PRE-APPROVED VERSION

L

TONY HINDS
(CONTROLLED DIST'N)

PROTECTED-COG R&D

COG-95-264

AECL

GUIDELINE FOR CATEGORIZATION OF SOFTWARE IN NUCLEAR POWER PLANT SAFETY, CONTROL, MONITORING AND TESTING SYSTEMS

Revision 1.0

by

G.H. Archinoff, D.K. Lau,
J. de Grosbois and W.C. Bowman

The work reported in this document was funded by the COG R&D Program:

Technical Committee No. 16 WPIR No. 1651

Control Centre Technology Branch
Instrumentation and Control Branch
Chalk River Laboratories
Chalk River, Ontario
Canada K0J 1J0

1995 May 24

PROTECTED-COG R&D COG-95-264

AECL

GUIDELINE FOR CATEGORIZATION OF SOFTWARE IN NUCLEAR POWER PLANT SAFETY, CONTROL, MONITORING AND TESTING SYSTEMS

Revision 1.0

by

G.H. Archinoff, D.K. Lau,
J. de Grosbois and W.C. Bowman

ABSTRACT

This document is a guideline for categorizing the safety criticality level of software in nuclear power plant safety, control, monitoring, and testing systems. The Guideline follows a risk-based approach, which recognizes that the criticality level of software depends on the safety significance of the plant system of which the software is a part, and on the worst possible consequences of failure of the software with respect to the safety function of the system. The categories of safety criticality are defined, and specific decision criteria and a procedure are presented for selecting the appropriate categorization level depending on the safety-related role of the software in the plant system. The Guideline can also be used to assist with computer system design decisions aimed at achieving the requisite confidence in the system and software while minimizing the criticality of any one software function.

Control Centre Technology Branch
Instrumentation and Control Branch
Chalk River Laboratories
Chalk River, Ontario
Canada K0J 1J0

1995 May 24

GUIDELINE FOR CATEGORIZATION OF SOFTWARE IN NUCLEAR POWER PLANT SAFETY, CONTROL, MONITORING AND TESTING SYSTEMS

Revision 1.0

G.H. Archinoff, D.K. Lau,
J. de Grosbois and W.C. Bowman

VALUE AND IMPLICATIONS

The fundamental challenge in the development and deployment of software and software-controlled computer systems in safety related nuclear applications is to ensure an adequate degree of system reliability and safety. This guideline defines a procedure for the categorization of a software or software-controlled system based on the safety significance of the plant system involved and the possible software failure impact on that system. The Guideline provides specific decision criteria that are to be used to define the level of integrity (i.e., the categorization) required of the system. System and software development standards can be specified within this framework to define a consistent definition of specific software and system engineering practices, methods, and procedures (i.e., to define the level of rigour) that is to be applied in the development of software and systems to achieve the desired level of system integrity for each category.

This report defines a process and procedure for categorization of software and software-controlled systems in nuclear power plant safety, control, monitoring and testing systems. The guidelines presented herein have been based on the principles and approach established in the previous version of the guideline, on the experience gained from several trial categorizations, and on current industry consensus. The intent of the guideline is to define a consistent, practical, and repeatable method of categorization.

L.R. Lupton R.R. Shah

Control Centre Technology Branch
Instrumentation and Control Branch
Chalk River Laboratories
Chalk River, Ontario
Canada K0J 1J0

1995 May 24

REVISION HISTORY

Revision	Date	Description	By
0	1991 June	Initial version of Guideline (Ontario Hydro version)	A.K. Lee
1.0	1995 May 24	Major re-write and enhancement of the basic approach defined in Rev. 0. Incorporates feedback from categorization experiences to date. Issued for trial use.	G.H. Archinoff, D.K. Lau, J. de Grosbois and W.C. Bowman

TABLE OF CONTENTS

PREFACE

This document is a re-write and enhancement of the general approach defined in Reference 1, which is the first version of this Guideline. The changes incorporated in this revision are based on user feedback from twelve different categorizations. International standards (References 2-10) were also consulted in the preparation of this version of the Guideline.

The major changes with respect to Revision 0 are as follows:

1. There is a more complete description of the underlying basis of the approach to categorization.

2. Increased guidance is provided in the areas of:
   - selecting the plant system type,
   - identifying the safety significance of the plant system, and
   - determining the impact of software failure on the plant system.

3. Design heuristics are provided to assist with design decisions aimed at ensuring overall system reliability and safety while avoiding unnecessarily stringent categorization of the software.

4. A suggested outline is provided for the categorization report.

5. Monitoring and Testing Systems have been explicitly addressed.

The results of the previous applications of Rev. 0 of the Guideline are summarized in Table i below. It should be noted that the categories identified in the table may be based on specific assumptions and/or conditions that are documented in each categorization report. The individual reports should, therefore, be consulted in order to understand the basis for each categorization.

Jim Hunton - BNGS A SSMC
Bill Bowman - ~~OH~~ HO examples.

892-1561

Categorized Software	Category
Pickering A Contact Scanners	Category 3
Bruce B MTC in the DCC	Category 2 (may be reducible to 3)
Bruce A DCC	Category 2
Bruce A PDS	Category 3
Bruce A CSCS	Category 3
Bruce A SSMC	Category 3
Bruce A Data links	Category 3
Pickering B BPC	Category 2
Pickering A SOR Drop Test software	Category 3
Darlington Fuel Handling and Protective Software	Fuel Handling Category 3, Protective software Category 2
Bruce A Standby Generator Startup Controller	Category 2
Darlington Digital Controllers for Deaerator Pressure Control, Steam Generator Level Control, and Reactor Vault Cooling Control	Category 3 for all three applications of the PDC

See Larry Austin

Table i: Summary of Previous Categorizations Based on Revision 0 of this Guideline

It is recognized that the categorization process will evolve over time as more experience is gained through the use of the Guideline. The Guideline currently uses plant system reliability information, combined with a systematic assessment of the impact of software failure, to determine the appropriate software category. It is possible that, in the future, the categorization process will become even more reliability-based, by isolating the reliability requirements of the computer system of which the software is a part, and relating these requirements to overall plant safety.

Users of this Guideline are encouraged to provide feedback on their experiences with it to COG I&C Technical Committee 16.

1. INTRODUCTION

The purpose of this guideline is:

- To help categorize software in nuclear plant safety, control, monitoring and testing systems with respect to the effect of its failure on nuclear safety. Software is categorized in order to select software engineering practices that achieve sufficient assurance of the adequacy of the software, and

- To help direct the system design process towards design decisions that minimize any unnecessary reliance on software or software controlled systems with respect to nuclear safety, and where such reliance is necessary or appropriate, to ensure the software system integrity required is clearly identified, understood, and achieved. By gaining a better understanding of the safety and reliability implications of software and system design decisions, system designers may also be able to reduce the cost, effort and risk associated with developing and maintaining software controlled systems.

Software categorization is usually initiated by system and software designers who must decide on appropriate software engineering standards for their software, or who are in the process of making system design decisions, such as the allocation of functions to hardware and software, or the architecture of a computer system. The categorization process generally requires knowledge and expertise in the following areas:

- nuclear safety,
- plant systems (particularly detailed knowledge of the plant system(s) of which the software being categorized is a part, and the role of the plant system(s) in the overall operation of the station),
- reliability analysis (including human reliability analysis),
- computer systems engineering, and
- software engineering.

It may be necessary to assemble a categorization team with several members sharing expertise in these areas.

The primary output of the categorization process is a document that details the analysis and assumptions used to determine the category for a particular software system design. Indirectly, the categorization process could influence design decisions, by identifying design alternatives which permit a reduction in the categorization requirement for all or parts of the computer system. Users of this information include the system engineer and the computer system and software designers.

