AMENDMENT OF SOLICITATION/MODIFICATION OF CONTRACT

BPA NO.	1. CONTRACT ID CODE	PAGE	OF PAGES
		1	2

2. AMENDMENT/MODIFICATION NO.	3. EFFECTIVE DATE	4. REQUISITION/PURCHASE REQ. NO.	5 PROJECT NO.(If applicable)
002	SEP 17 2007	RES-06-050	

6. ISSUED BY	CODE	7. ADMINISTERED BY (If other than Item 6)	CODE
U.S. Nuclear Regulatory Commission Div. of Contracts Attn: Adelis M Rodriguez, 301-415-5719 Mail Stop T-7-I-2 Washington, DC 20555	3100	U.S. Nuclear Regulatory Commission Div. of Contracts Mail Stop T-7-1-2 Washington, DC 20555	3100

8. NAME AND ADDRESS OF CONTRACTOR (No., street, county, State and ZIP Code)	(X)	9A. AMENDMENT OF SOLICITATION NO.
APPLIED PROGRAMMING TECHNOLOGY INC 240 MARKET ST STE 208 BLOOMSBURG PA 178151716		9B. DATED (SEE ITEM 11) 10A. MODIFICATION OF CONTRACT/ORDER NO. NRC-04-06-050
CODE 092982854 FACILITY CODE	X	10B. DATED (SEE ITEM 13) 08-08-2006

11. THIS ITEM ONLY APPLIES TO AMENDMENTS OF SOLICITATIONS

☐ The above numbered solicitation is amended as set forth in Item 14. The hour and date specified for receipt of Offers ☐ is extended, ☐ is not extended.
Offers must acknowledge receipt of this amendment prior to the hour and date specified in the solicitation or as amended, by one of the following methods:
(a) By completing Items 8 and 15, and returning _____ copies of the amendment; (b) By acknowledging receipt of this amendment on each copy of the offer submitted; or (c) By separate letter or telegram which includes a reference to the solicitation and amendment numbers. FAILURE OF YOUR ACKNOWLEDGMENT TO BE RECEIVED AT THE PLACE DESIGNATED FOR THE RECEIPT OF OFFERS PRIOR TO THE HOUR AND DATE SPECIFIED MAY RESULT IN REJECTION OF YOUR OFFER. If by virtue of this amendment you desire to change an offer already submitted, such change may be made by telegram or letter, provided each telegram or letter makes reference to the solicitation and this amendment, and is received prior to the opening hour and date specified.

12. ACCOUNTING AND APPROPRIATION DATA (If required)
B&R: 76015111205 Job: Y6851 BOC: 252A Approp: 31X0200.760
Obligation: $114,533.63 DUNS: 092982854 FFS: RES-C07-266

13. THIS ITEM APPLIES ONLY TO MODIFICATIONS OF CONTRACTS/ORDERS, IT MODIFIES THE CONTRACT/ORDER NO. AS DESCRIBED IN ITEM 14.

(X)	
	A. THIS CHANGE ORDER IS ISSUED PURSUANT TO: (Specify authority) THE CHANGES SET FORTH IN ITEM 14 ARE MADE IN THE CONTRACT ORDER NO. IN ITEM 10A.
	B. THE ABOVE NUMBERED CONTRACT/ORDER IS MODIFIED TO REFLECT THE ADMINISTRATIVE CHANGES (such as changes in paying office, appropriation date, etc.) SET FORTH IN ITEM 14, PURSUANT TO THE AUTHORITY OF FAR 43.103(b).
X	C. THIS SUPPLEMENTAL AGREEMENT IS ENTERED INTO PURSUANT TO AUTHORITY OF: FAR 52.243-2 Changes - Cost Reimbursement
	D. OTHER (Specify type of modification and authority)

E. IMPORTANT: Contractor ☐ is not, ☒ is required to sign this document and return 2 copies to the issuing office.

14. DESCRIPTION OF AMENDMENT/MODIFICATION (Organized by UCF section headings, including solicitation/contract subject matter where feasible.)

The purpose of this modification is to perform a within scope modification to the statement of work, increase the contract ceiling and increase the obligated amount.

See attached page 2 for description of modification.

Period of Performance: 8/8/2006 to 8/7/2008. (unchanged)
Obligated amount: $714,533.63(changed)
Total Contract Ceiling: $836,255.70 (changed)

Except as provided herein, all terms and conditions of the document referenced in Item 9A or 10A, as heretofore changed, remains unchanged and in full force and effect.

15A. NAME AND TITLE OF SIGNER (Type or print)	16A. NAME AND TITLE OF CONTRACTING OFFICER (Type or print)
Kenneth R Jones President	Stephen Pool Contracting Officer
15B. CONTRACTOR/OFFEROR (Signature of person authorized to sign)	15C. DATE SIGNED 9/17/07
16B. UNITED STATES OF AMERICA BY (Signature of Contracting Officer)	16C. DATE SIGNED 9/13/07

NSN 7540-01-152-8070
PREVIOUS EDITION NOT USABLE
STANDARD FORM 30 (REV. 10-83)
Prescribed by GSA - FAR (48 CFR) 53.243

TEMPLATE - ADM001
SUNSI REVIEW COMPLETE
SEP 26 2007
ADM002

The purpose of this modification is to:

(1) Modify the Statement of Work, thereby increasing the total estimated (ceiling) amount of the contract by $114,533.63 from $721,722.07 to $836,255.70.

(2) Provide funding in the amount of $114,533.63, thereby increasing the obligated amount under the contract from $600,000.00 to $714,533.63.

(3) Increase the option price for option year 2 by $92,537.36 from $374,411.94 to $466,979.26.

Accordingly, the following changes are hereby made to the contract:

1. Section B.3 CONSIDERATION AND OBLIGATION–COST PLUS FIXED FEE (JUN 1988) ALTERNATE I (JUN 1991), paragraphs (a), (c), (d) and (f) are revised to read as follows:

(a) The total estimated cost to the Government for full performance of this contract is $836,255.70, of which the sum of $760,232.45 represents the estimated reimbursable costs, and of which $76,023.25 represents the fixed fee.

(c) The amount currently obligated by the Government with respect to this contract is $714,533.63, of which the sum of $649,576.03 represents the estimated reimbursable costs, and of which $64,957.60 represents the fixed fee.

(d) It is estimated that the amount currently allotted will cover performance through May 8, 2008.

(f) (OPTION YEAR 2) The total estimated cost to the government for full performance of Option Year 2 is $466,979.26, of which the sum of $424,526.60 represent the estimated reimbursable costs, and of which $42,452.66 represents the fixed fee.

2. Section C. –DESCRIPTION/SPECIFICATION/STATEMENT OF WORK is deleted in its entirety and replaced with the attached revised statement of work.

All other terms and conditions remain the same.

