RESEARCH REPORT

VTT-R-00241-13

Quality assurance of a safety analysis software

Author: Tero Tyrväinen, Teemu Mätäsniemi

Confidentiality: Public

Report's title	
Quality assurance of a safety analysis software	
Customer, contact person, address Valtion ydinjätehuoltorahasto	Order reference 12/2012SAF
Project name FinPSA knowledge transfer	Project number/Short name 77288/FinPSA-transfer
Author(s) Tero Tyrväinen, Teemu Mätäsniemi	Pages 16/
Keywords Quality assurance, safety analysis software, probabilistic risk analysis	Report identification code VTT-R-00241-13

Summary

The performance of probabilistic risk analysis (PRA) is dependent on the use of computer codes dedicated to modelling and quantification of complex mathematical models. Due to the role of PRA in safety decision making, it is more than desirable to ensure the quality of the PRA computer code. This report gives an overview of quality assurance requirements which can be relevant to safety and risk analysis tools that are applied in the domain of nuclear power plant safety.

Confidentiality	Public

Espoo 11.1.2013

Written by	Reviewed by	Accepted by
Tero Tyrväinen Research Scientist	Jan-Erik Holmberg Team Leader	Jari Hämäläinen Technology Manager

VTT's contact address

Distribution (customer and VTT)
SAFIR TR8, FinPSA end user group

Contents

1 Introduction

The performance of probabilistic risk analysis (PRA) [1] is dependent on the use of computer codes dedicated to modelling and quantification of complex mathematical models. Due to the role of PRA in safety decision making, it is more than desirable to ensure the quality of the PRA computer code.

Quality means broadly the fitness of purpose of a product or a service. However, different user groups can require different properties from the same product. For instance, PRA analysts and the maintenance team can have different requirements for the software and these requirements can be conflicting. One of the key properties in the context of PRA software is the ability to quantify a reliability model and identify minimal cut sets correctly, but PRA computer codes have also other important quality properties such as user-friendliness and functional properties such as data base management.

Quality assurance (QA) can be defined as all those actions that provide confidence that quality is achieved [2]. This report gives an overview of quality assurance requirements which can be relevant to nuclear power plant safety analysis tools like PRA computer codes. The purpose is to provide a background study for QA of the Finnish PRA code FinPSA [3]. In this document, PRA can be understood to cover all the levels of PRA 1, 2 and 3.

The structure of the report is as follows. Chapter 2 gives the main definitions. Chapter 3 summarises the literature review by presenting the main QA and PRA references. Chapter 4 presents overall quality assurance requirements, while Chapter 5 concludes the report.

The report focuses on the QA of the PRA tool development process and QA of the PRA tool (the product). QA of the PRA modelling process and the PRA model are out of the scope. To some extent, same QA principles are however applicable for the PRA modelling, as well.

2 Definitions

Software: Computer programs, procedures, and associated documentation and data pertaining to the operation of a computer system. [4]

Software tool: A computer program used in the development, testing, analysis or maintenance of a program or its documentation. [5]

Safety software: Safety system software, safety or hazard analysis software or design software, or safety management or administrative control software. [6]

Safety analysis software: Software that is used to analyse nuclear facilities. This software is not part of a structure, system or component but helps to ensure proper accident analysis of nuclear facilities or a structure, system or component that performs a safety function. [6]

Quality Assurance: All those actions that provide confidence that quality is achieved. [2]

Quality assurance of process: All those actions that provide confidence that the process to develop a quality product is of good quality.

Quality assurance of product: All those actions that provide confidence that desired level of quality of product is achieved.

Software product quality: A measure of software that combines the characteristics of low defect rates and high user satisfaction. [7]

Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. [8]

Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. [8]

Configuration management: The process of identifying and defining the configuration items (e.g. source code or documentation) in software, controlling the release and change of these items throughout the software's life cycle, and recording and reporting the status of configuration items and change request. [5]

Acceptance testing: Formal testing conducted to determine whether or not software satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the software. [8]

3 Main references

This chapter summarises the literature review on safety analysis software quality assurance by presenting the main references. First, general software quality assurance references are presented and their applicability to safety analysis software assessment is addressed. Second, PRA guide references are discussed.

