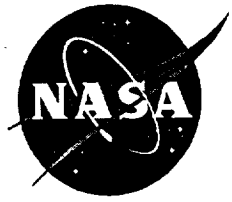
ADVANCED RISK REDUCTION TOOL (ARRT)
SPECIAL CASE STUDY REPORT

SCIENCE and ENGINEERING TECHNICAL ASSESSMENTS (SETA) PROGRAM

CONTRACT NUMBER: NAS2-98028
DOCUMENT NUMBER: 8521.04.15.05.001013
October 13, 2000

Prepared for:



NASA Independent Verification and
Validation (IV&V) Facility
100 University Drive
Fairmont, WV 26554

Prepared by:

D.N.AMERICAN
1000 Technology Drive, Suite 3220
Suite 3220
Fairmont, WV 26554

SAIC
An Employee-Owned Company
1213 Jefferson Davis Highway
Suite 1200
Arlington, VA 22202

Table of Contents

Foreword

This document describes SETA's analysis of the Advanced Risk Reduction Tool (ARRT)

Prepared by: Paul J. Kirsch, SAIC
Jane Hayes, SAIC
Lillian Zelinski, SAIC

Reviewed By: Laurie Smith, D.N. American, Inc.
Frank Huy, D.N. American, Inc. Program Manager

Technical Lead: Paul Kirsch, SAIC

Approved By: Frank Huy, D.N. American, Inc. Program Manager

1 INTRODUCTION

This special case study report presents the Science and Engineering Technical Assessments (SETA) team's findings for exploring the correlation between the underlying models of Advanced Risk Reduction Tool (ARRT) relative to how it identifies, estimates, and integrates Independent Verification & Validation (IV&V) activities. The special case study was conducted under the provisions of SETA Contract Task Order (CTO) 15, of NASA contract NAS2-98028, and the approved technical approach documented in the CTO-15 *Modification #1 Task Project Plan.*

Section 2 provides an ARRT project overview and executive summary of SETA's primary finding. Section 3 describes the analyses performed for this Special Case Study; provides examples of potential ARRT risk-methodology enhancements; and makes recommendations for the direction of future tool development. Section 4 compares the approach pursued by the Jet Propulsion Laboratory (JPL) research team to that pursued by Department of Defense (DoD) and the nuclear industry to reduce software acquisition and operational risks. SETA's findings and recommendations are summarized in Section 5.

2 ARRT PROJECT OVERVIEW

ARRT is a comprehensive software tool being developed by the JPL in collaboration with the Glenn Research Center, West Virginia University, Texas A&M University, and Miami University. When fully realized, ARRT is to provide a standard means for:

- Identifying software risks;
- Creating optimal risk-mitigation plans;
- Producing consistent cost and schedule risk-reduction budget estimates; and
- Creating equitably negotiated IV&V, software development, and Quality Assurance (QA) plans that are compliant with NASA policy and ISO 9000 process criteria.

Note that the last requirement is actually a compound of three requirements, in that the tool must be able to create three distinct types of plans: IV&V, software development, and QA.

2.1 Executive Summary

The evaluated version of ARRT is only capable of supporting software development-lifecycle planning; it does not support IV&V or QA planning. The balance of this report discusses why this is so, and provides recommendations for ways to expand ARRT capability into the domain of IV&V planning, which is the primary interest of the NASA Software IV&V Facility.

3 SPECIAL CASE STUDY EVALUATION

The Statement of Work for this Special Case Study prescribed four distinct analysis activities:

1. [Evaluate the] IV&V activities and how they correlate to Capability Maturity Model (CMM) Key Practice Areas (KPA) and Software Engineering Institute (SEI) Risk Element (RE).
2. [Determine what] risk mitigation activities are missing or are inappropriate.
3. [Evaluate resource] estimation efficiency.
4. [Analyze the] relationships between risk mitigation activities and the risks.

The third task was deleted from the Special Case Study when it was learned that AskPete, the cost-modeling database, is still being developed at the Glenn Research Center. Expectations for each of the remaining activities were clearly defined on 18 September 2000, via a teleconference attended by the SETA team, and Mr. Marcus Fisher and Mr. Ken McGill of the NASA Software IV&V Facility. The approved technical approach was captured in the CTO-*15 Modification #1 Task Project Plan*. The following sections define the revised tasks, present findings, and offer recommendations for the direction of future tool development.

3.1 ARRT Failure Mode (FM) Analysis

The goal of this analysis task was to determine the level of correlation between the Failure Modes (FM) defined in ARRT and SEI RE.