The following section describes the scope of this Guideline. Section 3 contains definitions of terms used in this Guideline. Section 4 describes the underlying principles of the approach to categorization, and provides an overview of the categorization process. Section 5 describes the activities in the categorization process that are undertaken at the level of the plant system, while Section 6 describes the analysis that focuses on the internal elements of the plant system. Section 7 presents the specific steps in the categorization process. Section 8 contains a suggested format for the categorization report. The final section contains heuristics that can be used in the design process when options are available to select a less stringent category.

2. SCOPE OF THE GUIDELINE

This Guideline is applicable to software that controls, monitors or tests nuclear plant processes or equipment. The methodology is suitable for categorization of software or software-controlled systems used in any nuclear plant design, although some specific categorization criteria identified make use of some CANDU-specific design principles.

The software categorization process discussed in this document does not explicitly take into account issues other than nuclear safety. It is recognized that there are other factors that could affect the software category, such as personnel safety, equipment damage, unit outage time, etc. However, users of this Guideline are not prevented from considering these factors in addition to nuclear safety, in order to arrive at a category with which to engineer the software. These factors may result in either the same or a more stringent software category.

3. DEFINITIONS

Computer System - A system composed of computer(s), peripheral equipment such as disks, printers and terminals, and the software necessary to make them operate together.

what about A/D and front end equip.

Design Authority - The person(s) responsible for ensuring that all appropriate requirements of a system are identified, and that all requirements, including functional, performance, safety, and reliability, are met.

Failure Probability per Demand - The probability that a system will fail to meet its minimum performance requirements given a demand to meet them. This term is normally applied to systems which are poised or dormant, such as special safety systems or testing systems respectively.

Fuel/High Level Waste (HLW) System - See Safety-Related Systems List.

Initiating Event (Serious Process Failure) - A malfunction of a plant system that would, in the absence of Special Safety System actions, lead to significant fuel failures or a large release of radioactivity.

Initiating Event Frequency Limit - The frequency of occurrence of an initiating event that is assumed in the licensing documentation for the plant, expressed in occurrences per year.

Minimum Performance Requirements - The minimum amount of equipment, and the minimum functional and performance characteristics of that equipment, necessary to achieve the plant system performance assumed in the safety analysis which supports the plant licence. For example, if a shutdown system has 30 shut-off rods, and the safety analysis assumes that 28 rods drop when the system is initiated, then 28 rods dropping becomes a minimum performance requirement. Similarly, the timing characteristic of the rod drop assumed in the safety analysis also becomes a minimum performance requirement.

Mitigating Provision - A part of a plant system that acts to prevent failure of the software in another part of the same plant system from causing the plant system to fail to meet its minimum performance requirements. The existence of independent mitigating provisions within a plant system is typically what permits relaxation of the software failure impact type from I to II or III.

Mitigating System - For the purpose of this Guideline, mitigating systems refer to safety-related systems that have nuclear safety-related functional requirements to reduce the consequences of an initiating event. The special safety systems are mitigating systems, but are distinguished and dealt with individually due to their more stringent unavailability requirements.

Monitoring System - A plant system that comprises data gathering equipment, data display equipment, possibly data processing capabilities, and the operator, and whose purpose is to monitor the status of safety-related equipment or processes, so that appropriate action can be taken by the operator via interaction with other plant systems, if necessary. The data gathering, processing and display portion of the monitoring system has either or both of the following roles:

- provides information from which the operator determines the acceptability of the behaviour of the system being monitored (e.g., helium pressure in LISS (Liquid Injection Shutdown System) tanks is within acceptable range, or it is out of acceptable range), and
- provides information which assists the operator's decision-making process during normal, abnormal, upset, or emergency conditions (e.g., heat transport system D_2O storage tank level is low, so operator begins to investigate possibility of small LOCA (Loss of Coolant Accident)).

If all operator interactions in response to the data display are with the same equipment used to process the field data and display it, and if the information from the operator is then used within the equipment or is transmitted to other equipment to control plant equipment or processes, then the system in question is not a monitoring system. It is either a safety/mitigating system, a process system, or a testing system. Expressed another way, the system is not a monitoring system if all operator feedback occurs within the system. The system can be considered solely a monitoring system if all operator control interactions are with equipment outside of the system. In most cases, there will be a mix of operator feedback modes, both within the system and with other systems. In such cases, the system under question should be classified as being of more than one type.

Monitoring/Testing System - A system that is either a monitoring system or a testing system. Monitoring systems typically provide continuous surveillance of the status or state of other systems (e.g., a Plant Display System) and typically have no (or very well constrained) functions that involve direct control of plant process systems. Testing Systems are typically used periodically to validate in service functionality of other process or special safety systems.

Nuclear Safety-Related Functional Requirements - The set of functional requirements associated with a safety-related system, which, if fulfilled, ensure that the system will fulfil its role in achieving acceptable levels of radiological safety with respect to the public and plant personnel.

Plant System - For the purpose of this Guideline, a plant system is a set of interconnected elements designed to achieve a functional goal. The elements may include physical process equipment, as well as computer hardware, software and humans (e.g., operators). A plant system may consist of a single element, or it may consist of several sub-systems. In this Guideline, the term plant system is not synonymous with the term computer system. A computer system may be a sub-system within a plant system.

Plant System Type - For the purposes of this Guideline, three plant system types are defined:

- Safety/Mitigating Systems,
- Process Systems, and
- Monitoring/Testing Systems.

Each of these plant system types is defined separately. Section 5 describes how the plant system classifications used in the station safety-related system list can be mapped onto the plant system types used in this Guideline. It should be noted that a plant or computer system can have more than one role. For example, a system can be both a process system and a safety/mitigating system.

Process System - For the purposes of this Guideline, a process system is a safety-related plant system whose role is to contribute to the production of electricity, either directly or indirectly. It should be noted that the operator can be part of a process system, if operator inputs are required for the control function to be implemented. In such cases, the overall system comprises the operator, the directly controlling portion of the system, and the means by which the operator interacts with the directly controlling portion of the system.

Process Control System - The portion of a process system that performs the control function. If the plant process system is controlled by software, then the software is a part of the process control system, which itself is part of the plant system.

Reliability - The probability that a system will meet its minimum performance requirements when required. Reliability is typically expressed in terms of the fraction of time a system is capable of meeting its minimum performance requirements, relative to the time the system is required to be capable of meeting its minimum performance requirements. Reliability is typically expressed as a fraction. For example, a system with a reliability of 0.999 will meet its minimum performance requirements 99.9% of the time it is required to be available. Reliability may be calculated on a "per demand" basis for some systems. See Unavailability for the converse definition.

Reliability Model - The same as the model used to determine unavailability (see Unavailability Model).

Risk Assessment - An assessment of the overall risk of the plant, usually comprising fault trees that are used to calculate the probability of severe core damage. The risk assessment can be the basis for unavailability models of safety-related systems, as well as the basis for predicting the frequencies of initiating events.

Safeguards System - See Safety-Related Systems List.

Safety-Related Function - A function of a safety-related system that the system must perform in order to meet its minimum performance requirements.

Safety-Related System - A plant system, and the components and structures thereof, that 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 plant personnel from the operation of the nuclear power plant.

Safety-Related Systems List - The operating stations (at least at Ontario Hydro) have classified safety-related plant systems according to their safety role in the plant. The following definitions are taken from the Pickering safety-related systems list (P-SRP-0.15-0, 86-03-25). An examination of the Bruce NGS B safety-related systems list indicates that the same system names are used, and that the definitions are almost identical.

- **Special Safety System** - A system designed specifically to prevent significant releases of radioactivity to the public in the event of a serious process failure (e.g., shut down system 1 and 2 (SDS1, SDS2), emergency coolant injection system (ECIS), and containment).

- **Safety Support System** - Those portions of the power, air and water process systems that are necessary to support the operation of the special safety systems (e.g., Class I, II, III or IV power).

- **Safety-Related Process System** - Process system whose failure can directly induce the operation of a special safety system to prevent regulatory limits from being exceeded (e.g., heat transport system (HTS) pipework).

- **Standby Safety System** - System that provides for ultimate cooling of the reactor following a design base earthquake or a total loss of normal plant power supplies, e.g., emergency water supply or emergency power supply.

