the process in a separate listing.

Within the NCCPM, engineering for Navy C^2 asset reuse, mission critical trust and near-real-time performance must be integrated into the overall system and software engineering process. This requires the integration of DoD, Navy, intelligence, war planning and other

Government and industry standards, practices, documentation, tools, and teams of specialists. Depending on the specific risk, the engineering process activities may be integrated and take place as part of one or more major spiral cycles. A Navy C^2 supportable SEE that guides and assists reuse and trust activities by employing integrated assurance tools, a knowledge-based process manager and a library facilitator may also be required. Where high trust is the mandate, the SEE must provide for formal, reasoning-based engineering methods and tools. Reuse of trusted assets will require high confidence in asset integrity as well as early agreements with accreditation officials on the role and acceptability of reuse in the trust assurance process.

There are common risks that are inherent in a reuse-based, trusted Navy C^2 application domain. These risks are identified and mitigation approaches are summarized in subsection 2.3 of this report. This risk identification led to a refinement of the process activities of the SCPM. The common pattern of activities that address trust and reuse risks in the earlier TRW trusted system process model work proved to be appropriate for the Navy C^2 domain. Additional domain-specific activities were interwoven into the earlier process descriptions to formulate the NCCPM. The process model description in this report includes the subsection 2.3 descriptive summary of risks, risk mitigation approaches and their correspondences to the defined NCCPM Spirals 0–5 along with the process activities lists and spiral illustrations in Sections 3 and 4.

4.2.1 Initial Project Plans and Analysis of Reuse, Trust and Performance Requirements – Spiral 1

Initial system requirements for reuse and trust in the Navy C^2 domain, including requirements for reuse approach, trust policy, assurances, asset qualification and trust evaluation, may be conceptually difficult, ambiguously stated, unrealistic, and in conflict with other requirements.

In particular, secure and mission-critical trust and performance requirements may be opposing, and the issues of reuse are further complicated by this conflict. Furthermore, the engineering consequences of reuse and trust requirements, especially with respect to near-real-time performance, are likely to be far-reaching and obscure, even to experienced software and system engineers. Consequently, the first set of activities advanced by the NCCPM includes analysis of the cost, implications, and achievability of initial reuse, trust and performance requirements.

Activities in Spiral 1 also include the preliminary planning for technical and management functions within the Navy C^2 domain under the risk-driven spiral process. The specific activities for the development contractor in Spiral 1 may include all or some of the activities listed in Figure 11.

These activities are also illustrated in Figure 12 which presents a conceptual view of Spiral 1 They define the early activities and planning required to address the Navy C^2 development risks. In actual practice, some of these activities may be combined, may not be required

Quadrant 1 - Objectives & Constraints	Quadrant 2 - Risk Analysis & Mitigation
• Clarification of trust policy, review of trust principles and their historical interpretation and application • Identification of reuse policy and goals • Determine how to apply the Process Model (PM) to the specific application • Project overview • Initial staffing and training	• Initial assessment of trusted and untrusted reusable assets (other than COTS products) and their component level and system level reuse implications • Assessment of emerging trusted and trust-compatible COTS products, including with vendors about plans for future products • Assessment of support capabilities of library and SEE and available technology for reuse and trust goals • Initial identification and analysis of major project risks • Critical task analysis • Dialogue with evaluation and accreditation authorities to clarify trust criteria and evaluation procedures and implications of reuse of trusted assets
Quadrant 3 - Development	**Quadrant 4 - Planning & Management**
• Requirements interpretation, including identification of unachievable or high-risk trust and performance requirements • Development of written interpretations of reuse, trust and performance requirements • Development of informal trust policy • High-level system architecture • Clarification of basis for assurance of trust policy enforcement in developing systems, particularly for reusable, trusted assets • Documentation: - Concept of Operation/Security Concept of Operation - System Specification (SSS, PIDS, and/or CIDS) - System/Segment Design Document (SSDD) - Preliminary Software Requirements Specification(s) (SRS) - Preliminary Interface Requirements Specification (IRS) - Software Development Plan (SDP)	• System Requirements Review (SRR) • System Design Review (SDR) • Development of a reuse plan (for current reuse and future reuse capabilities) • Development of a life-cycle plan that emphasizes approximate budgetary and schedule milestones, incorporates reuse and risk management strategies and describes the techniques and tools used to assess progress and to provide management visibility and control during subsequent spirals • Human Engineering Plan • System Engineering Management Plan (SEMP) • Quality Assurance (QA) Plan • Configuration Management (CM) Plan • Development of a risk management plan and establishment of transitioning criteria for next project spiral

Figure 11: Spiral 1 Activities

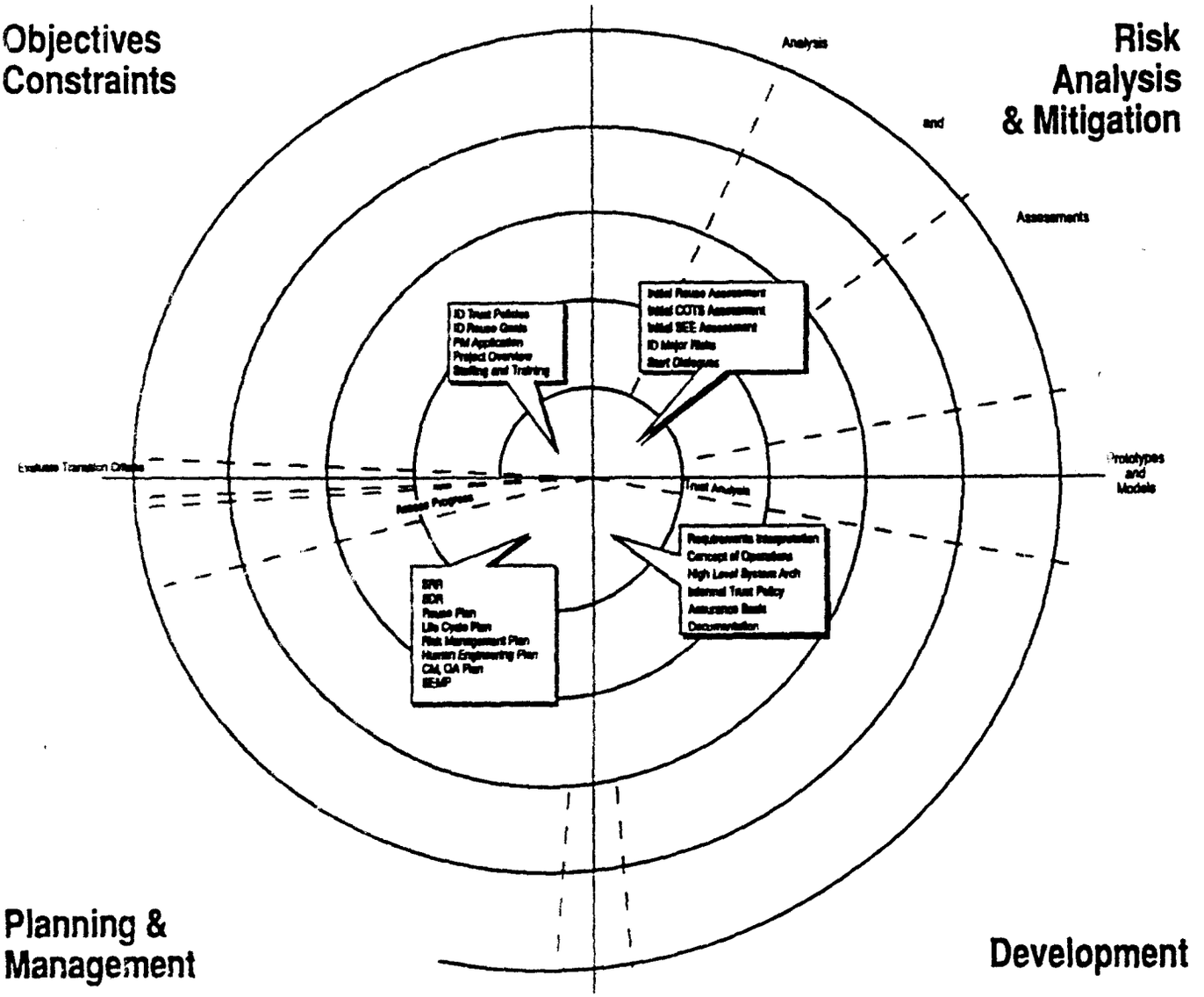


Figure 12: A Conceptual View of Spiral 1: Initial Project Plans and Requirements Analysis

or may be addressed in later spirals depending on project size and complexity and specific requirements.

The Navy C^2 Government activities that support the Spiral 1 risk-driven analyses, planning, and documentation are listed below. These activities include IV&V and SETA Support.

Spiral 1: Navy C^2 Government Activities

- Review CDRL items, including trust documents
- Attend SRR and provide comments and action items
- Attend SDR and provide comments and action items
- Respond to action items assigned to the Government
- Provide Document Trouble Reports (DTRs) to contractor
- Participate in DTR resolution meetings
- Maintain requirements traceability
- Negotiate Engineering Change Proposals (ECPs), as necessary
- Approve Specification Change Notices (SCNs), as necessary
- Provide inputs to contractor domain analysis
- Set up reuse library
- Approve, accept and support SEE
- Brief Designated Approval Authority (DAA)
- Determine accrediting authority
- Determine accreditation requirements
- Develop accreditation plan
- Coordinate certification and accreditation activities with operational sites
- Schedule and visit sites with development contractor
- Schedule and participate in user meetings with development contractor
- Provide inputs to critical task analysis
- Verify operator interface
- Provide Government Furnished Equipment and Information (GFE and GFI), as required

- Participate in Government, User and Contractor meetings (management and technical)
- Provide software benchmarks from Program A to Program B for reuse software
- Plan site implementation
- Plan Technical Evaluation (TECHEVAL)
- Plan Operational Evaluation (OPEVAL) with Operational Test and Evaluation Force (OPTEVFOR)
- Provide contract evaluation and grading
- Resolve funding and schedule issues
- Keep development contractor aware of changing threat, mission or requirements by documenting
- Review and reassess project risks
- Approve updated risk management plan
- Resolve and complete transitioning criteria

4.2.2 Reuse and Trust Enforcement Strategy and Basic Architecture – Spiral 2

After the initial Navy C^2 system reuse and trust requirements analysis, a strategy or philosophy for enforcing the reuse methodology and the trust policy must be developed. Additional assessments may be appropriate for technology considerations, process model application and SEE support including reuse library mechanisms, automated process management, and risk management, asset qualifier and tracker and language analysis tools.

The trust policy refinement is perhaps best accomplished by formulating a hypothetical trust enforcement architecture that embodies high-risk trust features and requirements and incorporates trusted, reusable assets as feasible. The hypothetical architecture is then evaluated for expected performance, robustness, functionality, and impact on untrusted component behavior and structure. The components of the hypothetical architecture may include existing hardware or software components that have been adapted for trust, emerging trusted COTS products, or entirely new custom-developed elements. Some of the Navy C^2 system trends toward specific COTS and GOTS are identified in the Appendix to this report. The evaluation of the hypothetical architecture may be limited to "paper and pencil" analysis, or more likely will involve hands-on experiments or prototypes to investigate key characteristics of potential components.

The use of formal methods to model and analyze the required trust and performance properties of the architecture may also be appropriate. An assurance plan is needed to define the appropriate assurance activities based on earlier assessments of reused components, trust needs and cost feasibility Unachievable trust and performance requirements, and high-risk

architectural decisions are identified. Interpretations of trust evaluation criteria that are non-trivial, or novel in approach, are outlined, the impacts of reuse are identified, and the rationale may require discussion with evaluation or accreditation authorities.

Initial performance budgets for key trust features may also be identified. Training standards and procedures for employees and future system users that emphasize reuse and trust principles must be developed. The project schedule as well as the SEMP, the risk management, reuse, CM and QA plans may need revision. The plans must consider such reuse and trust issues as re-evaluation of trusted components, reuse and integration of trusted assets in a Navy C^2 system environment and integration of heterogeneous trusted components. These plans establish the risk mitigation activities and transitioning criteria for the next project spiral(s). A project assessment is necessary before transitioning to the next spiral.

The activities described above are illustrated in Figure 13, and illustrated in Figure 14, A Conceptual View of Spiral 2.

The Navy C^2 Government activities that support the Spiral 2 risk mitigation for trust strategy and basic architecture are listed below. These activities include the IV&V and SETA contractor support.