3.1.1 Analysis Approach and Findings

The results of the analysis revealed that the ARRT Failure Modes map directly to the Risk Elements defined by the SEI; in fact, we have since learned that the FMs were copied verbatim from the SEI Technical Report "Taxonomy-Based Risk Identification," CMU/SEI-93-TR-6, ESC-TR-93-183.

3.1.2 Recommendations

Since 100% correlation exists between the two lists, no further analysis is required or recommended. Appendix A presents the SEI Risk Element Taxonomy versus the ARRT Failure Modes.

3.2 ARRT Preventive measures Analyses Control Tests (PACTs) Analysis

The goal of this analysis task was to determine the degree to which the PACTs[1] defined in ARRT represent the kinds of risk-mitigation activities performed by IV&V contractors.

1 PACT is an acronym for: Preventative measures (e.g. design rules, material and parts selection, architecture, redundancy), Analyses (e.g. structural, optical, chemical, electrical performance, FMECAs and other reliability analyses), process Controls (e.g. inspections, coupon sampling, standard procedures and processes) Tests (e.g. functional, environmental, stress screening)

3.2.1 Analysis Approach and Findings

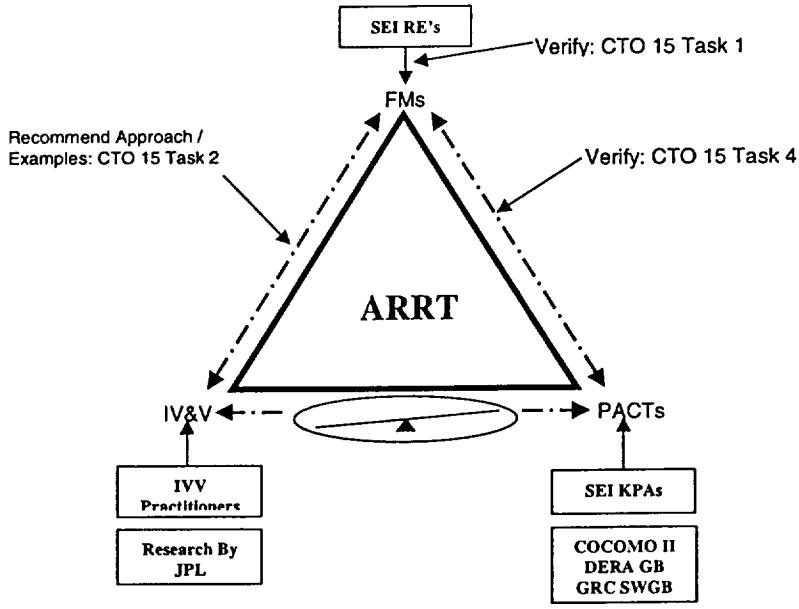
The SETA team first inspected the PACT detailed listing and hierarchy trees to (1) determine whether the activities described sufficiently represent the full scope of activities that are typically conducted by IV&V practitioners, and (2) identify missing or inappropriate activities.

Our inspection revealed that the PACTs are software development-best practices derived from SEI Key Practice Areas (KPAs) -- they are not typical IV&V activities. After discussing this finding with the NASA customer, it was decided that a more useful approach would be to develop examples of how the ARRT Failure Modes might be mapped to typical IV&V risk-mitigation activities. The result is provided in Appendix B and represents SAIC's best judgment based upon 15 years of experience in conducting IV&V for various projects.

3.2.2 Recommendations

A tradeoff between the cost of implementing a best practice (PACT) and its effectivity exists. Thus, while IV&V can always add unique value to a program due to its financial and managerial independence from the developer, the composition and required stringency of IV&V support is a function of the unique characteristics of the program and the development environment. To achieve the right balance requires that these factors be considered in a systematic way when tailoring an IV&V activity lifecycle. Figure 3-1 illustrates this concept as applied to ARRT (as well as a graphical depiction of the focus areas of this Special Case Study).

Figure 3-1 Balancing Best Practices, Risks, and IV&V



Because the current PACTs are derived from SEI KPAs, they are by definition good software development practices, rather than complementary or reinforcing IV&V activities. The "see-saw" at the bottom represents how a balance between good software development practices and supporting IV&V is required to efficiently minimize risk.

Thus, while the visual depiction of risks that ARRT now provides is useful for manually balancing costs versus specialized development and IV&V activities, the tool should also include heuristics or algorithms to find the most efficacious mix of PACTs, by weighing activity costs provided by AskPete against validated PACT effectivity weightings.

SETA understands that the current PACT effectivity weightings are preliminary. Once the fullest list of PACTs is assembled (i.e. a list that includes additional specialized software development and IV&V activities) the relative ranking of the activities and their effectivity values should be re-examined, preferably with the goal of achieving a consensus from amongst a panel of experts backed by empirical evidence, if available.