References [6, 9] indicate that ASME NQA-1 standard [5] is the most comprehensive quality assurance standard for safety analysis software and for safety software in general. The main parts that are applicable to safety analysis software in ASME NQA-1 are requirements 3 and 11 from Part I, Subpart 2.7 from Part II and Subpart 4.1 from Part IV. Subpart 2.7 provides requirements for computer software for nuclear facility applications. Requirements 3 and 11 define design and test control requirements, while Subpart 4.1 provides guidance on how the requirements should be used.

Other quality assurance standards and guides for safety software include IAEA technical reports series No. 397 [10], NNSA Safety software quality assurance guide [11] and IEC 61508-3 [12]. DOE has also specifically documented guidelines for QA of safety analysis and design software [13]. IEEE's more generally applicable software QA standards include IEEE Std 1012 [14], IEEE Std 730 [15] and IEEE Std 828 [16]. NASA's software QA documents [7, 17] are also worth examination in this context as well as ESA's [18, 19] even though they are not written for safety software. ISO/IEC 9126 standards 1-4 [20, 21, 22, 23] offer quite different point of view to software QA by defining metrics to measure the quality. The use of metrics is also addressed in IEEE Std 1061 [24].

There are no standards or guides written specifically for quality assurance of a PRA computer code. Finnish regulatory YVL guide [25] only states that guides for maintenance of PRA computer program used for assessing a Finnish nuclear power plant shall exist. However, the requirements that YVL guide sets for PRA should, of course, reflect in the functionality of a PRA computer program. Some other international and national PRA standards and guides contain sections and set some requirements for computer codes. A list of software requirements related quotations from these PRA guides is presented in Appendix 1.

Documents of IAEA [26, 27, 28, 29, 30, 31, 32, 33] address PRA computer programs most comprehensively. IAEA-SSG-3 [26] requires that computer codes must be capable to handle large and complex models and quantify them in a reasonably short timescale. IAEA-SSG-3 also demands that a level 1 PRA computer program has to provide the core damage frequency, frequencies of minimal cut sets, importance measures and results of uncertainty and sensitivity analyses. The requirements of IAEA-SSG-3 are also addressed in the requirements specification document of FinPSA. IAEA-TECDOC-832 [29] requires that the computer codes must be subject to a quality assurance programme to ensure that they are capable of correctly determining the minimal cut sets and correctly quantifying the PRA. IAEA-TECDOC-832 also states that the documentation of the computer codes including references should be extensive enough to assess the detail and verification. In addition, IAEA-SSG-4 [30] requires that level 2 PRA computer codes should be capable of modelling most of the events and phenomena that may appear in the course of the accident.

Other PRA guides include NRC Regulatory guide 1.200 [1], FANR RG 003 [34], ASME PRA standard [35], PNRA-RG-911.01 [36], ENSI –A05/e [37] and DOE PRA standard [38]. Most PRA standards require that computer codes must be validated and verified and suitable for the intended use. In most cases, no further explanation is given on what verification and validation should include. ASME PRA standard [35] states that the results should be compared to results that are obtained using accepted algorithms, and IAEA Review of PRA [33] requires that codes must first be applied to standard problems to gain experience. ASME PRA standard also remarks that the computer codes should be controlled to ensure consistent, reproducible results.

4 Overall requirements

This chapter discusses software quality requirements and quality assurance activities and overall requirements related to them. The chapter is divided into QA requirements of the development process and QA requirements of the software.

4.1 Requirements for the development process

4.1.1 Verification

IAEA-TECDOC-1135 [27] and IAEA-TECDOC-1229 [28] define the verification of a PRA computer code as “ensuring that the controlling physical and logical equations have been correctly translated into computer code.”

ASME NQA-1 [5] proposes that software must be verified by a competent individual that is not a developer of the computer program but can be from the same organization. Software verification methods can include reviews (design, code, results), alternate calculations, analyses, inspections, auditing, demonstration and tests (during development or after the program is finished). ASME NQA-1 remarks that the extent of verification and the chosen methods depend on the complexity of the software, the importance to safety, the similarity to previously proven software and the degree of standardisation.

It must be verified that the software requirements are traceable to user requirements [18]. During the development process, plans, software design, source code, documentation, changes and test results must be verified.

4.1.2 Validation

IAEA-TECDOC-1229 [28] defines the validation of a PRA computer code as "providing the theoretical examination to demonstrate that the calculational methods used in the computer code are fit for the intended purpose."