- **Standby Process System** - System that provides additional lines of defense to safety-related process systems, e.g., reactor setback system.

- **Fuel/High Level Waste (HLW) System**[1] - Systems or components associated with the management and control of high level radioactive waste (e.g., irradiated fuel bay (IFB)).

- **Safeguards System** - Equipment required to satisfy security and safeguard requirements.

- **Secondary System**[2] - Systems or structures that, if failed, could reduce the effectiveness or induce failure of other safety-related systems.

Safety/Mitigating System - A plant system that is either a special safety system or a mitigating system.

Safety-Related Process System - See Safety-Related Systems List.

Safety Support System - See Safety-Related Systems List.

Secondary System - See Safety-Related Systems List.

Software - A set of programs, associated data, procedures, rules, documentation, and materials concerned with the development, use, operation, and maintenance of a computer system. In the context of this Guideline, software refers to the software in a computer system that is part of a safety-related system. The software in question controls the computer system.

Software Category - The software category is a number from 1 to 4 inclusive. The software category is determined from Table 7.1 in this Guideline. Category 1 is considered the most important with respect to nuclear safety, while Category 4 is considered to have no

---

[1] The system type "Fuel/HLW" is not used in the safety-related systems lists. The system definition is given in the list, but no name is given for this type of system. The name "Fuel/HLW" was created for use in this Guideline.

[2] The system type "Secondary System" is not used in the safety-related systems lists. The system definition is given in the list, but no name is given for this type of system. The name "Secondary System" was created for use in this Guideline.

importance to nuclear safety. References 11, 12 and 13 should be used to engineer software and software controlled systems in Categories 1, 2 and 3, respectively.

Software Failure Impact - The software failure impact is the impact of failure of the software with respect to the nuclear safety-related functional requirements of the plant system of which the software is a part. The impact is described in terms of the degree of degradation of the plant system's ability to meet its minimum performance requirements.

Software Failure Impact Type - The software failure impact type is a classification of the impact of software failure with respect to nuclear safety. Three types (Type I, Type II and Type III) are used in this Guideline, with Type I being the most significant with respect to nuclear safety. The specific definition of each type depends on the nature of the system (safety/mitigating, process or monitoring/testing). The definitions are given in Section 6 of this Guideline.

Special Safety Systems - High reliability safety-related systems specifically incorporated into the plant design to limit or mitigate the consequences of initiating events, thereby ensuring that any resultant release of radioactivity to the environment and the public is kept within acceptable limits. Specifically, these systems are:

- shutdown systems (SDS1 and SDS2),
- emergency coolant injection system, and
- containment system.

Standby Process System - See Safety-Related Systems List.

Standby Safety System - See Safety-Related Systems List.

System Boundary - The system boundary is a conceptual envelope around the devices that make up a system. The boundary separates one system from another. Systems interact across the interfaces of their system boundaries. Within a system, there may be sub-systems which interact with each other in order to achieve the function of the larger system.

System Safety Significance - System safety significance is a classification of the plant system in terms of its importance to nuclear safety. Each type of system is classified as either High Significance, Medium Significance or Low Significance. The criteria used to determine the safety significance class are discussed in Section 5. Classification according to system safety significance is performed solely for the purpose of software categorization, and is an interim step in the categorization process.

Testing System - A safety-related system whose function is to test a special safety system or a mitigating system for the purpose of determining if the system under test meets its minimum performance requirements.

Unavailability - In the context of this Guideline, unavailability refers to the fraction of time a system is unable to meet its minimum performance requirements. Unavailability is typically expressed in units of y/y, and has a value less than or equal to one. It should be noted that if a system is not required to perform its safety function for a period of time, then unavailability has no meaning during that time (for example, the unavailability of a testing system applies only to that fraction of time when it is intended to be in service but is unable to meet is minimum performance requirements).

Unavailability Model - A fault tree that is used to predict the unavailability of a plant system. The fault tree identifies sub-systems and their inter-relationships which can lead to failure of the subject system to meet its minimum performance requirements.

Unavailability Requirement - A limit on the maximum permissible unavailability of a plant system to meet its minimum performance requirements. The unavailability requirement is typically expressed in units of y/y. For example, an unavailability requirement of 10^{-3} means that the system can be unable to meet its minimum performance requirements for no more than 8.766 hours per year.

4. AN OVERVIEW OF SOFTWARE CATEGORIZATION

4.1 Software Categories

Four categories of software are defined in this Guideline, with Category 1 being the most significant with respect to nuclear safety. Section 7 outlines the specific procedure for determining the software category.

Software Nuclear Safety Category 1 indicates that the software under consideration is critical to nuclear safety. Such software is also referred to as "safety critical software". Ontario Hydro and AECL have produced a software engineering standard (Reference 11) for developing safety critical software, and this standard should be applied to the engineering of Category 1 software. Failure of software in this category can result in either a system with a high safety-related reliability requirement (such as a special safety system) not meeting its minimum performance requirements, or a serious initiating event in a process system (i.e., an event for which the initiating event frequency limit is very low).

same ?

The worst consequences of failure of Category 2 software can result in a serious process failure, or a degradation in the performance of a mitigating system. However, the consequences can still be mitigated effectively by special safety system action. Hence, there is a distinct reduction in safety significance compared to Category 1 software. Reference 12 should be used as the governing standard for Category 2 software.

Category 3 software can affect nuclear safety, but in a less significant way than Category 1 or Category 2 software. The worst consequences of failure of software in this category either do not prevent the affected plant system from meeting its nuclear safety-related design intent, or if they do, the affected plant system has a low safety significance. Hence, there is a further reduction in safety significance compared to Category 2 software. Reference 13 should be used as the governing standard for Category 3 software.

Category 4 is assigned to software whose failure has no effect on nuclear safety.

The assignment of a software category is independent of the software engineering standard applied to that category. The current standards for Categories 1 to 3 are References 11 to 13, respectively. The content of these standards may change over time, but this should not affect the categorization.

The software engineering standards referred to above are applicable to custom developed software. However, the degree of rigour associated with qualification of predeveloped software should also take into account the results of categorization.

4.2 The Basis of the Categorization Process

The software engineering standard applied to Category 1 software provides the highest level of assurance that the software will meet its safety-related requirements. A relaxation in the degree of software engineering rigour for Categories 2, 3 and 4 is based on the reduced safety significance of the software in these categories. That is, the underlying premise of software categorization is that, as the safety significance of the software decreases, less effort needs to be expended to demonstrate that the software meets its requirements.

This approach is consistent with a risk-based approach to nuclear safety, where the risk associated with the failure of a system is a function of both the probability of failure and the consequences of failure. The higher the risk associated with system failure, the higher must be the assurance that the software will not contribute to that failure.

An acceptable level of plant risk is achieved by designing the plant to have a low probability of serious process failures, and by providing mitigating systems which minimize the consequences of serious process failures, should they occur. There is a finite probability that mitigating systems may also fail to perform their safety function when called upon, so redundant mitigating systems are provided, or when this is not practical, the mitigating systems are designed to very high reliability (e.g., special safety systems). An overall goal is to achieve a probability of severe core damage in the range of 10^{-5}-10^{-7} occ/y, where severe core damage can occur only through a combination of certain serious process failures in conjunction with failure of certain mitigating systems.

As shown in Figure 4.1, there exists in any nuclear plant, a hierarchy of systems established to ensure that the overall plant risk goals are achieved. Within this hierarchy of plant systems, process systems limit the frequency of serious process failures. Mitigating systems (including special safety systems as a particular type of mitigating system) represent additional barriers to the release of radioactivity in the event of a serious process failure.

Figure 4.1 also shows that monitoring and testing systems play a role in limiting the release of radioactivity. Monitoring systems, which monitor process and/or special safety systems, provide the operator continual information on the state-of-the-plant and may be relied upon by the operator to detect a serious process failure. Monitoring and testing systems associated with mitigating systems provide information on the ability of those systems to perform their safety functions. They also provide the operator with information on the status of those systems, and the plant parameters associated with those systems. Monitoring and testing systems do not represent physical barriers to release but do play an important role in the overall safety of the plant by providing information on both process and mitigating systems to assist the operator with diagnosis and corrective action.

