THE USE OF "GRADING" IN THE APPLICATION OF THE CSA N286.7 STANDARD FOR QUALITY ASSURANCE OF ANALYTICAL, SCIENTIFIC AND DESIGN COMPUTER PROGRAMS FOR NUCLEAR POWER PLANTS

J. Pascoe[1], D.J. Richards[2], E. Mileta[3] and J. Skears[4]
[1] AMEC-Nuclear Safety Solutions, Toronto, Ontario, Canada
[2] Atomic Energy of Canada Limited, Chalk River, Ontario, Canada
[3] Ontario Power Generation, Toronto, Ontario, Canada
[4] CANDU Owners Group, Toronto, Ontario, Canada

Abstract

Computer programs used for the design and safety analysis of nuclear reactors must comply with the Canadian standard CSA N286.7-99, "Quality Assurance of Analytical, Scientific, and Design Computer Programs for Nuclear Power Plants". Although "grading" is acknowledged in the N286 series of Standards, an appropriate level of guidance for implementation is not provided. A collaborative effort was undertaken by the CANDU Industry to provide guidance on the application of the Standard - based on Industry experience. This guidance included a graded approach to meeting the requirements of the Standard and that approach is described and illustrated with a worked example in this paper.

1. Introduction

The CSA N286.7-99 Standard [1] is incremental to the CSA N286.0 suite of standards and while it narrows the focus to analytical tools used to design, analyze or support safety related systems, it is still broad in scope. The Standard specifies high-level requirements for the development, qualification and use of scientific, engineering and safety analysis software used to design, analyze or support the continued operation of nuclear power plants. It is expected that owner organizations[1] will develop more detailed and more specific requirements in the form of procedures or governance. It is important to ensure that users of software that may have an impact on the design or continued operation at a nuclear facility be provided with some direction as part of the scope of an organization's governance.

Software covered under the Standard is used in diverse applications and the use of incorrect results obtained from that software could have an impact upon the environment, the safety of the public or plant staff, or the continued operation of the plant or facility. These impacts will range from relatively minor to significant and could also represent the potential for substantial expenditures. The events or circumstances assessed with software will have a likelihood or probability of occurrence that must be taken into account when considering potential impacts. These two factors: 1) impact and 2) probability of occurrence, taken together constitute risk. In order to be pragmatic, a software quality assurance process should allow for the extent of software qualification to be commensurate with the overall risk associated with the use of results obtained from the software.

[1] The owner of the nuclear power plant.

While the Standard acknowledges the potential for a graded application, the specifics are left to owner organizations and participants[2].

2. Scope of N286.7-99

The CSA-N286.7-99 Standard *specifies*[3] *the requirements for the quality assurance program applicable to the design, development, maintenance, modification and use of analytical, scientific, and design computer programs that are used in nuclear power plant applications to perform or support:*

a) design and analysis of safety-related equipment, systems, structures and components as identified by the owner;

b) deterministic and probabilistic safety analyses and reliability studies;

c) reactor physics and fuel management calculations, and

d) transfer of data between computer programs or pre- or post-processing calculations associated with (a), (b) and (c) above.

The Standard further provides the following definition of Safety-Related System:

Those systems, and the components and structures thereof, which, by virtue of failure to perform in accordance with the design intent, have the potential to impact on the radiological safety of the public or nuclear facility/plant personnel from the operation of the nuclear facility or nuclear power plant. Those systems, and the components and structures thereof, are associated with

(a) the regulation (including controlled start-up and shutdown) and cooling of the nuclear facility or reactor core under normal conditions (including all normal operating and shutdown conditions);

(b) the regulation, shutdown, and cooling of the nuclear facility or reactor core under anticipated transient conditions, accident conditions, and maintenance of the nuclear facility or reactor core in a safe shutdown state for an extended period following such conditions; and

(c) limiting the release of radioactive material and the exposure of nuclear facility or plant personnel and/or the public to meet the criteria established by the licensing authority with respect to radiation exposure during and following normal, anticipated transient conditions, and accident conditions.

As is seen, the scope of the Standard is quite broad, and encompasses a wide range of software.