Algorithms, equations, mathematical formulations and expressions must be validated with respect to the software requirements. The correctness of any constraints or limitations must be validated. [14]

4.1.3 Documentation

Documentation must be understandable, legible and unambiguous to the intended audience. All acronyms, mnemonics, abbreviations, terms and symbols must be defined. [14]

IEEE Std 730-1998 [15] requires that the documentation must include software requirements specification, software design description, software verification and validation plan, software verification and validation report, user documentation and software configuration management plan at minimum. Other possible documents may be software development plan, standards and procedures manual, software project management plan, test plans, test results, test verification reports, problem reports, software quality assurance plan, software concepts document, software interface specification, program structure document or software maintenance manual. Software review documents must include software requirements review, preliminary design review, critical design review, software verification and validation plan review, managerial reviews, software configuration management plan review and post-mortem review. All documents need to be reviewed.

4.1.4 Areas of the development process

In this section, different phases and areas of the software development process are discussed with regard to quality assurance. Figure 1 illustrates the software development process by presenting a simplified life cycle model. In reality, the phases of the life cycle model overlap a lot and the process is iterative. In this document, the phases of the life cycle model that are considered are design phase, development and implementation, testing and operation and maintenance. In addition to these, configuration management is performed at each phase. Training can also be performed throughout the whole life cycle to support the needed actions.

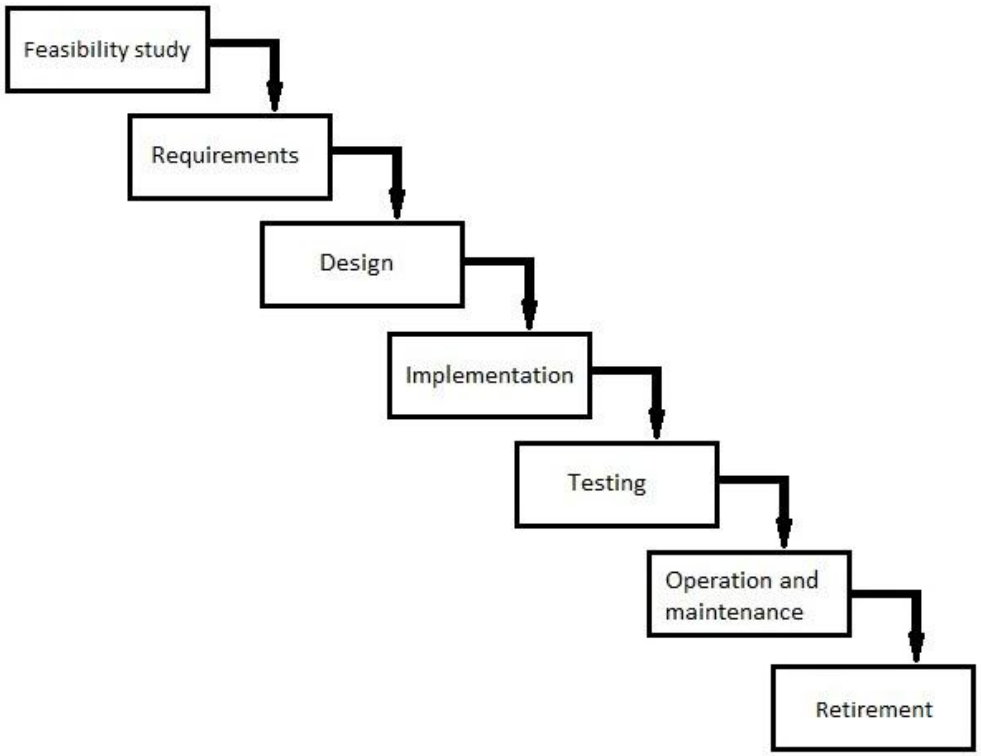

Figure 1: A waterfall lifecycle model of software

4.1.4.1 Design

Software design is a higher-level interpretation of what will be implemented in the source code [17]. Software design is usually divided into architectural design and module design. In architectural design, the software is divided into modules that are as independent as possible. Module design concentrates on what is inside the modules. ASME NQA-1 standard [5] requires that design analyses must be sufficiently detailed so that a person technically qualified in the subject can understand and review the analyses and verify the adequacy of results without help of developers. ASME NQA-1 also states that software design requirements must be traceable throughout the software development cycle.