In summary, SETA recommends:

1. Add IV&V-specific activities to the PACT list, as the current list does not include specific IV&V risk-mitigation activities. This will render the list compliant with the requirement to create IV&V plans.
2. Develop an expert consensus for the relative rankings of the PACTs via the Delphi technique, and assign effectivity values developed using a decision science method such as the Analytical Hierarchy Process (AHP). This research will render the output of ARRT defensible to potential Center customers.
3. Develop a strategy for optimally balancing IV&V and software development PACTs using validated PACT effectivities and AskPete cost data. Such research would add significant value to the ARRT tool by making its recommendations more analytical and repetitive.

3.3 ARRT FM-to-PACT Mapping Analysis

The goal of this task was to evaluate the suitability of the mapping of PACTs to FMs currently defined in ARRT.

3.3.1 Analysis Approach and Findings

SETA's original approach was to assess each link as being justified, marginal, or unjustified in SETA's judgment. The rationale for adding or deleting specific linkages would also be provided.

After analyzing the PACTs, SETA concluded that most of the PACTS should be performed on any project because they represent good engineering practices. Rather than report the obvious, SETA decided to do the following:

1. Categorize the current PACT tree into two classes: Standard Good Practices and Special Practices.
2. Define additional examples of Special Practices, which would reduce risk in key areas when the KPA-based PACTs are already in place.

The idea is that Standard Good Practices be considered the default list when creating a Software Development Plan, while the Special Practices will target project-specific areas of weakness.

Appendix C provides SETA's classification of the current, KPA-based PACTs into Standard Good Practices and Special Practices.

Appendix D presents 38 additional Special Practices conceived to complement the current list of generic, KPA-based PACTs, and the recommended mapping between them and the 81 Failure Modes currently in ARRT.

3.3.2 Recommendations

1. Divide the PACTs into Standard Best Practices (SBPs) and Special Practices (SPs). IV&V activities may also be grouped, with some considered SBPs and other, more detailed activities falling within the class of SPs.
   A. The SBPs would constitute the "default" practices NASA should expect, with the SPs providing domain or project specific, targeted risk reduction.
   B. PACT tailoring could then be performed to be consistent with or complimentary to the Capability Maturity Model - Integrated (CMMI).

4 OTHER PERSPECTIVES

This section outlines the risk-management approaches pursued by industry and government groups who, like NASA, have an interest in reducing software acquisition and operational risks. Where available, SETA has listed specific risk areas and PACTs that these groups have identified as being critical. The JPL team may wish to further investigate these perspectives for inclusion in the ARRT tool as specialized PACTs.

4.1 Department of Defense (DoD)

Within the DoD, no single set of Best Practices or methods for the systematic analysis of military system quality or reliability exists. The DoD has moved in recent times to developer-specified Best Practices in the name of Acquisition Reform. A DoD Program Office may scrutinize the processes via an SEI Audit that a potential contractor has in place before a contract is awarded, or may require that the contractor be certified at a certain SEI Level before a contractor is allowed to bid on a contract.

Currently, a major Navy Program uses several Risk Analysis Tools. The Program Office uses the Technical Risk Identification and Mitigation System (TRIMS) module available free from the Best Manufacturing Practices, Center of Excellence (http://www.bmpcoe.org). TRIMS is a tool designed to help identify, quantify, and track risks in a program, and then reduce or mitigate these risks to acceptable levels. It works through out all phases of a program's transition from initial concept to full production and life cycle support. TRIMS is based on proven risk models or published practices such as those of the NAVSO P-6071 Best Practices (Templates) and the SEI.

One of the contractors on this program uses several tools from the Software Program Managers Network (http://www.spmn.com/index.html). The mission of the Software Program Managers Network (SPMN) is to enable managers of large-scale, software-intensive development or maintenance projects to more effectively manage and succeed by identifying and conveying to them, management Best Practices, lessons-learned, and direct support. The tools, Risk Radar and Project Control Panel, are available free from the SPMN as well as excellent literature and videos on Best Practices in many key program areas.

However, when Weapons Safety is involved, more formality is apparent in the DoD. A formal Government Weapons Safety Review Board (WSRB) is involved in the safety validation of the system or sub-system before any live ordinance can be used with the system. The WSRB has well defined processes and checklists that are used throughout the program starting with requirements specification through formal testing.

4.2 Nuclear Industry

Four nuclear regulatory authorities worldwide (United Kingdom's Nuclear Installations Inspectorate, Canada's Atomic Energy Control Board, France's Direction de la Surete des Installations Nucleaires/Institut de Protection et de Surete Nucleaire, and the US Nuclear Regulatory Commission) have agreed on how a safety case should be handled for acceptance of computer-based instrumentation and control (I&C) systems in nuclear power plants. The following describes the good practices, i.e., PACTs, upon which they have reached consensus.