A summary of obligations, from the award date through the date of this action, is given below:

Total FY 06 obligation amount:	$300,000.00
Total FY 07 obligation amount:	$414,533.63
Cumulative Total of NRC Obligations	$714,533.63

This modification obligates FY07 funds in the amount of $114,533.63.

SECTION C - DESCRIPTION/SPECIFICATIONS/STATEMENT OF WORK

OFFICE OF NUCLEAR REGULATORY RESEARCH
RFPA RES-05-052
DRAFT STATEMENT OF WORK

TITLE: SNAP Maintenance and Development

A. BACKGROUND

The Symbolic Nuclear Analysis Package (SNAP) is a graphical user interface system designed to support U.S. Nuclear Regulatory Commission (NRC) nuclear analysis codes. SNAP includes pre- and post-processor components allowing code specific models to be built, exported, submitted to the analysis code, then receiving the output from the analysis code and displaying the results in a user-friendly fashion. SNAP is written primarily in Java, although some aspects of the post-processor component are written in C. SNAP is designed to be object-oriented and extensible. Currently, SNAP supports the following NRC-developed analytical codes: TRAC/RELAP (Reactor Leak And Power safety excursion code) Advanced Computational Engine (TRACE), RELAP, Version 5 (RELAP5), CONTAIN, MELCOR, FRAPCON and FRAPTRAN. TRACE and RELAP5 are system-level thermal hydraulic codes. MELCOR is a system-level severe accident code. CONTAIN is a code used to predict containment response and associated phenomena. FRAPCON and FRAPTRAN are fuel phenomena codes.

The following component applications make up SNAP:

1. Model Editor

The purpose of this application is to allow the analyst to create an input model (deck) using graphical tools. The model editor currently also has the ability to read and write RELAP5 American Standard Code for Information Interchange (ASCII) decks and read and write TRACE (Thermal-hydraulic Portable Restart and ASCII formats) decks. A deck that is read in (imported) will be parsed into model editor component objects and displayed on the model editor's canvas. The analyst may then edit the model using the model editor graphical tools and export it for use later.

Through the "animation model" plug-in, the model editor is also the primary application for viewing the results of an analytical code run. Using this application, the analyst may prepare a graphical "mask" that can then be used to display the results of the analytical code in a user friendly fashion. Both 2-D and 3-D masks are supported.

2. Execution Monitor

The Execution Monitor is an application that manages the running of the analytical codes. This is a "wrapper" for the analytical codes that permits the other SNAP applications to access and interact with the analytical codes as they are running. "Runtime" plug-ins (JAR files) are used to encapsulate this functionality for each code.

3. Configuration Tool

This tool is used to configure global options for the SNAP client applications. The Calculation and Database servers can also be configured and launched using this tool.

This tool contains a separate panel for Global Options, the Calculation Server, and the Database Server. The Calculation Servers configuration allows the user to set up any number of analytical codes running locally or on remote servers. Database server's configuration allows the user to specify connectivity to databases of results or experimental data.

4. Job Status Tool

This tool is used to allow the user to see the current status of running analytical codes. Limited control over the running codes is also provided by this application.

5. Analysis Code Graphing, Advanced Computation and Exploration of data (AcGrace)

Currently, this is the only application in SNAP that is not written in Java. AcGrace is actually an open source application that was modified to read and plot data for several NRC-sponsored analytical codes. AcGrace reads the files produced by the analytical codes and produces plots that can be saved as Portable Document Format (PDF), Scalable Vector Graphics (SVG), Maker Interchange Format (MIF), Postscript and several other formats suitable for publication applications.

6. SNAP Database Server

The SNAP Database Server provides a platform independent persistent store for models and system configuration information. The SNAP Database Server provides a standardized interface to an embedded relational database management system (RDBMS). The server utilizes Derby, an open source pure-Java Structured Query Language (SQL) database that can be embedded into an application.

7. Admin Tool

The SNAP system administrator manages the system configuration information contained in the database using an administrative tool. The tool provides the ability to create users, manage projects, assign privileges and manage system tables.

8. NRC Databank Server

The NRC Databank Server is similar in function to the Execution Monitor in that it provides a source of data to the post-processor with the difference being that it works with experimental data as opposed to calculations. The post-processor applications connect to the server to obtain the list of all experimental facilities, the list of experiments available for a given facility, and the list of data channels available for a given experiment. Multiplexed or de-multiplexed data sets can then be retrieved for a selected set of data channels and time ranges.

9. Calculation Archive Server

Calculations run on an Execution Monitor may be archived to the SNAP Archive server for long-term retention using the Job Status Tool. The post-processor applications may connect to this server to access archived data sets.

B. OBJECTIVES

1. To maintain and modify the SNAP code in a manner consistent with the current code design and general coding practices for SNAP (specified below).

2. To maintain compatibility with the current versions of the SNAP-supported analytical codes: RELAP5, TRACE, CONTAIN, MELCOR, FRAPCON and FRAPTRAN.

3. To maintain and improve the SNAP code documentation and help system to ensure that it is current with the current code capabilities.

C. TECHNICAL AND OTHER SPECIAL QUALIFICATIONS REQUIRED

The performance of this work scope requires very specialized knowledge on the SNAP code including its code architecture, programming details, and/or its usage. Personnel working on this project shall have an engineering degree or equivalent and shall have experience on programming and usage of the SNAP code (or a code similar to SNAP in terms of being Java-based and having an extensible user interface). Also, personnel working on this project shall demonstrate a complete understanding of (and preferably experience with) a "pluggable" code design and have experience with at least one other (not SNAP) large Java-based application.

D. SCOPE OF WORK

General Requirements for SNAP Code Development

1. Code quality assurance procedures outlined in "Software Quality Assurance Procedures for NRC Thermal Hydraulic Codes, NUREG-1737," will be used as a guide (available online through the ADAMS system: ML010170081 at http://www.nrc.gov/). "Software Requirements Specifications (SRS)" and "Software Design and Implementation Documents (SDID)," shall be prepared. Project Officer (PO) approval of these documents must be granted to the contractor before code work (for each code development task) is initiated. Testing shall be performed as approved in the SRS.

2. SNAP is to be maintained in the current version of the Java language.

3. SNAP support of user codes must be implemented using a "plug-in" design. By plug-in," it is meant that all Java codes specific to supporting any one analytical code is to be placed in its own "JAR" file and loaded into SNAP at runtime. Plug-in support is implemented separately for the post-processor and the pre-processor. This requirement is important to maintain SNAP's extensibility.

4. SNAP plug-ins must conform not only to the application interface (API) developed at Applied Programming Technologies, Inc. (APT), but also conform to the design patterns

used by APT. Documentation for the SNAP API will be made available upon written request to the Contracting Officer. The source code for SNAP will NOT be released.