The design approach must be technically adequate. It must be verified that the software design covers all the requirements before the program is approved for use.

4.1.4.2 Implementation

In the implementation phase, the computer program design is transformed into code. These transformations must be correct, accurate and complete. Techniques to develop error free code include, for example, unit level testing, refactoring, program slicing and use of coding checklists. Code analyses can include code logic analysis, formal inspection of source code, code data analysis, code interface analysis, unused code analysis, interrupt analysis and test coverage analysis [17]. Programming must be performed by conforming to a suitable programming language coding standard [12].

DOE G 200.1-1 [39] proposes that the programming team should meet regularly to discuss encountered problems and to facilitate program integration and uniformity. DOE G 200.1-1 also states that all codes should be backed up on daily basis and stored in an offsite location.

4.1.4.3 Configuration management

Configuration items to be controlled include documentation (software design requirements, plans, instructions for computer program use, test plans, problem reports, defect lists and results), computer program (source code, back-up files), training material and support software [5].

ASME PRA standard [35] requires that the PRA computer codes are controlled to ensure consistent and reproducible results. Software configuration management includes configuration identification, change control, status control and configuration reviews and audits [5, 6]. Criteria for these areas must be defined in the software configuration management plan [6]. ASME NQA-1 [5] proposes that a software baseline need to be established at the completion of each activity of the software design process. A baseline defines the most recent approved software configuration. A baseline labelling system must uniquely identify each configuration item, changes to configuration items by revision and each configuration [6]. The baseline labelling system must be used throughout the life of the software development and operation. The changes must be verified, tested and approved by the responsible organization and only approved changes are added to the baseline [5]. Reviews and audits need to be used to verify that the software product is complete and consistent with configuration item descriptions in the requirements [6].

4.1.4.4 Testing

Tests are needed to find defects, to validate the program and to verify that a computer program satisfies specified requirements and acceptance criteria. Test requirements and acceptance criteria must be based on design documents or other relevant technical documents and must be provided by the responsible organization. All tests have to be traceable to the requirements and all requirements must be tested. For a safety analysis program, the main requirement is that the calculated results are correct. The adequacy of test results can be evaluated by comparing the results to results provided by a comparable proven program, hand calculations, experiments, known solutions or empirical data and information from technical literature. [5, 6, 17]

Tests need to be planned beforehand. The planning can be started when the requirements document is complete. The test cases must be complete and the plan must specify pass/fail criteria for each test. Any special constraints, dependencies and procedures for performing tests have to be documented. NASA Software safety guidebook [17] states that the testing plan must include functional testing, acceptance testing and off-nominal testing at minimum.

A program can be tested as a whole or smaller units can be tested separately. Different types of testing include integration testing, functional testing, stability testing, resistance to failure testing, compatibility testing, performance testing, installation testing, regression testing and parallel operations testing [17]. ESA

PSS-05-11 [19] recommends that first unit tests should be white-box tests to ensure that the software is performing its functions as intended.

Test results must be documented. All test activity deliverables must be under configuration management [6]. Approval of changes is always needed before performing tests [5].

4.1.4.5 Operation and maintenance

The maintenance of the software includes modifications and migration, which is the movement of software to a new operational environment [14]. After the computer program is taken into use, its use is controlled in accordance with approved procedures and instructions, which include problem reporting and corrective actions, application documentation, access control specification, in-use tests and the configuration change control process [5].

When a problem is reported, the first step is to determine whether the problem is an error, user mistake or some other kind of problem. When an error is identified, it should be analysed how the error impacts past and present use of the computer program and how corrective actions impact development activities. The users need to be notified about an error, its impact and corrective actions as well as how to avoid the error. A system needs to be developed for documenting problems and prioritising corrective actions. [5]

Access control should address the security of the computer program. For example, unique user identification and a reliable password system can be used. [5]

4.1.4.6 Training

Software developers, designers, testing personnel and users must be qualified to perform their tasks. DOE's Safety software quality assurance functional area qualification standard [40] requires that safety software quality assurance personnel must have sufficient knowledge about safety software and software management, quality assurance, engineering, development, maintenance and assessment.