First, the nuclear industry is only concerned with *safety systems* and *safety-related systems.* Power plant systems are systematically analyzed for failure modes, with an assessment conducted of the acceptability of the risk associated with each FM. Only systems critical to safe operation are analyzed. *Safety systems* include emergency reactor core-cooling (protection) systems and decay heat-removal (safety actuation) systems. *Safety-related* systems include radiation monitors, fire detection systems, etc.

For either of these categories (safety or safety-related), a self-standing safety case is required to demonstrate that the:

1. Correctness and completeness of the overall requirements specification is justified relative to the intended system function;
2. System has been designed to standards compatible with its required safety integrity;
3. Delivered system meets all aspects of its requirements specification; and
4. Adequate means are specified to ensure performance of the system throughout its operational life.[2]

Early in the project the following must be considered and evaluated as part of the risk-mitigation plan:

1. Safety importance;
2. Defense-in-depth;
3. System boundaries;
4. Novelty;
5. Basis of the safety case;
6. Licensee/regulator interface;
7. Need for Independent Assessment;
8. Approach to be used if system licensed in another country.

The following are the main safety case requirements that must be met during production of the system. The following may be considered as the nuclear-industry required PACTs:

[2] The principal standards applicable to computer-based systems are found in 0 of IEC 880 (Software for computers in the safety systems of nuclear power stations).

1. General demonstration principles for safety and safety related systems;
2. Complete functional capability;
3. Correct and traceable specifications;
4. Minimizing faults in the design;
5. Fail safe design;
6. Operational testability;
7. Full system testing;
8. Well defined standards;
9. Competent staff and team organization;
10. Quality assurance;
11. Attention to security throughout development;
12. Controlled change process;
13. Well managed documents;
14. Consistency with operating plans and procedures;
15. Attention to human factors of operator interfaces.

Additionally, if it is a safety system, the following must also be developed/utilized/ evidenced:

1. Single failure criterion;
2. Common cause failures;
3. Structured software development process;
4. System design principles;
5. Complete V&V using both test and static analysis; and
6. Use of valid and controlled tools.

Regulators focus attention upon, or look for evidence of:

1. Independent assessment;
2. Defense in depth;
3. COTS;
4. Formal methods;
5. Performance feedback; and
6. Technological developments.

4.3 Aviation Industry

In the United States, the safety requirements for civil aviation systems are defined and published by the Requirements and Technical Concepts for Aviation (RTCA), which is an association of aeronautical organizations from both government and industry. RTCA is not an official agency of the U.S. Government, but its findings are followed as a standard guideline and are recognized by the Federal Aviation Administration (FAA). The findings of the RTCA are jointly developed with the European Organization for Civil Aviation Equipment (EUROCAE) WG-12 through a consensus process. RTCA Document RTCA/DO-178B, *Software Considerations in Airborne Systems and Equipment Certification,* provides guidance for airborne systems. An appendix to DO-178B, soon to be issued, shall provide similar guidance for ground-based systems.

DO-178B discusses several aspects related to software development and ways to ensure that risk is mitigated. This is achieved through redundancy suggestions and the use of dissimilar software running in parallel. It also classes software into five levels, A-E, based upon the contribution of software to potential failure conditions, which might provide a useful set of criteria for determining the level and stringency of IV&V that should be applied to a project. Also considered as PACTs that can mitigate risk are:

1. System Architectural Considerations – ways in which the system safety assessment process can determine whether the system architecture preludes anomalous behavior.
2. Partitioning – the technique by which isolation between functionally independent software components can reduce the software verification process.
3. Multiple-Version Dissimilar Software – a design technique that involves producing two or more software components that provide a function in a way that can avoid sources of common errors.

DO-178B also recommends PACTs associated with the software verification and quality assurance processes, as well as guidance for the purposes and contents of software lifecycle data such as software development and software verification plans. Finally, it discusses tool qualification, use of previously used software in safety critical applications, and alternative methods including the application of formal methods and exhaustive input testing, which could be borrowed and included as PACTs in ARRT.