Spiral 2: Navy C^2 Government Activities

- Review CDRL items including trust documents
- Attend SSR and provide comments and action items
- Respond to action items assigned to the Government
- Provide DTRs to contractor
- Participate in DTR resolution meetings
- Maintain requirements traceability
- Negotiate ECPs, as necessary
- Approve SCNs, as necessary
- Continue implementation of reuse library
- Brief DAA
- Coordinate certification and accreditation activities with operational sites
- Maintain Certification and Accreditation Plan
- Schedule and visit sites with development contractor
- Schedule and participate in user meetings with development contractor

Quadrant 1 - Objectives & Constraints	Quadrant 2 - Risk Analysis & Mitigation
• Refinement of trust strategy/philosophy into the Philosophy of Protection and refinement of reuse enforcement strategy for the Navy C^2 environment • Identify trust constraints • Objective determination, assessment and tracking of early Process Model (PM) application	• Additional assessments of technology • Analyze reuse capabilities • Assessment of initial SEE support • Attend user meetings; site visits • Initiation of any prototypes needed to validate/refine trust and reuse approaches
Quadrant 3 - Development	**Quadrant 4 - Planning & Management**
• Development of Security Policy Model (formal or informal) • Philosophy of Protection • Basic software architecture definition that provides required trust and applies reuse as feasible • Tailor SEE for Navy C^2 project-specific needs • Develop Software Requirements Specification(s) (SRS) and Interface Requirements Specification (IRS) • Trade-off studies • Document engineering notes • Conduct technical and management reviews and walkthroughs as needed	• Conduct Software Specification Review (SSR) • Establishment of training standards and procedures • Revisitation and update of project schedule and Lifecycle Plan with configuration management and reuse support • Development of assurance plan • Revision of the SEMP and reuse, CM and QA plans as needed • Revision of the risk management plan and establishment of transitioning criteria for the next project spiral • Assessment of project progress and transitioning criteria achievement

Figure 13: Spiral 2 Activities

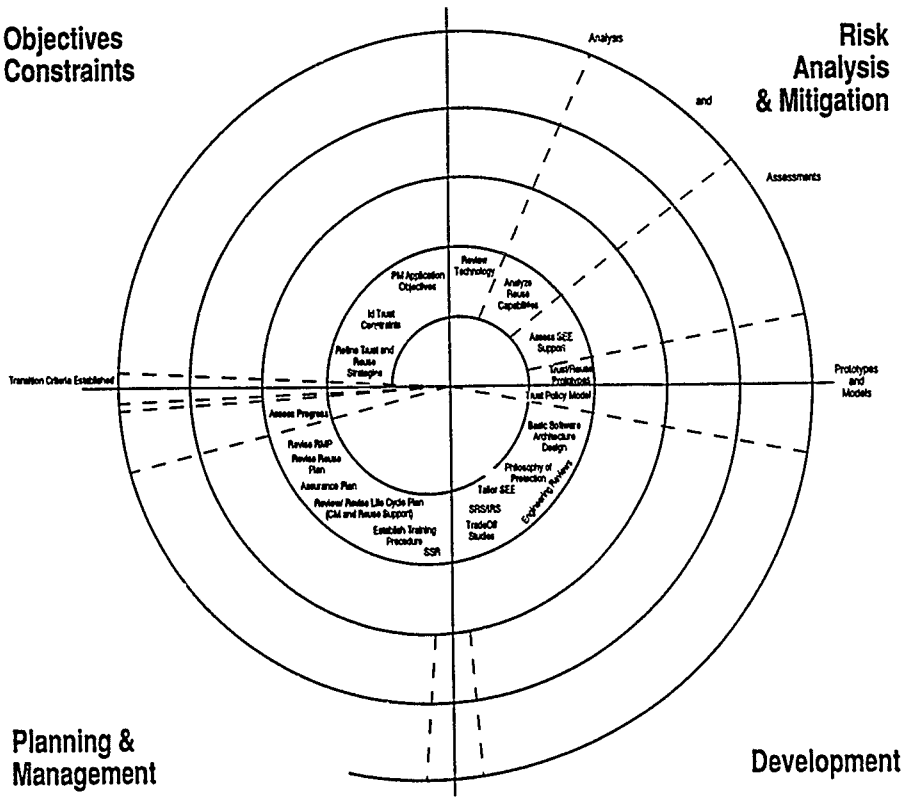


Figure 14: A Conceptual View of Spiral 2. Reuse and Trust Strategy and Basic Architecture

- Review critical task analysis
- Verify operator interface
- Provide GFE and GFI, as required
- Participate in Government, User and Contractor meetings (management and technical)
- Plan site implementation
- Plan TECHEVAL
- Plan OPEVAL with OPTEVFOR
- Provide contract evaluation and grading
- Resolve funding and schedule issues
- Keep development contractor aware of changing threat, mission or requirements by documenting them
- Review and reassess project risks
- Approve updated risk management plan
- Resolve and complete transitioning criteria

4.2.3 Critical Elements and Architecture Refinement – Spiral 3

This set of Navy C^2 system development risk-reduction activities verifies the achievability of reuse, trust and performance requirements, and establishes a foundation for system design This is accomplished by prototyping critical elements of a candidate policy enforcement architecture and/or experimenting with critical reusable assets. The system design must allow for evolvability and open architecture solutions in the Navy C^2 environment These activities are to provide empirical evidence that an architectural solution is within reach and to define its underlying approach. The prototype may be based on a Navy-supplied generic reuse architecture for the Navy C^2 domain with reusable assets or built from real components, stubs, or a combination of the two. Ada may be used even at this early stage The hypothetical architecture must show evidence of:

- Successfully applying and integrating reusable assets;
- Enforcing reuse methodology, designing for future reuse;
- Satisfying trust performance requirements and not preventing the satisfaction of other performance requirements;
- Enforcing trust policy; and

- Complying with trust assurance requirements, primarily well-structuredness.

The prototype evaluations may also assess the impact of the architecture's external interface on reusability and on both untrusted components and human users. An inability to hypothesize a satisfactory architecture may indicate that more drastic risk mitigation measures should be considered, such as the negotiated relaxation of reuse, trust or performance requirements, cost, or schedule (as acceptable by the Government).

Depending upon the sophistication and success of the prototype and the scale of other risks, the prototype may be a throw-away that simply verifies the feasibility of requirements, or it may become the base from which the system's architecture evolves and/or may consist of reusable assets that can be applied to future Navy C^2 system developments.

The spiral activities that may occur during preliminary design are described in Figure 15 and illustrated in Figure 16, a Conceptual View of Spiral 3.

The activities performed during early design and the number of spirals required will vary according to the needs and complexity of a particular project. In particular, once reuse technology is well established for the Navy C^2 application domain, the preliminary design activities may be simplified enough to require mainly reuse analysis. There may be an opportunity to reuse integration software assets that were developed on other projects to permit the repeated use of heterogeneous components and evolve toward a true open architecture while preserving trust characteristics for a specific Navy C^2 system development.

The Navy C^2 Government activities that direct and support Spiral 3 architecture refinement are listed below. These activities include IV&V and SETA contractor efforts

Spiral 3: Navy C^2 Government Activities

- Review CDRL items, including trust documents
- Attend PDR and provide comments and action items
- Respond to action items assigned to the Government
- Provide DTRs to contractor
- Participate in DTR resolution meetings
- Maintain requirements traceability
- Negotiate ECPs, as necessary
- Approve SCNs, as necessary
- Brief DAA
- Coordinate certification and accreditation activities with operational sites

Quadrant 1 - Objectives & Constraints	Quadrant 2 - Risk Analysis & Mitigation
• Incorporation of TCB and reuse constraints into Navy C^2 critical element considerations and plan critical element prototypes and experiments • Experimental integration of new and reusable Navy C^2 critical elements	• Assessment of Process Model application • Analyze Navy C^2 reuse qualifications of prototypes • Assess performance of Navy C^2 critical components • Reassessment of risks • Develop trust and reuse prototypes including the critical elements
Quadrant 3 - Development	**Quadrant 4 - Planning & Management**
• Enhance formal/informal Security Policy Model • Integrate critical elements • Refine the software architecture, including any revisions of the Software Requirements Specification(s) (SRS) and Interface Requirements Specification (IRS) that are needed • Conduct Preliminary Design, including the following documentation: - Software Design Document(s) (Preliminary Design) (SDD) - Software Test Plan (Test ID's) (STP) - Preliminary Interface Design Document (IDD) - User Interface Document (UID) • Compile and document design assurance evidence: - Descriptive Top-Level Specification(s) (DTLS) - Formal Top-Level Specification(s) (FTLS), if required - Formal proofs of correspondence, if required - Initial Covert Channel Analysis (CCA) • Document engineering notes • Conduct reviews and walkthroughs as needed	• Preliminary Design Review (PDR) • Review and revise resource allocation • Revise project schedule • Revise risk management plan (RMP) • Assess progress

Figure 15: Spiral 3 Activities

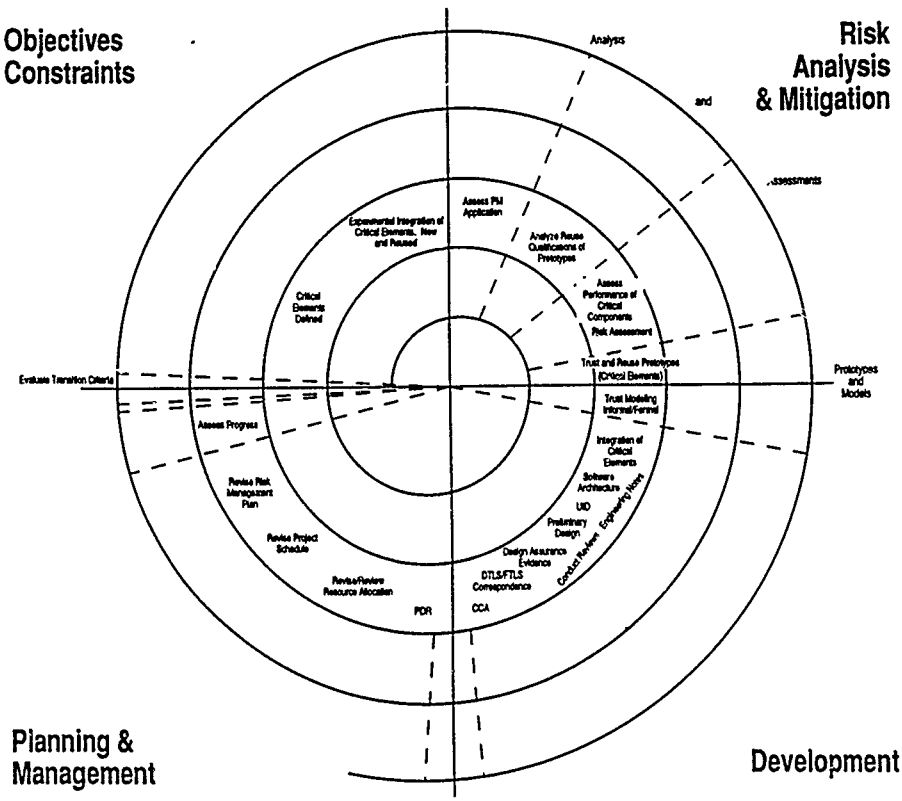


Figure 16: A Conceptual View of Spiral 3. Critical Elements and Architecture

- Maintain Certification and Accreditation Plan
- Continue implementation of reuse library
- Attend design walkthroughs
- Review SDFs
- Participate in configuration control board's activities
- Schedule and visit sites with development contractor
- Schedule and participate in user meetings with development contractor
- Verify operator interface
- Provide GFE and GFI, as required
- Participate in Government, User and Contractor meetings (management and technical)
- Plan site implementation
- Plan TECHEVAL
- Develop testing for TECHEVAL
- Plan OPEVAL with OPTEVFOR
- Develop testing for OPEVAL
- Develop tests for trust certification tests of software, system accreditation (ST&E)
- Provide contract evaluation and grading
- Resolve funding and schedule issues
- Keep development contractor aware of changing threat, mission or requirements by documenting
- Review and reassess project risks
- Approve updated risk management plan
- Resolve and complete transitioning criteria

4.2.4 System Development and Assurance – Spiral 4

The early spirals of the NCCPM deal with resolving major risks in the feasibility, requirements, scope, and reuse and conceptual approach to building a Navy C^2 system, while the later activities are concerned with actual product building. The Navy C^2 reuse planning and methodology of Spiral 0 strongly influence the actual development of new products both from a current use standpoint and the goals for future reuse. Reusable Navy C^2 components may be shown to be consistent with a new or reused specification in a new environment and/or with respect to new interfaces.

Approximations used in performance models may be validated as actual components become available. As in the traditional waterfall process model, the Navy C^2 system may be developed via the creation and validation of progressively detailed descriptions of the system, i.e., specifications for requirements, system architecture, high-level and detailed designs, etc., leading ultimately to executable machine code. However, the NCCPM differs from the traditional waterfall model in the following ways:

- The NCCPM recognizes the continuing need for risk-assessment and risk-mitigation activities (including reasoning-based analysis, modeling and prototyping), and explicitly calls for their presence throughout major portions of the development process In addition, to the extent possible, software development techniques and tools as well as reuse support are incorporated in the SEE to further reduce risks.
- The NCCPM allows concurrent threads of development activities that may traverse the traditional progression of software product-phases in loosely synchronized manner
- The NCCPM allows each thread to follow non-traditional progressions of activities where appropriate in the Navy C^2 domain.

The software may be incrementally developed and/or the system may ultimately be composed of integrated reusable components, COTS and GOTS engineered for trust and reuse in the Navy C^2 application.

Figure 17 describes the possible activities during the development and assurance stages of Spiral 4 and Figure 18 presents a Conceptual View of Spiral 4.

Although the Navy C^2 development and assurance activities are conceptualized as occurring in a fourth spiral, the required activities may occur over multiple spirals depending on the degree and number of project risks that occur or remain during system development. The required set of activities for a particular Navy C^2 development could be conducted within a phase-oriented process such as the standard waterfall paradigm if the development risks have been reduced to a very low level. Multiple, concurrent or phased spirals may also be used to represent incremental stages of coding and testing that may be separate or may depend on other spirals.