It is important that users of PRA computer code fully understand the limitations of the software [27, 28].

The training materials need to be readable, correct and complete. The training procedures need to be effective.

4.2 Requirements for the software

4.2.1 Requirements

Requirements must be unambiguous, complete, consistent, correct, clear, feasible, traceable, modifiable and verifiable [6, 17]. The functional and performance requirements need to be detailed enough so that the software design can be performed [13].

4.2.2 Design

The software design elements need to identify functions, the operating system, interfaces, performance requirements, design constraints, installation

considerations and design inputs [6]. The software design must be internally complete, consistent, understandable and correct [5].

4.2.3 Code

In the implementation phase, the descriptions of the higher abstraction level are transformed into code and data structures. These transformations must be correct, accurate and complete. The implementation must be traceable to the design. All interfaces and procedural logic must be complete and correct.

ISO/IEC 9126-1 [20] defines six quality characteristic of a software: functionality, reliability, usability, efficiency, maintainability and portability.

4.2.3.1 Functionality

A computer program must provide all functions that are listed in the requirements documentation. The functions must correspond to the needs of users. The produced results must be correct.

4.2.3.2 Reliability

The source code needs to be free from logic errors. The program must be capable to handle errors as well as possible.

4.2.3.3 Usability

The computer program must be understandable and attractive to the user. The program must be relatively easy to learn and use.

4.2.3.4 Efficiency

The computation time must be reasonable with respect to complication of the task. The computer program must use appropriate amount of resources when performing a function.

4.2.3.5 Maintainability

The source code must be easy to modify. The code must set the scene for the analysis of errors and deficiencies. The computer program must be stabile with regard to modifications and validity of modifications must be easy to test.

4.2.3.6 Portability

The computer program must function in all required environments. The installation must be possible and straightforward in all required environments. The computer program must be capable of sharing resources with other computer programs without hampering their use.

5 Conclusions

PRA guides typically address computer codes very briefly. They usually set very general requirements for the PRA tool, such as “a validated and verified computer code shall be used”, without further spelling out how the validation should be carried out. However, a moderate collection of general requirements and guidelines could be constructed when requirements obtained from different sources were put together.

Quality assurance of software is an increasingly studied area. Many authorities have published their own software QA standards and guides. General QA principles presented in software QA guides can very well be applied to safety analysis software.

This document reviewed literature on software quality assurance from the point of view of PRA computer codes. The findings of the literature survey were presented as overall QA requirements of the development process of safety analysis software and overall quality requirements of a safety analysis software product. The purpose was to provide background information for the QA planning of the PRA tool FinPSA. The document will be developed further and the next version will include QA requirements for the development of the FinPSA code.

References

[1] U.S. Nuclear Regulatory Commission, ”Regulatory guide 1.200: An approach for determining the technical adeguacy of probabilistic risk assessment results for risk-informed activities,” Washington, D.C., 2007.

[2] The Code of Federal Regulations, ”Nuclear Safety Management,” 2001. 10 CFR Part 830.

[3] Niemelä I, ”FinPSA,” Radiation and Nuclear Safety Authority (STUK), 2009. [Online]. Available: http://www.stuk.fi/ydinturvallisuus/ydinvoimalaitokset/en_GB/finpsa/. [Refered 18 September 2012].

[4] American Society of Mechanical Engineers, ”Quality assurance requirements for nuclear facility applications,” 2000. ASME NQA-1-2000.

[5] American Society of Mechanical Engineers, ”Quality assurance requirements for nuclear facility applications,” 2008. ASME NQA-1-2008.

[6] U.S. Department of Energy, ”Safety software guide for use with 10 CFR 830 subpart A, quality assurance requirements, and DOE O 414.1C, quality assurance,” Washington, D.C., 2005. DOE G 414.1-4.

[7] National Aeronautics and Space Administration, ”Software assurance standard,” 2004. NASA-STD-8739.8.

[8] The Institute of Electrical and Electronics Engineers, ”IEEE Standard Glossary of software engineering terminology,” New York, 1990. IEEE Std 610.12-1990.

[9] O'Kula K & Eng T, ”A "toolbox" equivalent process for safety analysis software,” U.S. Department of Energy, SAWG workshop, Washington, D.C., 2004. WSRC-MS-2004-00116.

