Pacific Northwest NATIONAL LABORATORY

PNNL- 29717

# Health Physics Codes Consolidation and Modernization

February 2020

Bruce K McDowell
Caitlin A Condon
Jeremy P Rishel
Bruce A Napier
Steve J Maheras
Russell D Swannack
Todd V Billow
Angela C Dalton
Eva E Hickey

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

PACIFIC NORTHWEST NATIONAL LABORATORY
*operated by*
BATTELLE
*for the*
UNITED STATES DEPARTMENT OF ENERGY
*under Contract DE-AC05-76RL01830*

**Printed in the United States of America**

**Available to DOE and DOE contractors from the**
**Office of Scientific and Technical Information,**
**P.O. Box 62, Oak Ridge, TN 37831-0062;**
**ph: (865) 576-8401**
**fax: (865) 576-5728**
**email: reports@adonis.osti.gov**

**Available to the public from the National Technical Information Service**
**5301 Shawnee Rd., Alexandria, VA 22312**
**ph: (800) 553-NTIS (6847)**
**email: orders@ntis.gov <https://www.ntis.gov/about>**
**Online ordering: http://www.ntis.gov**

PNNL- 29717

# Health Physics Codes Consolidation and Modernization

February 2020

Bruce K McDowell
Caitlin A Condon
Jeremy P Rishel
Bruce A Napier
Steve J Maheras
Russell D Swannack
Todd V Billow
Angela C Dalton
Eva E Hickey

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

# Summary

The U.S. Nuclear Regulatory Commission's (NRC's) Radiation Protection Computer Code Analysis and Maintenance Program (RAMP) develops, maintains, improves, distributes, and provides training on NRC-sponsored radiation protection and dose assessment computer codes. The RAMP codes provide dose assessment and licensing support for all the areas of radiation protection including nuclear power plant licensing, fuel cycle, and consumer and medical use of radioisotopes.

The computer codes in the RAMP program have been developed since the 1970s to address specific regulatory needs. The codes have numerous current and legacy issues that reduce the efficiency of operation and maintenance of the codes and increase associated costs. In their current state, the codes are also unable to fully assess radiological doses from advanced non-light water reactor designs. The RAMP suite of codes faces several issues, including the following:

- inefficiency of having and maintaining multiple codes.
- functional redundancy between codes.
- dated science within the codes.
- rigid and dated data transfer methods.
- lack of standardized code writing.
- limited ability to assess advanced reactor designs.
- a history of changing ownership and associated loss of code knowledge.
- inconsistent maintenance.
- lack of standardized quality assurance.

These current and legacy issues could be addressed by transforming the suite of single-purpose radiation protection and dose assessment codes into a single consolidated code that is modular, flexible, efficient, and user-friendly. This report describes this proposed new functional approach. The framework for the proposed consolidated code relies on three general approaches:

- **Consolidate Current Code Capabilities into One Code.** Extract the scientific functions of the current suite of RAMP codes and create a consolidated code that has functional code modules or "engines" that perform the same calculations. The engines would also be modified as needed to address unique aspects of advanced non-light water reactor designs.

- **Develop Standardized Data Transfer Schema.** Establish a standardized schema for data transfer into and out of the code, and between engines within the code, that would be flexible enough to encompass all existing information types and be easily expanded to incorporate unique inputs from advanced reactor designs.

- **Build a Single and Separate User Interface.** Establish a single user interface that would be separate from the functional engines and would access the functional engines as needed to address specific regulatory needs or perform research.

This report also identifies other issues to be considered during code development, including transition from the current suite of codes to a consolidated code, potential participants in code development, quality assurance, regulatory requirements, and recommendations for early prototype development to demonstrate proof-of-concept.

# Acronyms and Abbreviations

| | |
|---|---|
| ARCON | Atmospheric Relative Concentrations in Building Wakes computer code |
| ATD | atmospheric transport and dispersion |
| BWR | boiling-water reactor |
| CAP88 | Clean Air Act Assessment Package of 1988 |
| CCP | Calculation Control Package |
| COTS | Commercial off-the shelf software |
| CFR | U.S. Code of Federal Regulations |
| D/Q | relative deposition factor |
| DandD | Decontamination and Decommissioning computer code |
| DBA | design-basis accident |
| DOE | U.S. Department of Energy |
| EPA | U.S. Environmental Protection Agency |
| GALE | gaseous and liquid effluent computer code |
| GASPAR | dose analyses computer code for nuclear power plant radioactive effluents to the atmosphere |
| GENII | Generation II computer code |
| GOTS | government off-the-shelf software |
| GUI | graphical user interface |
| HA | High-Assurance |
| HABIT | control room habitability computer code |
| LADTAP | dose analyses computer code for nuclear power plant radioactive effluents to surface waters |
| LOCA | loss-of-coolant accident |
| LWR | light water reactor |
| MILDOS | radiological impacts computer code for airborne emissions from uranium milling facilities |
| MOU | Memorandum of Understanding |
| NARAC | National Atmospheric Release Advisory Center |
| NESHAP | National Emission Standard for Hazardous Air Pollutants |
| NPP | nuclear power plant |
| NRC | U.S. Nuclear Regulatory Commission |

| | |
|---|---|
| NRCDose | Not an acronym (Software suite that integrates the functionality of three individual Fortran codes: LADTAP II, GASPAR II and XOQDOQ) |
| NUREGs | (NRC's) NUREG-series publications |
| PAVAN | ground-level $\chi$/Q for accidental release computer code |
| PNNL | Pacific Northwest National Laboratory |
| PWR | pressurized-water reactor |
| QA | quality assurance |
| RadToolbox | Radiological Toolbox computer code |
| RADTRAN | Radioactive Material Transport computer code |
| RAM | radioactive material |
| RAMP | Radiation Protection Computer Code Analysis and Maintenance Program |
| RASCAL | Radiological Assessment System for Consequence Analysis computer code |
| RG | Regulatory Guide |
| SCP | Software Control Package |
| SNAP/RADTRAD | Symbolic Nuclear Analysis Package, Radionuclide Transport, Removal, and Dose Estimation computer code |
| SQA | software quality assurance |
| SCALE | Standardized Computer Analyses for Licensing Evaluation computer code |
| SFR | sodium-cooled fast reactor |
| SLAB | DOE denser-than-air releases computer code |
| SNAP | Symbolic Nuclear Analysis Package |
| SNL | Sandia National Laboratories |
| SOARCA | State-of-the-Art Reactor Consequence Analyses |
| SQAP | Software Quality Assurance Plan |
| SSC | structure, system, or component |
| UI | user interface |
| XOQDOQ | computer code for atmospheric dispersion modeling for routine releases |
| XML | extensible markup language |
| $\chi$/Q | relative air concentration |

# Contents

# Figures



# Tables

# 1.0 Introduction

The U.S. Nuclear Regulatory Commission's (NRC's) Radiation Protection Computer Code Analysis and Maintenance Program (RAMP) develops, maintains, improves, distributes, and provides training on NRC-sponsored radiation protection and dose assessment computer codes as described in SECY-14-0117, "The Radiation Protection Computer Code Analysis and Maintenance Program" (NRC 2014). The RAMP codes provide dose assessment and licensing support for all areas of radiation protection, including nuclear power plant (NPP) licensing, fuel cycle, and consumer and medical use of radioisotopes. RAMP includes radiation protection and dose assessment computer codes, which encompass NPP licensing (e.g., reactor siting, design-basis accidents [DBAs], and normal effluent releases), emergency response and severe accidents, atmospheric transport and dispersion (ATD), and site decommissioning.

The computer codes in RAMP have been developed since the 1970s to address specific regulatory needs. These codes today have numerous current and legacy issues that reduce the efficiency of operation and maintenance of the codes, thereby increasing the cost of their continued use. Although some of the codes have been updated recently and continue to be revised, they are still stand-alone codes and many of the legacy issues still exist. This report discusses a way to address these issues by transforming the current suite of single-purpose radiation protection and dose assessment computer codes into a consolidated functional and modern code that is modular, flexible, efficient, and user-friendly. This report describes this new functional approach.

Most of the current RAMP codes are designed for large light water reactors (LWRs). As part of the preparation for regulating a new generation of non-light water reactors (non-LWRs), the NRC has developed a vision and strategy to assure their readiness to efficiently and effectively conduct the NRC mission for these technologies, including fuel cycles and waste forms. In July 2016, the NRC published the draft vision and strategy document, "NRC Vision and Strategy: Safely Achieving Effective and Efficient Non-Light Water Reactor Mission Readiness," in the *Federal Register* (81 FR 47443). In December 2016, the NRC published the updated vision and strategy document to reflect stakeholder feedback (NRC 2016). The NRC is also developing *NRC Non-Light Water Reactor (Non-LWR) Vision and Strategy, Volume 4 — Licensing and Siting Dose Assessment Codes (RAMP)* (in draft), which describes the computer codes in RAMP and how they would be applied to each of the principal non-LWR design types, and summarizes the tasks necessary to resolve "gaps" in the capability to model and simulate those designs with the accuracy required by the regulator.

The proposed code consolidation approach would help the NRC prepare for their review and regulation of non-LWRs by addressing the gaps described in *Volume 4*. For instance, because many of the current versions of RAMP codes are rigid in design structure as well as coding, the codes do not have the flexibility to accept input parameters different from those of LWR technologies. These parameters include source term development considerations that will vary significantly for high-temperature gas-cooled reactors, sodium fast reactors, molten salt reactors, fluoride salt-cooled high-temperature reactors, liquid metal-cooled reactors, and heat pipe reactors. The existing codes would also need to be updated to address the following technical issues:

- core radionuclide inventory determination accounting for fuel form, geometry, and other relevant characteristics.
- near-field ATD modeling.

- selection of dose coefficients.
- environmental exposure pathways including tritium and carbon-14 modeling.
- chemical transport modeling.
- fuel reprocessing.

The RAMP codes considered in this report are generally codes that calculate atmospheric dispersion, normal and accidental releases, and dose and dose consequences to humans, nonhuman biota, and the environment. The codes described in this report are considered to have similar and, in some cases, redundant capabilities that could be consolidated. The proposed consolidation approach would also be flexible enough to add the functions of additional single-purpose codes in the future.

This report is organized as follows:

- Section 2 briefly describes the current codes in RAMP, the codes that are the focus of this report and a short description of their purpose, and a problem statement describing related challenges and issues.
- Section 3 describes the historical dose assessment approach, the proposed modernization/consolidation approach, an example of early prototype development, the scope of the consolidation effort, and the advantages of a flexible approach.
- Section 4 describes other considerations including meeting regulatory requirements, potential participants and roles, and software quality assurance.
- Section 5 describes code development and implementation considerations including agile software development, software transition management, and phased code development and implementation.
- Section 6 presents conclusions and recommendations.
- Section 7 lists references to supporting documents.
- Appendix A discusses NRCDose3 as an example of code consolidation.
- Appendix B presents an example graphical user interface.

# 2.0 RAMP Codes and Regulatory Requirements

Current computer codes developed by the NRC under RAMP to support regulatory compliance, and the challenges and issues they face relative to modernization needs, are described in the following sections.

## 2.1 Current Codes in RAMP

The RAMP NPP codes are designed and developed by the NRC and other agencies to support licensing, emergency response and severe accident dose assessments, calculation of ATD, and decommissioning at conventional LWRs (i.e., both pressurized-water reactors [PWRs] and boiling-waters reactors [BWRs]). Figure 2.1 shows the codes that are currently in RAMP and the codes that may be added to RAMP in the future.

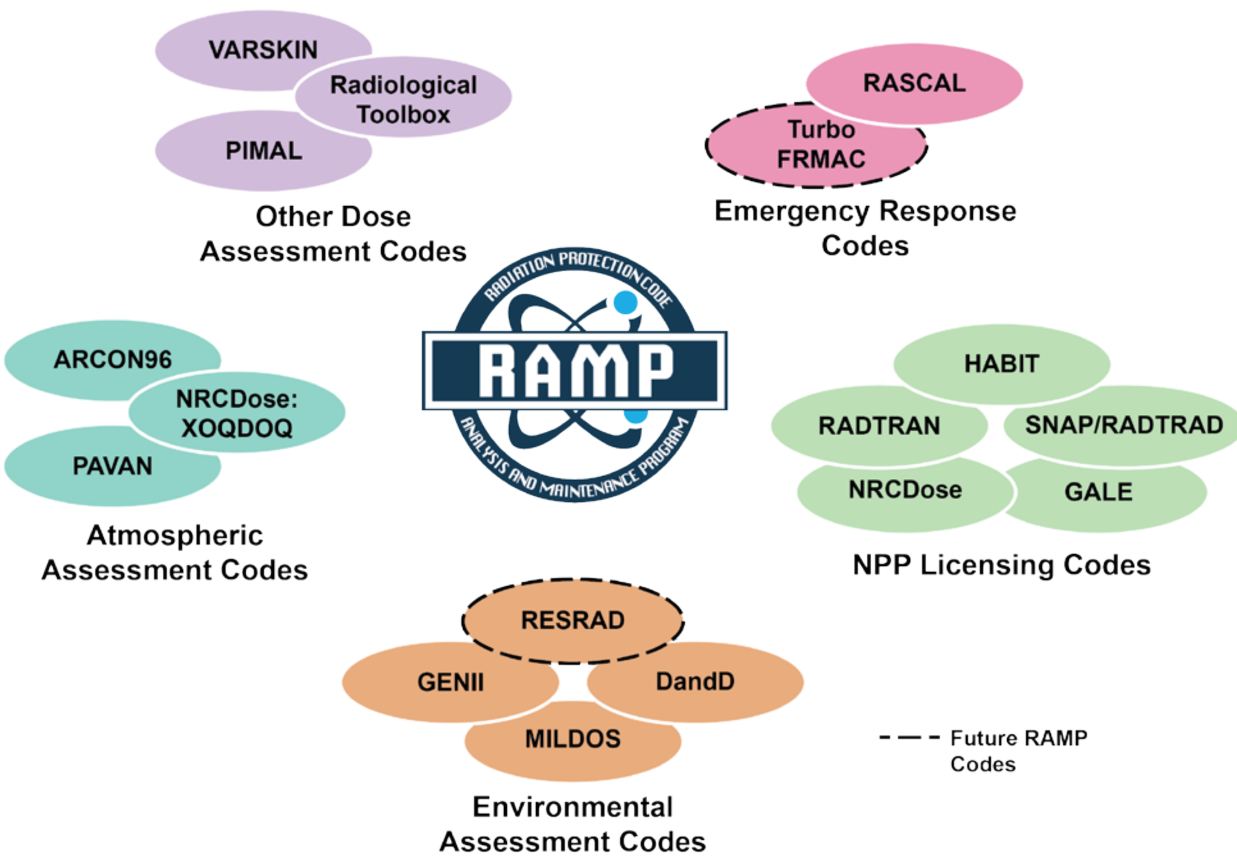



Figure 2.1. Current and Future RAMP Codes

The RAMP NPP computer codes provide tools for NRC staff to perform confirmatory calculations with respect to the regulations in various parts of Title 10 of the *U.S. Code of Federal Regulations* (10 CFR) and NRC Regulatory Guides (RGs). Figure 2.2, from *NRC Non-Light Water Reactor (Non-LWR) Vision and Strategy, Volume 4 — Licensing and Siting Dose Assessment Codes (RAMP)* (in draft), displays the relationship of some of the RAMP codes to the specific NPP dose assessment areas.