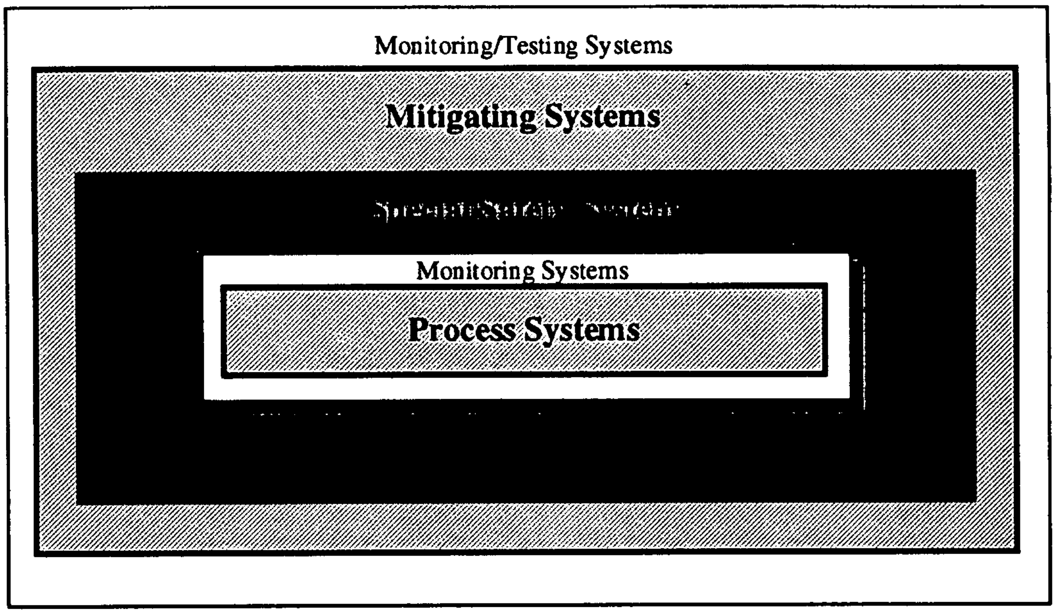

Figure 4.1: Hierarchy of Plant Systems

It is important to note that a given safety-related plant system may have one of several roles in the overall safety of the plant. As well, the significance of a given system with respect to plant safety will vary from system to system, and the role of any software in that system may, or may not, be linked directly to the plant system's overall role in the safety of the plant. The software categorization process takes all of these factors into account. The process of determining the software categorization involves two basic phases, each comprised of several steps.

Phase I, "Determining the Plant System Safety Significance", involves identifying the safety significance of the plant system of which the software to be categorized is a part. Three levels of plant safety significance are used in this Guideline: High, Medium and Low (a fourth level, no safety significance, would result in software of Category 4). The assigned safety significance is determined by following a step-wise (and sometimes iterative) procedure of determining the plant system type and its reliability requirements. Section 5 describes this phase in detail. Although a software category can be assigned based solely on consideration of the safety significance on the plant system (e.g., Category 1 to High significance, Category

2 to Medium and Category 3 to Low), the categorization process provides a means of identifying and taking credit for any factors that reduce the software's safety significance.

Phase II, "Determine the Software Failure Impact Type and Category", involves identifying and classifying the worst possible software failure modes and effects, in terms of their potential impairment of plant system safety functions. This is based on the possibility that the software may not actually implement or affect the functions which dictate the plant system safety significance, or there may be other provisions within the plant system which compensate for failures of the software. The software failure impact type is assessed to determine if the software category can be adjusted to a lesser category than that associated with the plant system's intended reliability design requirements. Achieving a lesser categorization result will result in applying a governing standard that imposes less rigorous software engineering methods, without compromising plant safety. Section 6. describes this phase in detail.

The philosophy and rationale behind the approach to determining the plant system safety significance (i.e., phase I of the categorization process) is based on a few key assumptions. If the plant system of which the software is a part is a special safety system or another type of mitigating system, the safety significance of the plant system can be determined from its unavailability requirements. For example, the unavailability of special safety systems must typically not exceed 10^{-3} y/y, whereas the unavailability of typical mitigating systems must not exceed 10^{-2} y/y. The more stringent unavailability requirements for special safety systems are indicative of their greater importance to safety. That is, in order to achieve an acceptable risk, the probability of failure of these systems must be the lowest of all safety-related systems, because the consequences of failure are high. Hence, for special safety systems, the unavailability requirement is used as an indicator of safety significance. The more stringent the unavailability requirement, the higher the system's safety significance.

Plant systems which perform a process role do not have an unavailability requirement related to nuclear safety. Instead, the initiating event frequency limit is used as an indicator of safety significance. Generally, plant systems with a lower initiating event frequency limit are more significant to nuclear safety. A plant system that is engineered such that its initiating event frequency limit is 10^{-3} occ/y, can be expected to have a greater consequence of failure than a plant system with an initiating event frequency limit of 10^{-1} occ/y. In addition, for high frequency events, there are usually multiple systems available to mitigate the consequences of an initiating event. In summary, for process systems, the categorization process is based on the assumption that the lower the initiating event frequency, the greater the system's safety significance.

Increasingly, monitoring and testing systems are being installed or modified to include software. A testing system is one which automates all or part of the process of testing a special safety system or a mitigating system. Software in such systems can be used to control the test and/or to interpret and report on whether or not the system passed the test. A monitoring system is one that samples data that is used or produced by a safety, mitigating or process system, and reports this information to the operator. The operator makes decisions, and may take actions, based on this information. Hence, the operator is an integral part of the monitoring system. Because monitoring systems provide the operator with information which the operator uses to make decisions, it is conceivable that some monitoring systems may fail in a way that causes the operator to take action which causes an initiating event. Similar to safety/mitigating systems, the safety-related unavailability requirement of monitoring and testing systems is used to indicate the safety significance of the plant system in which the software to be categorized is a part.

A given plant system may have more than one role, so it may be classified as more than one type of system. For example, a plant display system provides a monitoring role (i.e., it is a monitoring/testing system), but if the operator uses the plant display system to enter data that is then communicated to other plant systems which use the data to perform control functions, then the plant display system also has a process role (process system). Mitigating systems often have a dual role. For example, the shutdown cooling system is normally used to remove decay heat when the reactor is shutdown, but it may also be called into action as a heat removal system under accident conditions. Therefore, it has both a process and a mitigating role. The categorization process recognizes the possible multiple roles of plant systems.

Operating stations classify plant systems according to their safety-related function in order to identify those plant systems that need special emphasis because of their importance to safety. The system classifications are documented in each station's Safety-Related Systems List and can be used to identify the type of plant system for categorization purposes. Table 5.1 in Section 5 of this Guideline describes how the plant system types used in the safety-related systems list are mapped onto the plant system types used in this Guideline.

Phase II, an equally important phase of the categorization process, is the determination of the software failure impact type. There are three failure impact types defined. Section 6 describes the criteria for selection of the failure impact type in detail. If a system is determined to be type I, the underlying assumption is that the safety significance of the software should be treated as equal to the safety significance of the plant system of which it is a part. Assigning a Type I software failure impact type assumes that the software may fail in such a way as to maximize the safety-related consequences of failure of the plant system.

The criteria for software failure impact types II and III are based on the assumption that the worst possible failures of the software or software controlled system have a respectively lesser affect on the performance requirements of the plant systems (in which it is a part). Moving from software failure impact type I to II, or from type II to III usually allows a lesser category to be selected (see Table 7.1). Section 6 describes the process and criteria of determining software failure impact type in full detail. Thus the determination of the software failure impact type takes into consideration the role and interaction that the software or software controlled system has on the larger plant systems with which it interacts (or forms a part).