[10] International Atomic Energy Agency, ”Quality assurance for software important to safety,” 2000. IAEA technical reports series No. 397.

[11] National Nuclear Security Administration, ”Safety software quality assurance - Qualification standard reference guide,” 2011.

[12] International Electrotechnical Commission, ”Functional safety of electrical/electronic/programmable electronic safety-related systems,” 2008. IEC 61508-3.

[13] U.S. Department of Energy, ”Assessment criteria and guidelines for determining the adequacy of software used in the safety analysis and design of defence nuclear facilities,” Washington, D.C., 2003. DOE CRAD-4.2.4.1.

[14] The Institute of Electrical and Electronics Engineers, ”IEEE Standard for Software

Verification and Validation," 1998. IEEE Std 1012-1998.

[15] The Institute of Electrical and Electronics Engineers, "IEEE Standard for Software Quality Assurance Plans," 1998. IEEE Std 730-1998.

[16] The Institute of Electrical and Electronics Engineers, "IEEE Standard for software configuration management plans," New York, 2005. IEEE Std 828-2005.

[17] National Aeronautics and Space Administration, "NASA software safety guidebook," 2004. NASA-GB-8719.13.

[18] European Space Agency, "Guide to software verification and validation," ESA publications division, Noordwijk, Netherlands, 1994. ESA PSS-05-10.

[19] European Space Agency, "Guide to software quality assurance," ESA publications division, Noordwijk, Netherlands, 1995. ESA PSS-05-11.

[20] The International Organization for Standardization and The International Electrotechnical Commission, "Software engineering - Product quality - Part 1: Quality model," Geneva, Switzerland, 2001. ISO/IEC 9126-1.

[21] The International Organization for Standardization and The International Electrotechnical Commission, "Software engineering - Product quality - Part 2: External metrics," Geneva, Switzerland, 2003. ISO/IEC 9126-2.

[22] The International Organization for Standardization and The International Electrotechnical Commission, "Software engineering - Product quality - Part 3: Internal metrics," Geneva, Switzerland, 2002. ISO/IEC 9126-3.

[23] The International Organization for Standardization and The International Electrotechnical Commission, "Software engineering - Product quality - Part 4: Quality in use metrics," Geneva, Switzerland, 2004. ISO/IEC 9126-4.

[24] The Institute of Electrical and Electronics Engineers, "IEEE Standard for a software quality metrics methodology," New York, USA, 1998. IEEE Std 1061.

[25] Radiation and Nuclear Safety Authority (STUK), "Probabilistic safety analysis in safety management of nuclear power plants," Helsinki, 2003. Guide YVL-2.8.

[26] International Atomic Energy Agency, "Develoment and application of level 1 probabilistic safety assessment for nuclear power plants," Vienna, Austria, 2010. IAEA-SSG-3.

[27] International Atomic Energy Agency and the OECD Nuclear Energy Agency, "Regulatory review of probabilistic safety assessment (PSA) level 1," Vienna, Austria, 2000. IAEA-TECDOC-1135.

[28] International Atomic Energy Agency and the OECD Nuclear Energy Agency, "Regulatory review of probabilistic safety assessment (PSA) level 2," Vienna, Austria, 2001. IAEA-TECDOC-1229.

[29] International Atomic Energy Agency, "IPERS guidelines for the international peer review service, Second edition," Vienna, Austria, 1995. IAEA-TECDOC-832.

[30] International Atomic Energy Agency, "Development and application of level 2 probabilistic safety assessment for nuclear power plants," Vienna, Austria, 2010. IAEA-SSG-4.

[31] International Atomic Energy Agency, "Determining the quality of probabilistic safety assessment (PSA) for applications in nuclear power plants," Vienna, Austria, 2006. IAEA-TECDOC-1511.

[32] International Atomic Energy Agency, "Framework for a quality assurance programme for probabilistic safety assessment," Vienna, Austria, 1999. IAEA-TECDOC-1101.

[33] International Atomic Energy Agency, "Safety reports series No 25: Review of

probabilistic safety assessment by regulatory bodies," Vienna, Austria, 2002.

[34] Federal Authority for Nuclear Regulation, "Probabilistic risk assessment: scope, quality and applications - Draft," 2011. FANR RG 003.