RESEARCH QUALITY

The quality of NRC research programs are assessed each year by the Advisory Committee on Reactor Safeguards. Within the context of their reviews of RES programs, the definition of quality research is based upon several major characteristics:

- Results meet the objectives (75% of overall score)
  - Justification of major assumptions (12%)
  - Soundness of technical approach and results (52%)
  - Uncertainties and sensitivities addressed (11%)
- Documentation of research results and methods is adequate (25% of overall score)
  - Clarity of presentation (16%)
  - Identification of major assumptions (9%)

It is the responsibility of the contractor to ensure that these quality criteria are adequately addressed throughout the course of the research that is performed. The NRC project manager and technical monitor will review all research products with these criteria in mind.

ORGANIZATIONAL CONFLICT OF INTEREST

The servicing agency recognizes that Section 170A of the Atomic Energy Act of 1954, as amended, requires that NRC be provided with disclosures on potential conflicts when NRC obtains technical, consulting, research and other support services. The servicing agency further recognizes that the assignment of NRC work to the servicing agency must satisfy NRC's conflicts standards. Accordingly, when NRC enters into an agreement with the servicing agency to perform work for NRC, and during the life of the agreement, the servicing agency shall review its current work, planned work and where appropriate, past work for the servicing agency and others (meaning organizations, in the same/similar technical area as the NRC project scope of work, e.g., (included but not limited to), NRC licensees, vendors, industry groups or research institutes that represent or are substantially comprised of nuclear utilities) to determine whether such work is in the same or similar area as the proposed NRC project. Should that review reveal current or planned work for the servicing agency or others in the same or similar technical area as the proposed NRC work, the servicing agency shall provide name of organization, dollar value, and period of performance of the work identified as well as descriptions of such potentially conflicting present/planned/past work to NRC. NRC shall then determine whether a conflict would result and, if one does, determine, after consultation with the servicing agency, the appropriate action NRC or the servicing agency should take to avoid the conflict or when appropriate under NRC procedures, waive the conflict.

PERFORMANCE REQUIREMENTS

Task 1. Application testing, Interface Improvements, and Code Distribution.

1. SNAP plug-ins for analytical codes shall be tested to ensure that they can import and export the respective code input decks without causing changes in the calculated results. For each analytical code for which SNAP provides a plug-in,

the contractor shall maintain and exercise a relevant suite of input decks to assure that the SNAP plug-in is performing input and output operations correctly. The contractor shall at least include in these test suites any decks that are provided by the PO. The contractor shall correct errors discovered in the associated SNAP plug-in during testing as necessary.

2. Maintain a multi-platform, SNAP installation package to support code distribution. SNAP is to be distributed from a website under strict control of the contractor. Access to the SNAP code is to be coordinated through the PO.

3. Evaluate and correct errors reported by the SNAP user community.

4. Maintain and extend SNAP User's manuals and help files to reflect any code changes. Automate linkage to analysis code documentation.

5. Provide User Support and Model Editor User Interface Improvements. Implement user-supplied requests for improvements as reviewed and approved by the PO. User interface issues have been identified by "beta testers" and NRC staff using the TRACE, RELAP5 and CONTAIN plug-ins. Some of the reported issues have been addressed some have not. It is anticipated that many more user interface changes and improvements will be required after SNAP version 1.0 is released (expected in early 2006) and more people begin using and providing user suggestions. Also, the NRC is trying to establish a contract to provide usability feedback before SNAP version 1.0 is released. The contractor shall collect the suggestions for user interface changes and document them in the monthly progress reports. The contractor shall propose solutions as to how to address the user suggestions in SNAP and present them to the PO in a written report. After the PO sends either a written letter or an email to the contractor indicating approval, the contractor shall implement the proposed solution(s).

6. Develop and provide user training for USNRC staff. Training shall be provided to NRC staff in small classes (approximately 10 staff at a time) and consist of hands on tutorials, presentation of features and methods of use of the SNAP tools. Feedback from the tutorials shall used to focus training course developments. More then one course "type" may be specified, that is, there may be a desire to have advanced courses, or courses focusing on specific codes. No more than five courses per contract year are planned.

7. Develop NUREG ready manuals for user manuals for SNAP core, TRACE, PARCS, and RELAP5. The user manuals shall be written in a way that the subject matter will be identical to the on-line help provided by the SNAP application.

8. Provide support to NRC staff in their efforts to develop in-house expertise in plug-in development for SNAP. Specifically, the USNRC staff is planning to write a SNAP plug-in for the RADTRAD code. Support for this effort shall include providing assistance in plug-in and user interface design as well as providing recommendations regarding development.

9. Develop a K Factor to FRIC conversion tool. This tool shall allow the analyst to convert between K Factors and FRICs and also show the relevant intermediate information used in this conversion so that the analyst may easily verify the calculated results.

10. Develop a new elevation loop check system for the TRACE Plug-in. This system shall provide a detailed list of the components within hydraulic loops, along with the intermediate information used to calculate the component elevations. This information will assist the user in resolving loop check errors, and will also allow the loop check system to be easily verified. The elevation information for each loop shall be exportable to a Comma Separated Value (CSV) format file, which can be imported into a spreadsheet for further analysis.

11. Develop a Plug-in Reference Documentation Management tool. The plug-in documentation links are currently hardwired into each plug-in. In addition, they are limited to the Code User's Manual. This tool shall manage an external file that will be loaded at runtime to create reference links to the code manuals for a selected plug-in. This shall allow multiple reference documents (i.e. User's Manual and Theory Manual) to be available for a given component type and shall allow the reference links to be updated outside of the plug-in.

Base year:
Completion date: Two years after award of contract

Task 2. Maintain SNAP Capabilities

The supported analytical codes (MELCOR, CONTAIN, TRACE, FRAPCON/FRAPTRAN and RELAP5) are still undergoing modifications that will necessitate changes to their input format requirements. The contractor shall upgrade SNAP to fully support these modifications. The contractor shall maintain contact with the analytical code developers so that any changes can be identified and modifications made to the respective SNAP plug-in in a timely fashion. The PO shall be notified, either by letter or email, of any planned changes to any plug-in before such changes are actually made. Note that in cases where the analytical code has been updated or changed in a major way, the contractor may need to write an entirely new plug-in for the new version of the analytical code. If the need for an entirely new plug-in is encountered, the contractor shall not proceed to make changes to the existing plug-in, but instead inform the PO so that other actions may be planned.

Base year:
Completion date: One year after award of contract

Task 3. Provide Technical Support

This task provides technical support to the NRC. The work includes making presentations, attending meetings, reviewing technical reports, and providing technical consultation and support as requested by the PO. Also, this task is to provide support for "beta testers." "Beta testers" are SNAP users who are issued pre-release versions of SNAP for the purpose of evaluating SNAP in typical user environments (i.e., "real world" use of SNAP). Currently, there are about 30 beta testers for SNAP; however, only 5-10 are active.