[2] An organization required by the owner of the nuclear power plant to meet one or more of the Standards in the N286 series.
[3] The italics designate text from CSA N286.7-99

3. Grading

3.1 Introduction

To establish a graded approach to software quality assurance, the software used by an organization must be firstly decomposed into a hierarchical structure, with the importance of all software ranked or graded on potential impact arising from the use of incorrect results or other criteria. Secondly, requirements on the activities used to qualify the software must be flexible enough to allow the appropriate level of application of qualification that is commensurate with the use and type of software.

There are three primary requirements in establishing a graded implementation of software quality assurance procedures:

1) Creation of a scheme for grading and characterizing the software.

2) Setting software quality assurance requirements based on software grade.

3) Provision of a mechanism for software quality assurance requirement relaxation based on software type, characteristics and other mitigating factors.

This paper is focused on the first requirement. Guidance on requirements two and three must be contained within an organization's software governance.

3.1.1 Characteristics of international grading approaches

Characteristics of international approaches to grading can be summarized as follows:

1. Three software levels or grades are used.

2. The grade is determined by assessment of application risk and software characteristics.

3. No quantitative or mechanistic means of assigning level or grade is used.

3.2 Basis for establishing software grade

The following concepts are used to establish the basis for graded conformance with the Standard:

1. Software Rank - the importance of the software based on risk arising from use.

2. Software Characterization - defines the state or origin and applications of the software.

3. Mitigating Factors and Compensatory Actions – agents that can reduce the severity of an incorrect software result or can aid in the prevention of such a result.

These concepts are expanded upon in the following sections.

3.3 Risk as a basis for establishing software grade

In keeping with international best practices [2 – 5], it is recommended that risk be used as the basis for establishing software grade. The use of results obtained with software could potentially have an adverse impact on safety – of the public and plant workers, if they are inaccurate[4] or used outside of their range of application.

Risk, therefore, can be used as a measure of the importance of software and the degree to which it must be qualified to reduce that risk to an acceptable level.

In this paper, risk will be taken to mean the combination of the probability of an event occurring as a result of software use with the impact that event would have and with its different circumstances. Risk can be mitigated through compensatory actions.

To aid in the determination of software grade, the types of risks are broken down as shown in Table 1 [6]:

Owner and participant organizations may need to provide further and specific guidance for the identification of risks and risk levels.

3.3.1 Software characterization

Software characterization is the process of specifying traits or characteristics possessed by software that can be used to assess risk arising from use of the software and establish and refine quality assurance requirements based upon grade. Examples of software characteristics include:

- Application – what is the software used for?
- Development – was the software developed in compliance with a software quality assurance standard?
- State – is the software new to be developed or does it exist (legacy)?
- Extent of use – is the software used by many organizations for diverse applications?
- How it was acquired – commercial "off the shelf", custom built, through an agreement with another organization, General Public License.
- Conformance – was the software developed or qualified in accordance with a recognized software quality assurance program?
- Complexity - the number and type of models contained in the software or the number and type of plant systems, components or behaviour modeled by the software.

Examples of software use are listed in Table 2.

[4] The use of inaccurate results may be acceptable in those circumstances where an adequate level of margin exists between the code calculation and the critical value of a key acceptance parameter.

3.3.2 Software grade

Once the risks associated with the use of software results have been established and quantified, those risks and knowledge of how the software is applied can be used to determine grade.

It is proposed that three grades of software be established:

Grade 1. Software of this grade is used to assess the mitigating effect of or to determine setpoints for special safety systems and safety related systems as defined in the Standard. Incorrect results obtained with this grade of software could place the public or facility workers at risk of serious injury. Errors in analysis carried out using this type of software could result in immediate regulator notification and facility shutdown or reduction in power. Analysis rework costs or facility modifications have the potential to be extremely costly. This grade of software is used to establish the facility safety case. Each type of risk associated with the use of results generated by software of this grade has high impact.

Grade 2. Errors in results obtained with software of this grade have the potential to reduce the effectiveness of facility safety systems. The risk to facility and public safety is reduced from that of Grade 1 software. Use of results obtained with this grade of software could lead to noncompliance with terms of operating licenses or operating policies and principles. Financial losses arising from errors obtained are less than that for Grade 1 software, but could nonetheless be substantial. Risks associated with the use of results generated by software of this grade form a spectrum from high impact to low impact.

Grade 3. This grade of software is used in applications that have negligible impact on facility operation. Risk to the public, facility workers and the environment is also negligible.