[35] American Society of Mechanical Engineering Commitee on Nuclear Risk Management, "Standard on probabilistic risk assessment for advanced non-LWR nuclear power plant applications," 2008.

[36] Pakistan Nuclear Regulatory Authority, "Probabilistic safety assessment of nuclear power plant-level 1: regulatory guide," Islamabad, Pakistan, 2010. PNRA-RG-911.01.

[37] Swiss Federal Nuclear Safety Inspectorate, "Probabilistic Safety Analysis (PSA): Quality and Scope," Villingen, 2009. ENSI-A05/e.

[38] U.S. Department of Energy, "Development and use of probabilistic risk assessment in Department of Energy nuclear safety applications - Draft," Washington, D.C., 2010.

[39] U.S. Department of Energy, "Software engineering methodology TOC," 1996. DOE G 200.1-1.

[40] U.S. Department of Energy, "Safety software quality assurance functional are qualification standard," Washington, D.C., 2011. DOE-STD-1172-2011.

Appendix 1: Quotations from PRA standards and guides

Nr	Quotation	Source
1	5.9. The analysis should be carried out using a suitable computer code that has the following capabilities: (a) It should be capable of handling the very large and complex logic model of the nuclear power plant. (b) It should be capable of quantifying the PSA model in a reasonably short timescale. (c) It should be capable of providing the information necessary to interpret the Level 1 PSA, such as the core damage frequency, frequencies of cutsets (combinations of initiating events and failures and/or human errors leading to core damage), importance measures and results of uncertainty and sensitivity analyses.	IAEA-SSG-3, chapter 5.9. [26]
2	***2.4.6. Validation and verification of computer codes*** The computer codes used in the PSA should be validated and verified. In this context, **validation** is defined as providing the demonstration that the calculational methods used in the computer code are fit for purpose and **verification** is defined as ensuring that the controlling physical and logical equations have been correctly translated into computer code. The reviews should determine whether the codes which have been selected by the PSA team are fit for purpose and that the users of the codes are experienced in their use and fully understand their limitations. It is recommended that the regulatory authority and the utility should reach an agreement on the set of codes to be used.	IAEA-TECDOC-1135, chapter 2.4.6 [27]
3	***2.4.6. Validation and verification of computer codes*** The computer codes required for a Level 2 PSA include the codes which model the severe accident phenomenology, including the codes which model individual phenomena as well as the integrated codes (see Section 3.3), and the probabilistic codes for quantifying the events trees used to model the progression of the severe accident — see Section 3.6. As for the computer codes used in the Level 1 PSA, those used in the Level 2 PSA also need to be validated and verified. In this context, **validation** is defined as providing the theoretical examination to demonstrate that the calculational methods used in the computer code are fit for the intended purpose. This may also involve comparison with experimental evidence. **Verification** is defined as ensuring that the controlling physical and logical equations have been correctly translated into computer code. The reviewers need to check that the analysts have used the codes within their limits of applicability. In addition, they need to confirm that the predictions of the codes are consistent with the analysis carried out for similar plants and experimental information. Where integrated codes are used, their predictions are	IAEA-TECDOC-1229, chapter 2.4.6 [28]

Nr	Quotation	Source
	compared with those obtained using separate effects codes. It is necessary for the reviewers to determine whether the codes which have been selected by the PSA team are fit for the intended purpose and that the users of the codes are experienced in their use and fully understand their limitations. It is suggested that the regulatory authority and the utility reach an agreement on the set of codes to be used.	
4	The reviewer should check that the computer codes used are subject to a quality assurance (QA) programme to ensure that they are capable of correctly determining the minimal cut sets and correctly quantifying the PSA.	IAEA-TECDOC-832, chapter 6.7. [29]
5	5.13. The analysis of the progression of severe accidents should be performed using one or more computer codes for severe accident simulation (see Annex II). The computer code(s) chosen to perform detailed analysis and the number of calculations that should be performed depends on the objective of the PSA. Among the issues that should be considered in making these decisions are: (a) The code(s) should be capable of modelling most of the events and phenomena that may appear in the course of the accident. (b) Interactions between various physicochemical processes should be correctly addressed in the computer code. (c) The extent of validation and benchmarking effort and associated documentation should be satisfactory. (d) Computing time and resource requirements should be reasonable. The analysts should be aware of the technical limitations and weaknesses of the selected code(s). The analyses of severe accidents should cover all sequences leading either to a successful stable state, where sufficient safety systems have operated correctly so that all the required safety functions necessary to cope with the plant damage state have been fulfilled, or to a containment failure state.	IAEA-SSG-4, chapter 5.13. [30]
6	PRA quantification software, thermal/hydraulic codes, structural codes, radionuclide transport codes, human reliability models, common cause models, etc. are typically used in the PRA quantification process. These models and codes should have sufficient capability to model the conditions of interest and provide results representative of the Facility. They need to be used only within their limits of applicability. As errors in such programmes may significantly impact the results, it is necessary that the development and application of the computer programmes, spreadsheets or other calculation methods exhibit a high level of reliability as ensured through a documented verification and validation process. In addition, users should demonstrate the appropriateness of the models or codes selected for a specific application and of the way in which these programmes are combined and used to produce the needed results.	FANR RG 003 - Draft, chapter 14 [34]
7	(e) Process used to maintain software configuration control used	ASME PRA Standard,