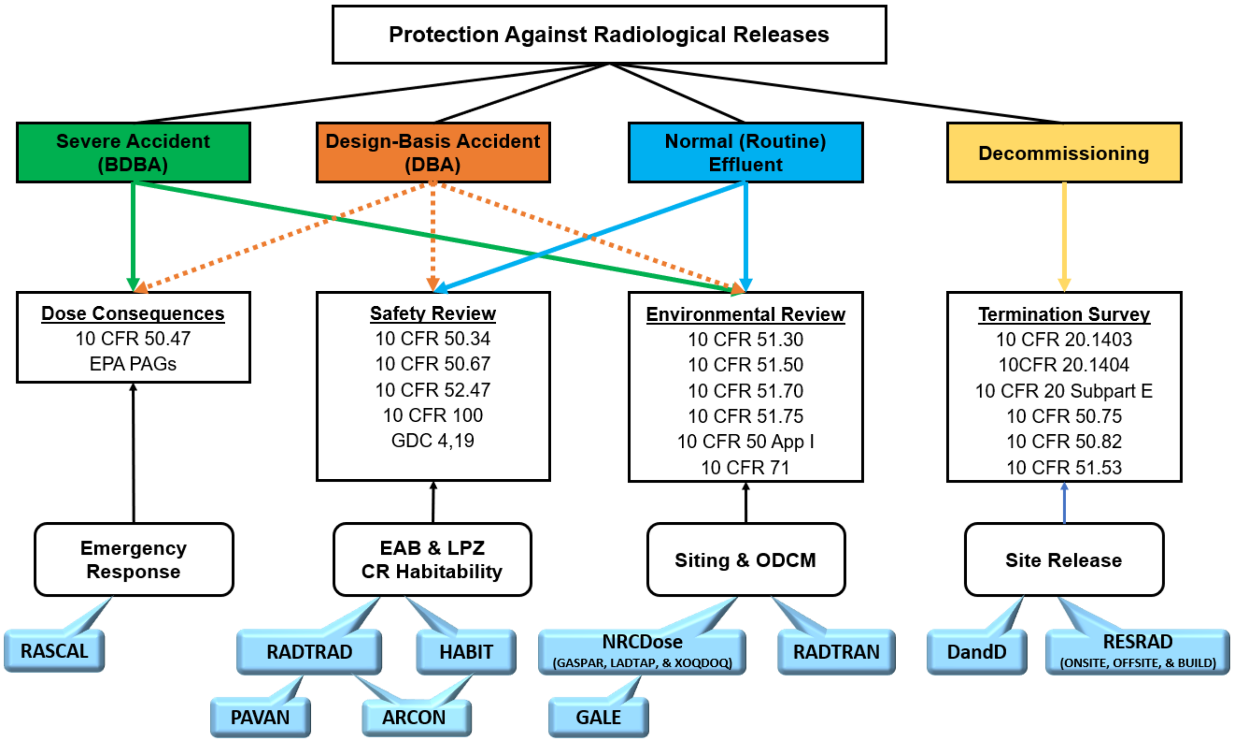



Figure 2.2. RAMP Code NPP Applications and Regulations

The codes considered in this report for consolidation and modernization are the codes that calculate atmospheric dispersion, accumulation in the environment, exposure to humans and nonhuman biota, and dose consequences. Table 2.1 lists the RAMP codes that are the focus of this consolidation and modernization approach and provides a brief description of their purposes.

Table 2.1. RAMP Codes

| Code | Description |
|---|---|
| ARCON96 | Used to calculate relative air concentrations ($\chi$/Qs) in support of control room habitability assessments required by general design criteria (GDC) 19 of 10 CFR Part 50 Appendix A and RG 1.194 (NRC 2003). Available to RAMP users for free. |
| DandD | Assesses compliance with the dose criteria of 10 CFR Part 20, Subpart E to allow licensees to perform simple estimates of the annual dose from residual radioactivity in soils and on building surfaces as described in NUREG-1700 (NRC 2018). Available to RAMP users for free. |
| GENII | Environmental Radiation Dosimetry Software (Napier et al. 2011). Models radionuclide release scenarios for atmospheric and surface water releases and how they would affect doses to nearby humans and nonhuman biota. Available to RAMP users for free. |
| MILDOS | Licensing confirmatory analysis software for the radiological impacts of airborne emissions from uranium mining and in situ recovery activities. Available to RAMP users for free. |

| Code | Description |
|---|---|
| NRCDose | Software suite that integrates the functionality of three individual Fortran codes: LADTAP II, GASPAR II, and XOQDOQ, which were developed by the NRC and have been in use by NPP licensees and the NRC staff for assessments of liquid radioactive releases and offsite doses, gaseous radioactive effluents and offsite doses, and meteorological transport and dispersion, respectively. LADTAP II implements the liquid pathway modeling described in RG 1.109 (NRC 1977a) and RG 1.113 (NRC 1977b). GASPAR II implements the gaseous effluent pathway modeling described in RG 1.109 (NRC 1977a); XOQDOQ implements the atmospheric dispersion modeling described in RG 1.111 (NRC 1977c). Available to RAMP users for free. |
| PAVAN | Used to estimate relative ground-level air concentrations ($\chi$/Q s) resulting from radioactive material releases from design-basis accidents at NPPs. Available to RAMP users for free. |
| RADTRAN | Confirmatory analysis code used for risk and consequence analysis associated with routine, incident-free, transportation of radioactive materials (RAM) and accidents that might occur during RAM transportation. Available to RAMP users for free. |
| RASCAL | Emergency response code used to perform independent dose and consequence projections during radiological incidents and emergencies (severe accidents) at NPP facilities. The RASCAL code is a fast-running emergency response code that calculates projected doses for various NPP severe accident scenarios and is used for protective action decision-making. Available to RAMP users for a fee. |
| SNAP/RADTRAD | Licensing confirmatory analysis code used, as described in NUREG/CR-7220 (Arcieri et al. 2016), for design-basis loss-of-coolant accident (LOCA) assessment calculations. Available to RAMP users for a fee. |

## 2.2 Problem Statement

The greatest problem facing the RAMP suite of codes is that there is not one single problem, but rather multiple, complex problems that make modernization of the codes a multifaceted challenge. Many of the codes listed in Table 2.1 were developed in the late 1970s and early 1980s to implement certain calculations documented in RGs that are either considered acceptable to the NRC for the licensing of NPPs or are required by *Federal Regulations*. From a licensing perspective, these codes assure certain calculations are performed in a consistent, technically defensible, and reproducible manner. A historical review of these codes reveals they were generally developed in a serial manner by independent contractors, as the RGs were written, and as funding became available for their development. This method of code development has led to multiple issues, which can be categorized as presented below.

**Redundancy:** Many of the RAMP codes perform similar functions, are structurally similar, or share common data. For example, the gaseous and liquid effluents (GALE) series of codes is broken into four separate codes for calculating annual average normal (routine) releases of radioactive material in gaseous and liquid effluents from PWR and BWR NPPs, even though these four codes could be designed to share similar computational routines. Other RAMP codes, like XOQDOQ, GASPAR II, and LADTAP II, which use the GALE source term to estimate dose from normal operations, were designed separately and did not share input and output data. To make the three codes more user-friendly, NRCDose was developed to make a software suite of the three codes. The atmospheric codes, principally ARCON96, PAVAN, and XOQDOQ, might be better served under a single “atmospheric engine” for performing near-, mid-, and far-field dispersion calculations. Each code has its own data management process leading to unique inputs, outputs, and ways of transferring data. This makes it difficult to share data between the codes, and in some cases, leads to overlapping functionality.

**Dated Science:** Given the continual development of RGs, additional inefficiencies exist. For example, Appendix A to RG 1.23 Revision 1 (NRC 2007b) is a more recent RG that provides a recommended format for providing hourly meteorological data to the NRC. The three legacy atmospheric models in RAMP—XOQDOQ, PAVAN, and ARCON96—cannot directly use this format. Instead, XOQDOQ and PAVAN use a highly simplified meteorological data set, called a joint frequency distribution, which requires additional, independent processing of the hourly data. ARCON96 uses hourly meteorological data that must first be converted to a special model-specific format. These various meteorological data formats and additional processing steps make the RAMP codes more difficult to use and can lead to errors and inefficient licensing reviews. As another example, in XOQDOQ, dry deposition is decoupled from the air concentration calculation and wet deposition is not treated at all.

**External Risks:** The RAMP codes also face external risks, such as not compiling in modern compilers, incompatibilities running on advanced (e.g., 32- and 64-bit) operating systems, and lack of legacy programming language(s) talent as more advanced computer languages are adopted. While incremental maintenance can keep these codes functional with computer system updates, these compatibility issues will only increase as the codes get older. Eventually, the codes may no longer operate on newer computer systems. Beyond just maintaining functional codes is the issue that the science behind these codes has continued to advance since their development, yet many of the codes are tied to specific regulations, which limits their ability to adapt to scientific changes.

**Lack of Quality Assurance (QA):** When the codes were developed in the late 1970s and early 1980s, no formal program like RAMP existed to assure the codes were continuously maintained. Thus, no mechanism existed to track and fix known code issues. Furthermore, the RAMP codes were not developed to the same QA standards and the existing QA depends largely on when and where the codes were developed and maintained. Some of the more recently updated and maintained RAMP codes have more up-to-date QA, but other codes that have undergone minimal maintenance have little-to-no documented QA. Having a comprehensive QA program will be critical moving forward to assure that RAMP codes are functional, accurate, and technically defensible. This will be critical as RAMP continues to broaden its domestic and international user base, as well as for licensing and regulating advanced nuclear reactors.

**Dated Code Practices:** Many of the RAMP codes reflect development practices, styles, and standards that were appropriate for the time, but are now considered outmoded and inconsistent with present-day programming practices. For example, many of the codes were developed in early versions of the Fortran programming language. Though still functional, these codes use deprecated conventions like “GO TO” statements, which make following code logic difficult, leading to the slang term “spaghetti code”—a tangled web of programming code in which control within a program jumps around. In addition, many of the codes lack formal variable declaration, often using “implicit declaration,” whereby the first letter of the variable name is used to declare the variable type (i.e., integer, real, or string). In some cases, variables are typed as “variants,” and can be used to hold integer, real, and character values, depending on the calling routine, which significantly increases the risk of a computational error. Yet other legacy issues, like using over-simplistic variable naming conventions (e.g., “X”, “Y”, or “Z”) to conserve computer memory, are no longer an important code development consideration. Because many of the RAMP codes were developed when these dated code practices were prevalent, maintaining them into the future becomes increasingly difficult.

**Multiple Code Owners and Custodians:** Many of the RAMP codes have gone through a change in ownership or at times a lack of ownership. Changing ownership between organizations leads to lost knowledge about the intricacies of the development of each code. Changing ownership can also lead to changes being made in the codes, but if these changes are not well documented, they may be lost, thereby leading to a code that was altered from its original state but without a lasting record. Furthermore, conflicts in intellectual property rights can create challenges when a code is changing custodian or ownership. Changing ownership leads to knowledge loss and different requirements for maintenance and QA, which over time can lead to computational errors, degradation in code value, or loss of capability.

**Inability to Assess Non-LWR Designs:** The RAMP codes were all developed to support the existing fleet of U.S. LWRs, but the nuclear industry is evolving and preparing for advanced reactors. The RAMP codes developed for LWRs are generally incapable of handling this new technology. While the questions for the reactors are generally similar (e.g., related to releases, dosimetry, decommissioning, etc.), the base assumptions used in advanced reactors are fundamentally different than for LWRs, and there is no easy way to address these differences in the existing codes. Furthermore, the science behind the RAMP codes is continuously evolving and improving, and many of the codes have not been updated to reflect the changes in scientific understanding since their original development.

**Continuing Maintenance of Multiple Codes:** Coupled with these challenges is the general challenge to be more efficient. Having a suite of different codes all requiring modernization and maintenance is inefficient given the overlapping capabilities. Most of the codes were developed in a vacuum, meaning they were developed with a specific input and output in mind. When stepping back to look at the bigger picture of overlapping capabilities within the codes it becomes apparent that maintaining and updating all of them does not make financial sense. Furthermore, because every code was developed individually, there is no standardized coding language, input format, or user interface between the codes in the RAMP suite—all of which make them difficult to consolidate in their current state. While all the capabilities and functions need to be retained, maintaining a suite of different codes to perform these functions is inefficient.

These current and legacy issues could be addressed by transforming the suite of single-purpose radiation protection and dose assessment codes into a single consolidated code that is modular, flexible, efficient, and user-friendly.

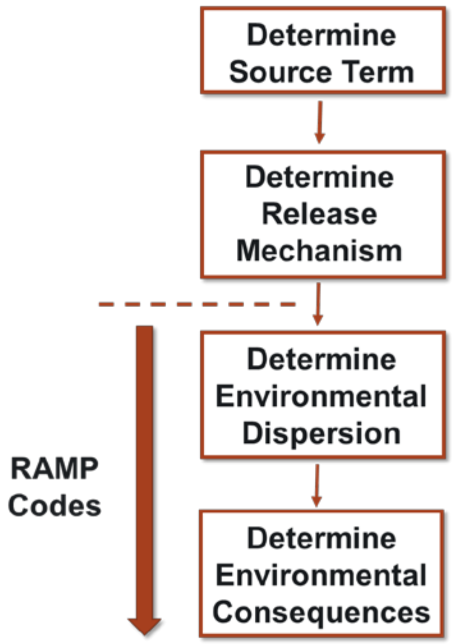
# 3.0 Consolidation/Modernization Approach

The proposed approach builds on NRC's historical dose assessment approach by consolidating the important functional elements of current RAMP codes into one modern and user-friendly code.

## 3.1 Historical Dose Assessment Approach

Dose assessment for NRC licensing efforts has historically consisted of four sequential steps:

1. Determine the source term: What radionuclides are available for potential release to the environment?
2. Determine the release mechanism: How are the radionuclides released, e.g., chemistry, particle size, buoyancy, building size, stack height?
3. Determine the dispersion of the release into the environment: Where do the releases go…how much, how far, how fast?
4. Determine the environmental consequences: What is the resulting dose to humans and nonhuman biota?




The RAMP codes considered in this report have been used primarily to evaluate Steps 3 and 4 in this dose assessment process.

The evaluations in the environmental dispersion and consequence steps can be further separated into four sets of calculations:

- **Dispersion.** Calculate the dispersion into the environment. This calculation includes dispersion in the atmosphere or through surface water (rivers and lakes) or groundwater.
- **Accumulation.** Calculate the location and accumulation of radionuclides.
- **Exposure.** Calculate the exposure to humans and nonhuman biota.
- **Consequence.** Calculate the dose to humans and nonhuman biota. Determine the appropriate dose coefficients and calculate the dose.

The current suite of RAMP codes has been developed to perform some or all of these calculations, depending on their regulatory purpose. Figure 3.1 illustrates examples of some of the calculations performed by the RAMP codes.

Figure 3.1 also illustrates various redundancies in the existing code suite. Although the RAMP codes were designed with unique approaches to each functional step in the assessment process, many codes perform similar functions. While each RAMP code has its own purpose and capability, many of the scientific principles are common to or shared with other RAMP codes. For example, several RAMP codes calculate atmospheric dispersion (i.e., NRCDose, RADTRAN, RASCAL, ARCON 96, PAVAN, GENII). While each code may have a different purpose, capability, input type, or endpoint of interest, they all rely on principles of atmospheric dispersion as part of their coding. Similar redundancies and overlaps can be identified in other functional areas, such as dispersion through bodies of water, environmental accumulation, nonhuman biota exposure, human exposure, and consequences.

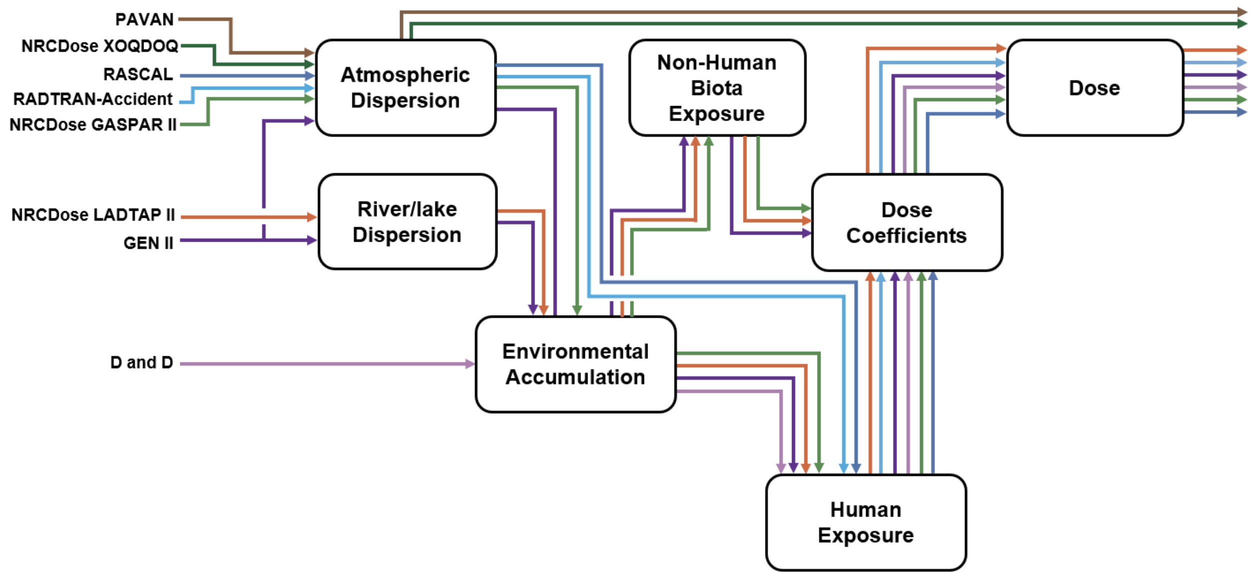



Figure 3.1. Example Current RAMP Code Calculations

## 3.2 Consolidated Code Approach

The proposed approach would develop code modules—or "engines"—that would consolidate the calculations performed by the current suite of RAMP codes and modernize the code language, data transfer methods between the engines, and user interface (UI). The approach for developing consolidated code engines would address current and legacy issues by doing the following:

- constructing code engines that consolidate functions to reduce redundancy and allow for flexibility and easy updates.
- leveraging both legacy and modern code languages, as needed, for efficiency.
- using consistent methodologies for passing data into and out of the code engine and between engines.
- developing a user-friendly UI that can be used to call functional engines as needed to meet a regulatory requirement.
- addressing technical issues needed for licensing advanced non-LWR designs.
- identifying clear owners and defined roles and responsibilities.
- developing and maintaining a documented QA program.