3.4 Process for determining software grade

The following specifies the basic elements of the process to determine software grade:

A) Specification of Software and Version Number: The software for which a grade is to be assigned should be clearly specified – including version number. In the circumstance in which multiple software versions are in use, all versions of the software to which the grade is to be assigned should be individually identified. The key software characteristics should also be listed.

B) Identification of Potential Software Applications: Using Table 2 as a guide, all organizational specific potential applications of the software are listed. Relevant characteristics (see Section 3.3.1) of the software are also specified as appropriate.

C) Consequence Assessment: For each software application identified in B), potential consequences or impacts are specified if software provides incorrect or deficient results.

D) Individual Risk Assessment: For each application and potential impact identified in B) and C), the type of risk associated with the use of the software using Table 3 is identified. It is noted that a single application may present more than one type of risk. For each application and type of risk, the

risk level is assessed taking into account the software characteristics. Consideration may be given to the likelihood of the event being analyzed or the probability of the software being used with an undiscovered error or defect. Risk levels will be High, Medium or Low.

E) Identification of Critical Application: From D) the application that presents the highest application risk is identified. This will be the application that has the highest risk level against any of the identified factors.

F) Initial Assignment of Software Grade: Using the information defined in E), the risk is decomposed into impacts and probability of occurrence. For the highest risk (chosen from the group of risk types) an initial grade of 1, 2 or 3 is assigned to the software. It is noted that the types of risk may be ranked in order of importance for a specific software and application. Risk levels (probability or impact) may be reduced through the use of mitigating factors or compensatory actions with documented and approved justification.

G) Final Assignment of Software Grade Considering Mitigating Factors and Compensatory Actions: If the assessor believes the grading determined in the previous step is too high, then for each factor having a risk level at the initial grade assessment level, he/she should provide the justification (mitigating factors or compensatory measures) for moving that risk level down to a lower level. Mitigating factors and compensatory actions are identified and their potential to reduce the likelihood or magnitude of the consequences of software application is considered. If as a result of the application mitigating factors or compensatory actions all risks are reduced to a lower level, then the software grade may be reduced to the next lower level.

H) Mitigating Factors and Compensatory Actions: Mitigating factors are those characteristics of software use that act to lessen the extent of, or make less severe the impact of errors that may be produced by the software. Compensatory actions are those activities performed to adjust or make up for shortcomings in the software. One common mitigating factor in the use of analysis software is that the results are part of a strictly controlled design quality assurance program (i.e., CSA N286.2) and are typically formally documented with independent review and approval. This also allows for more compensatory actions, like formal third party reviews or qualification test programs. These factors and actions tend to reduce the potential risk arising from software application. In contrast, software used outside of the design process, or software that is used as part of a process whose results are not separately documented, whose results are immediately acted upon, or whose results cannot be tested or verified, would tend to increase the potential risk arising from software application. Examples of compensatory actions and mitigating factors are given in Table 4.

3.5 Worked example

Table 5 presents a worked example of the software grading process applied to a hypothetical thermalhydraulic analysis code, CODEX.

4. Conclusions

A grading approach for the application of CSA N286.7-99 has been described and an example of the implementation presented. Using this grading approach allows for relaxation of requirements as the

resulting software quality assurance process allows for the extent of software qualification to be commensurate with the overall risk associated with the use of results obtained from the software.

5. Acknowledgement

The authors wish to acknowledge the other members of the CANDU Industry Team that contributed to this work: R. Chun, Bruce Power; R. Ghai, Atomic Energy of Canada; F. Iglesias, Candesco Corporation; O. Nainer, Bruce Power; M. Nguyen, Hydro Quebec; Y. Parlatan, Ontario Power Generation; B. Willemsen, New Brunswick Power.

6. References

[1] CSA N286.7, "Quality assurance of analytical, scientific, and design computer programs for nuclear power plants".

[2] DOE G 414.1-4, 6-17-05, "Safety Software Guide for Use with 10 CFR 830 Subpart A, *Quality Assurance Requirements, and DOE O 414.1C, Quality Assurance".*

[3] John Zepper, Kathy Aragon, Molly Ellis, Kathleen Byle, and Donna Eaton, "ASCI applications software quality engineering practices", SANDIA REPORT SAND2002-0121, Unlimited Release, January 2002.

[4] IAEA Safety Series No. 50-C-QA (1985 QA (1985-88)).