Similar standards, utilized by the European Community and Great Britain in the establishment of the safety case for air traffic control systems are IEC 61508, *Information Technology - Software Process Assessment,* and IEC 61508, *Functional Safety of electrical/electronic/programmable electronic safety-related systems.*

5 CONSOLIDATED RECOMMENDATIONS

SETA recommends that the JPL researchers:

1. Add IV&V-specific activities to the PACT list, as the current list does not include specific IV&V risk-mitigation activities. This addition will render the tool compliant with the requirement to create IV&V plans.
2. Divide the PACTs into SBP and SP -- IV&V activities might also be so grouped, with some considered SBPs and other more detailed activities falling within the class of SP.
    a. The SBPs would constitute the "default" practices NASA should expect, with the SPs providing domain or project-specific targeted risk reduction.
    b. PACT tailoring could then be performed along the lines of what the SEI is proposing for CMMI.
3. Develop an expert consensus for the relative rankings of the PACTs via the Delphi technique and assign effectivity values developed using a decision science method such as the AHP. This research will render the output of ARRT defensible to potential Center customers.
4. Develop a strategy for optimally balancing IV&V and software development PACTs using validated PACT effectivities and AskPete cost data. Such research would add significant value to the ARRT tool by making its recommendations more analytical and repeatable.
5. Consider adopting selected safety and mission critical PACTs that are endorsed by related industries such as nuclear power, civil aviation, and DoD.

Appendix A ARRT Failure Modes Mapped to SEI Risk Elements

ARRT FMs	SEI REs
1: SEI SRE	Appendix 2 of SEI CMU/SEI-93-TR-6
2: Product Engineering	A. Product Engineering
3: Requirements Risks	1. Requirements
4: Stability: Unstable requirements	a. Stability
5: Completeness: Incomplete requirements	b. Completeness
6: Clarity: Unclear requirements	c. Clarity
7: Validity: Invalid requirements	d. Validity
8: Feasibility: Infeasible requirements	e. Feasibility
9: Precedent: Unprecedented requirements	f. Precedent
10: Scale: Large size or high complexity system	g. Scale
11: Design Risks	2. Design
12: Functionality: Potential problems in meeting functional specs	a. Functionality
13: Difficulty: Difficult design to achieve	b. Difficulty
14: Interfaces: ill-defined or uncontrolled internal interfaces	c. Interfaces
15: Performance: Stringent response time or throughput requirements	d. Performance
16: Testability: Product difficult to test	e. Testability
17: Hardware Constraints: Tight constraints because of target hardware	f. Hardware Constraints
18: Non-Developmental Software: Problems with software used here, but developed elsewhere	g. Non-Developmental Software
19: Code and Unit Test Risks	3. Code and Unit Test
20: Feasibility: Implementation of design difficult	a. Feasibility
21: Unit Test: Level and time for unit test inadequate	b. Testing
22: Coding/Implementation: Problems for coding (difficult requirements, poor design, etc.)	c. Coding/Implementation
23: Integration and Test Risks	4. Integration and Test
24: Environment: Inadequate test and integration environment	a. Environment
25: Product: Integration of software components to each other and to hardware, and testing of the product.	b. Product
26: System: Problems with integration of product to interfacing systems or sites	c. System
27: Engineering Specialties Risks	5. Engineering Specialties
28: Maintainability: Implementation difficult to understand or maintain	a. Maintainability
29: Reliability: Problems with system reliability or availability	b. Reliability
30: Safety: Infeasible safety requirements	c. Safety
31: Security: Security requirements more stringent than state-of-practice or experience	d. Security
32: Human Factors: Poor human interface specification	e. Human Factors
33: Specifications: Feasibility of implementation and quality attributes of stability, completeness, clarity, and verifiability.	f. Specifications
34: Development Environment	B. Development Environment

ARRT FMs	SEI REs
35: Development Process Risks	1. Development Process
36: Formality: Problems with the formality of the development process	a. Formality
37: Suitability: the adequacy with which the development model, process, methods, and tools support required activities.	b. Suitability
38: Process Control: Problems in ensuring process is used, or with measurement or improvement of the process.	c. Process Control
39: Familiarity: Project members are inexperienced in process	d. Familiarity
40: Product Control: Problems with traceability of requirements from specifications to implementation.	e. Product Control
41: Development System Risks	2. Development System
42: Capacity: Insufficient work station processing power, memory, etc.	a. Capacity
43: Suitability: Development system does not support all phases, activities, functions	b. Suitability
44: Usability: Development system is not easy to use	c. Usability
45: Familiarity: Unfamiliarity with development system	d. Familiarity
46: Unreliable development system	e. Reliability
47: System Support: Problems with training, access to expert users, or repair by vendors	f. System Support
48: Deliverability: Lack of resources to deliver system.	g. Deliverability
49: Management Process Risks	3. Management Process
50: Planning: inadequate planning and agreement to plan	a. Planning
51: Project Organization: Problems with the effectiveness of program organization, effectiveness of roles and responsibilities.	b. Project Organization
52: Management Experience: Inexperienced managers	c. Management Experience
53: Program Interface: Poor communication between managers and customer, senior management	d. Program Interfaces
54: Management Methods Risks	4. Management Methods
55: Monitoring: Poor project monitoring by management	a. Monitoring
56: Personnel Management: Problems with selection and training and work according to plan, get help, etc.	b. Personnel Management
57: Quality Assurance: Lack of quality assurance procedures and resources	c. Quality Assurance
58: Configuration Management: Poor configuration management	d. Configuration Management
59: Work Environment Risks	5. Work Environment
60: Quality Attitude: Lack of orientation toward quality workmanship	a. Quality Attitude
61: Cooperation: Lack of team spirit	b. Cooperation
62: Communication: Poor technical communication among team and management	c. Communication
63: Morale: Low morale on project	d. Morale
64: Program Constraints	C. Program Constraints
65: Resources Risks	1. Resources
66: Schedule: Inadequate or unstable schedule	a. Schedule
67: Staff: Staff inexperience's, lacking knowledge or skills	b. Staff