This framework would reduce overhead costs associated with maintaining the individual RAMP codes, while retaining their critical functions and integrity within the nuclear regulatory field. The consolidated code engines would be developed in a new framework that relies on three general concepts, or "pillars," as shown in Figure 3.2 and described below.

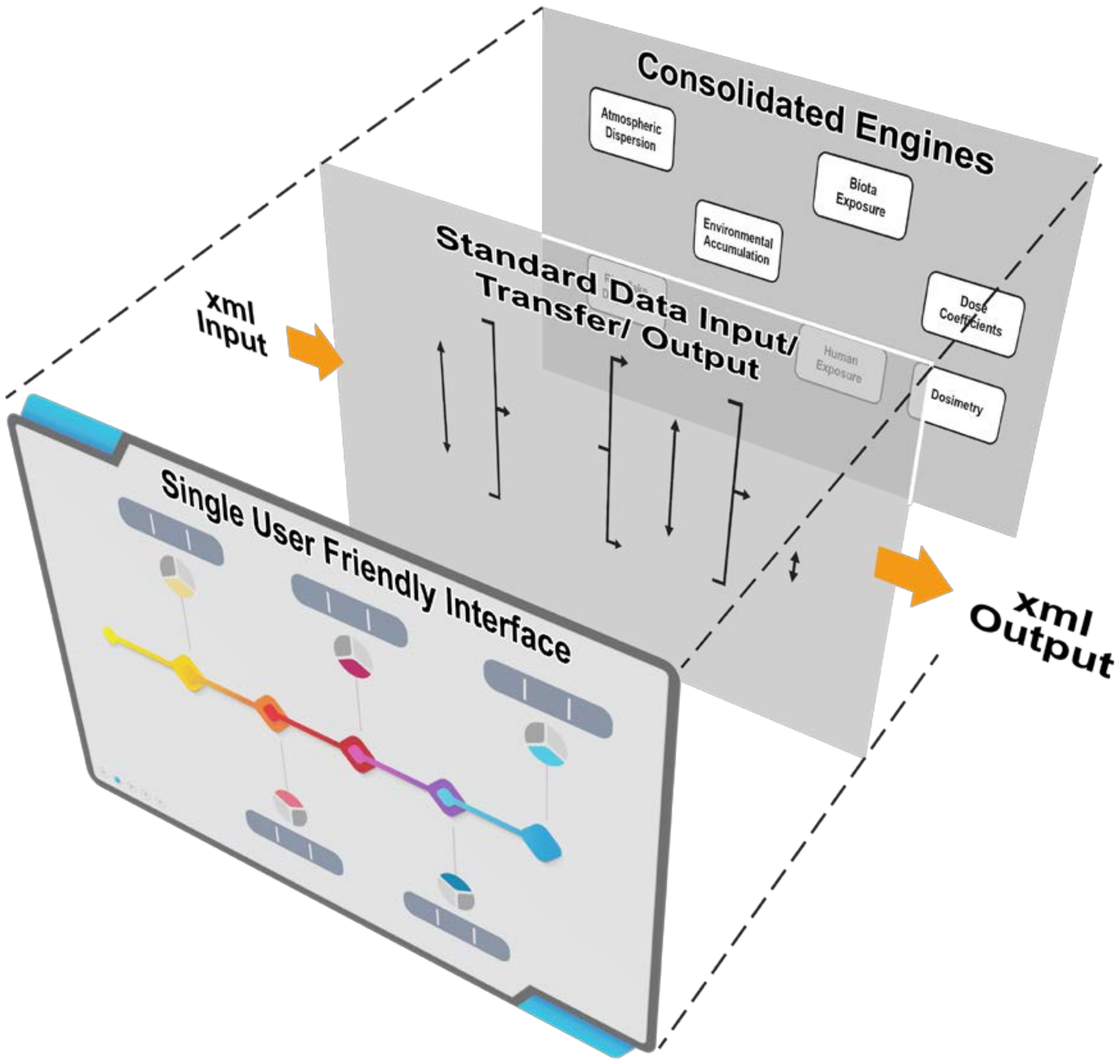



Figure 3.2. Three Pillars of a Consolidated Code Framework

- **Create Consolidated "Engines".** Extract the scientific functions of the current suite of RAMP codes and create a single consolidated code that has functional modules or engines that perform the same calculations of dispersion, accumulation, exposure, and consequence. Maintain the scientific functions of the RAMP codes but maintain them as consolidated functional engines rather than as separate, redundant, and constrained functional portions of individual codes. The engines could also be modified, as needed, to address unique aspects of advanced non-LWR designs.
- **Develop Standardized Data Input/Transfer/Output Schema.** Establish a standardized method for data transfer into, out of, and between engines that encompasses all existing data requirements and is readily expandable to accommodate future requirements, such as computational changes or unique inputs from advanced reactor designs. Accordingly, an extensible markup language (XML) schema would be developed for each engine, which would allow for seamless data movement between any of the functional engines or future development therein.

- **Build a Single User-Friendly Interface.** Establish a single UI that would be separate from the functional engines and would access the engines as needed to address specific regulatory needs or perform research. A UI will always require continual maintenance and separating the UI from the engines allows for continuous maintenance. In addition, grouping all engines under one interface has numerous benefits, such as a consistent user experience and simplified maintenance (i.e., a single versus multiple interfaces).

### 3.2.1 Pillar 1: Consolidated Engines

Pillar 1 is the development of code engines that consolidate the scientific functions of the RAMP codes listed in Table 2.1. The consolidated code would consist of a set of functional modules—engines—that would perform the same calculations required for dose assessment as those performed by the current RAMP code suite. Functional engines would isolate the basic science that is common to the RAMP codes in a stand-alone set of code modules written in a modern code language that is flexible enough to incorporate future improvements. The development of an engine framework would separate the fundamental scientific principles, making this framework adaptable and flexible for regulating both the existing fleet of nuclear reactors and advanced reactors. Figure 3.3 illustrates the concept of functional engines and their use in determining the consequences of a release of radionuclides.

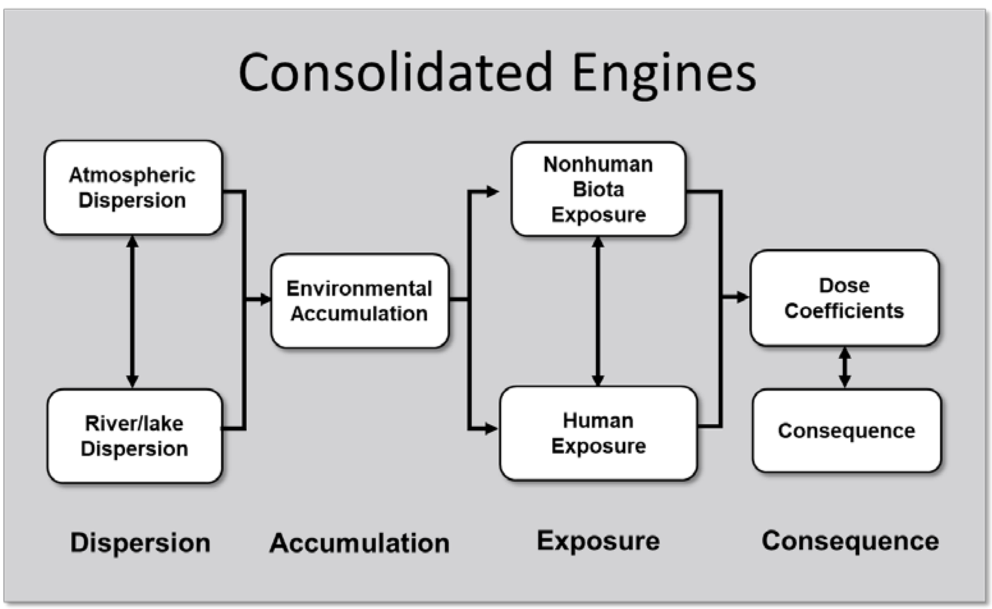



Figure 3.3. Functional Engines in the Code Consolidation Framework

Code for the functional engines would be developed using the following approach:

- Use legacy code language in current RAMP codes as applicable.
- Update codes with modern code languages and conventions as needed for efficiency and flexibility.
- Update codes for current scientific approaches and data availability.
- Update codes to address advanced non-LWR design issues.

A code user would be able to choose the functional engines and the assessment path during model setup and execution. The choice of engines used within the consolidated code would depend on the regulatory or research purpose of a given run. For example, a user that would have previously evaluated atmospheric dispersion using PAVAN or XOQDOQ would only use the atmospheric dispersion engine. An assessment that would have been completed using NRCDose would use the atmospheric and river/lake dispersion engines, the environmental accumulation engine, the human exposure engine, and the dose coefficient and dose engines. In this way, one consolidated set of functional engines would be able to serve multiple regulatory assessment needs. Figure 3.4 and Figure 3.5, respectively, illustrate the use of functional engines using PAVAN and NRCDose GASPAR II runs as examples.

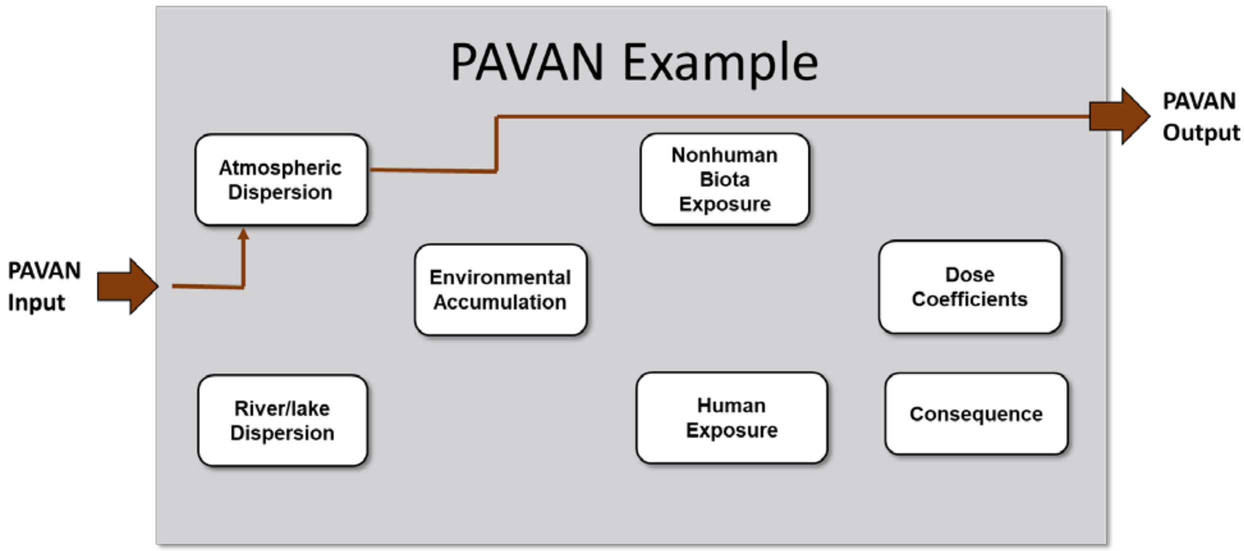



Figure 3.4. Functional Engines Used in a PAVAN Run

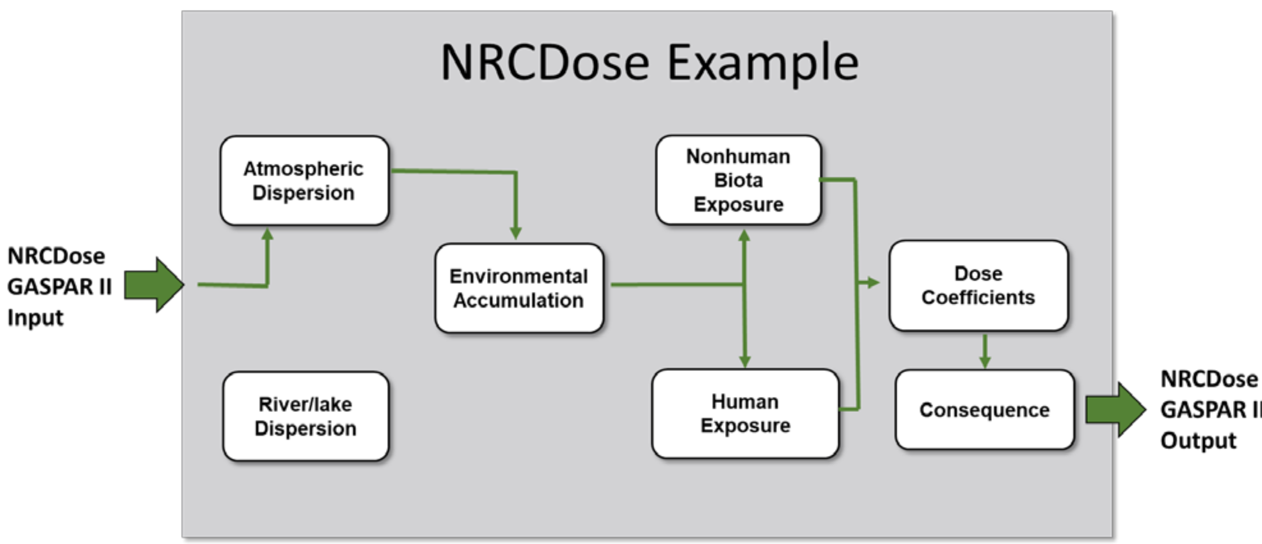



Figure 3.5. Functional Engines Used in a NRCDose GASPAR II Run

Appendix A provides an additional discussion of NRCDose as an example of how the functionality of an existing code would be implemented in a consolidated code, and how a user would interface with the consolidated code to meet the same regulatory purpose.

The functional engine approach improves development flexibility by allowing for future modifications and efficient data transfer. Furthermore, separating capabilities as stand-alone engines eliminates code redundancy and inefficiency. By maintaining fewer codes, more time can be spent on assuring that the engines remain scientifically current and relevant. Not only would this limit the burden of work to maintain the RAMP capabilities, but it would eliminate misapplication of the RAMP codes. For example, rather than users needing to understand which modeling distances are appropriate for PAVAN compared to ARCON96, the user-specified inputs would dictate which atmospheric model would be appropriate for the distance being evaluated. The functional engines would also be modified to address emerging advanced non-LWR design issues such as near-field ATD modeling, selection of dose coefficients, and new environmental exposure pathways, including tritium and carbon 14 modeling.

Because the consolidated approach separates out the scientific capabilities and is not designed to meet a specific regulatory purpose, this approach would also allow for dose assessment research to be conducted separate from any specific model or regulatory requirement. Inputs to the model would be generic source term and release mechanism outputs. Functional engines could be chosen based on the research interest. Outputs could be custom-designed to answer specific research questions.

### 3.2.2 Pillar 2: Standard Data Input/Transfer/Output

The second pillar of a consolidated framework is the use of a standardized method for data input, transfer, and output (Figure 3.6). The current suite of RAMP codes relies on formatted data input and output files for operation. In all cases, the format is fixed, meaning any change to the input file (e.g., a new variable or data type) requires an underlying change to the code and this change would generally not be backward compatible with earlier versions of the code. The proposed approach would rely on a modernized method— XML—for data handling.