Nr	Quotation	Source
	for PRA development The computer codes used to support and to perform PRA analyses shall be controlled to ensure consistent, reproducible results.	chapter 5.3. [35]
8	The documentation of the computer codes including references should be extensive enough to assess the detail and verification.	IAEA-TECDOC-832, chapter 7.3.3. [29]
9	The computer code used for solution and quantification of the PSA model is verified and validated, and is used only within its specified range of applicability. Specific limitations of the code are recognized.	IAEA-TECDOC-1511, Table 11.2-B. [31]
10	In order to ensure QA for the PSA, all computer codes used in the development of the PSA must be verified and validated, either in the course of their development or by the PSA group. Computer codes that are purchased commercially may be verified and validated by the code developer. For software that is not commercially procured but, for example, written internally in the PSA organization, a verification, validation and QA process should be performed. QA for computer software is described in Ref. [4].	IAEA-TECDOC-1101, chapter 4.2. [32]
11	The codes used to perform the analysis are validated and verified for both technical integrity and suitability.	NRC Regulatory guide 1.200, chapter 1.2.2 [1]
12	The quantification of the PSA should be carried out using a suitable computer code which has been fully validated and verified.	PNRA-RG-911.01, chapter 5.8.4 [36]
13	5.56. This paragraph provides recommendations on meeting Requirement 18 of Ref. [3] on use of computer codes for a Level 1 PSA. The computer codes used to justify the success criteria should be well qualified to model the transients, loss of coolant accidents and accident sequences being analysed and to obtain a best estimate prediction of the results. The computer codes should be used only within their established realm of applicability and should be used only by qualified code users.	IAEA-SSG-3, chapter 5.56. [26]
14	For the computation of the PSA results, a validated computer code shall be used. Limitations of the code or of the quantification method (e.g., missing capability to consider success probabilities in accident sequences) shall be discussed.	ENSI –A05/e, chapter 4.7.1. [37]
15	In particular, regarding the users of the codes, the audit should confirm that: (1) The users are experienced in the use of the codes and understand the code limitations. (2) Adequate guidance and training has been provided in the use of the codes. (3) The codes have been used to evaluate standard problems to gain experience.	IAEA Review of probabilistic safety assessment by regulatory bodies, chapter 2.5.1. [33]
16	The computer codes that support these analytical methods should be adequate for the purpose and scope of the analysis and the	IAEA-SSG-3, chapter 2.5. [26]

Nr	Quotation	Source
	controlling physical and logical equations should be correctly programmed in the computer codes (Ref. [3], para. 4.60).	
17	**4.1.4 Quality Assurance and Peer Review Plans** The PRA plan shall identify the applicable DOE QA requirements and describe how they will be met including DOE requirements for QA records and audits, the use of verified computer programs, document logs, a corrective action program, and the use of procedures, in addition to the following topics:	DOE Standard: Development and use of probabilistic risk assessment in Department of Energy nuclear safety applications - Draft, chapter 4.1.4 [38]
18	PERFORM quantification using computer codes that have been demonstrated to generate appropriate results when compared to those from accepted algorithms. IDENTIFY method specific limitations and features that could impact the results.	ASME PRA Standard Table 4.5.10-2 (b) ESQ-B1 [35]