Quadrant 1 - Objectives & Constraints	Quadrant 2 - Risk Analysis & Mitigation
• System Development (Incremental stages/component integration) • Reuse of acceptable assets within the system development	• Tracking the application of the Process Model (PM) • Analyses/assessments of any remaining issues • Assessment of component and system performance • Interpretation/proving the Security Policy Model(s)
Quadrant 3 - Development	**Quadrant 4 - Planning & Management**
• Event driven additional prototyping • Conduct detailed design including the following documentation: - Software Design Document - Software Development Files - Software Test Description - Interface Design Document • Application of reasoning-based assurance and revisions of the FTLS and the Security Policy Model to FTLS Correspondence • Coding - staged, incremental, etc. • User Documentation - Operation and Support Documents - Computer Resources Integrated Support Document (CRISD) - Computer System Operator's Manual (CSOM) - Software User's Manual (SUM) - Software Programmer's Manual (SPM) - Firmware Support Manual (FSM) - Version Description Document(s) (VDD) - Software Product Specification(s) (SPS) • Documentation of reusable assets • Documentation of maintainability and evolvability • CSCI Functional and Physical Configuration Audits • Assessment of Asset Qualifications • Security testing and documentation - DTLS and FTLS Correspondence to Trusted Computing Base - Covert Channel Analysis - Trusted Facility Manual - Security Features User's Guide - CM Plan • System testing, documentation (STD) and evaluation including: - Evaluation of all requirements for reuse, trust, performance • Component evaluation and certification • Document engineering notes • Conduct reviews and walkthroughs as needed	• System Accreditation Support • Critical Design Review (CDR) • Test Readiness Review (TRR) • Formal Qualification Testing (FQT) support • Planning for operation and maintenance • Tracking Configuration Management, including reuse and trust • Development of guidelines for maintenance and reuse • Review of lessons learned • Revision of the risk management plan (RMP) for operations and maintenance

Figure 17: Spiral 4 Activities

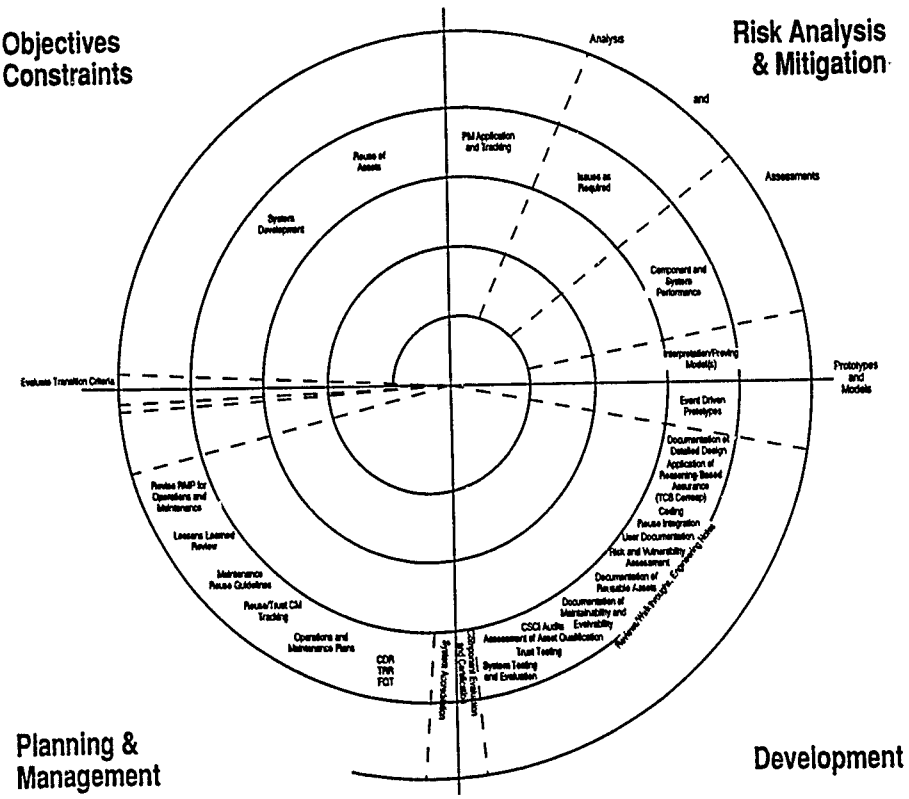


Figure 18: A Conceptual View of Spiral 4: System Development and Assurance (May be Incremental Over Multiple Spirals)

The Navy C^2 Government activities that oversee the Navy C^2 system development activities are described below. These activities include contractor participation for IV&V and SETA support.

Spiral 4: Navy C^2 Government Activities

- Review CDRL items, including assurance, trust documents (CCA, etc.)
- Attend CDR and provide comments and action items
- Provide DTRs to contractor
- Participate in DTR resolution meetings
- Maintain requirements traceability
- Attend design walkthroughs
- Attend code walkthroughs
- Attend (verification and assurance) trust (TCB)-based walkthroughs
- Review SDFs
- Attend TRR and provide comments and action items
- Respond to action items assigned to the Government
- Participate in configuration control board's activities
- Plan operations and maintenance
- Brief DAA
- Coordinate certification and accreditation activities with operational sites
- Maintain Certification and Accreditation Plan
- Set up Ships Parts Control Center (SPCC) sparing
- Plan and schedule training
- Negotiate ECPs, as necessary
- Approve SCNs, as necessary
- Incorporate software into reuse library
- Schedule and visit sites with development contractor
- Schedule and participate in user meetings with development contractor

- Verify operator interface
- Provide GFE and GFI, as required
- Participate in Government, User and Contractor meetings (management and technical)
- Schedule and participate in site surveys
- Plan site implementation
- Attend SIT and SPT
- Participate in system installation at site(s)
- Plan TECHEVAL
- Develop testing for TECHEVAL
- Plan OPEVAL with OPTEVFOR
- Develop testing for OPEVAL
- Develop tests for trust certification tests of software, system accreditation (ST&E)
- Participate in FCA, PCA, FQT
- Conduct TECHEVAL (DT&E)
- Write TECHEVAL final report
- Participate in OPEVAL (OT&E) with OPTEVFOR
- Participate in ST&E accreditation testing
- Perform accreditation
- Resolve and define accreditation issues – retesting, if necessary
- Write OPEVAL final report
- Provide contract evaluation and grading
- Resolve funding and schedule issues
- Keep development contractor aware of changing threat, mission or requirements by documenting them
- Accept system
- Approve for secure operation
- Turn system over to operations and maintenance personnel

- Review and reassess project risks
- Approve updated risk management plan
- Resolve and complete transitioning criteria

4.2.5 Maintenance – Spiral 5

For Navy C^2 systems, maintenance is the phase that continues to dominate the lifecycle costs. Maintenance has traditionally introduced risks, particularly those associated with system degradation caused by modifications that over time diminish the integrity and clarity of the system design. Attempting to control maintenance costs and activities has been the significant driver for much of software engineering research and development, particularly for DoD mission–critical systems.

The advance of successful Navy C^2 reuse technology should reduce greatly the traditional problems associated with costly maintenance. Engineering for reuse is analogous to engineering for ease of maintenance. The desirable characteristics of reusable Navy C^2 assets are much the same as those of maintainable assets. The availability of reusable assets and the associated information within a SEE containing a knowledge–based Navy C^2 reuse library will provide strong support for maintenance engineering.

Use of the NCCPM during maintenance follows the same pattern that is applied during development. Objectives, alternatives, and constraints are examined. Risks associated with the candidate modifications are assessed for reuse, trust and performance implications, and an approach with minimal impact to the Navy C^2 system application is selected. At this point, the use of formal models and specifications developed during the system construction may provide a method for evaluating the impact of proposed changes without the trial and error process that often accompanies maintenance efforts.

Maintenance modifications are achieved by updating all of the relevant development documents. Strict configuration management of the products is required for both reuse and trust. The implications of modifications should be well documented to support reuse qualification and to facilitate re–evaluation, if required. Maintenance activity, with modifications collected or grouped so the result is a new version of the system, represents additional spirals in the NCCPM.

Reuse issues may involve the qualification of both the old and new Navy C^2 asset versions and the provision of rationale for maintaining both in a reuse library. Reuse qualification and certification methodology must apply to maintenance of all assets, and the control of asset versions with rationale for maintaining older versions is a critical requirement for reuse.

Figure 19 and Figure 20, A Conceptual View of Spiral 5, illustrate the possible activities within the quadrants and sectors of a maintenance spiral for systems requiring trust and reuse.

Quadrant 1 - Objectives & Constraints	Quadrant 2 - Risk Analysis & Mitigation
• Maintenance of baselined Navy C^2assets • Implementation of plans created in previous spirals including: - Careful tracking of changes to reusable assets - Careful tracking and analysis of changes to trusted elements • Identification of potential maintenance risks and mitigation activities • Update of constraints for reusable, trusted components	• Reuse, trust and performance impact assessment of proposed changes • Analysis and assessment of technology enhancements, particularly in the areas of reuse and trust and in response to changing mission requirements • Analysis and assessment, of trust strategies including any new policies and mission requirements • Analysis and assessment of technology to support reuse and trust maintenance • Analysis and assessment of performance requirements • Assessment of asset qualification after modification • Development of any prototypes needed
Quadrant 3 - Development	**Quadrant 4 - Planning & Management**
• Modeling and interpretation of trust strategies including any new policies or mission requirements • Development of design revisions including software, hardware and documentation • Application of reasoning-based analysis and verification • Coding and integrating modified components • New Version Description Document(s) (VDD) and revision of any other documentation as needed • Trust testing • System retesting and re–evaluation including evaluation of all requirements for reuse, trust, penetration and performance • Support of re–evaluation and recertification of elements as required ◦ Document engineering notes • Conduct reviews and walkthroughs as needed	• Support reaccreditation of system trust as required • Revision of risk management and other plans for future operations and maintenance • Review and revision of lessons learned

Figure 19: Spiral 5 Activities

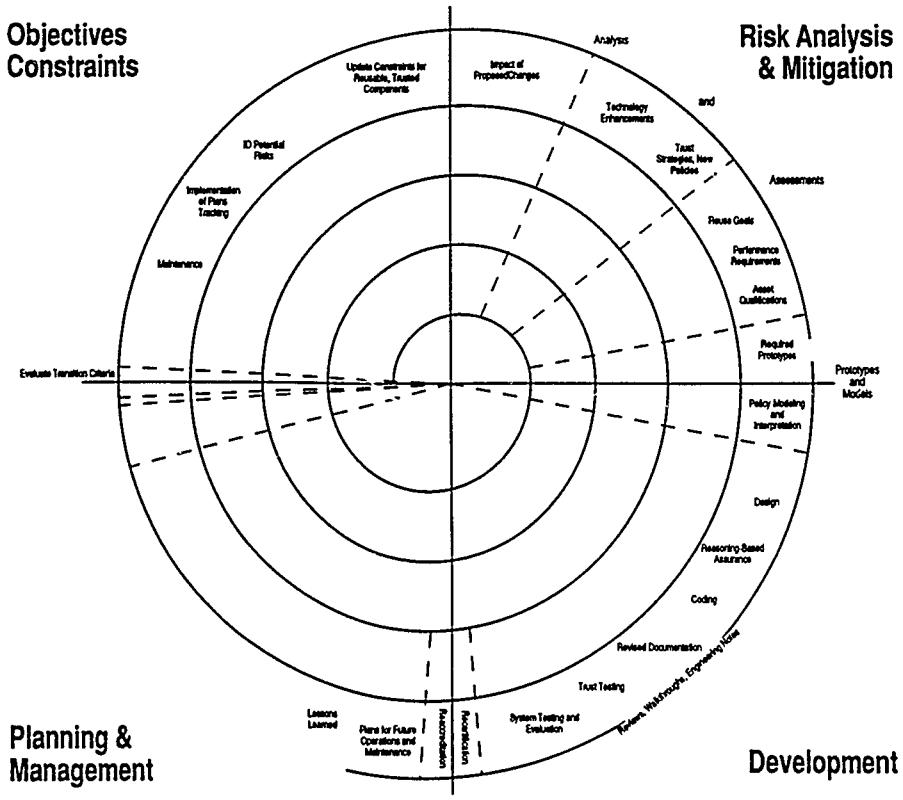


Figure 20: A Conceptual View of Spiral 5: Maintenance

In practice, the maintenance spiral could be partitioned into a number of spirals that address the specific risks associated with Navy C^2 system changes. Depending on the amount of effort involved and the degree of risk, the spirals may be similar to those used to address design and development risks in the initial system development.

Maintenance for trusted Navy C^2 systems is a challenging task since modification to the trusted portion of the system has the potential for invalidating the evaluation rating or certification of components and/or the accreditation of the system. Since implications of a modification are not readily determinable for most Navy C^2 systems, re-evaluation and recertification necessitated by maintenance may be a significant cost and risk factor for both developers and evaluators. Even a minor system change to a system that involves a life-critical or security-critical function has the potential for dangerous or unacceptable consequences without careful analyses and tests to assure that integrity, safety and security are maintained.

Reusability issues for trusted systems are associated closely with maintenance issues. Reuse theory and practice for highly trusted systems will require research advances in areas that are not yet well understood.

Maintenance for trusted, reusable Navy C^2 systems must be controlled and planned very carefully. The qualification of reusable assets may be affected by changes as well as the adherence to original trust properties. The implications of suggested modifications must be assessed carefully to determine the impact on asset reuse and Navy C^2 system trust and performance. Modifications to the trusted portion of the system will, in all likelihood, require modification to the analytic materials that have been developed to assure the trust characteristics of the system. For example, in a TCSEC trusted system [19], a modification to the Trusted Computing Base (TCB) will necessitate re-examination and possibly modification to the interpretation of the formal policy model and the covert channel analysis, as well as to the more directly related products, such as the design specification and the user documentation. Since for TCSEC products, the requirements for architectural constraints are so stringent, modifications introduce the risk of loss of evaluation rating Even if the rating can be maintained, the cost re-evaluation is a non-trivial aspect.

During maintenance, the Navy has primary responsibility for the operational system. The Government may be supported by the original development contractor or some other organization under contract for maintenance. Therefore, many of the fundamental maintenance activities are described here under the list of Government activities

Spiral 5: Navy C^2 Government Activities

- Conduct change assessment
- Provide site risk analysis support
- Test software and hardware upgrades or modifications
- Support recertification and reaccreditation resulting from upgrades or modifications

- Support other systems reusing the fielded software
- Continue training support to the site(s)
- Assess system modifications and technology enhancements for reuse, trust and performance implications
- Participate in configuration control board's activities
- Maintain reuse library with emphasis on older versions
- Attend reviews and walkthroughs
- Review and reassess project risks
- Approve updated risk management plan

5 REMARKS

This report describes the NCCPM, a full life-cycle process model for the development of trusted Navy C^2 systems. Relevant process model information is contained in risk summary tables derived from a TRW preliminary Navy C^2 domain analysis, in correspondence tables that relate certain standards to the major process model spirals and, principally, in the lists of activities defined in charts and through conceptual views for process Spirals 0 through 5

The NCCPM provides a top level description of the development process from the earliest stages of system concept through the maintenance stage. The descriptions incorporate the activities and products to be accomplished by Government, support contractors and development contractors in the development of trusted Navy C^2 systems. The NCCPM is risk-driven and is based on previous TRW process model work on the DARPA/ISTO Advanced Computing Systems project, on the reuse activities defined during this subtask and documented in the *Draft Composite Paradigm Report*, and on the preliminary domain analysis work contained in the Appendix of this report.