XML is a markup language that defines a set of rules for encoding data in tags that is both self-descriptive and computer-readable. Modern programming languages can read XML files natively. Thus, XML is a powerful and flexible data structure choice because it is development language agnostic (i.e., compatible with any coding language as a way of managing data). By using an XML schema, rules for encoding data developed for each engine would make data input universal and adaptable, while making it easy to pass output data between the different functional engines. By using XML as the data management system within the consolidated code framework, the entire system would be more robust relative to the advancements in the nuclear industry and any associated improvements (e.g., meteorological input data). In addition, this data management system would be more adaptable to variables that cannot be anticipated at this point, including those associated with advanced non-LWR designs.

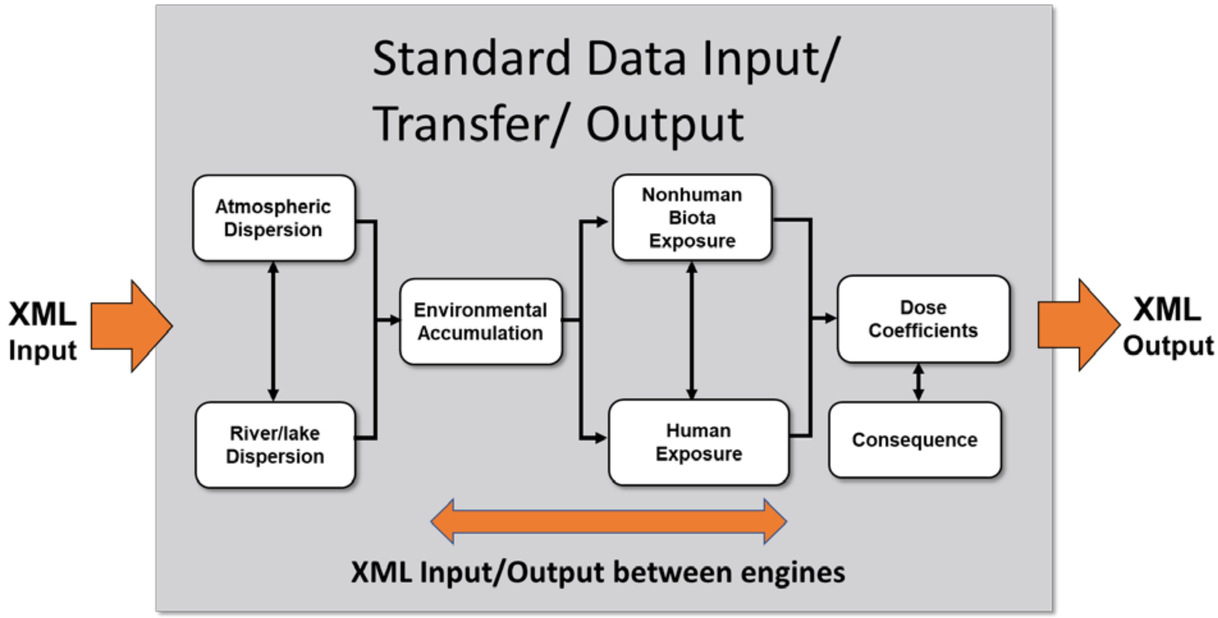



Figure 3.6. Standard Data Input/Transfer/Output

### 3.2.3 Pillar 3: Separate User-Friendly Interface

The new framework relies on a complete separation between the functional code engines and the UI. The UI would consist of a front end (i.e., a graphical user interface [GUI]) that interacts with users and a back end that communicates with functional engines to execute user-defined computational commands. The GUI would be designed to effortlessly guide users through the relevant code engine input screen(s), primarily through a series of questions about desired outputs (e.g., see Figure 3.7). All user entries and selections from the GUI screens would be used to develop the relevant XML data input files used by the back end of the UI and code engines. After the engines complete a run, the XML output would be used by the GUI for display and would also be available as text output to the user. Simplified GUI options are envisioned for infrequent (e.g., “once-a-year”) users who may want to quickly run certain routine regulatory compliance options (Figure 3.7).

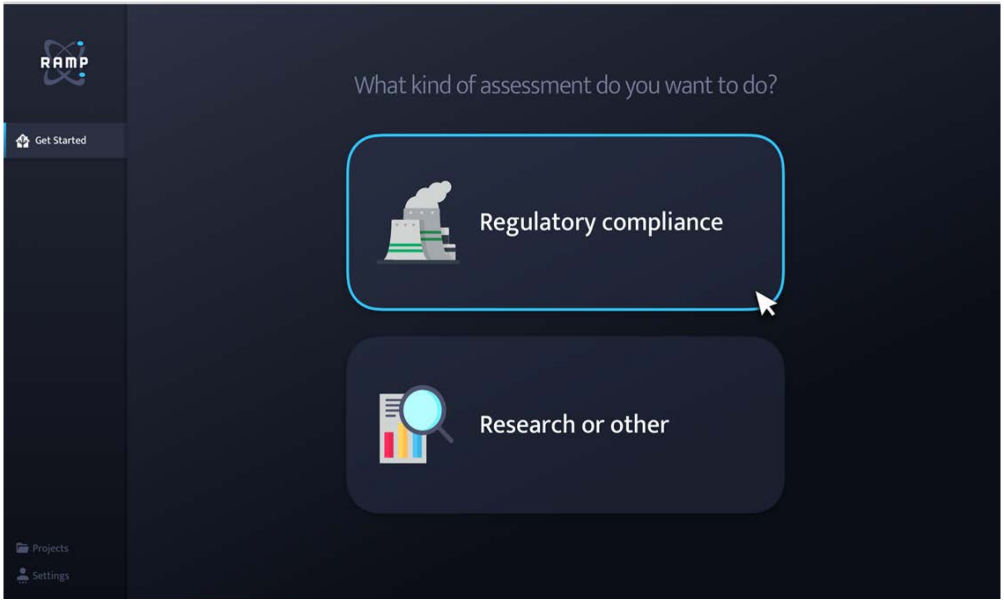



Figure 3.7. Example Graphical User Interface Entry Screen

The new RAMP UI would develop XML inputs and present outputs to users regardless of whether they have one functional engine or all the available RAMP functional engines. This way the NRC can still provide some functions of the RAMP codes to users free of cost, while other functional engines would require a fee for use (allowing the user to download the functional engine and make that portion of the UI active). An example high-level workflow of the UI is presented in Figure 3.8.

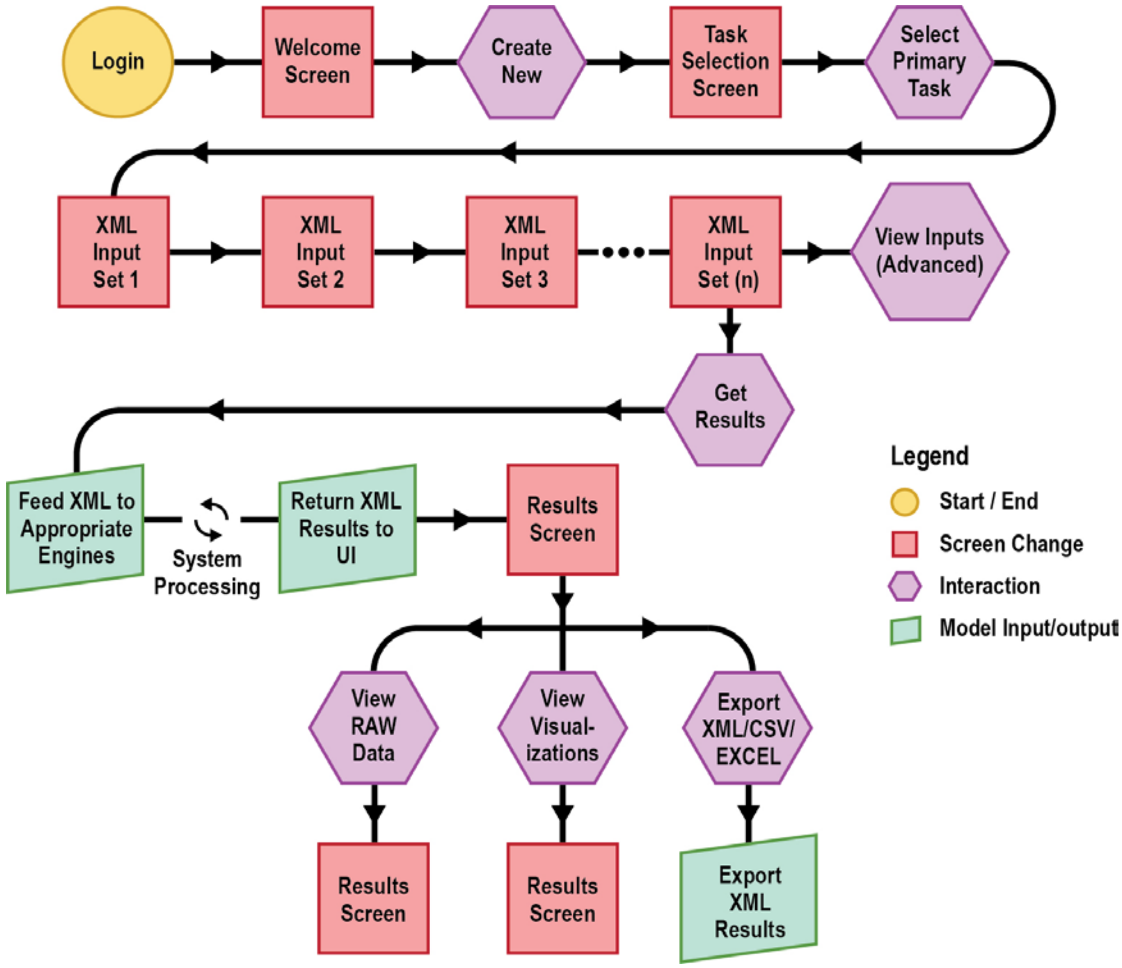



Figure 3.8. High-Level UI Workflow Scenario

By separating the UI from the functional engines, by way of a standardized XML data schema, the UI could be updated and changed without affecting the functional engines. Separating the UI from the functional engines would allow the NRC to offer the RAMP codes as either a web-based service or a local installation using the same functional engines. This separation would also allow for future development of interfaces compatible with mobile devices and other new technologies. In addition, separating the UI from the functional engines would allow for more efficient QA reviews of each component. Finally, maintaining one UI would be more cost-effective than trying to maintain the functionality of separate UIs for each code, as is currently the case under RAMP. An example UI approach is presented in Appendix B.

### 3.2.4 Early Prototype Development

Early development of a prototype is recommended to demonstrate proof-of-concept and guide subsequent full code development. The goal of a prototype would be to demonstrate the three pillars of the code consolidation and modernization approach, that is:

- Demonstrate how one core functional engine can perform the needed function and operate separately from the UI.
- Demonstrate the use of the XML data transfer schema.

- Demonstrate the functionality and experience of the new UI.

The new framework would be demonstrated by developing a prototype single functional engine and a new UI (which could be expanded to accommodate all functional engines) with a standardized XML data schema. Development of an atmospheric dispersion functional module is recommended as a demonstration of the consolidated code concept. As shown in the upper portion of Figure 3.9, near-, mid-, and far-field dispersion calculations are currently performed by ARCON96, PAVAN, and XOQDOQ (performed in NRCDose):

- ARCON96: Near-field (~100s of meters); DBAs at the control room and technical support center.
- PAVAN: Mid-field (~10 km); DBAs at the exclusion area boundary and in the low-population zone.
- XOQDOQ: Far-field (out to ~80 km); normal effluent releases for sensitive receptors and population.

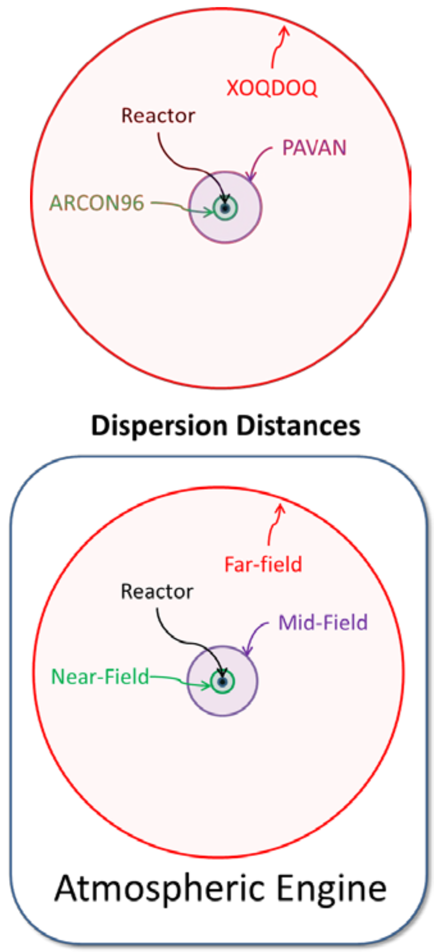



Figure 3.9. Dispersion Distances Calculated by an Atmospheric Engine

The lower portion of Figure 3.9 shows a single "atmospheric engine" that would perform these same calculations or calculations needed for any other distance as part of a review of an advanced non-LWR application. To develop this engine, the core computations performed in these existing codes would be reviewed and combined into one atmospheric engine (Fortran or other) that is comprehensive and flexible enough to calculate dispersion over near-field, mid-field, far-field, or any other required distance. An XML schema would be developed for standardized data transfer into the code, between functions in the code, and out of the code. The prototype would use the recommended format for providing hourly meteorological data described in Appendix A of RG 1.23 Revision 1 (NRC 2007b); currently, the three legacy atmospheric models in RAMP—ARCON96, PAVAN, and XOQDOQ—cannot directly use this format. Customized data output reports would be developed to present the results as required to address specific regulatory needs. A separate UI would be developed to demonstrate how a user would interact with the code for a specific regulatory purpose.

Prototype development would consist of two initial parallel tasks:

- Develop an atmospheric dispersion functional engine and XML schema for the variables of interest.
- Develop a functional UI.

Completion of the prototype would consist of pairing the atmospheric engine with the UI. Prototype testing would consist of focus group testing, user interviews, and/or usability testing of the UI and functional testing of the code to verify data transfer and confirm the code's accuracy and consistency with ARCON96, PAVAN, and XOQDOQ results. The prototype development would conclude with a report summarizing lessons learned for full consolidated code development.

## 3.3 Flexibility for Future Expansion

The proposed code consolidation would address current issues facing the RAMP codes and would be flexible for future upgrades and expansion as described below.

**Flexibility in Code Development**. By separating the UI from the functional engines, the consolidated code would allow for potential expansions of RAMP capabilities in the future with minimal changes to the UI. Additional functional engines could be created to address additional or unique assessments such as dose to the public from routine transportation of spent fuel. Functional engines could be improved as new scientific approaches are developed or additional data become available. Separating the UI would minimize the effort required for updates. The ability to add or improve functional engines would improve the NRC's ability to fully assess the release of radionuclides to the environment.

**Flexibility in Data Input and Output.** Using a standard XML input format (i.e., schema) would allow external data in this format to be easily imported into the model. Standard XML output data would also improve flexibility for creating reports that meet specific regulatory needs or creating customized reports to support research.

**Flexibility to Address Future Non-LWR Designs**. The consolidated code engine structure and the XML data transfer structure would allow for changes in calculations and inputs to address future non-LWR designs and other advanced reactor scenarios.

**Flexibility in Maintaining QA.** By separating the UI, the data, and the science, the UI experience can updated or changed without affecting the QA pedigree of the functional engines in the core code. QA would be maintained for each engine and pillar of the approach, such that a change to the UI would not threaten the quality of a functional engine, thereby allowing for flexibility in updates to engines or the UI.

**Flexibility in Creating Customized UI**. By separating the UI from the functional engines, users would have the option to potentially build their own UIs (e.g., another regulator could build a UI that only had the users go through a defined set of functional engine inputs to meet a specific regulatory requirement).

**Flexibility of Single Ownership**. The consolidated code would be owned and maintained by the NRC. This would simplify responsibilities for code maintenance and upgrades.

**Flexibility in UI Development.** Maintaining the functional engines separate from the UI would also allow for the development of a web-based framework in the future. This would be accomplished by creating a new UI that users could log into through the RAMP website, encrypting their XML input data, sending them to a remote server to go through the functional engines, encrypting the XML output data, and sending them back to the web-based interface for the user.

## 3.4 Code Consolidation Scope

The proposed consolidated code would perform the same calculations as the existing health physics codes in RAMP as well as some other codes used by other federal agencies. The code consolidation effort should prioritize the initial and subsequent efforts to consolidate the RAMP codes while also considering the potential for a consolidated code to meet the regulatory needs of other federal agencies.