ARRT FMs	SEI REs
68: Budget: Insufficient or unstable budget	c. Budget
69: Facilities: Inadequate facilities for building and delivering product	d. Facilities
70: Contract Risks	2. Contract
71: Type of Contract: Problematic contract	a. Type of Contract
72: Restrictions: Restrictive contract	b. Restrictions
73: Dependencies: Program is dependent on outside products or services	c. Dependencies
74: Program Interfaces Risks	3. Program Interfaces
75: Customer: Problems with customer's level of skill or experience in the technical or application domain, not having access to customer factions, etc.	a. Customer
76: Associate Contractors: Problems with associate contractors- conflicting political agendas, interface problems, lack of cooperation.	b. Associate Contractors
77: Subcontractors: Inadequate task definitions, subcontractor management mechanisms, lack of knowledge of the program or corporation, etc.	c. Subcontractors
78: Prime Contractors: Poorly defined task definitions, complex reporting, or dependencies on technical or programmatic information	d. Prime Contractors
79: Corporate Management: Poor communication and direction from senior management; non-optimum levels of support.	e. Corporate Management
80: Vendors: Unresponsive vendors- dependencies on deliveries and support for critical system components.	f. Vendors
81: Politics: Political problems for project	g. Politics

Appendix B. Example of IV&V Activities Mapped to Selected ARRT FM Areas

	Classic IV&V Activities																								
Selected FM Areas	Architecture Phase Review (APR)	Code Analysis	Critical Design Review (CDR)	Configuration Management	Change Assessment	Construction Phase Review (CPR)	Criticality Assessment	Document Inspection	Design Analysis	Functional Configuration Audit (FCA)	Interface Verification	On-Site Audit	Physical Configuration Audit (First Article PCA)	Preliminary Design Review (PDR)	Quality Assurance	Requirements Analysis	Requirements Tracing	System Design Review (SDR)	Special Analysis	System Requirements Review (SRR)	Test Readiness Review (TRR)	Tool Suitability Assessment	Test Execution	Test Monitoring	Test Planning
Requirements Risks	X				X		X	X			X				X	X	X		X	X		X			
Design Risks			X		X		X	X	X					X	X	X	X	X	X			X			X
Code and Unit Test Risks		X		X	X			X				X			X		X		X			X			
Integration and Test Risks				X	X			X		X		X	X		X				X		X	X	X	X	X

Appendix C. ARRT PACTs Classified into Two Groups: Standard and Specialized

PACTs		Standard Good Practices	Special Practices
1:	Requirements		
2:	Authorization to proceed	X	
3:	Identify design/coding standards	X	
4:	Maintain Software Development Folder	X	
5:	Software Assurance reviews Management Plan	X	
6:	Implement Problem report and corrective action system	X	
7:	Management Plan approval	X	
8:	Documented requirements	X	
9:	Peer review of requirements	X	
10:	Conduct formal inspection of requirements	X	
11:	Software Assurance reviews requirements	X	
12:	Requirements approval	X	
13:	Peer review of plans	X	
14:	Implement Formal configuration management	X	
15:	Conduct Product Assurance Audits	X	
16:	Conduct Formal Reviews	X	
17:	Document approval of requirements and formal review	X	
18:	Customer approval of certification procedures		X
19:	Conduct analyses of criticality and safety		X
20:	Plan and schedule IV&V activities		X
21:	Identify method for verification of safety critical functions and requirements		X
22:	Design		
23:	Document preliminary and detailed designs	X	
24:	Peer or management reviews of design meets requirements	X	
25:	Conduct peer reviews on design	X	
26:	Conduct formal inspections of design products	X	
27:	Conduct formal design review(s)	X	
28:	Record and maintain peer and formal review results	X	
29:	Document design changes	X	
30:	Approval of design changes	X	
31:	Baseline design	X	
32:	Place design under CM control after review changes incorporated	X	
33:	Implement formal change control	X	
34:	Prepare verification/validation		X
35:	Perform safety analyses		X
36:	Create verification procedures for safety critical functions and requirements		X
37:	Development		
38:	Record results of peer reviews and formal reviews	X	
39:	Conduct code walkthroughs, peer reviews or code inspections	X	
40:	Conduct formal inspections of code	X	