Base year:
Completion date: One year after award of contract

Task 4. Develop SNAP User Interface Test Suite

The SNAP developer tests each SNAP plug-in with a suite of test models. However, the user interface does not get "tested" in this process. Therefore, SNAP versions can sometimes contain errors that don't show up until a user finds them through usage. To "catch" these problems and improve the quality of the SNAP interface, the contractor shall develop a set methodology to test the various user level aspects of SNAP. Current technology indicates that Graphical User Interface (GUI) systems, such as SNAP, can be "made" to be testable by splitting the functional part of the GUI away from the visual portion so that automated testing can be performed in batch mode. Indeed, much of the current SNAP GUI is already done in this fashion. The effort under this task includes:

1. The contractor shall design a testing framework for the SNAP graphical user interface that will permit batch (non-visual) testing of the entire SNAP user interface.

2. Once the testing framework has been designed and reviewed for completeness and suitability and approved by the PO via an email or letter, the contractor shall implement the test framework in SNAP.

3. After the test framework has been implemented in SNAP, the contractor shall develop a full-coverage test suite for the SNAP GUI. "Full-coverage" here means that every visual portion of the GUI shall be tested.

The testing framework and test suite shall be placed in the SNAP code repository and become part of the standard "build" procedure for SNAP. The full test suite will be run (at a minimum) before every public release of SNAP. Standard SNAP development requirements apply to this task. Therefore, SRS and SDID documents are required. Also, a report describing the content and use of the test suite/framework shall be prepared by the contractor and delivered to the PO upon completion of the test suite implementation.

Base Year:
Completion date: 5 months after award of contract

Task 5. SNAP User Interface Testing

The contractor shall use the test suite implemented in Task 4 of this contract to ensure that SNAP code integrity is maintained. Furthermore, the contractor shall add to the test suite any tests that are deemed necessary during the ongoing development of SNAP and SNAP plug-ins. Status of the results of using the test suite shall be documented for each of the public releases in a report to the PO.

Base year:
Completion date: One year after award of contract

Task 6. Develop 3D Plotting Capability

3D surface plotting capabilities are needed for displaying kinetics information. The contractor shall identify the necessary parameters to plot through discussion with relevant analytical code users, code developers and the PO. Once the desired plot parameters are identified, the contractor shall develop a SRS and SDID describing the requirements and design for an application or plug-in, if determined more appropriate, the contractor can propose a modification to the existing animation plug-in. The product of this task is a fully functional user interface that provides a 3D plotting capability that meets the SRS stated goals. The implementation of the design shall be done in accordance with the general requirements for SNAP code development. The contractor shall test this plotting package on representative sets of data such as might be produced by the Purdue Advanced Reactor Core Simulator code and also with sets of data produced by the TRACE 3D vessel component. The data will be provided either by PO or other NRC contractors. The contractor shall correct any errors discovered during testing. The contractor shall provide a user's guide describing the use of this 3D plotting application.

Base year:
Completion date: 3 months after award of contract

Task 7. Develop Java Based "Lightweight" Plotting Tools

Integration of the Java applications in SNAP with the C-based application, AcGrace, continues to cause problems with the installation of SNAP. For instance, under MS Windows XP, AcGrace installation requires the installation and configuration of a substantial software package called "CYGWIN." The CYGWIN installation is many times the size of SNAP itself. Also, AcGrace users who are familiar with the Windows XP graphical interface are often confused by the different style dialog boxes and file requesters that it presents. Therefore, it is desirable to have a fast and relatively lightweight plotting interface for SNAP with a consistent user interface that can be installed through the same installation procedure as SNAP itself. Such a package shall provide for the ability to plot directly from the animation view displays (navigator and 2D view) and the ability to batch process to write printable plot files.

The contractor shall develop an advanced plotting tool that integrates the above stated requirements. The contractor shall also determine what other requirements such a plotting package should have based on the current usage of AcGrace with SNAP. Again, this plotting tool may be its own Java application, or may be integrated directly into the existing animation plug-in (JavaBean interface). The implementation shall be done in accordance with the general requirements for SNAP code development.

Base year:
Completion date: One year after award of contract

Task 8. Develop Advanced JavaBeans for the Animation Plug-in

The objects used to "build" an animation file mask are actually JavaBean components. In this work, the contractor shall design and add to the SNAP release package new

JavaBeans that can be used for displaying complex data that may be produced by analytic code runs. Currently, one such JavaBean that is envisioned is a "core melt" JavaBean. This JavaBean will be able to display an animation of a melting core including material heat up, fuel and structure relocation, combustion that may be occurring, etc. The contractor shall design and implement such a "core melt" JavaBean and up to ten other JavaBean components. The documentation requirement for these JavaBean components is reduced from the normal SNAP development. In this case, the contractor shall provide a description and design for the component all in one document. The document need not follow the requirements as stated for the other SNAP development documents. However, the contractor shall not proceed with development of any such JavaBean component until the PO or technical monitor provides an email or letter providing approval of the design.

The contractor shall develop a "Spreadsheet bean" that can be used to document as well as link specifiable fields into the model. The component should allow the user to define the size of the spreadsheet, enter input into the "cells", and do simple mathematical functions.

Base year:
Completion date: One year after award of contract

Task 9. Monitor Editor User Interface Improvements - DELETED

Task 10. Develop AVScript Style Analyst Support Interface

The NRC staff and several of its contractors who support the RELAP5 and TRACE development work use a tool called AVScript for automating code runs and plotting tasks. The AVScript tool is currently written in Perl and when used on the MS Windows platform requires a compatible version of Perl to be used there. To help alleviate operating system dependencies and to make the useful features of AVScript available to more analysts, a SNAP feature plug-in is proposed to provide these capabilities.

The contractor shall design and implement the feature plug-in following the general requirements for code development as specified previously in this document (i.e., prepare an SRS and SDID documents, obtain PO approval before implementation of design, etc.).

The contractor shall submit batch jobs to cluster machines. The cluster queuing software specifications will be sent to the contractor to better define the requirements for this work.

Base year:
Completion date: One year after award of contract

Task 11. Update CONTAIN Plug-in

The SNAP CONTAIN code plug-in was the first plug-in to use the Common Application Framework for Engineering Analysis (CAFEAN) plug-in interface. The CONTAIN plug-in would greatly benefit from modifications that have been made to other CAFEAN-based plug-ins since its release. User interface improvements that have been implemented for the TRACE code shall be reviewed and implemented where appropriate for CONTAIN.

The contractor shall document in SRS and SDID the requirements and any design changes needed for the CONTAIN plug-in code to be brought up to date with the TRACE plug-in code. The contractor shall obtain PO approval of the documents, either through an e-mail or letter, before proceeding to the implementation phase of the work.