The risk and correspondence tables summarized in Section 2 of this report provide process model guidance that supplements the process model descriptions of Sections 3 and 4. The broad scope of the NCCPM precludes detailed process descriptions within the constraints of this subtask. The current NCCPM does provide a prototype STARS-relevant process model description that can be used as a basis for the development of process building blocks and can help to define the complex dependencies between process participants, activities and products.

To achieve major advances in software productivity, further investigations are needed in a number of areas related to STARS goals. Many important open research issues relevant to the development of trusted, reusable systems and the STARS process, reuse and SEE goals were described in the *Draft Composite Paradigm Report*. Some of the issues discussed include

reuse methodologies and engineering support, broad trust, domain analyses, process automation, configuration and control within a complex and dynamic process, formal methods for assurance, architecting trust and reuse, and trusted software engineering environments for the development of trusted systems.

Much work needs to be done to accomplish the long term STARS goals for reliable, adaptable systems. The goal for automated process management within a SEE that supports reuse for a variety of application domains will require additional adaptation and the integration of related current and future work. The NCCPM represents the tailoring of previous process model work to the Navy C^2 domain to assist the goals for specifying and implementing automated process management within the STARS SEE in a particular application area. Through more specific process descriptions, an automated process concept of operations and support tool requirements can be better understood. Immediate goals beyond this subtask include a continuing effort to build on the process modeling activities and to conduct experimentation with process representations and process automation tools that exist today.

A TRUSTED NAVY C^2 RISKS AND CHARACTERISTICS REPORT

A.1 INTRODUCTION

This report is a documentation of preliminary domain analysis activities to support Navy Command and Control (C^2) domain enhancements of the process model tailoring work in STARS Task US40. It provides initial characteristics and major development risks for the trusted Navy C^2 Ada domain to be used to derive risk mitigation activities; these are included in section 2.3. Information derived from this report helps to define the process model techniques and transitioning criteria for Navy C^2 development risk resolution.

This task addresses the inadequacy of current software development paradigms, especially within the goals for reuse and for trusted systems. The task results focus on the adaptation of previous process model work and the initiation of Navy C^2 domain analyses for the purpose of domain tailoring to strengthen the STARS foundation for reuse process building blocks and automated process management.

The Navy C^2 risks and characteristics identified herein represent preliminary domain modeling efforts. This initial characterization can support the description of a top level domain model for the trusted Navy C^2 application domain. To fully characterize trusted Navy C^2 systems and apply reuse concepts, much more detailed analyses beyond the scope of the current task will be required and more information from reuse and domain experts will be needed.

A.1.1 Background

To achieve a first step in constructing reuse process descriptions and reuse resources as described in [6], this report required preliminary Navy C^2 domain analysis activities Reuse is not a feasible option without a clearly defined reuse methodology and process descriptions as well as available reuse assets, a support library and tools. Foundations and issues for the reuse libraries and the Software Engineering Environment (SEE) are described in [4] Before such a state of reuse technology can exist, successful domain analyses must be accomplished. Various approaches to domain analysis and ongoing research are presented in [3].

In reality, the effectiveness of reuse within the Navy C^2 domain will not be known until actual systems are implemented using reusable assets. For the development of trusted Navy C^2 systems in a reuse environment, there is a need for a high degree of confidence in the integrity of trusted Navy C^2 assets. The issues for trusted assets in a reuse library are addressed in [5], many of which are open areas of research. Management commitment and a clear and early understanding of the reuse process in the Navy C^2 domain are fundamental for reuse as a feasible process model driver.

A.1.2 Scope

TRW has identified domain-specific characteristics and risks for Navy C^2 systems, focusing on trust and reuse considerations in the Navy C^2 environment, particularly when the Ada programming language is used.

In defining the scope of this task, the Navy Tactical C^2 domain was divided into two categories: Navy C^2 systems and Navy tactical data systems. Navy C^2 system characteristics include near real-time, large data base, and long term data storage. Navy tactical data system characteristics include real-time, small data base, perishable information, and short term data storage. Navy C^2 systems are located ashore and afloat. Navy tactical data systems are located afloat.

The domain of interest for this report is Navy C^2 systems, with a concentration on those systems ashore. There are many similar characteristics between Navy C^2 ashore and afloat systems, and subsection A.2.1 includes decision aids and automated support functions used in Navy C^2 systems afloat as well as those used ashore. The following Navy C^2 programs were used as the source of knowledge for this report:

- The OSIS Baseline Upgrade (OBU) - currently in operation [25]
- The Antisubmarine Warfare Operations Center (ASWOC) [24]
- A Navy C^3I Internal Research and Development (IR&D) architecture
- A Navy Command and Control System (NCCS)-Afloat domain analysis [22]

A.1.3 Approach

After identifying the domain of interest for this task, TRW drafted a plan of discussion items for meetings with domain experts. These discussion items included:

- Overview of STARS effort
- Overview of TRW STARS subtasks
- Definition of characteristics to include activities and functions, who and what performs functions, results, and how they inter-relate
- Brainstorming to develop list of characteristics
- Consideration of trust issues
- Consideration of reuse issues
- Brainstorming to develop list of major risks

TRW then held meetings with Navy C^2 domain experts in TRW and NRL. We also reviewed Unisys domain documents.

Through these technical exchanges and analyses of real world projects and research, we developed a comprehensive list of Navy C^2 system characteristics and major risks. In addition to real world projects (OBU and ASWOC) and the Unisys NCCS–Afloat domain analysis, we analyzed a TRW Navy C^3I IR&D architecture which used a "levels and views" methodology for developing the architecture. This methodology is described in section 3.2.1.

A.2 CHARACTERISTICS

As the result of meetings with TRW "resident" domain experts and review of TRW and Unisys Navy C^2 system architectures, a set of preliminary Navy C^2 system characteristics has been identified. The emphasis of these characteristics is the human interface of Navy C^2 systems, particularly how the machine and human interact to support the human activities These characteristics are discussed in subsection A.2.1. To fully characterize trusted Navy C^2 systems, a task which is beyond the scope of the current work, much more detailed analyses will be required and more information from reuse and domain experts will be needed. The preliminary nature of this list of characteristics can be seen as one step toward the definition of a domain model in which objects, operations and their interrelationships are defined. No assumptions of the completeness of this characteristics list can be made at this early stage. The characteristics identified here help to define the primary issues in the development of Navy C^2 systems and support the goal of identifying major development risks.

The characteristic "decision aids and automated support functions" was determined to be one of the most important characteristics in supporting the operator as he performs the required analysis necessary to build effective plans to accomplish a mission. Thus subsection A.2.2 describes a list of Navy C^2 decision aids and automated support functions in greater detail. Subsection A.2.3 discusses issues related to the identified Navy C^2 systems characteristics Many of these characteristics are inter-dependent. Strict adherence to one characteristic may impose limits on the ability to optimize another characteristic (i.e., adherence to hardware standards may impose limits on the ability to adhere to open architecture goals).

A.2.1 Navy C^2 System Characteristics

The identified Navy C^2 characteristics include:

1. Secure/trusted system
2. Man-machine interface
3. Communications
4. Message handling

5. Open architecture
6. Adherence to hardware standards
7. Supportable by Navy logistics
8. Reliability, maintainability, and availability
9. Data fusion
10. Decision aids and automated support functions
11. Man-in-the-loop
12. Distributed architecture
13. Flexible architecture
14. Near real-time system operation

A.2.1.1 Secure/Trusted System Within the defense community, there is growing awareness of the potential benefits a multilevel secure (MLS) mode of operation would provide with respect to reduced requirements for user clearances and flexibility of applications. However, the Navy C^2 environment is one in which most site users must necessarily access the most sensitive information in normal applications. Therefore, the operational environment for Navy C^2 systems remains at system high in today's world.

Navy C^2 systems require MLS at the communications level, and MLS is desirable for other functionality as illustrated in the development of the OBU system. Multiple levels of classified information (messages and other data) must be handled correctly and managed and communicated properly by the system. Security labels must be trusted within the system, across system interfaces and for external communication of sensitive information. System trust is therefore required for Navy C^2 applications with respect to security to help enforce confidentiality and integrity of information and also in a broader sense to help ensure the correct behavior of functions that enforce security policy and functions that are critical to the system mission (assured service).

To ensure the security (confidentiality) of highly sensitive information in the Navy C^2 environment, a man-in-the-loop is required traditionally for "write down" and export operations and decisions. Basing trust in humans to perform operations that are exceptions to strict system security policy reduces the level of trust required for the MLS automated functions. Trusted automated functions are still necessary to support the human users. Humans are trusted but not reliable for large amounts of data. Machines can be both trusted and reliable, but only within limited and narrow confines to permit a reasonable level of trust assurance.

Data fusion is an aspect that is not well understood but an important of Navy C^2 applications function that is pushing the state-of-the-art in trusted systems technology Management

and tracking of security labels in accordance with DoD security policy is essential. Data fusion increases the problem of classification issues for data aggregation and trusted database management. Requirements for highly trusted data base management systems and fusion algorithms and for trusted knowledge-based support are among the drivers for trust technology research. More discussion on data fusion as a distinct characteristic follows in A.1.2.9.

While the Navy C^2 mission is the principal driver for security, Navy C^2 systems must satisfy National Policy mandates for overall information security, computer security and secure operations (e.g., the National Computer Security Act of 1987, the NTISSP-200 mandate for controlled access protection of sensitive federal computer systems by 1992, and the Privacy Act of 1974). Navy C^2 security requirements are derived from the defined mission and from broad National Policy statements and standards, DoD policies and standards, intelligence and war planning policies and standards, and Navy policies, instructions and standards. These top level security policies and standards must be incorporated in the overall requirements to define a secure, trusted Navy C^2 computing environment that satisfies the mission needs.

A.2.1.2 Man–Machine Interface Navy C^2 systems are user intensive systems whose man–machine interface (MMI) must provide graphical capabilities as well as resource–based interfaces. The MMI must be designed to allow maximum usability of the system with the minimum amount of experience on the system since the amount of time available for training operators as well as the time each operator spends using a particular system is limited due to frequent sea and shore duty rotation. Current trends for Navy C^2 system standards are for the X Window System and OSF/MOTIF as the MMI windowing and look and feel standards. The MMI must support the analyst's expert abilities and provide a strong base for the complex interrelationship between man and machine that is required in the performance of many Navy C^2 applications.

The symbology used in Navy C^2 afloat system displays is standard Navy Tactical Data System (NTDS) symbology, whereas Navy C^2 systems ashore have different symbology. An example is the Antisubmarine Warfare Operations Center (ASWOC) which uses ASWOC symbology in their graphics displays.

A.2.1.3 Communications The communications protocols used for external and internal (local area network) data channels in Navy C^2 systems are numerous and dynamic Local area network protocol requirements trends are for TCP/IP[28]. Many of the same protocols are common among these systems, but each system also has unique communications protocols. The design of these protocols in reusable software needs to be flexible to accommodate change (interoperability, open system goals, new security policies, etc.), particularly with the Copernicus–derived external networks on the horizon. The Copernicus–derived networks will use a set of TADIXS and GLOBIXS lines for communications.

The Copernicus architecture proposes to change the center of the universe. from many shore–

based sensor and fusion centers to a single Fleet Command Center (FCC). Ashore, the FCC will act as the intersection point with eight Defense Data Network (DDN)/Defense Satellite Communications System (DSCS) (i.e., Global Information Exchange Systems (GLOBIXS)), one each for Signal Intelligence (SIGINT), Space and Electronic Warfare (SEW), Anti-Submarine Warfare (ASW), Imagery, Database Management and High Command Communications Net (HICOM), and two support systems.

The warfare GLOBIXS would all use a common technology - the DTC II - a family of evolutionary computers, hosting the Fleet All-Source Intelligence Terminal (FASIT), with a high percentage of COTS software including X Window, MOTIF, UNIX, DeLorme, Vitec, TOPIC, Sybase, and WordPerfect. GOTS software will include Panther/PAWS correlation and MIIDS/IDB reference databases. The purpose of the warfare GLOBIXS is to provide a shore-based infrastructure for the Navy to capture sensor data efficiently, and forward that data for tactical use to the FCC as a sensor-to-shooter throughput or as value-added product.

The new centers of the universe - the FCC in "co-orbit" with the TFCC - will each share a common tactical picture through a series of 14 TADIXS. One major impact of the TADIXS will be to really eliminate the Navy message as an operational format, thereby cutting up to 80 percent of the message traffic forever. There will be a significant savings in communications capacity. This will eliminate the Navy's total dependence on high frequency (HF) and provide alternate backup to Navy satellite communications (SATCOM).

The technological key will ultimately be a common format and a common terminal prescribed centrally, using application software to suit the warfare area. The result will be innovation channeled into operations and doctrine, not into splintered technological efforts

A.2.1.4 Message Handling The key requirement is to provide information about forces and other assets from one system/user to another and formulating this information in such a way that it can be processed by application software in the support of the Navy C^2 mission The message format header and text standards used by the Navy C^2 systems are numerous and dynamic. Most of the message formats are common among Navy C^2 systems Some of the common message format header standards include ACP-126, ACP-127, and JANAP 128 (with modifications for OTH-T, DOI 103, CLI, and ASWCCCS). Some of the common message text formats include JINTACCS and USMTF. The design of these text formats for message generation and message parsing in reusable message system assets needs to be flexible to accommodate change. Traditionally, translators have had to be used in Navy C^2 systems until software to support new formats could be implemented.