Once both the safety significance of the plant system and the software failure impact type are determined, Table 7.1 (Section 7) is used to select the software category. The cell at the intersection of the relevant row and column indicates the category of the software under consideration that is applicable. Sections 5, 6, and 7 provide more complete details on the categorization procedure.

5. DETERMINING PLANT SYSTEM SAFETY SIGNIFICANCE

A plant system is comprised of sub-systems and/or discrete components whose combined purpose is to achieve a stated function or goal. The software or software controlled computer system being categorized is typically one of these sub-systems or components. The operator may also be considered a component of the system. It is important to note that the software being categorized may be part of or interact with more than one plant system, or indirectly affect other sub-components of the computer system which in turn affect other plant systems. Figure 5.1 illustrates these simple concepts, and shows two independent plant systems which have both local and shared components of a software controlled computer system. Dependencies may exist between the systems, sub-systems, and components involved which may result in common modes of failure.

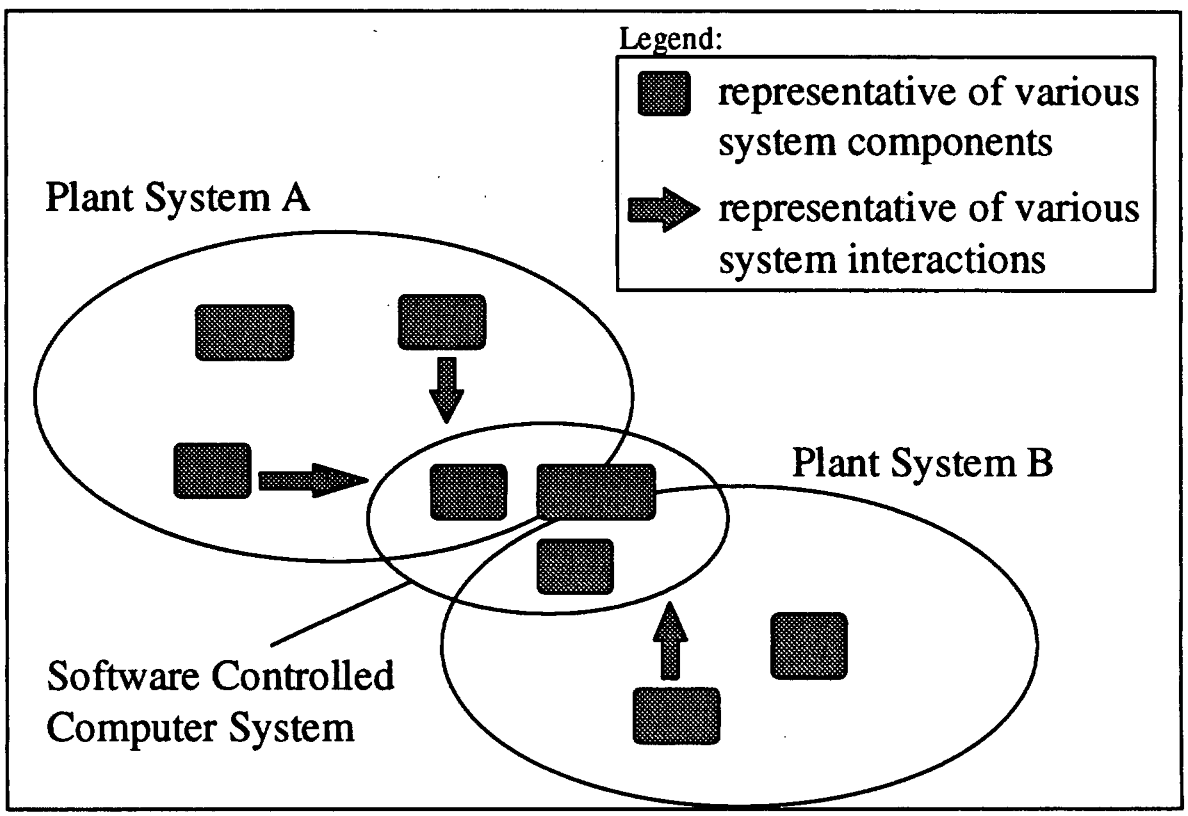

Figure 5.1: Typical Plant System and Software System Boundaries

Identification of the plant system boundaries is necessary in order to ensure a consistent approach to determining the safety significance of the plant system and the possible impact of software failure with respect to nuclear safety. It also provides a framework for systematically addressing the interdependencies and interactions of systems and sub-systems. Establishing the plant system boundaries is achieved by determining the plant system(s) in which the software forms a part or interacts with and understanding the role(s) of that system with respect to plant safety functions.

For the purpose of determining the safety significance, the system boundary is usually drawn around a well recognized plant system that has an identified unavailability requirement or an associated initiating event frequency limit (depending on the type of the system). These two parameters are typically available from existing safety and licensing documentation, and are used to determine the plant system safety significance. If the software performs a function that is not the prime function of the "larger" plant system, the boundary would likely still be drawn around the larger plant system, simply because the unavailability requirement or initiating event frequency limit information is typically more readily available for major plant systems. The assessment of the software failure impact type accounts for the fact that the software may perform only a secondary function within the plant system.

As an alternative choice, the boundary can be drawn around the sub-system of which the software is directly a part. In this case, failure of the software usually results in failure of the chosen system to perform its primary function, which can then impact on the larger system. This approach can be taken if there is information on the safety-related unavailability requirement or initiating event frequency limit of the sub-system.

If the plant system involved is a monitoring system, draw the boundary to include the operator, because the operator is essential if the monitoring system has an impact on safety. If the plant system involved is a testing system, the boundary may or may not include the plant system under test within the boundary. Including the system under test would probably result in a conservative assessment of the plant system safety significance, as the reliability requirement of the system under test would normally be more stringent than that of the testing system. The analyst should be aware that in some cases this may not be an option, as in the case where a testing system directly interfaces to and/or controls the system under test, and could damage the system or leave it in an inoperative state.

Identification of the type of system is important because the software category depends on the safety role of the plant system. Different plant system types have different safety roles. Some plant systems can have more than one role. For example, a plant system can have a mitigating and a process role. In such cases, the analysis should be carried out for both roles/types of plant system.

The safety role of the plant system in question can often be determined as well from the station's Safety-Related Systems List. Each station has such a list, which is used to identify those systems which require special emphasis (in varying degrees) in the operation of the station. All activities performed in the station are subject to approved work procedures. For some safety-related systems, additional requirements are imposed so the plant system design intent and availability can best be maintained. The Safety-Related Systems List identifies these systems.

This Guideline is consistent with the concept of the Safety-Related Systems Lists, in that additional requirements are imposed on the computer system and software engineering process, depending on the safety significance of the software. The Safety-Related Systems List can often be used to identify the plant system type. Table 5.1 shows how the plant system types defined in the Safety-Related Systems List can be mapped to the plant system types described in this Guideline. Definitions of the safety-related system types are provided in Section 3.

Table 5.1: Plant System Classification

Safety-Related Systems List System Type	Guideline Plant System Type
Special Safety System	Safety/Mitigating
Safety Support System	Safety/Mitigating
Standby Safety System	Safety/Mitigating
Safety-Related Process System	Process
Standby Process System	Process
Fuel/High Level Waste (HLW) System	Process
Secondary System	Safety/Mitigating or Process or Monitoring/Testing

It is possible that a given plant system may not be on the list, even if it is safety related. Note also that some plant systems may be designated as being more than one type. For example, it may occur that a plant system has both a mitigating and a process role. If it can be determined the software is part of a plant system on the Safety-Related Systems List, then Table 5.1 can be used to identify which type of plant system type is applicable to perform the software categorization. If the safety-related system type is a Secondary System (see Table 5.1), the role of the plant system could be a safety/mitigating system, a process system, and/or a monitoring/testing system. In this special case, the exact role(s) and thus type(s) should be determined from other station documentation, and the appropriate plant system type selected for categorization. For all other safety-related system types, determining the role(s) (or type(s)) of the plant system from Table 5.1 is straightforward.