Base year:
Completion date: 7 months after award of contract

Task 12. Restore Functionality of the RELAP5 to TRACE Vessel Conversion Tool

The SNAP RELAP5 plug-in has been, or will be, converted into a CAFEAN format; however, the existing RELAP5 to TRACE vessel conversion feature plug-in is still based upon the old RELAP5 plug-in and will therefore not function anymore. The contractor shall design and implement a new RELAP5 to TRACE vessel conversion feature plug-in based on the new CAFEAN format SNAP RELAP5 plug-in. All of the features of the current vessel conversion feature plug-in shall be reproduced in the new feature plug-in. All of the features of the new plug-in should be documented in SRS. As for all SNAP development work, the contractor shall document in SRS and SDID the requirements and design for this work. The contractor shall obtain PO approval of the documents, either through an e-mail or letter, before proceeding to the implementation phase of the work.

Base year:
Completion Date: 3 months after award of contract

Task 13. Develop "Animation Export" Capabilities

Users of the SNAP animation model plug-in have expressed a strong desire to be able to export animations produced by SNAP. The contractor shall add to the animation plug-in the ability to save the animations that it displays to a file that can be played back with third party software or with freely distributable software. As a minimum, playback in MS PowerPoint must be achieved. Some form of playback on the LINUX/UNIX platform must also be provided. As for all SNAP development work, the contractor shall document in SRS and SDID the requirements and design for this work and obtain PO approval of the documents, either by email or letter, before proceeding to the implementation phase of the work.

Base year:
Completion Date: 5 months after award of contract.

Task 14. Application Testing and Distribution

1. SNAP plug-ins for analytical codes shall be tested to ensure that they can import and export the respective code input decks without causing changes in the calculated results. For each analytical code for which SNAP provides a plug-in, the contractor shall maintain and exercise a relevant suite of input decks to assure that the SNAP plug-in is performing input and output operations correctly. The contractor shall at least include in these test suites any decks that are provided by the PO. The contractor shall correct errors discovered in the associated SNAP plug-in during testing as necessary.

2. Maintain a multi-platform, SNAP installation package to support code distribution. SNAP is to be distributed from a web site under strict control of the contractor. Access to the SNAP code is to be coordinated through the NRC project manager.

3. Evaluate and correct errors reported by the SNAP user community.

4. Maintain and extend SNAP user's manuals and help files to reflect any code changes. Automate linkage to analysis code documentation.

5. Provide user support. Implement user supplied requests for improvements as reviewed and approved by the PO.

Option year 1:
Completion date: One year after start of option year one

Task 15. Maintain SNAP Capabilities

The supported analytical codes (MELCOR, CONTAIN, TRACE, FRAPCON/FRAPTRAN and RELAP5) are still undergoing modifications that will necessitate changes to their input format requirements. The contractor shall upgrade SNAP to fully support these modifications. The contractor shall maintain contact with the analytical code developers so that any changes can be identified and modifications made to the respective SNAP plug-in in a timely fashion.

The PO shall be notified either by letter or email of any planned changes to any plug-in before such changes are actually made. Note that in cases where the analytical code has been updated or changed in a major way, the contractor may need to write an entirely new plug-in for the new version of the analytical code. If the need for an entirely new plug-in is encountered, the contractor shall not proceed to make changes to the existing plug-in, but instead inform the PO so that other actions may be planned.

Option Year 1:
Completion date: One year after start of option year one

Task 16. Provide Technical Support

This task provides technical support to the NRC. The work includes making presentations, attending meetings, reviewing technical reports, and providing technical consultation and support as requested by the PO. Also, this task is to provide support for "beta testers." "Beta testers" are SNAP users who are issued pre-release versions of SNAP for the purpose of evaluating SNAP in typical user environments (i.e., "real world" use of SNAP). Currently, there are about 30 beta testers for SNAP; however, only 5-10 are active.

Option year 1:
Completion date: One year after start of option year one

Task 17. SNAP User Interface Testing

The contractor shall use the test suite implemented in Task 4 of this contract to ensure that SNAP code integrity is maintained. Furthermore, the contractor shall add to the test suite any tests that are deemed necessary during the ongoing development of SNAP and SNAP plug-ins. Status of the results of using the test suite shall be documented for each of the public releases in a report to the PO.

Option year 1:
Completion date: One year after start of option year one

Task 18. Develop Java Lightweight Plotting Tools–DELETED

Task 19. Develop Advanced JavaBeans for the Animation Plug-in

The objects used to "build" an animation file mask are actually JavaBean components. In this work, the contractor shall design and add to the SNAP release package new JavaBeans that can be used for displaying complex data that may be produced by analytic code runs. Currently, one such JavaBean that is envisioned is a "core melt" JavaBean. This JavaBean will be able to display an animation of a melting core including material heat up, fuel and structure relocation, combustion that may be occurring, etc. The contractor shall design and implement such a "core melt" JavaBean and up to ten other JavaBean components. The documentation requirement for these JavaBean components is reduced from the normal SNAP development. In this case, the contractor shall provide a description and design for the component all in one document. The document need not follow the requirements as stated for the other SNAP development documents. However, the contractor shall not proceed with development of any such JavaBean component until the PO provides an email or letter providing approval of the design.

Option year 1:
Completion date: One year after start of option year one

Task 20. Model Editor User Interface Improvements

User interface issues have been identified by "beta testers" and NRC staff using the TRACE, RELAP5 and CONTAIN plug-ins. Some of the reported issues have been addressed, some have not. It is anticipated that many more user interface changes and improvements will be required after SNAP version 1.0 is released (expected in early 2006) and more people begin using and providing user suggestions. Also, the NRC is trying to establish a contract to provide usability feedback before SNAP version 1.0 is released. The contractor shall collect the suggestions for user interface changes and document them in the monthly progress reports. The contractor shall propose solutions as to how to address the user suggestions in SNAP and present them to the PO in a written report. After the PO sends either a written letter or an email to the contractor indicating approval, the contractor shall implement the proposed solution(s).

Option year 1:
Completion date: One year after start of option year one

Task 21. Develop "Modernized" MELCOR Plug-in

The SNAP MELCOR plug-in that was completed in 2005 addressed versions 1.8.5 and 1.8.6 of MELCOR. Since then, MELCOR developers have reimplemented the MELCOR code in FORTRAN 95 and have significantly changed the input deck format and organization. Therefore, a plug-in for the modernized version of MELCOR will need to be created. The contractor shall create the modernized MELCOR plug-in for SNAP providing support for input model editing and creation, runtime support and restart abilities, and connection to the animation model capabilities of SNAP. The contractor shall test the plug-in on a set of input decks provided to them by the NRC and confirm that the plug-in supports the modernized MELCOR code fully. The contractor shall follow the general requirements for code development as specified previously in this document (i.e., prepare SRS and SDID documents, obtain PO approval before implementation of design, etc.). The contractor shall write and deliver MELCOR plug-in user manuals.