A.2.1.5 Open Architecture A current trend for new Navy C^2 system developments is to require open architectures that comply with the Government and international initiatives and standards for product and system interoperability and open system interconnections An open architecture is an integrated hardware and software system that provides for mod-

ification and expansion of system functions without requiring major changes to the central hardware and software set or its architecture. This includes an architecture that consists of COTS, Government off-the-shelf (GOTS), pieces of systems, and pieces of prototypes. For example, the Navy has open architecture goals to achieve the porting of multiple software applications to various vendors' desktop workstation computers within the Navy standards

True open architectures will provide vendor and implementation independence where porting of applications and tools to various platforms will be readily achievable. The international community as well as the U.S. government and much of industry have become interested in achieving open systems and true interoperability. The focus on open system standards, products, methods and general research is now international. A primary goal for STARS is to foster and promote the achievement of open architectu.es within the broader goal of advancing the software technology for adaptable, reliable systems.

In the Navy C^2 application domain, the Operations Support System (OSS) is an example of a planned system development that will use an open architecture based on lessons learned from experiments and prototypes. The OSS will emphasize reuse, transportability and an evolutionary development process. A primary emphasis is on standardization and the use of COTS and GOTS. Standards proposed for use in OSS include:

Graphic/Windowing	X Window System
Man Machine Interface/Look and Feel	MOTIF
Operating System	UNIX System V
Network	IEEE 802.3
Network Protocol	TCP/IP
Network File	SQL
Languages	Ada and C
Interprocess	Applications interface standards - SOE

To meet the needs for rapid replacement of today's aging systems and to support the reuse of prototypes and experiments with "plug in" components, Navy C^2 applications are being developed and fielded with the plans and goals of open architectures The technology for integrating the components of such an architecture and achieving both interoperability and high trust is still evolving.

A.2.1.6 Adherence to Hardware Standards Navy C^2 systems may be required to adhere to certain hardware standards, particularly for workstation hardware The Navy standard desktop computers are examples of hardware standards that may be required in Navy C^2 systems, particularly in systems involving reuse. The Navy standard desktop computers are as follows: DTC I is a Hewlett Packard 9038, DTC II is a SUN Sparcstation, and DTC III or TAC III is still under procurement. Navy C^2 systems may require that software be transportable from DTC I to DTC II to DTC III

A.2.1.7 Supportable Navy Logistics Navy C^2 systems designed for reuse must incorporate Navy requirements for Navy organic support including the current trend toward reduced manning requirements. Navy technician reduced manning requirements involve both a reduction in the number of people available at a site as well as reduction in skill level of these people. The amount of time available for training maintenance personnel as well as the time spent maintaining a particular system is limited due to frequent sea and shore duty rotation. Navy C^2 systems must be designed for ease of maintenance to minimize maintenance downtime because they include a requirement to operate 24 hours per day, 7 days per week.

A.2.1.8 Reliability, Maintainability, and Availability (RMA) Navy C^2 systems require operation 24 hours/day, 7 days/week. RMA, personnel and training requirements must be tailored to support these system operation requirements. The systems must also support watches that change every 8 to 12 hours. Reduced manning requirements must be considered when planning system operation.

The trusted system requirements mandate user accountability. Thus, Navy C^2 trusted systems require incorporation of user roles and the capability to add and delete users and their specific roles. These systems must be designed to allow a user to have multiple roles and to allow multiple users assigned to identical roles.

A.2.1.9 Data Fusion Data fusion is the combination of information from diverse sources to create a complete, coherent set of information or picture from multiple sources of information. Navy C^2 systems maintain very large data bases of information from dissimilar sources for long periods of time. Classified information must be managed within some of these databases. Data at all classification levels is fused and the integrity of the security labels must be maintained. These data bases are queried frequently and require quick access. Data aggregation is the major security problem for data fusion. Other security issues include the problem of determination of sensitivity of unlabeled text data. Some of these Navy C^2 systems require trusted relational data base management systems. Currently, the technology of MLS data base management systems which satisfy these requirements is immature. Some of the categories of data fused by Navy C^2 systems include:

- Environmental analysis products
- Operations analysis products
- Intelligence data from overhead sensors
- Surveillance information from underwater sensors
- Tactical force surveillance information
- Readiness

- Tactical operations information

A.2.1.10 Decision Aids and Automated Support Functions Since decision aids and automated support are such critical aspects of Navy C^2 systems, we will expand this discussion below. Decision aids are required for assisting the Navy C^2 operator in making decisions about all the data described in data fusion above. Navy C^2 decision aids and automated support may include the use of enhanced man-machine interface, expert systems, color graphics, and data purging techniques. The following common decision aids and automated support functions for Navy C^2 systems ashore are described in more detail in subsection A.2.2:

- Automatic message correction
- Land-mass avoidance algorithm
- Closest point of approach calculation
- Data fusion tools
- Correlation and tracking tools
- Automatic message routing
- Planning tools
- Historical analysis and projection

Navy C^2 systems afloat require tactical decision aids for the user to perform "what if" analysis by creating hypothetical situations and applying the decision aids to the situations. As described in Unisys NCCS-Afloat Information Object Model [22], these tactical decision aids for afloat systems include:

1. General decision aids - These decision aids are generally used by all of the warfare areas. The general decision aids support formation planning, route planning, intercept planning, closest point of approach analysis, track analysis, and communications planning.
2. ASW decision aids - The ASW decision aids support barrier planning and evaluation, area search, and asset allocation planning. They provide statistical analysis tools for evaluation detection probability, performing track analysis, and analyzing contact reports and associating platforms with contacts.
3. ASUW decision aids - The ASUW decision aids support TASM cruise missile planning, multi-unit HARPOON planning, area search planning and assessment, barrier planning and evaluation, and SEATAK planning.

4. AAW decision aids – The AAW decision aids support F4/F14/F18 intercept and chain saw planning. They also provide radar range predictions for determining aircraft altitude assignments.

5. Strike decision aids – The Strike decision aids provide information concerning enemy air and coastal defenses, imagery of the target area and analysis for shore bombardment planning.

6. EW decision aids – The EW decision aids assist in determining satellite vulnerability, OPDEC planning, and EMCON planning.

A.2.1.11 Man-in-the-Loop Even though decision aids and automated support functions are a central part of Navy C^2 systems, National Policy constrains the actions of these systems to making recommendations. Navy C^2 systems must have human interaction to make actual decisions and generate orders. Every human interaction requires the use of trusted functions to provide assurance that the operator is authorized access to that particular information. System performance may then be affected since trust requirements potentially impact performance.

A.2.1.12 Distributed Architecture The trend for Navy C^2 systems is to require a distributed architecture within one environment consisting of numerous workstations, often in a small physical space due to space limitations. Distributed Navy C^2 systems exist within separate facilities and become part of a much larger distributed architecture, whereby communications links are used to pass information between the systems. This distributed architecture may increase security requirements for the protection of information in such an open, dispersed environment.

At the system level, Navy C^2 hardware and software elements combine to form a distributed system of interconnected processors. Software allocates processing functions to computers and logically connects peripherals and terminals to processing functions. The distributed systems software can reconfigure the network to respond to hardware failures, to cope with crisis mode operations, to schedule preventive maintenance, and to add new computers for increased performance or added functionality. The software that controls the distributed system adds a level of complexity and additional trust requirements above earlier centralized systems. Trust technology for distributed systems remains an area with many open issues where research is needed.

A.2.1.13 Flexible Architecture The Navy C^2 threat is constantly changing in response to world events (particularly recently) resulting in dynamic system requirements as well as generating requirements for both fixed and mobile sites. These systems must be designed to be easily adapted to include information about the new threat, and about new tactics and weapons for own force. The systems also must be designed to allow flexibility in the number of workstations and peripherals so that a system with a smaller footprint could

be deployed in a contingency situation. Navy C^2 software needs to be flexible to hosting on hardware that can be used afloat as well as ashore.

A.2.1.14 Near Real-time System Operation To be responsive to queries and platforms being supported, Navy C^2 systems ashore and afloat require that operation be near real-time. Due to trusted system requirements for such functions as security auditing, performance of the C^2 systems is often affected. Navy tactical data systems controlling weapons require real-time operation. The need for both near real-time performance and system trust creates a challenge for Navy C^2 system development.

A.2.2 Decision Aids and Automated Support Functions

This subsection provides a more detailed level of discussion of decision aids and automated support. Navy C^2 systems ashore and afloat handle large quantities of data received from numerous sources and are required to maintain these large data bases for long periods of time. These systems must be able to access specified data quickly since they are near real-time systems. This subsection describes eight common decision aids and support functions used by Navy C^2 systems ashore. System unique decision aids are not included.

A.2.2.1 Automatic Message Correction Navy C^2 systems receive numerous messages that contain message errors causing problems for automatic parsers. Automatic message correction support functions can help correct message errors without operator intervention. Some errors that can be corrected or for which compensation can be made include:

- Year-end or month-end transition errors in date time group
- Missing BAUDOT shift in numeric fields
- Invalid field delimiters
- Incorrect format for transmission path
- Spelling errors

A.2.2.2 Land-Mass Avoidance Algorithm Navy C^2 systems are involved in the correlation of contact reports to existing tracks. A land-mass avoidance (LMA) algorithm is used to determine if a contact and track pair is LMA geofeasible. If the updated track state estimate would be on land, the contact is not assigned to the track. Using the LMA decision aid prevents assignment of such a contact to a track and sends an alert to an operator.

A.2.2.3 Closest Point of Approach Calculation Navy C^2 systems use computational aids such as closest point of approach (CPA) to calculate position and time at which a specified unit is at the closest point of approach to another specified unit, or operator-specified point on a land mass, bottom contour, or restricted area. CPA decision aids compute range, bearing, position (LAT and LONG), and time of the CPA. The CPA computation is based on the best estimated position or operator-specified position, course and speed advanced to CPA.

A.2.2.4 Data Fusion Tools Navy C^2 systems receive attributes about a target from dissimilar sources and require automated support to organize and handle the fusion of all the information. The sources typically include tactical operations, surveillance, and National intelligence sources. Some of the advanced technologies supportive of data fusion and analysis include fuzzy logic (best guess), knowledge-based and expert systems, reasoning under uncertainty, neural networks and natural language processing.

A.2.2.5 Correlation and Tracking Tools Navy C^2 systems use automatic correlation and tracking tools to develop and maintain track and track history information on surface, subsurface and air platforms. Automated correlation and tracking tools use predefined and operator-definable filters along with correlation algorithms to correlate contact reports to track and to initiate new tracks.

In some cases, contact reports that cannot be correlated to a track automatically may require manual correlation. Correlator/Tracker computational and correlation aids are used by the operator in manual correlation. The computational aids perform single and multiple unit calculations and projections of platform position and area of uncertainty (AOU). Correlation aids provide the operator with assistance in correlation by calculating a numerical score or measure of confidence. The measure of confidence is computed on spatial or other platform characteristics data elements from the contact report and is used to evaluate manually the likelihood that a candidate contact and track pair should be correlated.

A.2.2.6 Automatic Message Routing Navy C^2 systems receive numerous formatted and narrative messages. Most of the messages are parsed by the message type. Automated support functions are used to determine the message type, such as contact report, narrative, query/response, sortie report, etc. Narrative messages require routing to particular operators. These functions use pre-defined criteria to route automatically the narrative messages and other message types to the correct destination.

A.2.2.7 Planning Tools Navy C^2 systems use "what if" situation planning tools to support readiness (resource utilization and asset optimization.) The planning decision aids use information on manning availability, platform availability, equipment configuration and

installation, weapon and sensor availability, and casualty reports to plan the state of readiness during a particular scenario at any point in time.

A.2.2.8 Historical Analysis and Projection Navy C^2 systems maintain target attributes from fused data over long periods of time. The data includes contact and track data, intelligence on the target, and red unit doctrine. This historical data is maintained by the Navy C^2 system and used to project target movement and intentions. Expert system decision aids are used for both short-term and long-term behavioral analyses of targets.

A.2.3 Issues

The current list of Navy C^2 characteristics represents a first attempt to identify generic application concepts to determine reusable in the Navy C^2 application domain. More detailed representations of objects, operations, and their interrelationships are needed to define clearly candidates for reuse. One major issue is the level of granularity or how detailed the characteristic descriptions need to be to provide adequate reuse guidance.

There are multiple ways to view Navy C^2 partitions, and there will necessarily be controversy over the best way to partition a "generic" Navy C^2 application description. A "levels and views" approach, which is described in subsection 3.2.1 of this report, offers a means to analyze all aspects of the system. The Information Object Model described in [22] gives a methodology to derive hierarchical object-based descriptions for a specific system application. Various other approaches to domain analysis also exist and have their strong proponents.

With reuse as the primary motivator, the partitioning of a trusted Navy C^2 system and its generic architecture may be different from the current instantiations of trusted Navy C^2 systems today. Detailed domain analyses and the feasibility of obtaining reusable assets will drive the formulation of generic architectures. Controversy is likely to remain in defining a generic partition and functional architecture for the Navy C^2 application domain Very low levels of granularity may be needed to determine the adequacy of some of the higher level functions for reuse while the complexity of the more detailed levels hinders the necessary system-wide view.

A.3 RISKS

This section presents the major risks identified for a trusted Navy C^2 system development There are many risks associated with the development of trusted Navy C^2 systems today that need to be addressed. Within the constraints of today's technology and human resources, common risks can be associated with any large, complex system development The most crucial risk for any system development is the potential for misunderstanding or misinterpreting system requirements. In some cases, the system customers and/or end users may be unsure of what they really want or need, and requirements may be fuzzy or poorly defined from

the start. Since the system is designed for the machine to support the human, the user's needs must be well understood. In the dynamic Navy C^2 environment, it is not unusual for requirements to change frequently during system development and during maintenance of operational systems. A detailed listing of these risks and potential mitigation activities are included in Section 2.3.