PACTs		Standard Good Practices	Special Practices
41:	Implement formal software configuration management	X	
42:	Periodically perform backups	X	
43:	Approval of documented design	X	
44:	Document lower level test procedures	X	
45:	Document test plans and procedures	X	
46:	Document verification/validation results	X	
47:	Document and maintain test results	X	
48:	Capture and document final unit tests	X	
49:	Product assurance witnesses testing of safety critical and security functions		X
50:	Document design changes	X	
51:	Track change requests, problem reports and corrective action reports	X	
52:	Document approval of version and build of software for release to system test	X	
53:	Approval for product release	X	
54:	System Testing		
55:	Conduct system level integration	X	
56:	Conduct testing to test procedure and record results	X	
57:	Software assurance approval of all tests	X	
58:	Baseline software and related documentation after passing tests	X	
59:	Place test suites, simulators and test results under formal configuration control	X	
60:	Document problems, changes and corrective actions found during acceptance testing	X	
61:	Track problems, changes and corrective actions to closure	X	
62:	Implement documented problem report and corrective action system for baselined software	X	
63:	Acceptance obtained from customer		?
64:	Acceptance/Release		
65:	Project lead verifies all requirements are met or waiver approved	X	
66:	Validate project via customer witness final demonstration	X	
67:	Validate product via system level test or product demonstration	X	
68:	Safety Critical software approval by safety board/safety manager		X
69:	Final hazard reports reviewed and approved		X
70:	Product Assurance conducts FCA/PCA	X	
71:	Acceptance review prior to release	X	
72:	Acceptance approval documented	X	
73:	Written customer approval of demonstration for release authorization.		X
74:	Copies of all software products and documentation delivered to customer	X	
75:	Copy of released code and any relevant documentation, plans, reports, papers, test cases provided to customer	X	
76:	Software executables and necessary documents made available from configuration management system	X	
77:	Formal notice of release		X
78:	Identified, labeled copies of software deliverables maintained in agreed location	X	
79:	Copy of software kept by developers	X	

PACTs	Standard Good Practices	Special Practices
80: User maintains software after release	X	
81: Record design changes	X	
82: Record approved implementation changes	X	
83: Reverify and revalidate changes	X	
84: Commercial release IAW NPG 2210		X
85: Support		
86: Changes documented and tracked using agreed configuration management process	X	
87: Changes formally controlled and approved prior to implementation	X	
88: Problem report and corrective action system continues through working life of product or until turned over to another organization.	X	
89: Pull, label and archive all current releases of software and documentation once working life of product is complete	X	
90: Independent Verification and Validation		X
91: Perform IV&V reviews, analyses and tests		X
92: Required Documents		
93: Management Plan	X	
94: Development Activities Plan	X	
95: Verification Plan	X	
96: Validation Plan	X	
97: Organizational and Technical Interface Descriptions	X	
98: Requirements Documentation	X	
99: Design Documentation	X	
100: Testing Procedures	X	
101: Assurance Plan	X	
102: Risk Management Plan	X	
103: Configuration Management Plan	X	
104: Version Description	X	
105: Certification Procedures		X
106: Training Development Plan	X	
107: Delivery and Operational Transition Plan		X
108: Concept Documentation		X
109: Safety Assurance Procedures		X
110: Security and Privacy Procedures		X
111: Acquisition Activities Plan		X
112: Users Guide	X	
113: Operational Procedures Manual		X

Appendix D. Examples of Additional Specialized Software Development PACTs Mapped to Current ARRT Failure Modes