Option year 1:
Completion date: One year after start of option year one

Option year 2:
MELCOR 2.0 plug-in development

Task 22. Add TRACE ECI Support to SNAP TRACE Plug-in Model Editor Capabilities

The TRACE Exterior Component Interface (ECI) provides TRACE with a method of transferring runtime calculation data between processes. This is meant to be used for "breaking" a calculation into pieces for running under separates threads possibly on separate machines and for interfacing specialized component models compiled as separate code. The SNAP TRACE plug-ins runtime interface can currently control and monitor the ECI "pieces." However, it is desirable to have a graphical interface that would allow a user to split an input model into the separate pieces that will be split off as separate ECI components. The contractor shall design and implement such an interface into the SNAP TRACE plug-in. As for all SNAP development work, the contractor shall document in SRS and SDID the requirements and design for this work and obtain approval of the documents from the PO, either through an email or letter, before proceeding to the implementation phase of the work.

Option year 1:
Completion date: 3 months after start of option year one

Task 23. Implement a "pluggable" Layout Engine for the Model Editor and Improve the Current Layout Algorithm

A semi-sophisticated algorithm that usually does an adequate job for small simple models currently handles layout of components on the 2D views. However, for plant size models where such a layout tool is really needed, the layout algorithm often produces unsatisfactory results. The analyst is then left to layout the model himself by hand. However, the model editor provides only minimal help with regard to manual layout.

Features such as grouping components, alignment of components and other structured drawing tools would greatly improve the analyst's ability to layout a model by hand.

1. To improve the layout features of SNAP, the contractor shall develop a layout engine plug-in interface so that new layout plug-ins can be implemented independent of the SNAP core code.

2. In addition, the contractor shall implement the current layout algorithm as a plug-in that is compatible with the new layout plug-in design. Improvements to the current algorithm to improve layout of large plants models shall also be investigated and recommendations shall be made to the PO as to the best options. Upon approval from the PO, the contractor shall implement the recommended layout engine option.

3. Features for manual layout shall be designed and implemented in SNAP. Features such as grouping components, alignment tools, etc., shall be proposed by the contractor and presented either orally or in a written document to the PO. In either case, the presentation shall have a "story board" visual to demonstrate the concepts involved. Upon approval from the PO, either via email or written letter, the contractor shall develop the standard SRS and SDID documents and implement the design. As for all SNAP development work, the contractor shall document in SRS and SDID the requirements and design for this work and obtain approval of the documents from the PO, either through an email or written letter, before proceeding to the implementation phase of the work.

Option year 1:
Completion date: 10 months after start of option year one

Task 24. Develop an Advanced Renodalization Tool

Renodalization of hydraulic components in TRACE or RELAP5 (and probably other such codes) can be very problematical when there are heat structures, control systems, or other hydraulic components attached. There are several areas where the current renodalization functionality can be expanded to improve efficiency. The current 1D renodalization functionality is restricted to operating on one component at a time. The ability to uniformly renodalize a group of selected components shall be added to assist with nodalization studies. The ability to scale the cross-sectional flow area of a selection of 1D components shall also be provided. A hydraulic loop could have every component's flow area scaled. This would allow an analyst to select a loop, copy the loop into a model, and adjust its flow areas to lump unbroken loops or extract a loop from a lumped loop. The ability to split TRACE TEE components into separate pipes and split and merge pipe components shall also be provided. Heat structures, signal variables, and steady state controllers that refer to hydraulic components will automatically adjust to take into account the new component numbers and node indexes. As for all SNAP development work, the contractor shall document in SRS and SDID the requirements and design for this work and obtain approval of the documents from the PO, either through an email or written letter, before proceeding to the implementation phase of the work.

Option year 1:
Completion date: 6 months after start of option year one.

WORK UNDER OPTION YEAR 2

Task 25. Application Testing and Distribution

1. SNAP plug-ins for analytical codes shall be tested to ensure that they can import and export the respective code input decks without causing changes in the calculated results. For each analytical code for which SNAP provides a plug-in, the contractor shall maintain and exercise a relevant suite of input decks to assure that the SNAP plug-in is performing input and output operations correctly. The contractor shall at least include in these test suites any decks that are provided by the PO. The contractor shall correct errors discovered in the associated SNAP plug-in during testing as necessary.

2. Maintain a multi-platform SNAP installation package to support code distribution. SNAP is to be distributed from a web-site under strict control of the contractor. Access to the SNAP code is to be coordinated through the PO.

3. Evaluate and correct errors reported by the SNAP user community.

4. Maintain and extend SNAP user's manuals and help files to reflect any code changes. Automate linkage to analysis code documentation.

5. Provide user support. Implement user-supplied requests for improvements as reviewed and approved by the PO.

Option year 2:
Completion date: One year after start of option year two

Task 26. Maintain SNAP Capabilities

The supported analytical codes (MELCOR, CONTAIN, TRACE, FRAPCON/FRAPTRAN and RELAP5) are still undergoing modifications that will necessitate changes to their input format requirements. The contractor shall upgrade SNAP to fully support these modifications. The contractor shall maintain contact with the analytical code developers so that any changes can be identified and modifications made to the respective SNAP plug-in in a timely fashion. The PO shall be notified in writing by letter or email of any planned changes to any plug-in before such changes are actually made. Note that in cases where the analytical code has been updated or changed in a major way, the contractor may need to write an entirely new plug-in for the new version of the analytical code. If the need for an entirely new plug-in is encountered, the contractor shall not proceed to make changes to the existing plug-in, but instead inform the PO so that other actions may be planned.

Option year 2:
Completion date: One year after start of option year two.

Task 27. Provide Technical Support

This task provides technical support to the NRC. The work includes making presentations, attending meetings, reviewing technical reports, and providing technical consultation and support as requested by the PO. Also, this task is to provide support

for "beta testers." "Beta testers" are SNAP users who are issued pre-release versions of SNAP for the purpose of evaluating SNAP in typical user environments (i.e., "real world" use of SNAP). Currently, there are about 30 beta testers for SNAP; however, only 5-10 are active.

Option year 2:
Completion date: One year after start of option year two

Task 28. SNAP User Interface Testing

The contractor shall use the test suite implemented in Task 4 of this contract to ensure that SNAP code integrity is maintained. Furthermore, the contractor shall add to the test suite any tests that are deemed necessary during the ongoing development of SNAP and SNAP plug-ins. Status of the results of using the test suite shall be documented for each of the public releases in a report to the PO.

Option year 2:
Completion date: One year after start of option year two

Task 29. Develop Java Based "Lightweight" Plotting Tools–DELETED

Task 30. Develop Advanced JavaBeans for the Animation Plug-in

The objects used to "build" an animation file mask are actually JavaBean components. In this work, the contractor shall design and add to the SNAP release package new JavaBeans that can be used for displaying complex data that may be produced by analytic code runs. Currently, one such JavaBean that is envisioned is a "core melt" JavaBean. This JavaBean will be able to display an animation of a melting core including material heat up, fuel and structure relocation, and combustion that may be occurring, etc. The contractor shall design and implement such a "core melt" JavaBean and up to ten other JavaBean components. The documentation requirement for these JavaBean components is reduced from the normal SNAP development. In this case, the contractor shall provide a description and design for the component all in one document.