There are two primary categories of system development risk: technical or programmatic The risks identified here for trusted Navy C^2 developments were not always cleanly partitioned into either category since some strongly contain both technical and programmatic elements. Therefore, the risks are categorized as:

- Both technical and programmatic
- Technical
- Programmatic

Identified risks are discussed below within each category

A.3.1 Both Technical and Programmatic Risks

Within a trusted, reuse-based Navy C^2 development, three risk areas are defined as both technical and programmatic due to the important technical constraints and human and sociological factors that comprise the risks. These risk areas are:

- Reuse
- Trust policy
- Evaluations, certifications, and system accreditations

A.3.1.1 Reuse The goals of reuse within the Navy C^2 community to provide desired capabilities in an accelerated and low cost manner are not without risks and compromise Considering the following examples, reuse can be a risk to any system within the development process. The potential of incorporating obsolete rather than state-of-the-art design exists This is particularly true if a systems' architecture is chosen as the foundation for future system's adaptations.

Rarely are all parties in total agreement as to the best design A COTR may be forced to reuse a design that is not considered optimal for the system in question Often a reuse plan has been devised without complete knowledge or understanding of all systems required to incorporate the reusable software This was a major challenge in the goal for OBU System reuse on the ASWOC C^3 Upgrade system development project

Since reuse technology is relatively young, often the process in the past has been more cumbersome and costly than starting from scratch. A goal of reuse in the STARS environment is to reverse this trend. Navy C^2 systems employing reuse technology have tended to integrate large blocks of software, rather than only making parameter changes relevant to the specific system. This focus on integration could entail major new development to support the reuse.

To support the process of reuse, there is a need to have a viable reuse asset library Reusable assets may include prototypes, software subsystems, components or elements, support documentation, analysis results, test results, certification results, environment descriptions and any product or document that supports reuse. A reuse asset library must provide easy access to assets and support flexible, appropriate descriptions within multiple environments. For the Navy C^2 domain, the reuse asset library must support the Navy C^2 asset descriptions and provide definitions and translations to help with the determination of potentially reusable assets from other domains. The asset library must provide integrated, intelligent tool support for reuse potential determination and must support asset integrity and certification. NOSC has initiated this work within the NCCS Afloat program. Research in reuse asset definition, storage, retrieval, management and tool support within a STARS Software Engineering Environment (SEE) is ongoing.

Crucial to the success of reuse is definition of reuse requirements in the initial development process Consideration and planning must be given to the reuse requirements because the foundation of the system design is dependent upon them. Reuse must be managed as any significant requirement is managed.

A.3.1.2 Trust Policy A major risk for Navy C^2 systems is the lack of understanding of the role of system mission and its relationship with the trust requirements A security and trust policy must incorporate both mission and trust needs to satisfy the system omission

A Navy C^2 system must be trusted to enforce a policy or a set of restrictions on the operations allowed by users and internal processes. Systems for TCSEC [19] B2 and higher assurance, for instance, require that the trust policy must be stated in terms of a formal policy model. It is essential that a formal policy model by accurate with respect to the intent of the informally-stated policy it represents, and that it includes all critical components of the informal policy.

The formulation of an informal trust policy and its expression via a formal policy model are development risk-mitigating activities that are themselves inherently risky. The policy may need to be expressed via a formal policy model in order to analyze its characteristics and implications, and it is possible that it may be found to include unreachable states, deadlock, and other unintended behavior. Such defects, which may be subtle, can lead to undesirable system behavior if uncorrected.

Policy modeling and the insight it provides can help mitigate potential risks posed by a flawed policy While creating a formal model may reveal policy defects, the model formulation process may itself pose risks If the model is inaccurate and is used for formal analysis or

verification, there is a risk that errors in the model may be inadvertently forced into the design and implementation of the system. Also, the model needs to mesh with the doctrine and concept of operations for the particular system.

A.3.1.3 Evaluations, Certifications, System Accreditation, Reaccreditation, and Recertification Trusted Navy C^2 systems for mission-critical applications require certification and system accreditation before they are allowed to operate in a classified, safety-critical, or life-critical environment. Trusted commercial products for classified or sensitive applications require certification; in particular, TCSEC trust requires product evaluation through the National Computer Security Center (NCSC) to achieve the desirable designation of a "trusted product" at a specific TCSEC level. Similarly, safety-critical systems must be certified prior to their operational use. Software, hardware, and environmental certification for security, and other important system requirements, are necessary activities that support the final accreditation of a mission-critical system. Lack of concurrence, misunderstandings, and/or absence of agreement on the ultimate accreditation requirements have posed high risk for many Navy C^2 system developments in the past.

When trusted systems are modified or revised, the certification or accreditation accorded the original system is often nullified. This imposes a serious risk associated with reuse of trusted components or systems. Frequently, the process of recertification or reaccreditation may be almost as extensive as the original activities. Technical means to illustrate the implications and ramifications of system changes are still weak or non-existent. Modifications may have subtle consequences that undermine basic trust mechanisms or assurance. Until technology is strengthened in this area, the possibility of renewing approval for a trusted system introduces significant risk. STARS tasks UQ18 and US18 address the issues of trust, assurance, certification and reuse.

A.3.2 Technical Risks

The risks identified for trusted Navy C^2 system developments are principally technical in nature while there may be some lesser elements of human and sociological risk that are involved. Technical risks are more concrete and frequently better understood then the more subjective human aspects of system development. Nevertheless, technical risks may be critical for a system development and may be very difficult to manage, especially if only addressed late in the development. Some of the subtle dependencies between technical risks and related human aspects are addressed. Eleven technical risks are discussed here. They are

1. Understanding and Communicating Requirements
2. Frequently Changing Requirements
3. Assurance
4. Trust Skill Specialization

5. Architecture
6. Technology
7. Performance
8. Ada-related
9. Documentation
10. Standards
11. Trust Assurances During Maintenance

A.3.2.1 Understanding and Communicating Requirements Through meetings with Navy C^2 domain experts, understanding and communicating requirements was determined as the number one risk area. Understanding and communicating requirements may be impacted by political issues, but is believed to be primarily a technical risk This is likely to be true for all application domains. This determination applies to all areas of requirements, however, user interface requirements surfaced more frequently than others. A reason for this occurrence is that the user interface reflects an understanding of the way the operator would use the system and in turn this affects the division between automatic and manual functions and the resulting software design.

The Government states the needs of their development system through high-level Type A Specification requirements. Unfortunately, these requirements are subject to interpretation by various interested parties both inside and outside the Government For instance, an operational user may interpret a system functional requirement, such as sortie replay, differently from a contractor interested in developing the software to perform that particular system function. Often, the user does not understand his own requirements until he actually tries using a system that incorporates them. Likewise, requirements may be interpreted differently among the many Navy communities. The process of message fusion, for example, can take on differing meanings between intelligence and communications experts The risk of misinterpreted requirements is potentially a system that cannot communicate with external commands, centers, and systems, does not perform the functions needed to meet the Navy's mission, and does not provide the capabilities for the operator to perform required duties Such misinterpretations have a potentially serious impact on reuse goals.

For all persons involved in the initial development phases to attempt to have a uniform understanding of the requirements, concise definition of terms and functions must be conveyed in the requirement specification. This step along with scheduled meetings to answer questions and allay conflicting or incorrect requirements interpretation will help to provide a base on which requirements can be better communicated and understood by the different organizations and various interests. The actual meetings used to obtain user's ideas often result in requirements change.

A.3.2.2 Frequently Changing Requirements Many reasons exist to support the basis of frequently changing requirements in the Navy C^2 domain. One is the risk mentioned above, misunderstanding the intended requirements and miscommunication between individuals and groups. A more legitimate reason is that the mission has changed. This can occur due to fluctuations in the political environment or due to the fact that threats and risk to the system are now different. Frequently changing requirements may be impacted by political issues, but is believed to be primarily a technical risk. In addition, budget reductions and forced reductions in scope can have a serious impact on requirements. This is discussed further in subsection A.3.3. The risk of frequently changing requirements delays delivery of the operational system, impacts cost and schedule, and adds confusion as to what the current requirements are.

Delaying the fielding of a system could have great impact on other Navy and military operations. Vital missions may be placed on hold or valuable resources reserved for other purposes may be required as stand-ins until the new system becomes operational. Costs typically increase rather than decrease as a result of changing requirements; however, in recent years the Government has made changes to requirements as a cost savings effort. The same reasoning can be applied to schedules, too. The key is that if requirements are changed frequently, any desired cost and schedule savings may not be met. One of the greatest areas of concern is the confusion factor caused by multiple versions of requirements. This is especially true as the development process progresses. The development team may be off designing or coding to a set of requirements that are, in fact, not the set of desired or current requirements. Studies have shown that in latter stages of system development, costs increase exponentially as modifications are made to the design; hopefully, this would not be true in a component driven reuse development paradigm.

Methods to reduce this risk must focus on pinning down, as firmly as possible, what is needed to support the mission, then communicating this to all affected organizations. Using this strategy will support credibility of the Program Office should the requirements change again. One way to achieve better understanding is to employ prototypes and incremental builds and releases during design and development.

A.3.2.3 Assurance Assurances are special measures taken to increase the confidence that the implemented system enforces the trust policy. Assurances are intended to reduce the risk of a policy breach, and therefore act to reduce the risk that a system development effort will produce a low-quality or unacceptable product. Nevertheless, assurances may be difficult to carry out successfully within cost, schedule, and available technology constraints, because assurance techniques for trust-critical systems vary widely and some assurances may conflict with other important system requirements. Technology limitations, the knowledge base of the safety analysts, and the pervasiveness of safety-critical functions within a system increase the risks of safety assurance. For the purposes of Navy C^2 systems, safety-critical is interpreted as mission-critical. Examples of a mission-critical function risks in a Navy C^2 system would include the inability for a ground support facility to communicate with a supported aircraft or the inability to provide the aircraft with correct data and operational

programs to support the mission.

Both the TCSEC [19] and the recent draft interim standard for safety-critical systems (MOD 00-55) [27], issued by the British Ministry of Defense, define assurance techniques that carry substantial risks under today's practices. Compliance with system architecture requirements such as minimizing the extent of mission-critical software, defensive programming, etc., is a major risk for trusted system development because it depends on highly-skilled system architects. If the system architecture is deficient, other kinds of assurances may become unobtainable, for example, successful system testing, security testing and formal verification. Risks also occur in the use of formal verification. These risks include the weakness of current verification tools, the lack of verification systems that support proofs about programs written in widely-used languages such as Ada, and the fact that the current paradigm for building trusted systems limits the gains that can be achieved from verification.

A.3.2.4 Trust Skill Specialization Since trust is a relatively new technology, there are only a limited number of software professionals who have training and experience in the development of trusted Navy C^2 systems. These people are likely to be considered a scarce resource best employed as a team of specialists. As a result, the practice of building trusted systems today usually involves a trust engineering team and a software development team, each with specialized skills, and with little skill overlap between them. The current situation for a secure system development is illustrated in Figure 21.

Typically, the trust engineering team helps define assurance-related design, and reviews the system design as it evolves to ensure that the standards are followed. In addition, the trust engineering team may be responsible for producing such trust-related deliverables as top-level specifications and covert channel analyses. The software development team is responsible for on schedule, within-budget delivery of a system that meets all of its requirements, including some subset that concerns trust. This division of labor poses the following risks:

1. The development team may lack sufficient understanding of trust principles contributing to an inability to incorporate adequate trust into the design process. This creates a potential for failure to meet trust requirements and may cause rework to retrofit trust after deficiencies are found that could lead to cost and schedule overruns.

2. The trust engineering team may be able to veto a potential design on grounds that it violates trust principles, but may lack the design experience or skill to propose credible design alternatives. Furthermore, the trust engineering team may feel its proper role is to emphasize trust exclusively, without respect to adverse effects on other system requirements. This creates a potential for failure to meet the performance or other non-trust requirements.

3. Both the trust and development teams and management must have a thorough understanding of the implications of the reuse requirements on trust and mission needs

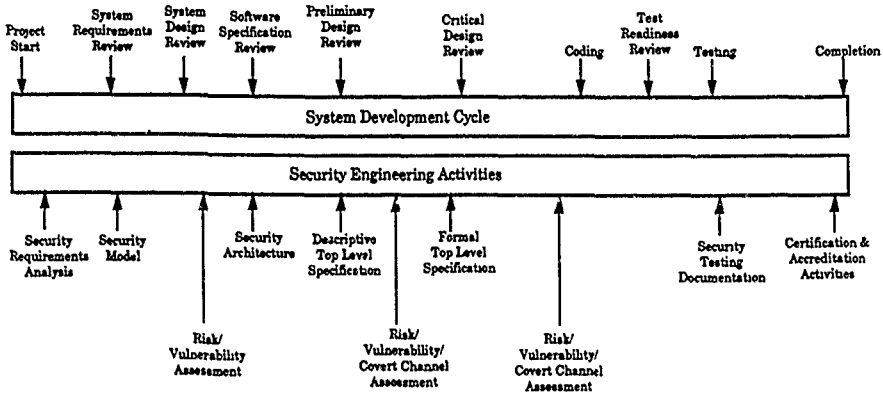


Figure 21: The Current Paradigm

within the development. A careful reuse plan supported by the asset library and automated tools are essential to help mitigate the high risks associated with reuse of trusted assets.

4. There may not be a single chief architect with sufficient authority and design insight to solve apparent conflicts between trust, reuse and other requirements, creating a potential for inconsistencies in design approach and lower product quality due to the "design by committee" syndrome.

A.3.2.5 Architecture Given a well-formed policy and an accurate and complete formal policy model, formulating a Navy C^2 architecture to enforce the policy constitutes another risk factor. The architecture may be constrained by COTS limitations, hardware instruction set characteristics, performance requirements, or requirements for compatibility with an existing untrusted system. Given a limited trust experience base, it may be difficult for system designers to assess the effect of architectural decisions on application developers or end users. For example, poor architectural decisions may:

1. Cause severe distortion to the "natural" structure of application programs, leading to high development and maintenance costs or loss of run-time efficiency;
2. Be incompatible with existing COTS products or other available Navy C^2 reusable software; and
3. Cause the user interface to become unacceptably awkward.