[5] Licensing of Safety Critical Software for Nuclear Reactors, Common Position of Seven European Nuclear Regulators and Authorised Technical Support Organizations, Revision 2007.

[6] Spallation Neutron Source Quality Assurance Plan, SNS-QA-P01, ORNL, March 2004.

Table 1 - Types of Risk	
Functional	Risk of a system or component to fail performing its functions as a result of software use.
Environment	Risk to the environment presented by use of the software. The impact of this type of risk is related to the severity of adverse effects experienced in the environment surrounding the plant.
Health and Safety	Risk to plant workers, and the public presented by use of the software. An example of the impact arising from this type of risk would be the potential for higher than expected radiological exposure of plant workers conducting routine plant maintenance.
Compliance	Risks for non-compliance with federal, provincial and organizational laws, statutes, regulations and procedures. The impact of this risk type could be fines or forced shutdown.
Cost	Potential financial risks associated with use of the software. Impacts associated with this type of risk include loss, waste or poor value for money.

Table 2 - Examples of Software Uses		
• Safety system set point determination	• Release/Dose calculations	• Safety Analysis
• Engineering Assessment Calculations	• Risk Assessment	• Design
• Reliability Studies	• Fuel Management	• Release Monitoring
• Radiation Protection	• Outage Support	• Environmental Qualification
• Engineering Design Calculations		

Table 3 - Use of Risks in Software, Engineering and Safety Analysis Software Grade Determination

	Grade 1	**Grade 2**	**Grade 3**
	High	Medium	Low
Risk Type	If results produced by the code are incorrect::		
Functional	Serious impact on continued plant operation including need for prompt shutdown or reduction in power. Trip setpoint may be incorrect with no opportunity or basis for operator action.	Degradation in support for continued plant operation. Shutdown or reduction in power level may be required. Trip setpoint may be incorrect but basis and opportunity exists for operator action.	Negligible impact on plant operation. Management notification required
Environment, Safety and Health	Potential for 1) severe adverse impact on health and safety of plant workers or the public or 2) environmental damage that could exceed regulatory limits or involve significant cleanup costs.	Potential for injury or illness requiring hospitalization, temporary or partial disability. Moderate adverse impact on the environment or the health and safety of a plant worker or the public.	Potential for 1) minimal impact on the health and safety of plant workers or the public, or 2) negligible impact on the environment.
Compliance	Potential for non-compliance with Canadian Safety and Control act and federal laws or regulations. Regulator notification required within a short period of time.	Potential for non-compliance with terms of operating license or plant operating policies and principles. Regulator notification may be required.	Potential for minor non-compliance with established management practices or corporate procedures.
Cost	Potential for a financial loss of $500K or more through high cost of analysis rework, plant modifications or lost revenue.	Potential for a financial loss of $50K or more (but less than $500k).	Potential for financial loss less than $50K.

Table 4 - Example of Compensatory Actions and Mitigating Factors

Compensatory Actions and Mitigating Factors are measures to be taken to account for shortcomings in the software, or barriers to be introduced to reduce the severity of the errors resulting from software application. These include, but are not limited to the following:

- Measures to prevent misuse or human error and includes such items as:
  - Extra Reviews of Software Use and/or Results
  - Qualified Input Data
  - Independent Parallel execution of code
  - Extra Training
  - Extra Documentation
- If the software is generally used but only has safety significance in limited cases then do not use the software for those few safety significant applications.
- Measures to qualify the answer produced or redundancy in the affected system and includes such items as:
  - Alternate Calculation(s)
    - Manual Calculation
    - Independently Developed Software
  - Testing
  - Benchmarking against known solutions
  - Historical Trend Analysis
  - Use of large design or safety margins
  - Ensure that Software Results are only one of several inputs to a decision or act
  - Use of conservative assumptions, data or models