The document need not follow the requirements as stated for the other SNAP development documents. However, the contractor shall not proceed with development of any such JavaBean component until the PO provides an email or letter providing approval of the design.

Option year 2:
Completion date: One year after start of option year two.

Task 31. Model Editor User Interface Improvements

User interface issues have been identified by "beta testers" and NRC staff using the TRACE, RELAP5 and CONTAIN plug-ins. Some of the reported issues have been addressed, some have not. It is anticipated that many more user interface changes and improvements will be required after SNAP version 1.0 is released (expected in early 2006) and more people begin using and providing user suggestions. Also, the NRC is trying to establish a contract to provide usability feedback before SNAP version 1.0 is

released. The contractor shall collect the suggestions for user interface changes and document them in the monthly progress reports. The contractor shall propose solutions as to how to address the user suggestions in SNAP and present them to the PO in a written report. After the PO sends either a written letter or an email to the contractor indicating approval, the contractor shall implement the proposed solution(s).

Option year 2:
Completion date: One year after start of option year two.

Task 32. Develop "Modernized" MELCOR Plug-in

The SNAP MELCOR plug-in that was completed in 2005 addressed versions 1.8.5 and 1.8.6 of MELCOR. Since then, MELCOR developers have reimplemented the MELCOR code in FORTRAN 95 and have significantly changed the input deck format and organization. Therefore, a plug-in for the modernized version of MELCOR will need to be created. The contractor shall create the modernized MELCOR plug-in for SNAP providing support for input model editing and creation, runtime support and restart abilities, and connection to the animation model capabilities of SNAP. The contractor shall test the plug-in on a set of input decks provided to them by the PO, and confirm that the plug-in supports the modernized MELCOR code fully. The contractor must follow the general requirements for code development as specified previously in this document (i.e., prepare SRS and SDID documents, obtain PO approval before implementation of design, etc.).

Option year 2:
Completion date: One year after start option year two

Task 33. Develop Water and Materials Properties Interface

Currently, several plug-ins require water and material properties data. As of now the water property data is built into SNAP and is based on the RELAP5 "new" water properties package. Materials data for the analytic codes is also built into the respective SNAP plug-ins. This reproduction of the materials properties and water properties in SNAP is dangerous from the point of view that if the analytical code should change its properties tables without the plug-ins being updated, then the input models that SNAP writes would be incorrect. It has been proposed that SNAP simply "talk" to the respective analytical codes to obtain the properties in question.

Under this task, the contractor shall develop an interface specification that can be used in the respective analytical codes that will permit SNAP to "talk" directly to the respective code to get the water properties or material properties as needed. The contractor shall publish the interface specification in a letter report to the PO and make the interface specification available to other codes developers, as specified by the PO. The contractor shall also write the interface code for the existing plug-ins and communicate these changes to the respective analytical code developers. As for all SNAP development work, the contractor shall document in SRS and SDID the requirements and design for this work and obtain PO approval, either through an email or written letter, before proceeding to the implementation phase of the work.

Option year 2:
Completion date: One year after start of option year two

Task 34. Add Engineering Data to SNAP

Currently, construction of piping components for TRACE and RELAP5 models requires inputting all of the geometric data associated with the piping being modeled. The geometric cross-sectional data for piping is typically obtained from a standard American Society of Mechanical Engineers (ASME)/American National Standards Institute (ANSI) piping schedule and must be entered into SNAP.

This task involves automating this process by allowing the analyst to construct piping components by simply selecting the type of pipe, (schedule, nominal diameter, and material) from a dialog displaying the standard ASME/ANSI piping schedule. The pipe component will then be created along with an optional pipe wall heat structure based on the entered pipe length and number of parallel pipes being modeled. As for all SNAP development work, the contractor shall document in SRS and SDID the requirements and design for this work and obtain PO approval of the documents, either through email or a letter, before proceeding to the implementation phase of the work.

Option year 2:
Completion date: 7 months after start of option year two

Task 35. Develop a "BWR stability plug-in".

The plug-in should automate the methodology that has been developed for stability analysis. The plug-in would run the TRACE PARCS input decks with the defined perturbations (noise, control rod, or pressure) and perform the signal processing needed to determine the decay ratio and the natural frequency.

The contractor shall implement the required features in SNAP, submitting a SRS, and SDID (possibly combined since this is considered a small effort) and then making the code changes after NRC PM approval, as per the usual procedure for SNAP development tasks.

Task 36. "RELAP5 to TRACE Model Conversion."

The purpose of this task is to begin implementation of a general deck conversion plug-in. The contractor shall write a plug-in that shall take RELAP5 decks and convert them to TRACE decks. The contractor shall also develop a user interface for this plug-in that can help an analyst make decisions regarding how deck translation should occur.

The contractor shall implement the required features in SNAP, submitting a SRS, and SDID (possibly combined since this is considered a small effort) and then making the code changes after NRC PM approval, as per the usual procedure for SNAP development tasks.

Task 37. "PARCS to TRACE coupling feature plug-in."

The purpose of this task is to develop a plug-in to support coupling PARCS and TRACE models. Another option is to modify the PARCS plug-in to allow

development of a mapping between the PARCS components and the TRACE code components. The requirement for this task is the development of a user friendly interface that allows the mapping of the PARCS code kinetics model onto a TRACE code Thermal/Hydraulic model.

The contractor shall implement the required features in SNAP, submitting a SRS, and SDID (possibly combined since this is considered a small effort) and then making the code changes after NRC PM approval, as per the usual procedure for SNAP development tasks.

Task 38. "TRACE model documentation feature enhancements."

Model documentation is very important to model quality assurance. To help an analyst document an input model the SNAP model editor (and possibly the individual plug-ins) shall be enhanced and modified.

The required features are:

- Ability to include reference documentation, including links to web pages, links to PDF documents and other common forms of electronic documentation.
- Ability to output a printable document (PDF and/or HTML) that contains all of the embedded model annotations and model statistics.

The contractor shall implement the required features in SNAP, submitting a SRS, and SDID (possibly combined since this is considered a small effort) and then making the code changes after NRC PM approval, as per the usual procedure for SNAP development tasks.

Task 39. "CAFEAN API NUREG"

The contractor shall submit the CAFEAN API specification in NUREG ready format for publication in 2008. The bulk of the API (Application Programmer Interface) shall follow the standard JavaDoc standards, however there shall be an introductory part that describes the use for the API for creating a simple plug-in, as an example of how to use the API. The documentation must follow NUREG/BR formatting requirements as well. The NUREG specifications will be provided to the contractor when needed.