### 3.4.1 NRC Codes

The proposed framework would consolidate functions performed by some of NRC's current suite of RAMP codes and would be flexible enough to allow future expansion and addition of modules designed to serve other regulatory or research needs. The consolidated code approach would also be flexible enough to add modules designed for stand-alone or special assessment cases. Table 3.1 lists codes that would be initially consolidated, codes that could be incorporated in the future, and possible stand-alone and special case codes.

Table 3.1. Codes Considered in the Consolidation Scope

| Initial Codes New Approach | Additional Codes or Inputs That Could Be Incorporated | Possible Stand-Alone/Special Cases |
|---|---|---|
| ARCON96 | PIMAL (MCNP) | RASCAL (Separate Code for EOCs) |
| DandD | HABIT (Control Room LWR) | |
| GENII | Radiological Toolbox | |
| MILDOS - Dispersion and Dose | GALE (Source Term for LWR) | |
| NRCDose (GASPAR II, LADTAP II, XOQDOQ) | MILDOS (Mining Specific Source Term) | |
| PAVAN | VARSKIN | |
| RADTRAN - Accident | RADTRAN (Routine Shipments) | |
| RASCAL | | |
| SNAP/RADTRAD (Source Term for LWR) | | |

### 3.4.2 Other Federal Agency Codes

Other federal agencies use codes similar to the RAMP codes to perform dose assessments. For example, the U.S. Environmental Protection Agency (EPA) has also developed codes as regulatory compliance tools for the emission of radionuclides into the environment. The codes developed by the EPA have functions that are complementary to those within the RAMP suite of codes to show compliance with the National Emission Standard for Hazardous Air Pollutants (NESHAP). The Clean Air Act Assessment Package of 1988 (CAP88) is an example of an EPA code whose functions overlap with those in the RAMP suite of codes. CAP88 is an air emission modeling tool that uses a Gaussian plume distribution to estimate exposure and risk from low-level chronic exposures (EPA 2013). The COMPLY EPA tool is a screening assessment tool for effective dose equivalents from stack emissions of radionuclides used to demonstrate compliance with regulations (EPA 1989).

These EPA codes were developed to demonstrate compliance with NESHAP, but the scientific functions of these codes overlap with the capabilities seen in the RAMP suite of codes. The functional engines in the consolidated code could be designed to calculate the same emissions, exposures, and risks as those currently determined using CAP88 and COMPLY codes. Similarly, the new UI could include choices for an EPA user to easily access a code run that meets CAP88 or COMPLY requirements. This same approach could be used to include other codes used by other federal agencies such as the U.S. Department of Energy.

# 4.0 Other Considerations

When considering how to best consolidate the current suite of single-purpose radiation protection and dose assessment computer codes into a functional and modern suite of codes, the need to meet regulatory requirements, identify potential participants and roles, and assure software QA should also be considered.

## 4.1 Meeting Regulatory Requirements

The current suite of RAMP codes was developed to support licensing, emergency response and severe accident dose assessments, and decommissioning at conventional LWRs. The regulatory need and guidance for the RAMP codes is found in regulations, RGs, NUREG-series publications (NUREGs), user manuals, and other documents. For example, Table 4.1 lists regulations and guidance associated with ARCON96.

Table 4.1. Regulations and Guidance Associated with ARCON96

| | |
|---|---|
| **Regulations** | 10 CFR Part 50, Appendix A, General Design Criterion 19, Control Room |
| | 10 CFR Part 50, Paragraph IV.E.8 of Appendix E, to Emergency Facilities and Equipment |
| **Guidance** | RG 1.206, C.I.2.3.4: “Short-Term Atmospheric Dispersion Estimates for Accident Releases” (NRC 2007a) |
| | SRP 2.3.4: *Short-Term Atmospheric Dispersion Estimates for Accident Releases* (NRC 2007c) |
| | RG 1.194: Atmospheric Relative Concentrations for Control Room Radiological Habitability Assessments at Nuclear Power Plants (NRC 2003) |
| | NUREG/CR-6331: Atmospheric Relative Concentrations in Building Wakes (Ramsdell and Simonen 1997) |

The consolidated code described in this report would perform the same key calculations performed by the RAMP codes and would provide an alternative approach to addressing the same regulatory needs. However, the guidance in RGs and the associated NUREGs is specific to the current suite of codes. Currently, there is no option for using an alternate consolidated code to meet the same regulatory purposes. Additional guidance and code updates will also be necessary to assess non-LWR designs.

The transition to a new consolidated code would have to first consider the NRC’s continuing need to demonstrate compliance with the appropriate regulations as the new code is developed, and second consider the need to develop new or modified guidance that addresses the use of the consolidated code. The transition to a new consolidated code would begin with an evaluation of the regulations and guidance documents to determine where they are aligned and where they need to be revised. The evaluation would be used to develop recommendations for updating the guidance to allow use of the consolidated code to meet the regulations. To maintain the continuing viability of the current RAMP codes, it may be necessary to perform the code consolidation in steps, making upgrades incrementally while necessary changes to RGs and NUREGs are implemented.

## 4.2 Potential Participants and Roles

The RAMP codes have been developed and continue to be developed and maintained by the NRC, U.S. Department of Energy (DOE) national laboratories, other federal agencies, universities, and commercial contractors (Figure 4.1). In addition, international entities (for example Public Health England) have relevant expertise in radiological code development.

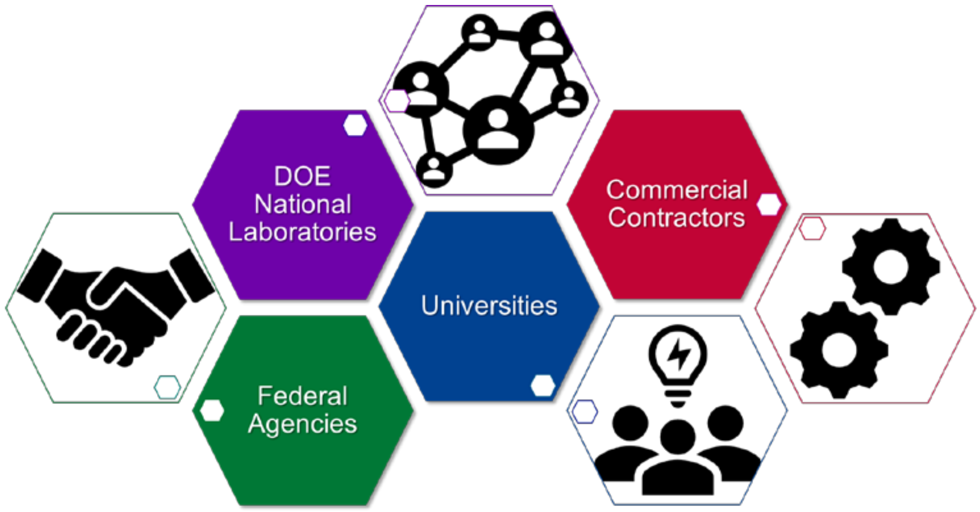



Figure 4.1. Code Development Participants

The proposed approach (described in Section 3.0) would consolidate the functions of several of these codes into one set of functional engines and update these functions to address anticipated non-LWR designs. Accordingly, the consolidated code would not require continuing support from individual custodians. However, individual code developers could provide the following general support during code development, depending on the aspect of the code being consolidated and updated:

- Review and provide input on the consolidation approach.
- Provide information about previously implemented legacy models and code changes.
- Provide information about past QA, configuration management, and verification and validation.
- Assist in integrating the code into the functional approach described in Section 5.0.
- Assist in identifying areas of code that are no longer used.
- Assist in reviewing coding changes for consolidation, modernization, and consideration of non-LWR designs.
- Assist in verifying the calculations.
- Assist in documenting default parameter values.
- Assist in developing a suite of sample problems that exercise all areas of the consolidated code and anticipated non-LWR designs.
- Assist in developing a user guide and technical manual.

Also, in *NRC Non-Light Water Reactor (Non-LWR) Vision and Strategy, Volume 4 – Licensing and Siting Dose Assessment Codes (RAMP)* (in draft), the NRC recommends that the individual RAMP codes be independently evaluated by the code's current contractor relative to various non-LWR technologies to provide the NRC with a roadmap for code updates and changes necessary to support non-LWR licensing. As noted in Section 2.0, recommendations that are developed during individual code evaluations should be considered in the development of the final approach to code consolidation and modernization. Depending on the recommendations, capabilities of current custodians and other health physics organizations may be required.

Future code maintenance and development would benefit from continuing engagement with the code developer community. A comprehensive project plan should include engagement mechanisms (e.g., Memoranda of Understanding [MOUs], subcontracts, etc.), an organization structure, and roles and responsibilities for current code custodians and other participants, as needed.

## 4.3 Software Quality Assurance

The NRC software QA approach is described in NUREG/BR-0167, *Software Quality Assurance Guidelines* (NRC 1993). As noted in the Abstract of NUREG/BR-0167, the document offers guidance to both NRC organizations and NRC contractors in the development and maintenance of software for use by NRC staff. The document was based on various industry standards applicable during its development, but it is now over 25 years old. Also, it was noted that this document may provide guidance but does not address the complex issue of software quality for software used in nuclear plants. As noted above in Section 2.2, no set QA standard has been applied to the existing suite of RAMP codes.

### 4.3.1 Software Quality Assurance Plan

Development of a new suite of RAMP codes would require a rigorous and fully documented approach to QA. This approach is typically described in a Software Quality Assurance Plan (SQAP) that defines the QA activities and identifies the documentation that shall be created and maintained during the entire software development life cycle process. The SQAP describes project requirements for how to perform tasks, such as configuration control of software while in development. The goal of the SQAP is to provide adequate confidence that the software development process is controlled, and that the software products shall meet established requirements.

Existing coding and documentation for RAMP codes that were not developed within an approved QA program would be brought into a SQAP for further development, consolidation, and modernization. In preparing a software baseline for release for use that complies with requirements in the SQAP, all software requirements would be demonstrated to be met during integration testing prior to software release. Where no prior approved baseline exists, then a comprehensive test planning and execution phase would be necessary to demonstrate that the requirements are addressed, because no credit for past requirements demonstration outside of an approved Quality Assurance Program could be taken.

Code validation is an important element of a SQAP. Proper validation of physical models encoded into analytical tools is essential to give developers the necessary guidance for developing and improving algorithms and numerical methods for describing physical processes. Moreover, validation results are essential for code users to gain confidence in applying the code to real-world applications. It is important that such validation exercises be performed objectively by developers, who may better understand the nuances of models, as well as users, who may have a more distant knowledge of the internal models but may have a greater knowledge of real-world applications.

### 4.3.2 Implementation

The procedures identified in the SQAP would be implemented in five phases:

1. Develop the SQAP and QA procedures, and train project staff to QA procedures.
2. Develop coding standards and XML naming conventions.
3. Implement procedures during software development life cycle phases.
4. Implement procedures as needed for continuing code updates and maintenance.
5. Conduct periodic audits to confirm continuing QA.

In addition to defining the procedures necessary to implement the QA program, the SQAP would define the documentation required for each step in the process of code consolidation, development, and ongoing maintenance. Documentation of compliance with the SQAP would continue as needed throughout the life of the code. Documentation would be maintained by the NRC as part of the project records and would be provided to code users as requested.

### 4.3.3 Benefits of Quality Assurance Program

The SQAP developed for a consolidated code would meet the NRC's minimum QA requirements and would provide the following additional benefits:

- **Consistent and Recognized QA Pedigree.** The SQAP would be developed under an accepted DOE program using a rigorous nuclear QA approach. This approach would be recognized as an acceptable standard by the nuclear community.
- **Continuing QA Pedigree.** Implementing the SQAP for code development would include continuing adherence to the plan during code maintenance and upgrades. NRC and RAMP users would have continuing confidence that the consolidated code meets the established QA standards for the consolidated code.
- **Easier Adoption.** The consolidated code would include QA documentation and verification tests for installation. This documentation would improve acceptance by user organizations and would provide easier methods for installation, validation, and verification.
- **Preparation for Advanced Non-LWR applications.** The consolidated code would provide a mechanism for assessing doses from anticipated non-LWR applications and would address the issues identified in NRC *Non-Light Water Reactor (Non-LWR) Vision and Strategy, Volume 4 – Licensing and Siting Dose Assessment Codes (RAMP)* (in draft). The QA program would assure non-LWR issues are addressed and rigorously tested during code development.
- **Additional User Fees to RAMP.** Implementation of a QA program would create a consolidated code that would have greater value to RAMP users than current codes. Accordingly, RAMP could charge users higher fees for an improved product. Additional fees would support continuing code improvements.

# 5.0 Code Development and Implementation

Code development and implementation would rely on development and use of an agile software development process, a software transition plan, and an early prototype to prove the concept prior to full code development.

## 5.1 Agile Software Development

For software development of user-facing applications, the current best practice is to employ an agile development process. Rather than defining all the features and requirements up-front, software is developed in an iterative fashion that features regular release cycles. Typically, this takes shape through initial user experience research (user interviews, focus groups, surveys, etc.) and interface design activities. The outcomes of these activities inform the product development cycle wherein software designs are implemented, tested with users, and further iterated upon. This process promotes adoption of the product and reduces costs by assuring that development activities are focused on features that deliver the functionality that end users need. The conceptual approach to agile software development is shown in Figure 5.1.

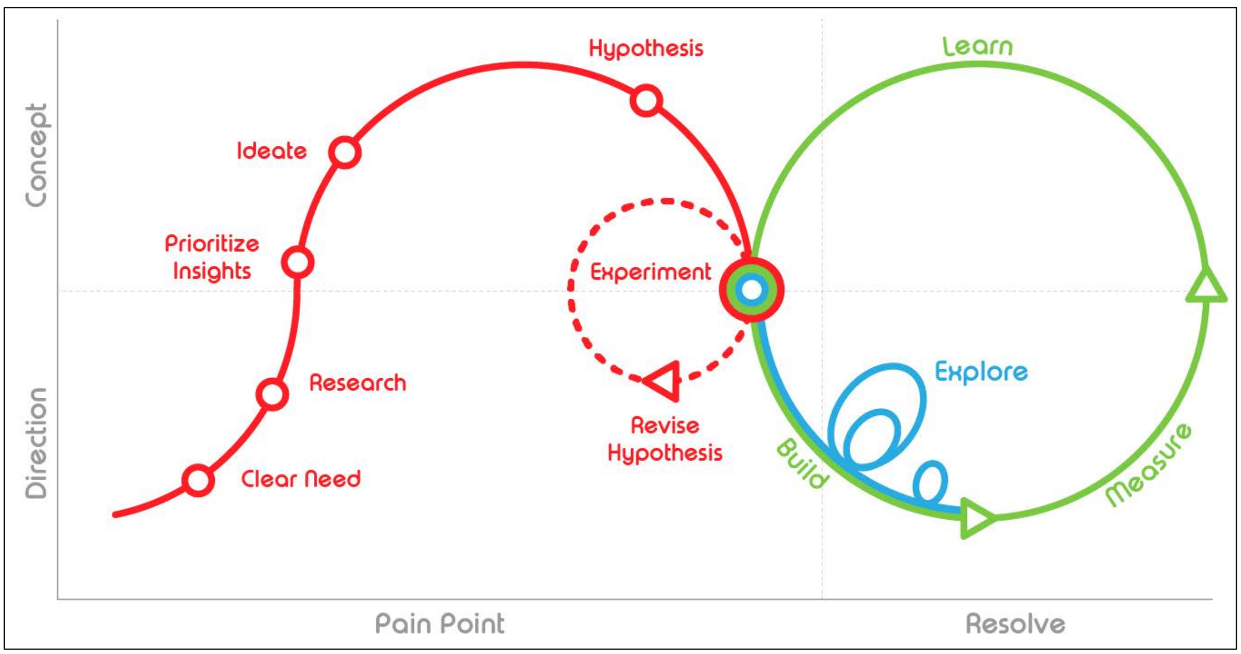



Figure 5.1. Conceptual Agile Development Process

RAMP code consolidation and modernization would use this agile development process to focus on shifting from individual code capabilities to overarching RAMP capabilities based on the use of functional engines. This code modernization approach is centered on maintaining the functional capabilities of codes rather than maintaining individual codes. This would be achieved by standing up a new framework of functions that would gradually replace the existing suite of RAMP codes.