When referencing Table 5.1, the analyst should always verify the correctness of the result by confirming the safety-related role of the plant system as given in the current plant safety analysis. This check ensures that the selection of the plant system type is based on the most current information. It may be, for example, that the current work for which the categorization is being performed affects the safety-related role of the plant system. This may require a re-evaluation of the safety-related system type. The analyst should also determine if the plant system is classified by the station under more than one safety-related system type. In such cases, the categorization should be carried through for all types relevant to the plant system under consideration.

If the role of the plant system is not known and the system is not on the station's Safety-Related System List, then the plant system type must be determined from other information, such as the Safety Report and other safety analysis documents or operating documentation such as the Operating Policies & Principles (OP&P), Abnormal Incidents Manuals (AIMs), or the Emergency Operating Procedures (EOPs).

5.1 Special Safety Systems and Mitigating Systems

The assigned system safety significance for special safety systems and mitigating systems is derived from the unavailability requirement (Q) identified for the plant system, where the unavailability requirement is with respect to the system's safety-related function. The following definitions are applied:

u ?

High Significance[3]	$Q \leq 10^{-3}$ y/y
Medium Significance	$10^{-3} < Q < 10^{-1}$ y/y
Low Significance	$Q \geq 10^{-1}$ y/y

(Note: For Pickering NGS A, the value of 10^{-3} y/y for High Significance is replaced with a value of $3x10^{-3}$ y/y, to reflect that station's reliability requirement for special safety systems).

The delineation between High and Medium safety significance is based on the distinction in the CANDU design between special safety systems and other mitigating systems. The former have the most stringent unavailability requirement of all systems which have a protective or mitigating function. Hence, special safety systems are considered in the High safety significance category.

The unavailability requirement can usually be obtained from the plant risk assessment or the reliability model of the plant system. If such information is not available, it is appropriate to use the information for a similar plant system at another plant for which the information is available, provided that differences in the system design are not significant, and that the component failure rates are similar. If there is still doubt about the plant system safety significance, and there is certainty that the plant system is not a special safety system, the Medium safety significance level should be selected. This will result in a conservative categorization for the software.

5.2 Process Systems

For process systems, the system safety significance is based on the initiating event frequency limit for failures of the process system. Data are available from plant risk assessments, which show expected initiating event frequencies.

If such information is not available from the risk assessment, it is acceptable to use the initiating event frequency limit for the same plant system for another plant, provided that the system design is similar and the component failure rates are expected to be similar. In the absence of such information, it may be possible to determine the initiating event frequency based on Table 5.2, which is taken from the Darlington Safety Report (Chapter 3, Table 1.1).

---

3 An unavailability requirement of 10^{-3} y/y is interpreted to mean that there is a 1 in 1000 chance that the system will not meet its minimum performance requirements. In practical terms, this means that the system must be able to meet its minimum performance requirements for all but 8.766 hours per year.

Table 5.2: Initiating Event Frequency Limits

Event Class from C-6	Qualitative Event Frequency Criteria	Quantitative Event Frequency Limit, f (expected frequency in occurrences/reactor-year)
1	Greater than 50% chance of occurring in the lifetime of a single reactor, or more frequently than twice in the lifetime of a 4-unit station	$f > 10^{-2}$
2	About once in the lifetime of an 8-unit station	$10^{-2} \geq f > 10^{-3}$
3	About once in the lifetime of a population of one hundred similar reactor units	$10^{-3} \geq f > 10^{-4}$
4	Low probability postulated failure	$10^{-4} \geq f > 10^{-5}$
5	Very low probability postulated failure	$f \leq 10^{-5}$

Table 5.2 refers to AECB consultative document C-6 [14], which identifies 5 event classes for evaluation. There is a list of initiating events associated with each event class. Ontario Hydro has identified a range of initiating event frequency limits corresponding to each event class. Therefore, if the initiating event for the plant system under consideration is identified, the event class for that event can be determined from C-6, and the initiating event frequency limit can be obtained from Table 5.2. If there is more than one initiating event associated with the plant system, the analysis should be carried through for all initiating events, and the most restrictive applied.

For the purposes of categorization, the following definitions of process system safety significance are applied, based on the initiating event frequency limit (f in occurrences/y):

High Significance	$f \leq 10^{-3}$	(C-6 Class 3, 4, 5)
Medium Significance	$10^{-3} < f \leq 10^{-2}$	(C-6 Class 2)
Low Significance	$f > 10^{-2}$	(C-6 Class 1)

High safety significance corresponds to event classes 3, 4 and 5. These represent the most severe initiating events, such as large break Loss of Coolant Accident (LOCA), as well as event combinations (which are not relevant for the purposes of categorization). Medium safety significance corresponds to less frequent events, and Low safety significance to events of relatively high frequency, but low consequence. It is expected that there will be no process systems containing software that have High safety significance. This is because process systems with such low initiating event frequencies are usually passive (e.g., piping). Hence Table 5.1 shows that, typically, no plant systems from the safety-related systems list map onto the High safety significance classification for process systems. However, the possibility of a high safety significance for a process system is not precluded in the categorization methodology.

5.3 Monitoring/Testing Systems

The approach for monitoring/testing systems is similar to that for safety/mitigating systems, in that the safety significance is based on the unavailability requirement of the monitoring/testing system. Although the unavailability requirement for the monitoring/testing system may not always be readily available, it may be possible to identify an appropriate requirement by examining the risk assessment model or the safety system reliability model. Failure of the monitoring/testing system to perform its function can be added as a failure mode, and an unavailability requirement determined such that the overall unavailability of the plant system being monitored or tested is not significantly affected.

To establish the unavailability requirement of a testing system, the unavailability model of the system being tested can be expanded to include failure of the testing system as a contributor to failure of the system being tested (if it is not already included in the model). The consequences of failure of the software in the testing system can be treated in the unavailability model as equivalent to increasing the test interval, if the software failure can cause the test results to be erroneous but not detected as such. The test interval should be increased in the unavailability model to a value equal to the time required to detect that the software error has occurred, using other system surveillance techniques. If there is no creditable mechanism that could detect a latent software error, then the test interval should be considered equal to the plant design life.

It should also be noted that the unavailability requirement for a testing system should be based on its failure probability per demand. This is because such systems are required to operate only for limited time periods, but when they do operate, they are required to be reliable. For example, if the testing system unavailability requirement is $<10^{-2}$ y/y, but the system only operates 26 hours per year, then the system can be out of service at other times and not impact the unavailability. However, when the system is required to operate, it would meet its unavailability target if it was unavailable for < 0.26 h of the 26 hours.

If the operator is within the boundary of the monitoring or testing system, then it is important to include the operator in the reliability model of the system. Human reliability models are typically incorporated into plant risk assessments, to ensure that the role of the operator is properly recognized in the overall plant risk model. For the purposes of software categorization, the role of the operator with respect to the nuclear safety-related functions of the monitoring/testing system must be determined, along with the types and possibilities of human error during interaction with the software (see Appendix B). Fault trees that include failure modes attributable to the operator can then be included in the overall reliability model of the plant system being monitored or under test. The human reliability must then be estimated, and appropriately combined with the reliabilities of other system components. Once this is done, an appropriate reliability requirement for the monitoring/testing system can be determined.

As a general rule, if the unavailability requirement for the monitoring/testing system cannot be obtained, the unavailability requirement for the plant system being monitored or tested should be used (as the default value). For example, applying this rule to a plant system that tests a special safety system results in a High safety significance, whereas applying the rule to a plant system which tests a mitigating system would typically yield a Medium safety significance. If the plant system being monitored or tested is a process system, then it would typically yield a Medium safety significance result.