Task 40. "TRACE Error Assistant Development, Phase 1"

The purpose of this task is to develop an error processing feature in SNAP to assist in debugging TRACE input models. Developing a new input model for TRACE currently involves a trial-and-error methodology. A TRACE input model can be very complex and involve many different input specifications. Even with the assistance of the SNAP model editor it is still possible to develop a TRACE input model that TRACE cannot run. An analyst now must submit the input model to be run under TRACE and then, if the model fails to run, look at the TRACE error output in a separate text editor to find where TRACE crashed. Once identifying where the model crashed the user must then examine the input model and try to ascertain what went wrong. The purpose of this task is to

develop the capability to automatically have SNAP relate errors identified by TRACE fed back into the model editor and then have the model editor assist the user through various visual queues to fix the modeling errors.

This task requires changes to the TRACE error reporting. Therefore, several phases are foreseen to accomplish this task. The first phase will be to examine the output that TRACE produces to examine the details that it provides with the goal of using those details in SNAP to locate the input model problems. To the degree possible, an error processing assistant shall be created that identifies to the user where errors exist in the input model. This first phase shall also identify any needed information that TRACE will need to be modified to provide to fully isolate errors in the input model.

E. DELIVERABLES AND DELIVERY SCHEDULE

The deliverables and delivery schedule set forth in this section are for reporting requirements as described in the individual tasks under Paragraph D, Scope of Work above. Other code deliverables are required as discussed under the individual task descriptions in Paragraph D, Scope of Work, above.

Task	Description	Due date
1.5	User interface improvement recommendations and design report(s)	Recommendations as needed in monthly progress reports Design report due 60 working days after request by PO
1.6	Training Manuals and media containing the software required for the training. One set of manuals and media (enough for up to 15 people) will be required for each type of training (up to 3 types).	The delivery date depends on the actual training schedule which will be determined on an "as-needed" basis. However, the contractor will be given at least two weeks to prepare the required materials for any given course.
1.7	The following **user** documentation formatted ready for publication as a NUREG/BR: (1) SNAP Core; (2) SNAP TRACE plug-in; (3) SNAP RELAP5 plug-in; (4) SNAP PARCS plug-in.	Draft version of the NUREG/BR documentation is due June 29, 2007. Final version is due by August 31, 2007.
1.9	The K-Factor/FRIC conversion tool is Documentation of this feature shall be included in the SNAP TRACE plug-in user manual.	Due to be included in the SNAP TRACE plug-in by August 31, 2007.
1.10	The new loop elevation check system Documentation of this feature shall be included in the SNAP TRACE plug-in user manual.	Shall be incorporated into the SNAP TRACE plug-in by August 31, 2007.
1.11	The Plug-in Reference Documentation Management tool. Documentation of this tool shall be included in the SNAP Core user guide.	Shall be completed and functional by October 31, 2007.
4	SRS and SDID for the testing interface	Due 3 months after award of contract

4	Report documenting the content and use of the test suite/framework	Draft due 4 months after award of contract Final due 5 months after award of contract
5	Status report for user interface tests	Report due prior to each public release
6	SRS and SDID and user's guide for 3D plotting additions	Draft due 2 months after award of contract Final due 3 months after award of contract
7	SRS and SDID for lightweight plotting tools	Draft due 11 months after award of contract Final due 1 year after award of contract
8	Requirement/Design/Completion document for each new "JavaBean"	Draft due 30 working days after requirement identified by PO Final due 1 year after award of contract
10	SRS and SDID for AVScript Interface Plug-in	Draft due 11 months after award of contract Final due 1 year after award of contract
11	SRS and SDID for Updated CONTAIN Plug-in	Draft due 6 months after award of contract Final due 7 months after award of contract
12	SRS and SDID for RELAP5 to TRACE Vessel Conversion Tools	Draft due 2 months after award of contract Final due 3 months after award of contract
13	SRS and SDID for Animation Export Tools	Draft due 4 months after award of contract Final due 5 months after award of contract
17	Status report for user interface tests	Report due prior to each public release
18	SRS and SDID for lightweight plotting tools	Documented in Monthly Technical Progress Report
19	Requirement/Design/Completion document for each new "JavaBean"	Draft due 30 working days after requirement identified by PO Final due 1 year after start of option year one
20	User interface improvement recommendations and design report(s)	Recommendations as needed in monthly progress reports Design report due 60 working days after request by PO
21	SRS and SDID for Modernized MELCOR Plug-in	Due September 30, 2008.
23	SRS and SDID for layout engine and model editor improvements:	Draft due 8 months after start of option year one. Final due 10 months after start of option year one
24	SRS and SDID for Advanced Renodalization Tool(s)	Draft due 5 months after start of option year one. Final due 6 months after start of option

		year one
28	Status report for user interface tests:	Report due prior to each public release
29	SRS and SDID for lightweight plotting tools	Documented in Monthly Technical Progress Report
30	Requirement/Design/Completion document for each new "JavaBean"	Draft due 30 working days after requirement identified by PO Final due 1 year after start of option year two
31	User interface improvement recommendations and design report(s)	Recommendations as needed in monthly progress reports Design report due 60 working days after request by PO
32	SRS and SDID for Modernized MELCOR Plug-in	Documented in Monthly Technical Progress Report
34	SRS and SDID for adding engineering data into SNAP	Draft due 5 months after start of option year two Final due 7 months after start of option year two
35	The "BWR Stability Plug-in" SDID and SRS documentation.	Documentation due September 28, 2007. Plug in due no later than December 21, 2007.
36	The "RELAP5 to TRACE Model Conversion" plug-in SDID and SRS documentation	Documentation due July 31, 2007. Plug in due no later that October 31, 2007.
37	The "PARCS to TRACE coupling feature plug-in" SDID and SRS documentation	Documentation due December 21, 2007 Plug-in due no later than March 31, 2008
38	The "TRACE model documentation feature enhancements" SDID and SRS documentation	Documentation due June 29, 2007. Enhancements due no later than August 31, 2008.
39	"CAFEAN API NUREG" documentation	Due no later than April 30, 2008.
40	The "TRACE Error Assistant" SDID and SRS documentation Note: for this phase of the TRACE error assistant development only a preliminary design is required. Further work on the error assistant will be decided on after reviewing the results of the first phase of the project.	Due no later than November 30, 2007.

The contractor shall review technical reports to ensure their high quality. The content of format technical reports should follow generally accepted technical writing practices; see NUREG-650, Revision 1, "Publishing Documents in the NUREG Series." The author must consider the audience who will read the document; link ideas in sentences and paragraphs to create an easy-to-follow logical transition; and ensure consistency of terminology, format, and style throughout. The reports should be well focused; i.e., they should not be too wordy and the prose should flow in a logical manner. The author must provide necessary information to avoid prose where logic would be incomplete. Technical reports should not include policy, administrative, managerial, or fiscal information unsuitable for wide dissemination. They should not contain proposals for additional work and words should be carefully selected to avoid marketing of contractor capabilities.