Table 5 - Assessing of Software Grade for CODEX Code[5]	
A. Software Name and Version Number and key software characteristics. (software characteristics are defined in Section 3.3.1)	1. CODEX Version 2 2. CODEX - MOD 2 The two versions of CODEX were developed by CompanyA and CompanyB. Although there are functional differences between these versions of CODEX, they are not relevant to software quality assurance grading. Characteristics: • CODEX is used by CANDU Owners and Designers; • CODEX, in various incarnations, has been in use since the mid 1970s; • At CompanyA, the functionality of CODEX has been migrated to the CODEY code. As part of the qualification effort, results from the two codes were compared for consistency; • CODEX models steady state, single-phase liquid flow in the HTS of CANDU power plants, and possesses fairly straightforward models of a small number of phenomena; • CODEX makes use of correlations that have been extensively qualified and are in wide-spread use; • The CompanyB version of CODEX has been validated.
B. Software Use (from Table 2 list all potential applications of the software)	1. Design 2. Safety System Assessment 3. Safety Analysis 4. Engineering Calculations
C. Consequence Assessment (For each software use identified in Section B, specify potential consequences or impacts if software provides incorrect or deficient results). Consider probability of occurrence and mitigating actions, as appropriate.	1. Use of CODEX to size feeders. - Reactor performance not optimized. - As a mitigating action, CODEX results are required to be verified/crosschecked as part of the design process by the use of system thermalhydraulic code or hand calculations. 2. Use of CODEX to assess trip effectiveness for a slow Loss of Regulation event. - trip setpoint incorrect. 3. Use of CODEX to assess the thermalhydraulic performance of a CANDU reactor loaded with alternative fuel bundles. - reactor power lower than expected.

[5] CODEX is a fictitious code used for illustrative purposes

<table>
<tr><td></td><td colspan="3">4. Use of CODEX to support feeder thinning assessment.
- pipe wall thickness lower than calculated,
- incorrect pipe selected for inspection.</td></tr>
<tr><td rowspan="5">D. Individual Risk Assessment (For each application and potential impact identified in Sections B and C, identify the type of risk associated with the use of the software using Table 1.
Note: A single application may present more than one type of risk. In the determination of the level, consideration may be given to the likelihood of the event being analyzed or the probability of the software being used with an undiscovered error or defect. Risk levels will be High, Medium or Low. For grade 2 software, these levels may be used in requirement relaxation.)</td><td>Application</td><td>Risk(s)</td><td>Levels</td></tr>
<tr><td>1. Use of CODEX to size feeders.
- Reactor performance not optimized.</td><td>Compliance
Cost</td><td>Low
Low</td></tr>
<tr><td>2. Use of CODEX to assess trip effectiveness for a slow Loss of Regulation event.
- trip setpoint incorrect.</td><td>Functional
Environmental
Compliance
Cost</td><td>Medium
Medium
Medium
High</td></tr>
<tr><td>3. Use of CODEX to assess the thermalhydraulic performance of a CANDU reactor loaded with alternative fuel bundles.
- reactor power lower than expected.</td><td>Functional
Cost</td><td>Low
High</td></tr>
<tr><td>4. Use of CODEX to support pipe thinning assessment.
a) pipe wall thickness lower than calculated,

b) incorrect pipe selected for inspection.</td><td>
a) Functional
Environment, Health and Safety
Compliance

b) Environment, Health and Safety
Compliance</td><td>
Low
Medium
Low

Medium
Medium</td></tr>
<tr><td>E. Identification of Critical Application (from Section D, specify the application that presents the highest application risk. This will be the application that has the highest number of High or Medium risk identifications.)</td><td colspan="3">Use of CODEX to assess trip effectiveness for a slow Loss of Regulation event.</td></tr>
<tr><td>F. Assignment of Initial Software Grade (Using the information in Section E, assign a grade of 1, 2 or 3 to the software.)</td><td colspan="3">Highest risk level for critical application is HIGH. Based upon this, CODEX should be assigned a grade of 1.</td></tr>
<tr><td>G. Assignment of Final Software Grade consideration of Mitigating Factors and Compensatory Actions
(Identify any mitigating factors or</td><td colspan="3">For the critical application, the sole high risk factor is cost.

Mitigating Factors: cross check of CODEX calculation with other thermalhydraulic codes.</td></tr>
</table>

compensatory actions that may reduce the likelihood or magnitude of the consequences of software application. Refer to characteristics of the software that may also impact on the application.)	Compensatory Actions: slow loss of regulation accident allows for operator intervention. CODEX has been widely used, is not complex, and has been validated. Given these considerations, the software grade assigned to CODEX can be reduced from grade 1 to grade 2.
Additional Comments / Recommendations (This field is required for Grade 2 software)	It should be noted that largely as a consequence of the use of CODEX to assess trip effectiveness determination, CODEX should be considered to be at the “higher end” of grade 2.