For monitoring/testing systems, the following definitions of plant system safety significance apply:

High Significance	$Q \leq 10^{-3}$ y/y
Medium Significance	$10^{-3} < Q < 10^{-1}$ y/y
Low Significance	$Q \geq 10^{-1}$ y/y

where Q is the unavailability target in y/y of the monitoring/testing system. (Note: For Pickering NGS A, the value of 10^{-3} y/y for High Significance is replaced with $3x10^{-3}$ y/y, to reflect that station's reliability requirement for special safety systems.)

The definitions of High, Medium and Low safety significance for monitoring/testing systems are the same as for safety/mitigating systems. This is consistent with the underlying principle that the unavailability requirement of a plant system is an adequate indicator of the safety significance of the system.

6. DETERMINING THE SOFTWARE FAILURE IMPACT TYPE

The software failure impact type is a classification of the effect of the worst-case failure of the software on the ability of the plant system to meet its minimum safety related performance requirements. The criteria that define the software failure impact types ~~are~~ chosen to be consistent with the approach to classifying safety ~~system~~ impairments and significant events at the operating stations (such as defined in the AIM). Three software failure impact types are defined for each type of plant system, with TYPE I representing a failure with the greatest consequences with respect to nuclear safety, and TYPE III the least. If the worst-case software failure is not TYPE I, then it is possible to reduce the stringency of the software category, because the role of the software within the plant system is less significant from a safety perspective than the role of the overall plant system.

Assessment of the software failure impact type should identify all possible safety-related impacts of software failure on the plant system, for all possible plant system types relevant to the plant system in question. For the purposes of categorizing the software, the most severe software failure impact type should be used. However, it is important to recognize that the limiting software failure impact type may occur only in specific circumstances, and that minor design modification or reconfiguration of the plant system could reduce the software failure impact type.

Within the plant system there may be sub-systems that perform multiple functions. The assessment of the software failure impact type must consider the impact of each function on all other functions. For example, when software that performs a particular function in the DCC's (Digital Control Computers) is being categorized, the impact of failure of this software on the other functions performed by the DCCs must be considered.

The assignment of a software failure impact type other than Type I must be based on an analysis of the role of the software with respect to the safety-related function of the system, and on the independent mitigating provisions within the plant system which can mitigate the consequences of failure of the software. The following sub-section provides guidance on performing this analysis, both at a level that examines the interactions with other plant level systems and/or sub-systems (Section 6.1.1), and at a level which examines the computer sub-system and the software itself (Section 6.1.2). Sections 6.2 to 6.5 provide definitions of the software failure impact type for the different types of plant system.

6.1 Consideration of Mitigating Provisions

6.1.1 Mitigating Provisions at the Plant System Level

When carrying out the assessment of software failure impact type, redundant or mitigating provisions within the plant system boundary should be accounted for, as they may lessen the impact of software failure. The following factors should be considered when assessing the software failure impact type:

(a) A redundant or mitigating provision within the plant system can be credited if it is independent of the software. If the cause of the software failure can also cause the redundant or mitigating provision to fail or be ineffective (i.e., if there is a common mode failure), then the redundant or mitigating provision cannot be credited. An example of a feature which could be credited would be a hardware interlock that prevents unsafe action by the plant system even if the unsafe action is commanded by the software.

(b) If the operator is a part of the plant system, and there are specific operating procedures which (independent of the software) would clearly make the operator aware of the need for action, and if the operator reliability for such action is commensurate with the overall reliability of the plant system (see discussion in Section 5.3 and Appendix B), such action can be credited with reducing the software failure impact type.

(c) The same version of the software performing the same function and running on a different computer cannot be considered to provide a mitigating function, because the same cause of software failure (e.g., a heretofore undetected software error) could manifest itself on both computers. (Note: Redundant systems do improve overall reliability due to decreased dependency on a single hardware failure.)

(d) Mitigating provisions outside of the plant system boundary should not be credited in the assessment of software failure impact type. Although other plant systems that can reduce the safety-related impact of failure of the plant system of which the software is a part, they have already been taken into account in the classification of the plant system safety significance (i.e., the presence of such plant systems will have been factored into the unavailability requirement or the initiating event frequency limit parameters for the plant system in question). Taking credit for these external mitigating provisions in the assessment of software failure impact type would, therefore, be taking double credit for a single mitigating provision. If, however, it can be shown that a mitigating provision outside of the plant system has not been taken into account in the risk assessment, then it is acceptable to credit such a provision in reducing the software failure impact, as long as such credit is consistent with licensing restrictions (such as not crediting the first trip parameter of a shutdown system).

(e) Consider the existence of any independent system component (e.g., software) that mitigates failure of the software being categorized. For example, if failure of the software being categorized can cause an initiating event, implement software that independently checks for such an event and mitigates it effectively. Ensure that the mitigating component is independent of the software being categorized. When considering this, past experience with similar systems may be consulted to identify examples of unintended interactions, which could be taken into account.

If the Guideline is being used to assist with design decisions, then the above discussion can be viewed from another perspective. The plant system can be designed to include independent redundancy or a mitigating provision, thereby reducing the software failure impact type and, hence, the software criticality level (i.e., the software category). The designer can assess the tradeoffs associated with providing such provisions, as compared to designing the software to a higher criticality level. The assessment of software failure impact type may show that the most stringent impact type results from a specific cause, which can be remedied relatively easily. Therefore, the categorization process can also serve to identify to the design authority mechanisms by which the software category can be reduced without compromising safety (see Section 9).

6.1.2 Mitigating Provisions at the Software and Computer System Level

This section describes how a computer system's design attributes may be credited as mitigating factors in determining the failure modes and impact type, and may result in a change in categorization. This is an additional level of analysis in the categorization process, based on failure modes and effects analysis performed at the computer system and software function level. It is done if there appear to be justifiable arguments to change the categorization for all or parts of the computer system, based on creditable design attributes. If successful, this additional effort may allow the analyst to change the categorization requirement for all, or parts, of the computer system, as appropriate. Note that this process will typically require interaction and iteration between the analyst and the computer system designer or design authority. The following is a framework for analysis and decision making in considering computer system design attributes in the categorization process.

A basic characteristic of computer systems is the inter-dependency between all software and hardware components in that system. Software inter-dependencies may be due to shared computer system resources, or common interfaces. Identifying and understanding these inter-dependencies are a key factor in assessing the impact of a possible software failure mode. The failure probability of an individual software function will be in part determined by the reliability of these inter-dependencies. Obvious examples of such inter-dependencies between components in a computer system would include shared hardware, a common operating system, shared data files, shared memory, shared peripheral devices, and common interfaces.

In general, for the purposes of categorization, the design attributes to be considered for their mitigating role in software failure include:

a) the effectiveness and reliability of failure handling mechanisms in the computer system, and,

b) the degree of isolation or inter-dependency between the various software functions, computer system components, and/or failure handling mechanisms.

Design attributes should always considered in the context of common points of failure as imposed by the system design and architecture. Failure handling mechanisms may provide failure detection, failure prevention, failure isolation, and failure recovery functions within the computer system. They should be considered in combination with hardware or software functionality, architecture, and human interaction within the system.

The design attributes in a computer system may be considered in the evaluation of the failure modes and impact types. For most failure modes identified, the focus will be placed on establishing "creditable" isolation and independence between the software functions in question (including any failure handling mechanisms).

Creditable isolation and independence between two functions is either:

a) complete isolation and independence as achieved when two software functions have no interaction and exist on autonomous systems with no shared resources, interfaces, or common points of failure, or

b) a well-constrained degree of inter-dependency exists, but which is guarded against by qualified failure handling mechanism(s). In this context, qualified means establishing that the reliability and maintainability of the failure handling mechanism(s) is (are) commensurate with the category of the most critical software function(s) in the computer system.

Note that for installed (existing) systems, it may be possible to document an extensive usage history to qualify a failure handling mechanism. Obviously, the requirements for the qualification will be higher in cases where the failure handling mechanism is used to safeguard a Category 1 system.