## 5.2 Software Transition Management

Development of a consolidated code would require a transition plan to communicate the roles and responsibilities of the code modernization and consolidation program, the consolidation approach, and steps involved in transitioning from the current suite of codes to a consolidated code. RAMP currently distributes some codes free of charge and distributes other codes for a fee. Table 2.1 lists the codes considered in this consolidated code approach that are free and those that require payment of a fee. A plan for transition to a consolidated code should consider the following software management issues:

- **Continuing Regulatory Compliance.** Determine the steps necessary to transition from the current suite of codes that are designed to meet specific regulatory requirements to a consolidated code that would meet multiple regulatory requirements.
- **Code Implementation.** Determine the steps for code beta testing and incremental implementation, including defining the roles and responsibilities of owners, developers, and representative users, and the implementation schedule. Determine the viability of and schedule for releasing functional modules separately from the total code package.
- **Fee Structure.** Determine the fee structure for obtaining the code, obtaining assistance for using the code, and for supporting continuing code maintenance and upgrades.
- **Ownership, Licensing and Distribution.** Determine who owns the consolidated code, what the RAMP requirements are for licensing the code, and how the code would be distributed.
- **Support.** Determine what level of continuing code support would be provided by RAMP or the RAMP contractor to licensees after installation.
- **Documentation and User Manuals.** Determine what level of documentation and user manuals would be provided as part of licensing the consolidated code and as part of continuing support.
- **Quality Assurance.** Determine what documentation would be provided to demonstrate the QA standard was met during development and would continue to apply during maintenance and upgrades.
- **Updating and Maintenance.** Determine the roles and responsibilities for providing continued maintenance of the code.
- **User Group.** Determine the structure and membership of a consolidated code user group. Determine methods for interaction between the user group, RAMP, and the code maintenance contractor.
- **Public Engagement.** Determine the methods of and schedule for engagement with code users, industry, and other stakeholders during code development.
- **Advanced Computing.** Determine options for future use of advanced computing options such as cloud-based computing and allowing for mobile access.

## 5.3 Phased Code Development and Implementation

Development of a modern consolidated code or suite of codes would require development of a project plan implemented in multiple phases over several years. For initial planning purposes, three project phases would consist of the following general elements:

1. Phase 1: Project Planning

   a. Develop a Code Development Project Plan.
   b. Develop a Software Transition Plan.
   c. Develop a SQAP and QA procedures.
   d. Develop and implement MOUs or other stakeholder agreements.
   e. Develop a knowledge management site for developer collaboration.
2. Phase 2: Code Development
   a. Train staff and implement QA procedures.
   b. Develop Code Requirements.
      i. Develop a Regulatory Requirements Document.
      ii. Develop Non-LWR Requirements Document.
      iii. Develop Coding Standards and Naming Conventions Document.
   c. Develop a functional prototype.
      i. Develop an atmospheric dispersion functional engine.
      ii. Develop a functional UI.
      iii. Conduct initial beta testing.
   d. Develop a complete working consolidated code.
   e. Validate and verify the consolidated code.
   f. Disseminate codes to the user group for beta testing.
   g. Develop a User Manual, test cases, and other documentation.
3. Phase 3: Code Implementation, Dissemination, Maintenance, and Continuing Development
   a. Release the consolidated code through RAMP.
   b. Establish a consolidated code user group.
   c. Develop mechanisms for continuing code maintenance and development.

# 6.0 Conclusions and Recommendations

The RAMP suite of codes was originally developed nearly 40 years ago and has had minimal incremental maintenance completed since then. These codes have numerous current and legacy issues that reduce the efficiency of operation and maintenance of the codes and increase the associated costs. In their current state, the codes are also unable to fully assess radiological doses from advanced non-LWR designs.

The current and legacy issues could be addressed by transforming the suite of single-purpose radiation protection and dose assessment computer codes into a consolidated functional and modern suite of codes that is modular, flexible, efficient, and user-friendly. The proposed approach would develop a single code that would consolidate the calculations performed by the current suite of RAMP codes and modernize the code language, data transfer methods, and UI. The consolidated code would be developed in a new framework that relies on three general concepts:

- **Consolidate.** Create a single consolidated code that has functional modules—or engines—that perform the same calculations as the current suite of RAMP codes.
- **Standardize.** Establish a standardized XML schema for data transfer into and out of the code, and between engines within the code.
- **Separate the UI.** Establish a single user-friendly interface that would be separate from the functional engines.

A modern consolidated code would address the following current and legacy issues:

- **Inefficiency of having and maintaining multiple codes.** A consolidated code would reduce ongoing upgrade and maintenance costs.
- **Functional redundancy between codes.** A consolidated code would allow the NRC to meet multiple regulatory needs with one code.
- **Dated science within the codes.** A modernized code would incorporate current scientific approaches and would be flexible for incorporating future changes.
- **Rigid and dated data transfer methods.** A modernized code would incorporate standard data transfer methods to be flexible for accommodating future changes in data inputs, data transfer between engines, and output reports.
- **Lack of standardized code writing.** The core code language would be written using more efficient, modern code approaches.
- **Limited ability to assess advanced non-LWR designs.** A modernized code would address known technical challenges with non-LWR designs and would be flexible enough to allow for unique future inputs and computational needs.
- **The history of changing ownership and associated loss of code knowledge.** A consolidated code would be owned and maintained by the NRC.
- **Inconsistent maintenance.** A consolidated code would eliminate the potential for multiple inconsistent maintenance approaches.
- **Lack of standardized QA.** A consolidated code would be developed and continually maintained under a QA program developed using a graded approach.

The RAMP codes have been developed and continue to be developed and maintained by the NRC, DOE national laboratories, EPA, and other federal agencies, universities, and commercial contractors. In addition, international entities (for example IMBA) have relevant expertise in radiological code development. Input from supporting participants would be important in developing a plan for transitioning from the current suite of RAMP codes to a consolidated code. A consolidated code approach would also be able to capture the calculations performed in some health physics codes used by other federal agencies such as EPA codes CAP88 and COMPLY. A comprehensive project plan should include engagement mechanisms (e.g., MOUs, subcontracts, etc.), an organization structure, and roles and responsibilities for current code custodians and other participants, as needed.

*NRC Non-Light Water Reactor (Non-LWR) Vision and Strategy, Volume 4 – Licensing and Siting Dose Assessment Codes (RAMP)* (in draft) recommends that the individual RAMP codes be independently evaluated by the code's current custodian relative to various non-LWR technologies to provide the NRC with a roadmap for code updates and changes necessary to support non-LWR licensing. Recommendations that are developed during individual code evaluations should be considered in the development of the final approach to code consolidation and modernization.

Early development of a prototype is recommended to demonstrate the proof-of-concept and guide subsequent full code development. The goal of a prototype would be to demonstrate the three pillars of the code consolidation and modernization approach:

1. Demonstrate how one core functional engine can perform the needed function and operate separately from the UI.
2. Demonstrate the use of the XML data transfer schema.
3. Demonstrate the functionality and experience of the new UI.

# 7.0 References

10 CFR Part 20. Code of Federal Regulations, Title 10, Energy, Part 20, "Standards for Protection Against Radiation." U.S. Nuclear Regulatory Commission.

10 CFR Part 20, Subpart E. Code of Federal Regulations, Title 10, Energy, Part 20, "Standards for Protection Against Radiation," Subpart E, Radiological Criteria for License Termination. U.S. Nuclear Regulatory Commission.

10 CFR Part 50. Code of Federal Regulations, Title 10, Energy, Part 50, "Domestic Licensing of Production and Utilization Facilities." U.S. Nuclear Regulatory Commission.

10 CFR Part 50 Appendix A. Code of Federal Regulations, Title 10, Energy, Part 50, "Domestic Licensing of Production and Utilization Facilities," Appendix A, General Design Criteria for Nuclear Power Plants. U.S. Nuclear Regulatory Commission.

10 CFR Part 50 Appendix E. Code of Federal Regulations, Title 10, Energy, Part 50, "Domestic Licensing of Production and Utilization Facilities," Appendix E, Emergency Planning and Preparedness for Production and Utilization Facilities. U.S. Nuclear Regulatory Commission.

10 CFR Part 50 Appendix I. Code of Federal Regulations, Title 10, Energy, Part 50, "Domestic Licensing of Production and Utilization Facilities," Appendix I, Numerical Guides for Design Objectives and Limiting Conditions for Operation To Meet the Criterion “As Low as is Reasonably Achievable” for Radioactive Material in Light-Water-Cooled Nuclear Power Reactor Effluents. U.S. Nuclear Regulatory Commission.

10 CFR Part 51. Code of Federal Regulations, Title 10, Energy, Part 51, “Environmental Protection Regulations for Domestic Licensing and Related Regulatory Functions.” U.S. Nuclear Regulatory Commission.

10 CFR Part 52. Code of Federal Regulations, Title 10, Energy, Part 52, "Licenses, Certifications, and Approvals for Nuclear Power Plants." U.S. Nuclear Regulatory Commission.

10 CFR Part 71. Code of Federal Regulations, Title 10, Energy, Part 71, "Packaging and Transportation of Radioactive Material." U.S. Nuclear Regulatory Commission.

10 CFR Part 100. Code of Federal Regulations, Title 10, Energy, Part 100, Reactor Site Criteria.” U.S. Nuclear Regulatory Commission.

81 FR 47443. July 21, 2016. “NRC Vision and Strategy for Non-Light Water Reactor Mission Readiness.” Federal Register, U.S. Nuclear Regulatory Commission.

Arcieri, W.C., D.L. Mlynarczyk, and L. Larsen. 2016. SNAP/RADTRAD 4.0: Description of Models and Methods. NUREG/CR-7220, U.S. Nuclear Regulatory Commission, Washington, D.C.

EPA (U.S. Environmental Protection Agency). 1989. *User’s Guide For the COMPLY Code (Revision 2)*. EPA 520/1-89-003, Washington, D.C.

EPA (U.S. Environmental Protection Agency). 2013. *CAP88 Version 3.0 User Guide*. Washington, D.C.

Napier, B.A., D.L. Strenge, J.V. Ramsdell, P.W. Eslinger, and C. Fosmire. 2011. *GENII Version 2 Software Design Document*. PNNL-14584, Revision 3, Pacific Northwest National Laboratory, Richland, Washington.

NRC (U.S. Nuclear Regulatory Commission). 1977a. *Calculation of Annual Doses to Man from Routine Releases of Reactor Effluents for the Purpose of Evaluating Compliance with 10 CFR Part 50, Appendix I*. Regulatory Guide 1.109, Revision 1, Washington, D.C.

NRC (U.S. Nuclear Regulatory Commission). 1977b. *Estimating Aquatic Dispersion of Effluents from Accidental and Routine Reactor Releases for the Purpose of Implementing Appendix I*. Regulatory Guide 1.113, Revision 1, Washington, D.C.

NRC (U.S. Nuclear Regulatory Commission). 1977c. *Methods for Estimating Atmospheric Transport and Dispersion of Gaseous Effluents in Routine Releases from Light-Water-Cooled Reactors*. Regulatory Guide 1.111, Revision 1, Washington, D.C.

NRC (U.S. Nuclear Regulatory Commission). 1993. *Software Quality Assurance Program and Guidelines*. NUREG/BR-0167, Washington, D.C.

NRC (U.S. Nuclear Regulatory Commission). 2003. *Atmospheric Relative Concentrations for Control Room Radiological Habitability Assessments at Nuclear Power Plants*. Regulatory Guide 1.194. Washington, D.C.

NRC (U.S. Nuclear Regulatory Commission). 2007a. *Combined License Applications for Nuclear Power Plants*. Regulatory Guide 1.206, Revision 0, Washington, D.C.

NRC (U.S. Nuclear Regulatory Commission). 2007b. *Meteorological Monitoring Programs for Nuclear Power Plants*. Regulatory Guide 1.23, Revision 1. Washington, D.C.

NRC (U.S. Nuclear Regulatory Commission). 2007c. *Short-Term Atmospheric Dispersion Estimates for Accident Releases.* Standard Review Plan 2.3.4, NUREG-0800, Washington, D.C.

NRC (U.S. Nuclear Regulatory Commission). 2014. *The Radiation Protection Computer Code Analysis and Maintenance Program*. SECY-14-0117. Washington, D.C.

NRC (U.S. Nuclear Regulatory Commission). 2016. *NRC Vision and Strategy: Safely Achieving Effective and Efficient Non-Light Water Reactor Mission Readiness*. ML16356A670. Washington, D.C.

NRC (U.S. Nuclear Regulatory Commission). 2018. *Standard Review Plan for Evaluating Nuclear Power Reactor License Termination Plans*. NUREG-1700, Revision 2, Washington, D.C.

Ramsdell, J.V. and C.A. Simonen, Jr. .1997. *Atmospheric Relative Concentrations in Building Wakes*. NUREG/CR-6331, U.S. Nuclear Regulatory Commission, Washington, D.C.

# Appendix A – NRCDose3 Consolidation

NRCDose3 contains the codes endorsed by the U.S. Nuclear Regulatory Commission (NRC) for evaluating exposure pathway doses from reactor effluents to demonstrate compliance with Title 10 of the *Code of Federal Regulations* Part 20 (10 CFR Part 20), Appendix I to 10 CFR Part 50, 40 CFR Part 190, and 10 CFR Part 51. NRCDose3 applies the models from NRC Regulatory Guide (RG) 1.109 Revision 1, *Calculation of Annual Doses to Man from Routine Releases of Reactor Effluents for the Purpose of Evaluating Compliance with 10 CFR Part 50, Appendix I* (NRC 1977a); RG 1.111, Revision 1, *Methods for Estimating Atmospheric Transport and Dispersion of Gaseous Effluents in Routine Releases from Light-Water-Cooled Reactors* (NRC 1977c); and RG 1.113, Revision 1, *Estimating Aquatic Dispersion of Effluents from Accidental and Routine Reactor Releases for the Purpose of Implementing Appendix I* (NRC 1977b).

NRCDose3 was added to the Radiation Protection Computer Code Analysis and Maintenance Program (RAMP) suite of codes in 2019 after a significant version update and is a graphical user interface (GUI) for the XOQDOQ, GASPAR II, and LADTAP II Fortran codes. The update included a Windows GUI, an expanded radionuclide library including 203 radionuclides with dose conversion factors (DCFs), and user-modifiable parameters for bioaccumulation factors, consumption rates, usage factors and other parameters. NRCDose3 allows the user to select DCFs from International Commission on Radiological Protection Publication 2 (ICRP-2; the default), ICRP-30, or ICRP-72 (for six age groups). NRCDose3 also added the ability to calculate dose to nonhuman biota in GASPAR, which previously was not available.

Although NRCDose3 is more user-friendly than the previous versions of NRCDose or the stand-alone codes XOQDOQ, GASPAR II, and LADTAP II, the updated version still does not address needs for models to support the reactor licensing process for safety and environmental reviews for non-light water reactors (non-LWRs). In addition, there are still some inadequacies in NRCDose3, such as the need to enter each radionuclide and quantity individually when building the source term.

With the consolidation and modernization of NRCDose3, the remaining issues related to using the codes would be resolved, such as providing for easy transfer of data like the source term into the GASPAR II and LADTAP II codes. Also, as information about non-LWR designs becomes available and the inadequacies of dose modeling are identified, the consolidated codes would be modified and improved to meet any new guidance, current regulations, or changes to regulations.

Although the look of the consolidated code for NRCDose3 would change, it would still have a user-friendly GUI that would walk the user through obtaining or entering the data needed for running XOQDOQ (or the atmospheric code that would implement RG 1.111 [NRC 1977c]), GASPAR, and LADTAP. As part of the consolidation, rather than having multiple codes that perform atmospheric dispersion, these codes would be among those consolidated, but would still implement the modeling provided in the guidance in RG 1.111 (NRC 1977c).

Other dispersion codes that may have different guidance, would use the different consolidated engines shown in Figure 3.3, as chosen by the user.

Because NRCDose3 has recently been revised, the code language in GASPAR II and LADTAP II should be reviewed for potential inclusion in functional engines developed as part of the consolidated code. Depending on the language chosen for the functional engines, some of the existing GASPAR II and LADTAP II code could be imported directly.

The gaseous and liquid effluents (GALE) series of codes in RAMP calculate the gaseous and liquid effluent releases from pressurized-water reactors and boiling-water reactors. These releases are currently used as input to NRCDose3. *NRC Non-Light Water Reactor (Non-LWR) Vision and Strategy, Volume – Licensing and Siting Dose Assessment Codes (RAMP)* (in draft), recognizes the need for existing codes, including the GALE codes, to be updated to address the core radionuclide inventory, fuel form, geometry, and other relevant characteristics of advanced non-LWRs. Because any updates to the GALE codes for non-LWR designs are not included in the initial consolidation of codes, the source term and release mechanisms calculated in a new non-LWR version of GALE would be an XML input to the consolidated code, as shown in Figure 3.3.