A.3.2.6 Technology Overall technological immaturity underlies most of the specific risk areas associated with developing reuse-driven, trusted Navy C^2 systems. Since they are emerging disciplines, trust and reuse are not yet supported by solid conceptual foundations. While certain principles have emerged, there remain important topics for which the issues are ill-understood. For example, although the definition of confidentiality stemming from well-established DoD regulations governing handling of classified material is relatively well-understood, there is little consensus that current definitions of integrity as a trust characteristic are useful in practice. Assured service as a trust objective has a clearer intuitive meaning; how assured services should be manifested in functionality or architecture is much less clear. The conceptual foundation for trusted systems is also weak in the areas of TCB extensibility and reusability, formal methods and system verification. The domain analysis process and early planning for reuse are in the research stage with no well established approaches and widely accepted practices that can be employed.

Even in areas where the conceptual foundation is relatively firm, engineering techniques and practices are not yet well established. Although a number of trusted and reuse-driven systems have been built and studied by experts, few if any pedagogical examples have been produced and targeted to the broader software engineering community. The vast majority of software professionals lack exposure to the reuse and trust concepts, principles, design

techniques, and examples, and organizations desiring training in trusted system development may find limited offerings from training firms and seminars and little or nothing in the trusted reuse area. Another indication of the technological immaturity is the limited availability of trusted commercial off-the-shelf (COTS) products, reusable components and support tools from which trusted systems can be built.

Due to the fact that Ada is a relatively new language, the immaturity of Ada technology is another technological limitation. The initial largest problem with Ada was compiler risk, i.e., correct programs not compiling, incorrect compilation or inefficient code. The immature Ada support environments also pose a high-risk issue, particularly for trusted systems development. Required to may be missing or available only in prototype versions and tools may be incompatible or inefficient. A third Ada technology risk is the current inadequate support for the use of Ada throughout the life cycle. Although tools to support Ada as a design language are available, support for Ada reasoning for trust and performance, and integrated configuration management and control for t Ada life cycle, are inadequate at best.

A.3.2.7 Performance The development of any reuse-driven, trusted Navy C^2 system using the Ada programming language may be significantly linked with its performance characteristics, including system availability. System performance modeling through the design/development period is a needed risk-reduction mechanism. These performance risks cannot be eliminated through the use of Ada or any other programming language, which may itself incur additional performance risks. First, system trust may add functionality, without regard for the programming language, in that trusted systems require access checking, data and output labeling, auditing, erasure of disk and memory areas, and user identification and authentication. Second, although some of this functionality is localized and is used only occasionally or only on command (such as the login function), much of it is pervasive throughout the system architecture and is used continually as the system operates. For example, access checking of a user's or process' authorization against the classification of data being accessed takes place whenever files are opened, and data and output labeling, auditing, and erasure of disk and memory areas take place continuously. Third, language-specific performance risks exist; for although Ada was designed for use in meeting the performance risks of real-time systems, risks do remain which inhibit the most effective use of Ada. Ada real-time performance issues include several sub-issues, many of which are vendor-or tool implementation-specific. Ada's powerful features can contribute to degradation of system performance if the compiler and run-time systems are immature or unfamiliar Also, Ada is a complex language, and indiscriminate use of its features may require large amounts of memory, reducing the availability of system resources and performance.

A.3.2.8 Ada-related For a trusted Navy C^2 system, the primary Ada project risk items fall into the following categories. technological immaturity; performance risks; inexperienced staff; inadequate resources; integration of Ada and non-Ada code [23]; and conversion of non-Ada code to Ada [23], particularly as they relate to achieving trust and performance

The specifics of the issues of technological immaturity and performance are discussed in the generalized topic sections above.

The risk regarding Ada-inexperienced staff comes about because there is not at present a significant base of Ada experience for trusted systems. The Ada language includes advanced features not available in other commonly-used languages which are seductive and easy to misuse, e.g., Ada tasking, generics, packages, exceptions, elaborations, and limited private types. It is also easy to over-assess the advantages of Ada, i.e., to expect that any code written in Ada will be portable and easy to maintain or that errors and sloppy code will be prevented by the compiler. There is a great misuse of Ada features in the pursuit of trust and performance which may preclude formal verification, as well as the inappropriate use (or non-use) of software engineering aspects of Ada, i.e., choosing a poor set of objects using object-oriented design.

Risks encompassing inadequate Ada resources include inadequate provision for requirements of resources: people, budget, computer, and schedule for an Ada project. Ada compilers are more powerful and have greater functionality than other common language compilers and need additional computing resources, requiring more mass memory and a more powerful CPU. The immaturity of tools from vendors, and the current lack of commercial library packages may cause schedule problems. Adequate training for personnel can divert resources from the development effort and access to "Ada gurus" is a critically scarce resource. Mismatches between pre-Ada budget, schedule, milestones, and cost drivers, and the reality of actual Ada developments (especially given the lack of sufficiently trained personnel) could result in inadequate resources for successful project performance, especially in support of the trust and performance requirements of the system. This risk should be addressed explicitly and early in the project by the project manager and resolved with upper management support.

Due to the limited quantity of existing Navy C^2 Ada code, initial trusted Navy C^2 systems involving reuse may be required to reuse some non-Ada code. To integrate non-Ada code with Ada code, source code such as global common data must be converted to Ada for compatibility with the new Ada code.

Integration of Ada and non-Ada code is another risk area that can be addressed with a management and design approach developed from past experience on Ada projects and on large, team-oriented software development projects. This approach involves using Ada packages to encapsulate related modules within a formally defined unit. Non-Ada modules are segregated from the Ada code and accessed through interface packages. This design approach provides a proven means of integrating code from dissimilar sources, provides a means of quickly generating a testbed to perform integration and performance testing, and supplies a structured decomposition of the system into units that are used as the basis of progress and configuration management.

As stated above, non-Ada source code, such as common global data, must at times be converted to Ada for compatibility with the new Ada code. A code translation tool could be used to quickly convert the code to Ada but this introduces large maintenance risks since

code translators do not produce easily readable and modifiable code. If non-Ada code cannot be integrated as-is with the Ada code (e.g., with interface programs) it is usually better to reuse the code design and have good Ada designers and programmers create design and code from the non-Ada design. This is a trade-off which minimizes the maintenance risks but increases the time required for the code to be produced.

A.3.2.9 Documentation A number of the trust-related deliverable documents are closely related to traditional non-trust deliverables. If the trust-related documents are produced solely by the security engineering team, the isolation can cause contradiction or redundancy with regard to the non-trust documents produced by the system development team, as well as unnecessary expense due to duplication of efforts. The risk that trust-related documents will drift into inaccuracy due to the ongoing evolution of the system during design and implementation is an even greater risk. If this should happen, it may be necessary to reconduct extensive analyses, or to rework the design to comply again with trust assurances.

Examples of closely related non-trust and trust documents requiring close coordination or integration include the following:

- System requirements versus trust policy and rationale;
- Preliminary and detailed design versus formal and descriptive top-level specifications, covert channel analysis, or hazard fault tree analyses;
- Test plans, procedures, and results versus trust testing or safety analysis;
- Manuals for the user, operator, facility manager, and maintainer versus manuals for trust administrator, safety operator, trusted system programmer, and trusted facility manager.

An additional documentation-related risk relative to reusing existing software in Navy C^2 systems is the availability, completeness, and standardization of documentation. Due to funding and schedule constraints, software documentation is often not updated to "as-built" status. Also, the level of the documentation is often not standardized between Navy C^2 programs.

Another serious problem exists in that current software documentation standards are not particularly useful to software maintainers or reusers. These standards are aimed at controlling the development process and reducing the inherent risks. The standards are not optimized towards rapid understanding of the software design. Furthermore, the standards do not encourage a trust-oriented approach to software system design.

A.3.2.10 Standards One manifestation of trust assurances is the imposition of special design, coding and naming standards such as the avoidance of global variables, pointer

types, and designated operating systems services, or the use of module naming conventions to differentiate TCB and non-TCB components for TCSEC trusted systems. TCB components may be subject to other special standards such as analysis by automated code auditing tools, more extensive testing requirements, earlier baselining and submission to configuration management control, and/or review of changes by a special security configuration control board (SCCB). If special standards are not clearly identified to the development team and integrated into regular standards and practices manuals, they will be inconsistently observed, leading to confusion, rework, and lower product quality.

A.3.2.11 Trust Assurances During Maintenance Maintenance introduces significant and continued risk into the development cycle. Modifications to the trusted portions of the system risk invalidation of the architectural constraints that provide assurance for the trusted system. Unless very carefully controlled, modifications can, over time, undermine the architectural integrity of the system that is fundamental for trust. Implications of the modifications on system performance must be carefully monitored and analyzed. Tracking the implications of the modification necessitates re-examination of the analysis performed to provide assurance for the trust characteristics of the system. For example, in TCSEC systems, modifications to the trusted computing base invalidate the trust rating of the system, and re-evaluation must be performed to achieve a rating for the modified system. For safety-critical systems, modifications may invalidate the results of software safety analyses performed during development. Software upgrades involving safety in a high-priority-emergency-fix situation must be carefully managed to ensure that no significant shortcuts of safety and maintenance methodologies occur.

Re-evaluations introduce considerable risk and cost to the continued system or product life cycle. Maintenance of trusted systems remains a research area, and thus has the risks associated with a domain that is not well-understood. Since system maintenance activities are frequently carried out by personnel other than the original development team, additional risks are introduced. If maintenance personnel are not provided with trust training and rigorous trust-supportive standards, risks of violation of trust assurance and potential loss of certification accrue. There exists a need for verification tools that are easy to use and rely on persistent storage of earlier proofs and verification conditions to speed reverification.

A.3.3 Programmatic Risks

Programmatic risks that are associated with project management and the human and sociological aspects of system development are extremely important issues in the development of a large, complex system such as a Navy C^2 application. Frequently, the failures or resulting problems uncovered in the final analysis of a completed system can be traced to the human aspects of the development. Some of the human aspects are closely related to the technical ones. For example, while requirement satisfaction is largely technical in nature, the roles of the humans involved and their political viewpoints, communication abilities and mechanisms are paramount. There are five programmatic risk areas identified for Navy C^2 system

developments. They are:

1. Programmatic, Political and Sociological
2. Opposing Interests
3. Cost Constraints
4. Schedule Constraints
5. Program Coordination, Management and Assurance

A.3.3.1 Programmatic, Political and Sociological The sociological risks within a Navy C^2 system development represent the human aspects of the process and include communications methods, standards, procedures, the Navy and contractor cultures, and the impact to staffing continuity and stability. The skill mix, the understanding of basic trust principles and reuse goals, and Ada experience may vary considerably. Skill specialization is a necessity for a Navy C^2 system development project and cross training of personnel will be necessary for project success and cost effectiveness.

Risks associated with poor communication remain high throughout the system development, and are of highest priority in the early stages of the development process when concepts and requirements that drive the system implementation are formulated.

Retainment of military personnel on a particular system is difficult within the Navy environment where rotations within a two to three year interval are common. This lack of personnel stability both for system users and program management is an inherent risk for Navy system developments.

A.3.3.2 Opposing Interests Political ramifications represent significant risks for reuse–driven, trusted Navy C^2 system developments. The high performance, trusted system development must deal not only with contractor technical and management interests and Navy user community and program management, but also with external evaluation, certification and accreditation groups. Each of these groups has a specific goal and these goals may not be in total conformance with one another.

A.3.3.3 Cost Constraints Cost is a significant risk area both in terms of resources and scheduling. There is a reluctance to commit resources on the front end of a project. On both the contractor and Navy sides, tough problems may tend to be ignored or de–emphasized until a time when they have become very costly to correct. High priority technical risk issues relating to trust, high performance and reuse need to be identified early in the Navy C^2 project so that adequate resources (perhaps additional funds) can be applied early in the life cycle. This is a crucial risk area with respect to the adequacy of reuse planning.

More recently, with the cost reduction and budget pressures facing the Navy (this is a Government-wide concern), there has been a trend toward Fixed Price contracts and a sacrificing of some requirements to keep down procurement costs. This is a risk area for both the Navy and the development contractor with respect to fully satisfying the Navy C^2 system requirements under the constraints of limited resources.

A.3.3.4 Schedule Constraints Schedule constraints are closely related to the risks associated with the tight Government resources of today. There must be flexibility in project scheduling. Inadequate provision of resources for realistic scheduling of a project that must achieve high performance, reuse and trust goals places the development at risk at its onset. The nature of trusted Navy C^2 system development risks requires an early emphasis on analyses, prototyping and modeling to help assure the fulfillment of requirements and the ultimate success of the implementation. This means that scheduling of system engineering activities early in the project is crucial.

Due to the continuing need for cost reductions, some Navy C^2 systems have remained in use beyond their expected lifetimes, and there is a growing need to rapidly deploy upgrades and system replacements. This need places a heavy emphasis on meeting a tight schedule, sometimes at the expense of functionality and/or maintainability and may even create a higher cost burden in the long run.

A.3.3.5 Program Coordination, Management and Assurance The complexity of a trusted Navy C^2 system development creates risks associated with the management and control of parallel activities, the management of irregular progress and the provision of adequate trust assurance in the resulting system. A Navy C^2 development requires accurate tracking of resources and progress by contractor and Government management. If the budget is realistic, it is not as difficult to determine the status of a project and determine how "complete" it is. However, in today's environment project management is extremely complex, especially under tight resources and within reuse goals. Support tools are essential to help monitor and track the project progress, the system baselines and the assurance activities and products. The lack of adequate, integrated support tools and process management automation is a significant risk for Navy C^2 system development.