Potential Specialized SWD PACTs	PACT Description	Mapping to ARRT Failure Modes
S1	Use evolutionary/incremental development model	4 [Unstable Requirements] 5 [Incomplete requirements] 9 [Unprecedented requirements] (See 0 for additional definitions)
S2	Baseline stable requirements and limit work to those	4-9, 31-32
S3	Prototype/simulate requirements and get user/client feedback	6, 9, 13, 15, 17, 20, 31, 32
S4	Perform independent JAD or paper exercise with users or client to validate requirements	7-9, 30
S5	Identify high-complexity items early and prototype	10, 12, 13, 15, 17, 20
S6	Perform additional V&V on complex items	10, 12, 13, 15, 17,
S7	Establish an internal interface working group	14, 76, 77
S8	Define testability requirements as an integral part of top-level requirements specification	16
S9	Establish test working group at concept stage	16
S10	Develop generic wrapper to interface with NDS such that a different COTS product can be substituted	18, 73
S11	Identify functional and performance requirements for capabilities to be provided by NDS and qualify the NDS against this early lifecycle	18, 73
S12	Extend schedule to allow for unit testing	21
S13	Improve design	22
S14	Acquire additional equipment	24
S15	Acquire additional personnel or borrow software engineers	24
S16	Begin interface testing as early as practical, using simulations if necessary	26, 73
S17	Identify maintainability issues early in lifecycle and prototype if possible	28
S18	Perform additional V&V on maintenance items of concern	28
S19	Identify reliability issues early in lifecycle and prototype if possible	29
S20	Perform additional V&V on reliability items of concern	29
S21	Initiate independent process reviews at outset	36, 38, 39
S22	Review appropriate plans, e.g., Management Plan and Software Development Plan	37
S22A	Perform tool suitability assessment	37
S23	Perform process training	39, 56, 60, 67
S24	Perform independent traceability verification	40
S25	Perform analysis to define needed development environment	42-44, 69
S26	Acquire the right development environment	42-44, 69
S27	Get training on development system from outset	45

Potential Specialized SWD PACTs	PACT Description	Mapping to ARRT Failure Modes
S28	Begin using DE as early as possible	46
S29	Make vendor deliver reliable product (warranty)	46, 47, 80
S30	Develop a fallback plan	46, 47, 50, 80
S31	Perform an organizational review and implement recommendations	51
S32	Utilize consultants and/or experienced managers to augment	52, 56
S33	Train managers	52, 54-56
S34	Add more Technical Interchange Meetings and informal staff meetings	53-55, 62, 76, 77
S35	Establish a User Working Group	53, 62
S36	Management by Example (MBE); instill quality spirit	60, 61, 63
S37	Conduct team building workshop	60, 61, 62, 63, 76, 77
S38	If early in SDLC, develop a staff training plan and implement	67, 53
S39	If late in SDLC, augment with experienced staff/mentors	67

6 ACRONYMS

Acronym	Expansion
AHP	Analytical Hierarchy Process
ARRT	Advanced Risk Reduction Tool
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integrated
CO	Contracting Officer
COTR	Contracting Officer's Technical Representative
CTO	Contract Task Order
DoD	Department of Defense
FM	Failure Mode
I&C	Instrumentation and Control
ISO	International Organization for Standardization
IT	Information Technologies
IV&V	Independent Verification and Validation
IWG	Interface Working Group
JPL	Jet Propulsion Laboratory
KPA	Key Practice Area
NASA	National Aeronautics and Space Administration
PACTs	Preventive measures Analyses Controls Tests
PM	Program Manager
POC	Points of Contact
QA	Quality Assurance
RE	Risk Element
RTCA	Requirements and Technical Concepts for Aviation
SAIC	Science Applications International Corporation
SBP	Standard Best Practices
SDP	Software Development Plan
SEI	Software Engineering Institute
SETA	Science and Engineering Technical Assessments
SLP	System Level Procedure
SOR	Statement of Requirements
SOW	Statement of Work
SP	Special Practices

Acronym	Expansion
SPMN	Software Program Managers Network
VA	Virginia
WSRB	Weapons Safety Review Board
WV	West Virginia

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 13, 2000	Final

4. TITLE AND SUBTITLE
Task Order 15 Advanced Risk Reduction Tool (ARRT) Special Case Study Report for the Science and Engineering Technical Assessments (SETA) Program

5. FUNDING NUMBERS
NAS2 -98028

6. AUTHORS
D.N. American, Inc and SAIC

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
D.N. American, Inc
Suite 3220, 1000 Technology Drive
Fairmont WV 26554

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
NASA Ames Research Center
Moffett Field CA

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
See NPG 2200.2A

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document presents the Science and Engineering Technical Assessments (SETA) team's findings and recommendations from the special case study of the ARRT Tool Developed by the Jet Propulsion Laboratory under the Office of Safety and Mission Assurance (OSMA) Software Assurance Research Program. This is the D.N. American response to *Contract Task Order #15 – ARRT Special Case Study* for the Science and Engineering Technical Assessments (SETA) contract.

14. SUBJECT TERMS
SETA, CTO 15, Special Case Study, Advanced Risk reduction Tool (ARRT)

15. NUMBER OF PAGES
24

16. PRICE CODE

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

NSN 7540-01-280-5500 Computer Generated STANDARD FORM 298 (Rev 2-89)
Prescribed by ANSI Std 239-18
298-102