The code consolidation for NRCDose3 would allow the user to apply the codes for purposes other than regulatory compliance, including accessing data such as the DCFs in ICRP 72, which may not be used for licensing. A user would be able to easily adjust input values for research or other uses, such as comparing results from non-NRC-developed and -approved codes.

## A.1 References

10 CFR Part 20. *Code of Federal Regulations*, Title 10, *Energy*, Part 20, "Standards for Protection Against Radiation." U.S. Nuclear Regulatory Commission.

10 CFR Part 50 Appendix I. *Code of Federal Regulations*, Title 10, *Energy*, Part 50, "Domestic Licensing of Production and Utilization Facilities," Appendix I, *Numerical Guides for Design Objectives and Limiting Conditions for Operation To Meet the Criterion “As Low as is Reasonably Achievable” for Radioactive Material in Light-Water-Cooled Nuclear Power Reactor Effluents*. U.S. Nuclear Regulatory Commission.

10 CFR Part 51. *Code of Federal Regulations*, Title 10, *Energy*, Part 51, “Environmental Protection Regulations for Domestic Licensing and Related Regulatory Functions.” U.S. Nuclear Regulatory Commission.

40 CFR Part 190. *Code of Federal Regulations*, Title 40, *Protection of Environment*, Part 190, "Environmental Radiation Protection Standards for Nuclear Power Operations." U.S. Environmental Protection Agency.

NRC (U.S. Nuclear Regulatory Commission). 1977a. *Calculation of Annual Doses to Man from Routine Releases of Reactor Effluents for the Purpose of Evaluating Compliance with 10 CFR Part 50, Appendix I*. Regulatory Guide 1.109, Revision 1, Washington, D.C.

NRC (U.S. Nuclear Regulatory Commission). 1977b. *Estimating Aquatic Dispersion of Effluents from Accidental and Routine Reactor Releases for the Purpose of Implementing Appendix I*. Regulatory Guide 1.113, Revision 1, Washington, D.C.

NRC (U.S. Nuclear Regulatory Commission). 1977c. *Methods for Estimating Atmospheric Transport and Dispersion of Gaseous Effluents in Routine Releases from Light-Water-Cooled Reactors*. Regulatory Guide 1.111, Revision 1, Washington, D.C.

# Appendix B – The Graphical User Interface

The graphical user interface (GUI) is part of the human-computer interface (i.e., UI) and is what the users see, touch, control, and manipulate. It typically consists of icons and widgets, functions to process user and application events, associations between specific users and specific functions, and an event-loop to process user events. UI elements typically include input controls, navigation components, information components, and containers. Design principles for the UI include simplicity, consistency, purposefulness in layout, strategic use of color and texture, use of typography to create hierarchy and clarity, and feedback responsiveness to user action (HHS n.d.). Influential design principles such as those used in Apple iOS emphasize aesthetic integrity (appearance and behavior integrate with function), consistency (standard icons, text styles, terminology), direct manipulation (onscreen content engaging users), feedback (perceptible feedback responding to user action), metaphors (using metaphors of familiar experience to allow users to physically interact with the screen), and user control (app designed to maintain user sense of control) (Apple, Inc. n.d.).

As Table B.1 shows, the development of a UI consists of the development of the front end (GUI or client side) and back end (server side). Front-end development consists of web design and client-side development (static content and/or dynamic web application). The back end handles databases, data storage, client-side event execution, and output transmission.

Table B.1. Common UI Tools and Components

| Components | | Function | Common Tools |
|---|---|---|---|
| Front End | Web Design | Look and feel | InVision, Adobe XC, User Testing, UXPin, etc. |
| | Front-End Development | Client-side functionality including implementing web design | HTML, CSS, JavaScript, frameworks such as Angular, Boostrap, JavaScript libraries such as REACT and CSS extensions |
| Back End | Back-End Development | Implemented on server; storing, organizing data, executing client-side events and returning information interpretable for the front end | Ruby, PHP, Java, .Net, and Python |
| | Database | Structuring and storing data including data generated from dynamic web application/user inputs | MySQL, SQL Server, PostgreSQL, Oracle |

## B.1 Preliminary Attributes of GUI Design

Based on the user functional requirements described in Section 3.2.3, the key attributes of the RAMP GUI are captured in Table B.2. These key attributes would guide the design, build, and implementation of the GUI through an agile development process.

These key attributes require extensive interaction between the user and the GUI. These interactions may range from logging into the application, defining the input values about applicable compliance requirements, and determining parameters for output, to viewing the output from the GUI. Figure 3.8 (in Section 3.2.3) depicts the high-level workflow in which a user interacts with the GUI and the functional engines. After logging in, the user defines a task for the functional engines by defining input values and selecting the model(s). The GUI communicates with the appropriate functional engines to execute calculations based on user input. The functional engines output data and return the results to the GUI. The results are then displayed on the screen for the user. The user can view the results, view the geospatial visualization based on the results, and export the results (e.g., as XML, CSV, Excel, text).

Table B.2. RAMP GUI Attributes

| Attributes | Description |
|---|---|
| Secure web login (if GUI is web-based) | The UI allows users to request and create a secure account and use a secure password to log in (e.g., two-factor authentication). |
| Role-based access control and privileges | The GUI provides differential access and privileges to different types of users:<br>• Basic users follow scripted instructions to enter inputs, execute calculations, and view the results.<br>• Advanced users have more complex input fields and more options for executing calculations. |
| Decoupled/loosely coupled front end and back end | The GUI is separated from the data module and functional engines and is the single overarching GUI for all the consolidated code.<br>The GUI and the back end might use the same or different language(s), but the two sides use an agreed-upon method for access, format transport, and data transmission and transformation.<br>Updates to the GUI would have no impact on the data module or functional engines. Updates to the data module and functional engines would not affect the GUI as long as the data access, transmission, and transformation remain mutually workable. |
| User-defined and dialog-prompted model selection and calculation | Input values from users prompted by the GUI dialog determine which model(s) among the competing models should be used to render the calculations. Users can also define desired output.<br>• For once-a-year users, the GUI prompts users to answer a series of questions and directs users to the applicable regulatory compliant options. The GUI displays relevant inputs based on their answers to compliance questions and for each functional engine that is selected.<br>• Basic users enter inputs to execute the calculation to obtain their required outputs.<br>• Advanced users can explore more detailed input options available with each functional engine. |
| Ease of use | The GUI is intuitive, easy to use, does not require a steep learning curve, and can enable both experienced and next-generation users to easily produce desired output. The GUI can be maintained, updated, and version controlled easily. |

## B.2 Draft GUI Mockup

To illustrate the work process, a preliminary GUI mockup design has been developed to provide an artistic representation of the GUI design vision. Note the figures presented below are not an actual GUI design, and the examples as well as results shown are examples and do not represent actual models or calculations. The purpose of the figures is to convey the look and feel that could be created for the GUI. The actual design would be based on elicited input and preferences from decision-makers and stakeholders. The vision for the GUI could evolve as the consolidation/modernization effort progresses. The agile development process would enable the development team to gather timely feedback and implement modifications to the design to meet the needs of the decision-makers, stakeholders, and users.

### B.2.1 Regulatory Compliance Example with GASPAR

As shown in Figure B.1, after logging into the RAMP GUI, the user is presented with two main assessment options: Regulatory Compliance or Research and Other. The Regulatory Compliance option directs users to a screen listing existing RAMP codes. This option is suited for users who have familiarity with RAMP codes and have clear plans for specific calculations they wish to run.


Figure B.1. Example RAMP Login Screen

After selecting the Regulatory Compliance option from the login screen, the user is directed to a code menu. The menu lists all existing RAMP codes. Each code is represented as a clickable button. Code associated with NRCDose (e.g., LADTAP, GASPAR, XOQDOQ) is organized under NRCDose on the menu. In the example below (Figure B.2), the user chooses GASPAR under NRCDose.


Figure B.2. Example RAMP Regulatory Compliance Code Menu

Then, the user is directed to the GASPAR code page. The screen asks the user to provide information about the project for which the user plans to run the code. In the example below, the screen prompts the user to enter a name for the project and the dose factor of interest (Figure B.3).

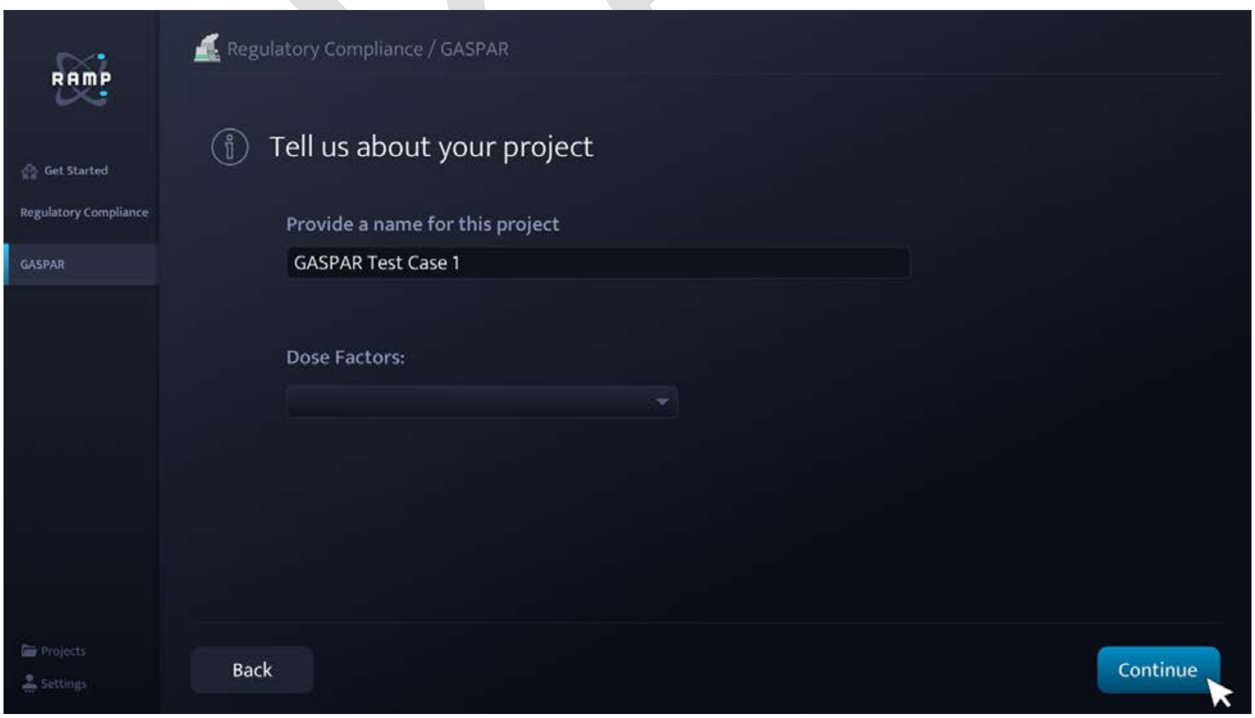



Figure B.3. Example of GASPAR User Input

After the project name and dose factor are specified, the user is directed to the Options screen under GASPAR (Figure B.4) where the user can choose to calculate individual doses only, print cumulative dose reports only, print dose-factor library data, or perform calculations, etc. The user is also prompted to enter values for various input parameters, such as the fractions characterizing the year for which the calculation would be done, the relative humidity, and the average temperature throughout the growing season. When the appropriate input fields are filled in, the user can click on the Continue button to proceed to the Source Term screen.

Figure B.4. Example of GASPAR Options

On the Source Term screen, the user can determine the source term to be used in the calculation (Figure B.5). The user can enter the value for source multiplication factor and release time for purges. The Source Term menu on the screen lists applicable source terms.


Figure B.5. Example GASPAR Source Term Selection

In addition to the readily listed source terms, the user can also add nuclides to the Source Term menu or clear all the listed source teams with a single click. The user also would be able to upload the source term output from GALE.

By clicking through the technical input tabs under GASPAR on the navigation menu on the left side of the screen, the user can similarly specify input values for parameters related to Population/Production, Location, Dose, Pathway, and Consumption. Then the user can proceed to calculate the desired results by clicking the Continue button. The GUI would communicate the user's command to the back-end functional engine(s) to execute the calculation using user-defined input values (Figure B.6).

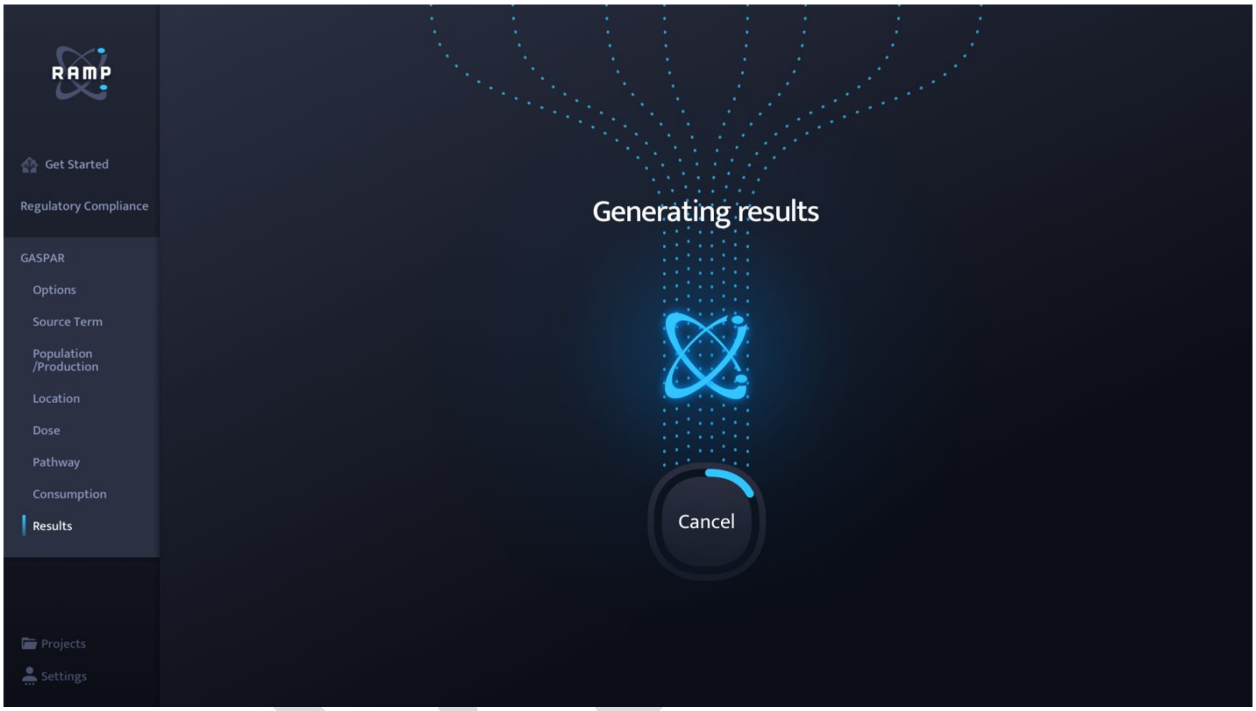



Figure B.6. Example of RAMP GUI Result Generation

The functional engines would render the calculation and return the results to the GUI. As the example GASPAR results show (Figure B.7), the GUI would display GASPAR output values in tables, including plume exposure due to noble gases and doses from gaseous effluents. The user can view additional results such as the FSAR (Final Safety Analysis Report) and Supplemental Report by clicking on the tabs displayed on the screen. The user can also print or export the displayed results.