A.4 FUTURE APPLICATION OF RESULTS

This initial identification of characteristics and risks for the trusted Navy C^2 system domain supports the primary US40 tasking to tailor the previous TRW process model work to the STARS goals for reuse within a Navy C^2 application domain. Information within this appendix has been used to enhance the STARS Composite Process Model (SCPM) and incorporate domain-specific risk mitigation activities identified for the development of reuse-based, trusted, high performance Navy C^2 systems. Defined Navy C^2 characteristics can be used to help derive top level objects, operations and their interrelationships and provide a

basis for further domain analysis and modeling. One goal of this task and follow-on work is to experiment with process model representations for aspects of the Navy C^2 domain with an ultimate goal of process automation. This appendix provides domain information for the initiation of the domain-specific process model representations.

A.4.1 Refinement of the Process Models

Based on the Navy C^2 domain risks described here and on previous process model guidance, corresponding risk mitigation approaches have been derived and are included in subsection 2.3. We have introduced more domain specificity into the SCPM spirals of activity and have provided more prescriptive sets of activities for the development of trusted, high performance Navy C^2 systems. In addition, we have incorporated into this final report process spirals for domain analyses and pre-contract activities within the Navy C^2 domain, a set of "spiral 0" activities.

This work has yielded significant lessons learned which could be used to enhance the original SCPM. For example, a more generic version of the NCCPM "spiral 0" activities should be incorporated into the SCPM. In addition, if the NCCPM were employed on a pilot Navy C^2 development project, feedback from that effort would provide substantial guidance for refining the NCCPM (and, by extension, the SCPM) to better accomodate production needs

A.4.2 Navy C^2 Domain Model and Process Model Representations

The Navy C^2 risks and characteristics identified for this report represent preliminary domain modeling work that can be applied to the development and enhancement of previous domain modeling efforts. This initial characteristics determination can support the description of a top level domain model for the trusted Navy C^2 application domain and help refine the descriptions of objects, operations and their interrelationships. A domain model comparison and enhancement with the Unisys NCCS-Afloat Information Object Model [22] (derived in STARS UQM-15 Phase II, December 1989) would be a useful exercise, although beyond the scope of the current task.

Enhanced domain model descriptions will support the goals for process representation and automation by providing a precise structure and basis for process descriptions within the application domain. The process representation exercises will require refinements of domain-specific process and model descriptions and analyses of automation capabilities. Time and technology constraints preclude extensive experimentation within the current task and early follow-on efforts. Process programming languages and process automation specifications are new areas of investigation. The future goals of the domain specific tasking include trade-off analyses, in-house experimentation with candidate process representations and automated capabilities, and more work toward automated process specification.

A.5 ACKNOWLEDGMENTS AND REMARKS

Our work records the information obtained through both document research and Navy C^2 domain meetings with domain experts. We especially thank James Hartz, Stephen Hertz, John Johnson, and Gary Rogers from TRW for their technical support, experience sharing, and for their time at domain meetings. We also thank Gary Rogers and Stephen Hertz for their sanity check reviews of our work and for their comments and contributions.

This appendix represents a two month, part time effort for an initial characterization of trusted Navy C^2 systems and a determination of the major development risks for such systems. As described in Section A.4, the risks have been analyzed further to determine risk mitigation approaches and activities for the trusted Navy C^2, reuse-driven process model description in our final report. The risks and mitigation activities are included in Section 2.3. To fully characterize trusted Navy C^2 systems and apply reuse concepts, much more detailed analyses will be required and more information from reuse and domain experts will be needed.

B ACRONYMS

A

AAW	Anti-Air Warfare
ACP	Allied Communications Publication
ACS	Advanced Computing Systems
AOU	Area of Uncertainty
AP	Acquisition Plan
ARB	Acquisition Review Board
ASUW	Anti-Surface Warfare
ASW	Anti-Submarine Warfare
ASWCCCS	Antisubmarine Warfare Centers Command and Control System
ASWOC	Antisubmarine Warfare Operations Center

B

BAFO	Best and Final Offer

C

C^2	Command and Control
C^3	Command, Control, and Communications
C^3I	Command, Control, Communications and Intelligence
CCA	Covert Channel Analysis
CDR	Critical Design Review
CDRL	Contract Deliverable Requirements List
CECOM	Center for Software Engineering
CIDS	Critical Item Development Specification
CLI	Communications Line Interface
CM	Configuration Management
COTR	Contracting Officer's Technical Representative
COTS	Commercial Off-the-Shelf
CPA	Closest Point of Approach
CRISD	Computer Resources Integrated Support Document
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSOM	Computer System Operator's Manual
CSU	Computer Software Unit
CVA	Clandestine Vulnerability Analysis

D

DAA	Designated Approval Authority
DARPA	Defense Advanced Research Projects Agency
DBMS	Data Base Management System
DDN	Defense Data Network
DoD	Department of Defense
DSCS	Defense Satellite Communications System
DT&E	Developmental Test & Evaluation
DTC	Navy Standard Desktop Tactical Support Computer
DTLS	Descriptive Top-Level Specification
DTR	Document Trouble Report

E

ECP	Engineering Change Proposal
EMCON	Emissions Control
EW	Electronic Warfare

F

FASIT	Fleet All-Source Intelligence Terminal
FCA	Functional Configuration Audit
FCC	Fleet Command Center
FQT	Formal Qualification Testing
FSED	Full Scale Engineering Development
FSM	Firmware Support Manual
FTLS	Formal Top-Level Specification

G

GFE	Government Furnished Equipment
GFI	Government Furnished Information
GLOBIX	Global Information Exchange System
GOTS	Government Off-the-Shelf

H

HARPOON	Over-the-horizon cruise missile
HF	High Frequency
HICOM	High Command Communications Net
HOL	High Order Language

I

IDD	Interface Design Document
IR&D	Internal Research and Development
IRS	Interface Requirements Specification
ISTO	Information Science and Technology Organization
IV&V	Independent Validation and Verification

J

JANAP	Joint Army Navy Air Force Publication
JINTACCS	Joint Interoperability of Tactical Command and Control Systems

L

LAN	Local Area Network
LAT	Latitude
LCDR	Lieutenant Commander
LMA	Land-Mass Avoidance
LONG	Longitude

M

MENS	Mission Element Needs Statement
MIIDS/IDB	Military Integrated Intelligence Data System/Integrated Database
MLS	Multilevel Secure
MMI	Man-Machine Interface
MOD	Modification
MOU	Memorandum of Understanding

N

NCCPM	Navy Command and Control Process Model
NCCS	Navy Command and Control System
NCSC	National Computer Security Center
NDCP	Navy Decision Coordinating Paper
NDI	Non-Development Item
NDS	Non-Developmental Software
NOSC	Naval Ocean Systems Command
NRL	Naval Research Laboratory
NTDS	Navy Tactical Data System

O

OBU	OSIS Baseline Upgrade
OPDEC	Operational Deception
OPEVAL	Operational Evaluation
OPTEVFOR	Operational Test and Evaluation Force
OR	Operational Requirement
OSIS	Ocean Surveillance Information System
OSS	Operations Support System
OT&E	Operational Test & Evaluation
OTH-T	Over-the-Horizon, Targeting

P

PCA	Physical Configuration Audit
PDL	Program Design Language
PDR	Preliminary Design Review
PE	Program Element
PED	Program Element Description
PIDS	Prime Item Development Specification
PM	Process Model
POM	Program Objective Memorandum

Q

QA	Quality Assurance

R

RFP	Request for Proposal
RLF	Reusable Library Framework
RMA	Reliability, Maintainability, and Availability
RMP	Risk Management Plan

S

SATCOM	Satellite Communications
SCCB	Security Configuration Control Board
SCMP	STARS Composite Paradigm Report
SCN	Specification Change Notice
SCPM	STARS Composite Process Model
SDD	Software Design Document
SDF	Software Development Files
SDL	Software Development Library
SDP	Software Development Plan
SDR	System Design Review
SEE	Software Engineering Environment
SEI	Software Engineering Institute
SEMP	System Engineering Management Plan
SETA	System Engineering and Technical Assistance
SEW	Space and Electronic Warfare
SIGINT	Signal Intelligence
SIT	System Integration Test
SPM	Software Programmer's Manual
SPS	Software Product Specification
SPCC	Ships Parts Control Center
SPT	System Performance Test
SRR	System Requirements Review
SSDD	System/Segment Design Document
SSR	Software Specification Review
SSS	System/Segment Specification
ST&E	Security Test & Evaluation
STARS	Software Technology for Adaptable Reliable Systems
STD	Software Test Description
STD	Standard
SUM	Software User's Manual

T

TAC	Tactical Computer
TADIXS	Tactical Digital Information Exchange System
TASM	TOMAHAWK Anti-Ship Missile
TCB	Trusted Computing Base
TCP/IP	Transmission Control Protocol/Internet Protocol
TCSEC	Trusted Computer System Evaluation Criteria
TECHEVAL	Technical Evaluation
TEMP	Test and Evaluation Master Plan
TFCC	Tactical Fleet Command Center
TRR	Test Readiness Review

U

UID	User Interface Document
USMTF	United States Message Text Formats

V

VDD	Version Description Document

References

[1] Boehm, Barry W., "Spiral Model of Software Development Enhancement," *IEEE Computing Surveys*: 61–72, May 1988.

[2] Royce, Walker W. "Incremental Development of Large Ada Systems: An Ada Process Model," *Proceedings of the 1989 ACM Tri-Ada Conference*, October 1989.

[3] *Impact of Domain Analysis on Reuse Methods*, Software Productivity Solutions, Inc., prepared for U.S. CECOM Army Center for Software Engineering, November 1989.

[4] *STARS UR40 – Repository Integration: Review of Existing Repository Technology*, Unisys Defense Systems, Paoli, PA, February 1989.

[5] *Framework of Issues in the Reuse of Trusted Software*, Unisys Defense Systems, Paoli, PA, and Trusted Information Systems, Glenwood, MD, STARS-RC-01550/001/00, July 1990.

[6] Holibaugh, Robert et.al., "Reuse: Where to Begin and Why" *Proceedings of Tri Ada 1989*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, September 1989.

[7] *Process Model for High Performance Systems in Ada, Phase I Technical Report*, TRW Systems Division, Fairfax, VA, August 1989.

[8] Shu, Christine, *Experience with Using V base and APPL/A for Process Modeling and Programming*, TRW Arcadia Project, January 1990.

[9] Kitaoka, Beverly J., "Repository Support for a Reuse Process," *Proceedings of the 8th National Conference on Ada Technology*, Atlanta, GA, March 1990.

[10] Sutton, Stanley M., et.al., *Language Constructs for Managing Change in Software Process Programs*, University of California, Irvine, CA. and University of Colorado, Boulder, CO, August 1989.

[11] Creps, Richard "Unisys STARS Reuse Technology," Unisys Defense Systems briefing, September 1990.

[12] Solderitsch, James J., et.al., "Constructing Domain-Specific Ada Reuse Libraries," *Proceedings of the 7th Annual National Conference on Ada Technology*, March 1989.

[13] Balzer, Robert, et.al., "Software Technology in the 1990's: Using a New Paradigm", *IEEE Computer*, November 1983.

[14] McCracken, Daniel and Jackson, Michael, "Life-Cycle Concept Considered Harmful," *ACM Software Engineering Notes*: 29-32, April 1982.

[15] Royce, Walker W., *Ada Process Model*, TRW Systems Engineering & Development Division, Carson, CA, November 1989.

[16] Solderitsch, James and Payton, Teri, *A Basis for Domain Specific Support Environments*, Unisys Defense Systems, Reston, VA, May 1990.

[17] *URAO: Repository Integration: Draft Definitized Repository Specification*, Unisys Defense Systems, Paoli, PA, STARS-RC-01240/001/00, April 1990.

[18] Layman, Gene, "Ada Repository Program," Naval Research Lab Briefing, October 1990.

[19] National Computer Security Center, Department of Defense, *Trusted Computer System Evaluation Criteria*, DoD 5200.28-STD, December 1985.

[20] *SDI-BMS Security Accreditation Study*, Phase 1 Report, Technical Report TM-(L)-8361/004/00, Unisys Corp., Camarillo, CA, March 1988.

[21] *Draft Composite Paradigm Report for Software Technology for Adaptable Reliable Systems*, TRW Systems Division, Fairfax, VA, STARS SC-03068/001/00, 10 November 1990.

[22] *STARS UQM15 - Phase II Final Report NCCS-Afloat Information Object Model Structured Specification*, Unisys, Reston, VA, STARS-RC-01450/001/00, 15 December 1989.

[23] *Proposal for Anti-submarine Warfare Operations Center (ASWOC) Command, Control & Communications (C^3) Upgrade Full Scale Engineering Development Technical Proposal Software Development Plan*, TRW Federal Systems Group, Fairfax, VA, 3 April 1987.

[24] *Anti-Submarine Warfare Operations Center (ASWOC) Command, Control & Communications (C^3) Upgrade Program Performance Specification*, TRW Federal Systems Group, Fairfax, VA, 28 April 1989.

[25] *The OSIS Baseline Upgrade (OBU) Product Baseline Description*, CDRL C037, TRW Systems Integration Group, Fairfax, VA, 31 August 1990.

[26] Department of the Navy, *Acquisition of Software-Intensive C^2 Information Systems*, SECNAVINST 5200., 5 January 1988.

[27] British Ministry of Defence, *Requirements for the Procurement of Safety Critical Software in Defence Equipment*, Interim Defence Standard 00-55, May 1989.

[28] LCDR M.S. Loescher, USN, Copernicus Project Office, Director Space and Electronic Warfare, Office of the Chief of Naval Operations, *The Copernicus Architecture* (U), SECRET, December 1990.

[29] Benzel, Terry C. Vickers, *Developing Trusted Systems using DoD-STD-2167A*, Proceedings of the Fifth Annual Computer Security Applications Conference, Tucson, AZ, December 1989.