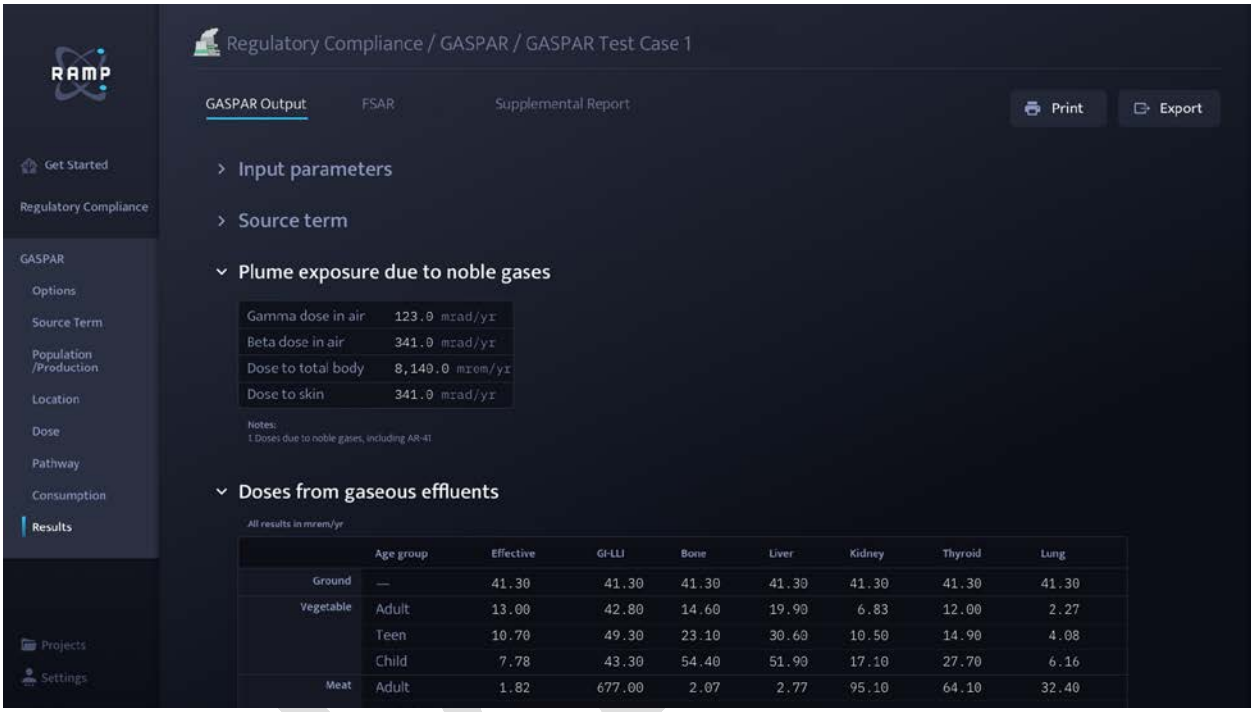



Figure B.7. Example of GASPAR Results

### B.2.2 Using GUI for Research and Other Purposes

The Regulatory Compliance option, as described above, is intended for users who have technical familiarity with RAMP codes and plan to run specific calculations. For users who are less familiar with individual RAMP codes and are interested in obtaining results to address noncompliance-driven research questions, the GUI offers the Research and Other option. Instead of focusing on individual RAMP codes, this option prompts the user to answer a series of questions and uses the answers to characterize the project of interest and to automatically select the most suitable RAMP code(s) to execute the calculation based on the user's input. In the example shown below, we illustrate how the RAMP GUI could be designed to support research and other purposes. Please note the design, example, values, and results shown in this subsection, as elsewhere in this appendix, are an artistic representation and are nominal in nature.

After a user logs into the home page, the user chooses the Research and Other option from the screen (Figure B.8).


Figure B.8. Example of RAMP Research and Other Option

On the new screen (Figure B.9), the user is prompted to enter a name for his/her project and specify the type of exposure to which the project is related (e.g., chronic or acute).

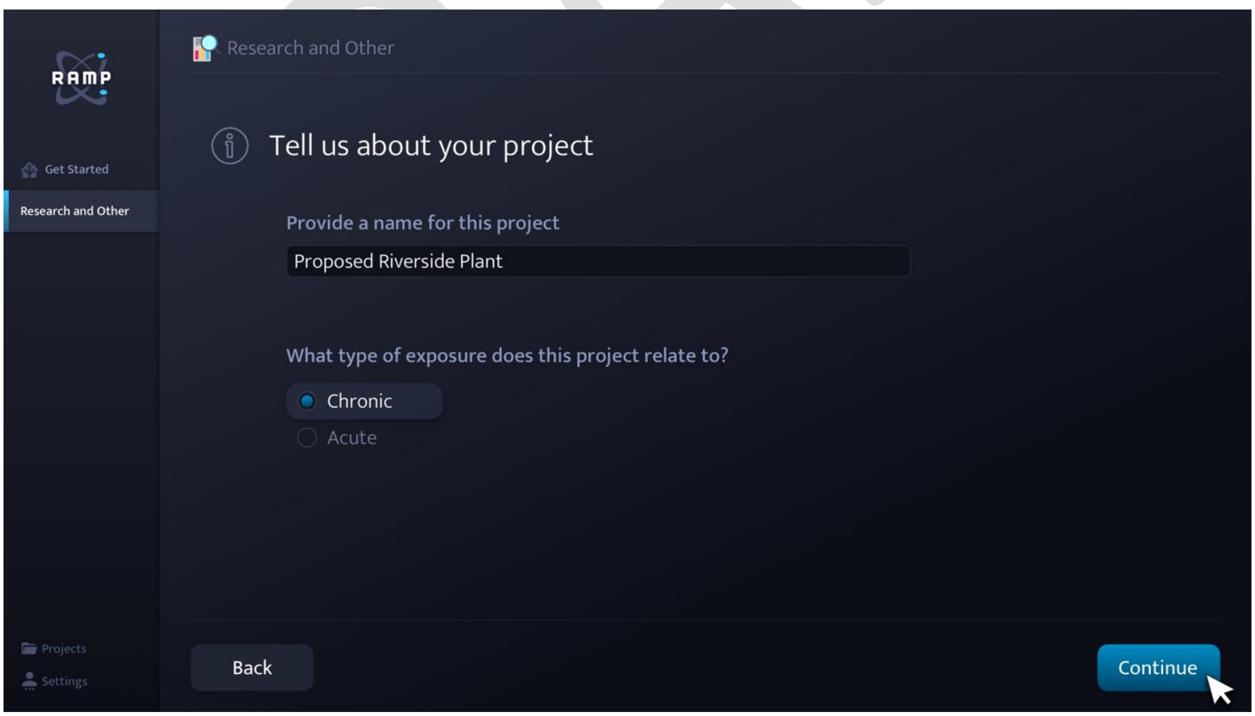



Figure B.9. Example of Additional Research and Other Options Menu

Next, the user is prompted to specify from a list of options the data that are available (Figure B.10). For each data option chosen, the GUI displays the associated available outputs.

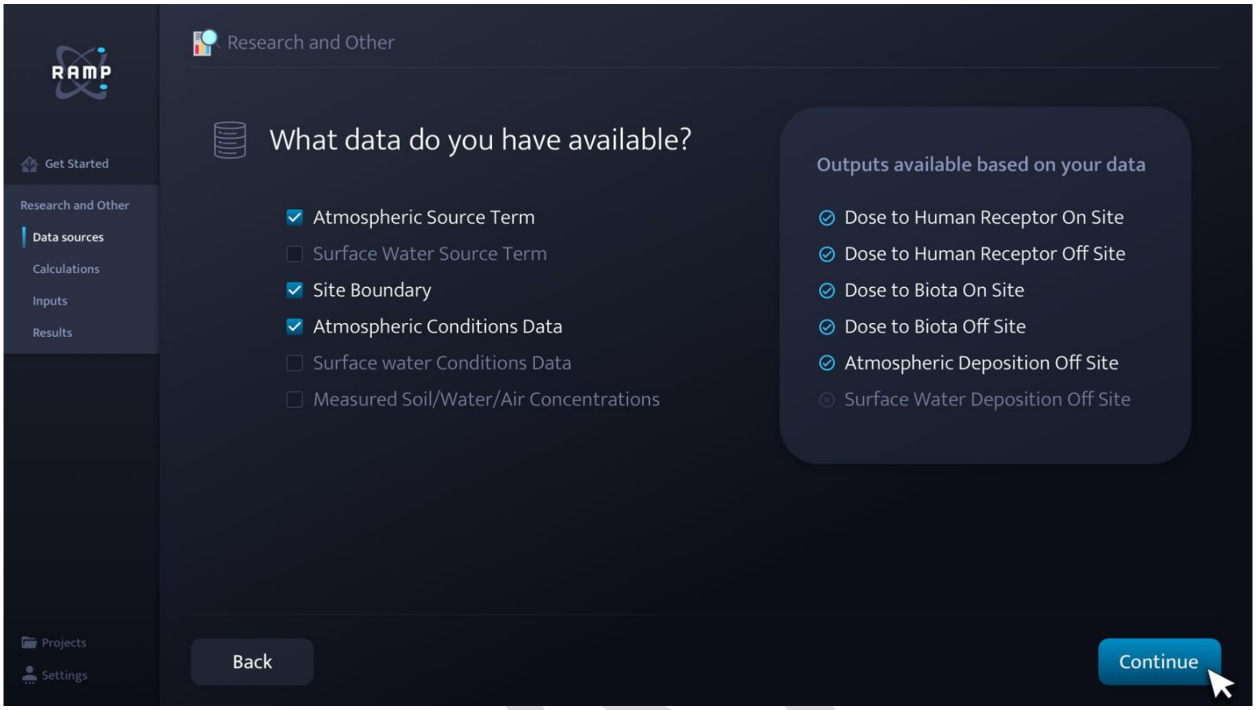



Figure B.10. Example of Research and Other Input Data Options

By clicking on the Continue button, the user proceeds to the next screen to choose desired output options (Figure B.11).

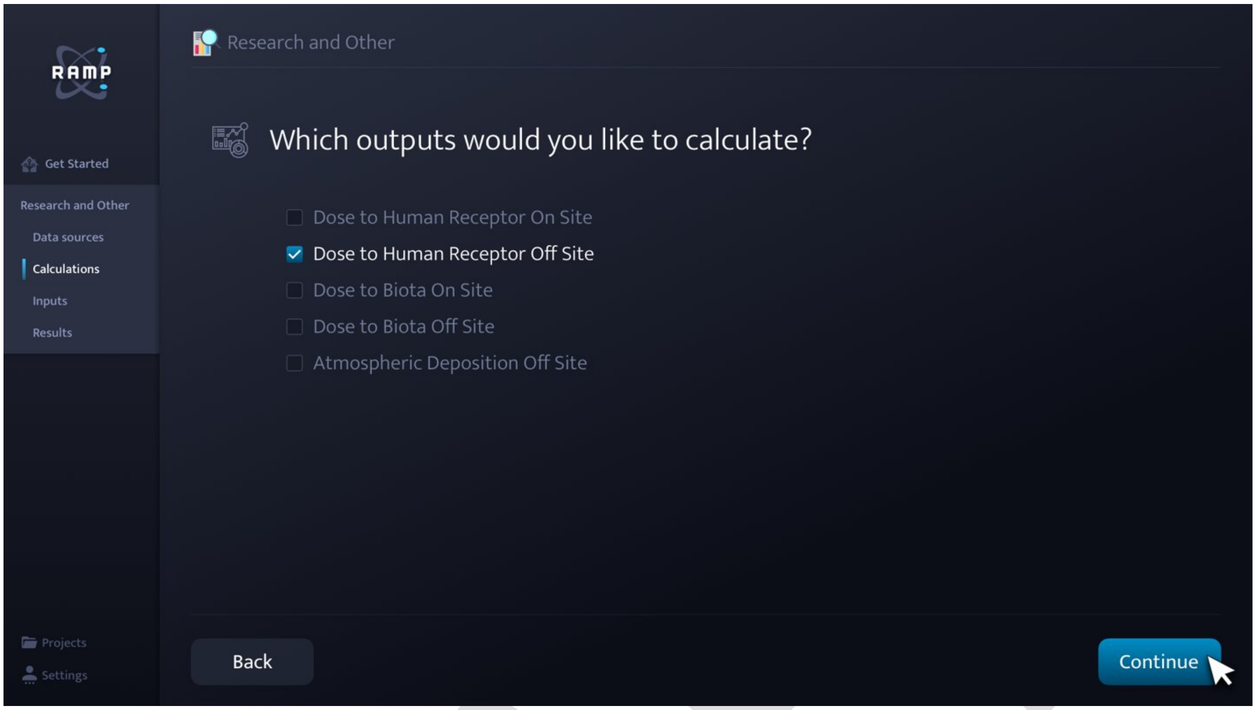



Figure B.11. Example of Research and Other Output Options

In addition, the user can click on the additional technical tabs in the navigation menu on the left side of the screen. For instance, the user can click on Inputs and go to the Inputs page where he/she can specify additional input values such as Source Term, Site Boundary, Atmospheric Release Conditions, and Atmospheric Conditions Data. When done, the user can proceed to calculate the results by clicking the Continue button. The GUI then transmits the user's command with user-specified inputs to the back-end functional engine, and the functional engine executes the calculation and returns the results to the GUI.

The results can be displayed under several tabs on the screen, including the raw output and output embedded in geospatial visualization (Figure B.12). The geospatial visualization allows the user to select any location of interest on the map of a site and drill down on the location to view site-specific information such as dose or meteorology. In addition, the user can download or export the results. Please note the example, values, and results shown in this graphic are nominal in nature and do not portray the models and results based on the actual RAMP codes.

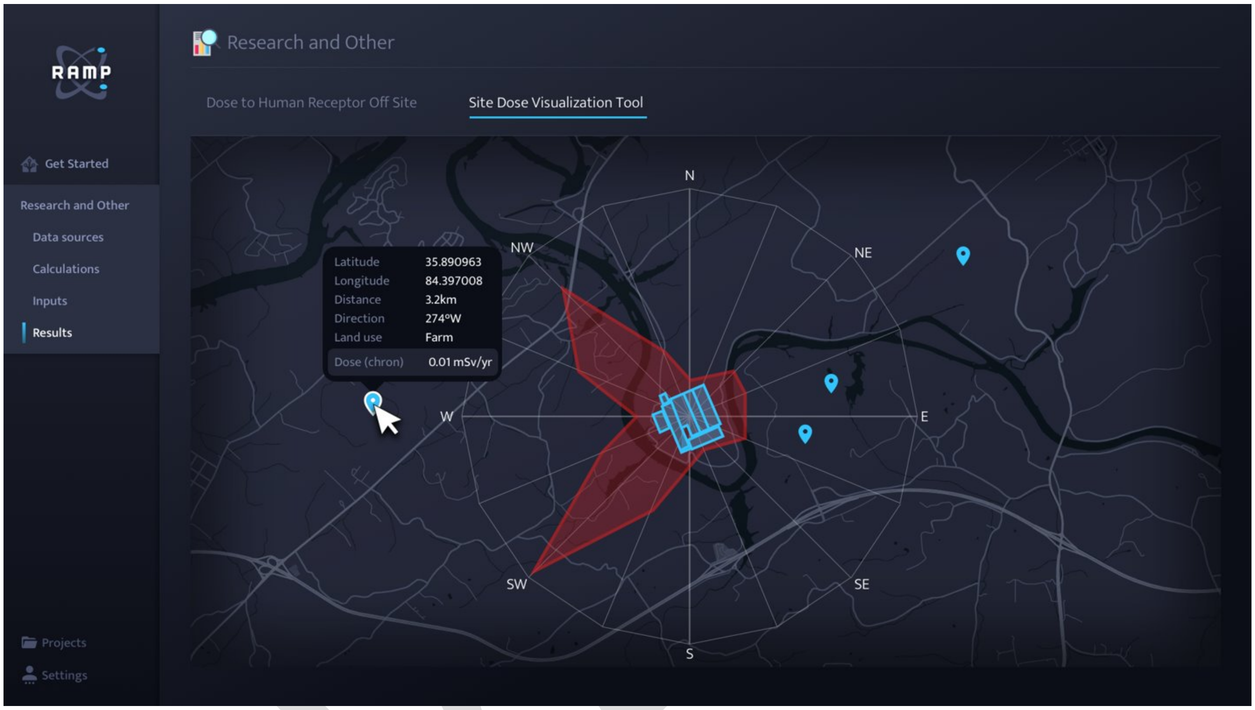



Figure B.12. Example Research and Other Results and Visualization

The design philosophy underlying the GUI is to provide nuanced capabilities to users who have different needs in an easy-to-use and intuitive way. The above examples are meant to provide readers with a primary look and feel for the GUI. The actual design and agile development would be responsive to the perspectives and needs of decision-makers, stakeholders, and users.

## B.3 References

Apple, Inc. n.d. “Human Interface Guidelines iOS Design Themes.” Retrieved from https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/ (Feb 4, 2020).

HHS (U.S. Department of Health and Human Services). n.d. “User Interface Design Basics.” Retrieved from https://www.usability.gov/what-and-why/user-interface-design.html (Feb 2, 2020).

**Pacific Northwest National Laboratory**

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354
1-888-375-PNNL (7665)

***www.pnnl.gov***