Appendix A contains three tables that can be used to assist in the identification and consideration of computer system and software design attributes that may be credited in determining the software failure impact type:

- Table A.1: Common Inter-dependency Issues to be Considered,
- Table A.2: Common Failure Handling Mechanisms to be Considered, and
- Table A.3: Specific Issues to Consider in Software Failure Mode and Impact Analysis.

As a simple example of crediting a software and computer system design attribute, consider the case of software in a computer system relied upon to alert the operator that a safety-related action is required. Credit can only be taken if it can be shown that the software being categorized cannot fail in such a way as to cause the operator to fail to be alerted. Active or passive mitigating provisions (ie. design attributes) within the computer system on which the software is running may be credited if it can be shown that there are no failure modes of the software which can render these mitigating provisions ineffective (See Section 6.1.2).

6.2 Special Safety Systems and Mitigating Systems

For special safety systems and mitigating systems, the software failure impact types and their criteria for determination are:

TYPE I: The designed nuclear safety functions will not be available for some or all process system failures (i.e., the safety-related function is not performed at all).
OR
The minimum performance requirements of the plant system will not be met for some or all process system failures (i.e., the safety-related function is performed, but is not effective enough to meet the minimum performance requirements).

TYPE II: The system's functional performance is degraded for some or all process system failures. However, the minimum performance requirements of the plant system will be met for all process system failures (e.g., the timing of a safety-related action may be slower than the nominal design value, but still fast enough to meet the minimum performance requirements)
OR

The system's designed redundancy is lost such that the probability of the plant system not meeting its minimum performance specifications is increased (e.g., one shut-off rod fails to drop, but the minimum performance requirements are met as long as no more than 2 rods fail to drop).

TYPE III: The software failure has no impact on the nuclear safety functions of the plant system. The built-in redundancy of the plant system is not compromised and the safety-related functional performance is not affected.

If the operator is defined to be within the system boundary, and if software failure can cause erroneous operator action to be taken (including no action when action is required), then such a failure mode of the software should be included in the assessment of software failure impact type.

6.3 Process Systems

For process systems, the software failure impact types and their criteria for determination are:

TYPE I: The software failure can directly or indirectly cause significant fuel failures or a large release of radioactivity in the absence of special safety system or other mitigating system actions.

TYPE II: The software failure can affect the plant system such that the fuel temperature can be raised directly or indirectly as a result of the software failure. However, fuel failures or a large release of radioactivity cannot occur even in the absence of special safety system or other mitigating system actions.
OR
The probability of a plant system failure that can cause fuel failures and large release of radioactivity in the absence of special safety system or other mitigating system actions is increased.

TYPE III: The software failure has no impact on the safety-related reliability performance of the nuclear safety functions of the plant systems.

If the operator is defined to be within the system boundary, and if software failure can cause erroneous operator action to be taken (including no action when action is required), then such a failure mode of the software should be included in the assessment of software failure impact type.

6.4 Monitoring Systems

For monitoring systems, the software failure impact types and their criteria for determination are as follows:

TYPE I: Failure of the monitoring software can directly cause an initiating event in a process system.
OR
Failure of the monitoring system software results in incorrect operator action, which results in the failure of a special safety or mitigating system to meet its minimum performance requirements.

TYPE II: The software fails in its monitoring function in a manner that causes the operator not to take action that is credited with mitigating a serious process failure.
OR
The software failure causes the operator to take action which initiates a serious process failure.
OR
The software failure increases the probability of incorrect operator action leading to the impairment of a special safety or mitigating system, such that the system fails to meet its minimum performance requirements.

TYPE III: The software failure has no impact on the information provided to the operator.

A Type I impact involves the initiation of a serious process failure or the impairment of a safety related system by monitoring software. It is not expected that present monitoring system designs can result in serious process failures, but this possibility in the future cannot be precluded. If, for example, intrusive monitoring systems are implemented in the future, such that the monitoring system temporarily opens a valve in a process system to obtain data, it is conceivable that the monitoring system can fail in such a way as not to close the valve, thereby initiating a small LOCA. If the monitoring system is controlled by software, then such a failure mode could initiate a serious process failure.

There are examples in present plant system designs where monitoring system software failure could impact on safety related system performance through the operator. For example, the value for reactor thermal power is used to calibrate shutdown system in-core detectors. Incorrect information could lead to failure of the reactor to trip on high power in the event of an uncontrolled reactor power increase. The operator relies on a monitoring system to obtain the thermal power value, and therefore failures of software in the monitoring system could contribute to a safety system impairment.

6.5 Testing Systems

For testing systems, the software failure impact types are as follows:

TYPE I: Failure of the testing software causes the designed nuclear safety functions of the system under test not to be available for some or all process system failures (i.e., the safety-related function is not performed at all).
OR
Failure of the testing software causes the minimum performance requirements of the plant system under test not to be met for some or all process system failures (i.e., the safety-related function is performed, but is not effective enough to meet the minimum performance requirements).

TYPE II: The software failure causes an inaccurate test result (e.g., by reporting inaccurate test results such as when a test fails but is reported as passing or when a test is not carried through to completion).
OR
Failure of the testing software causes the functional performance of the system being tested to be degraded for some or all process system failures, however the minimum performance requirements of the plant system will be met for all process system failures (e.g., the timing of a safety-related action may be slower than the nominal design value, but still fast enough to meet the minimum performance requirements).
OR
Failure of the testing software causes the designed redundancy to be lost such that the probability of the plant system not meeting its minimum performance specification is increased (e.g., one shut-off rod fails to drop, but the minimum performance requirements are met as long as no more than 2 rods fail to drop).

TYPE III: Failure of the testing software has no impact on the test or on the safety-related performance of the system under test.

A Type I impact involves impairment of the system under test as a result of failure of the testing software. This could occur, for example, if the testing software valved out important components of the system under test, and did not return the system to its original configuration after the test. Hence, it is conceivable that this failure effect could occur.

7. THE CATEGORIZATION PROCEDURE

The categorization procedure assumes that as the body of categorization work increases over time, categorization information will be re-used and reports will reference each other. As an example, the categorization of a sub-function within a computer system for which a categorization report has already been completed may only require a short summary which cites the original work and highlights any additional issues.

The categorization procedure is shown schematically in Figure 7.1 (below). The basic steps are as follows:

Phase I: Determining the Plant System Safety Significance:

1. **Identify the Plant System(s) Involved**. This involves determining which plant systems the software or software controlled systems form a part of or interact with. If more than one plant system is identified, carry out the remaining steps in the analysis for each plant system. This also involves determining the role of the software by evaluating the functions and interaction of the software or software controlled computer system in the plant system identified. Specifically, any functions in the software or software controlled computer system that may affect the function or performance of the plant system nuclear safety functions must be identified.

2. **Determine the Plant System Type(s)** (i.e. role(s)). This is achieved by identifying the plant system's nuclear safety functions and determining whether it is a special safety system or mitigating system, a process system, or a monitoring/testing system. If a plant system has more than one role, carry out the remaining steps in the analysis for all relevant roles.

3. **Establish a Suitable Plant System Boundary.** The selection of the plant system boundary may be influenced by available data from the plant Safety Analysis or Risk Assessment Reports for the station. Typically, either the unavailability requirement (for special safety systems or mitigating systems and monitoring/testing system) or the initiating event frequency limit (for process systems) is available for major plant systems. (See Sections 4.2 and 5)

4. **Determine the Plant System Safety Significance.** Using the definitions of system safety significance as appropriate for the plant system type (see Sections 5.1. 5.2, and 5.3), determine the plant system safety significance as High, Medium or Low (see Section 5 for detailed guidance).

Phase II: Determine the Software Failure Impact Type and Category:

5. **Identify All of the Software Failure Modes/Effects.** This step involves an assessment of the interactions of the relevant sub-systems that comprise the plant system, to determine the possible failure impacts of the software being categorized. All conceivable failure modes of the software or software-controlled computer system should be identified and examined to determine their possible impact on any plant system nuclear safety functions. An additional optional step (i.e. Step 5.b in Figure 7.1 below) may be taken whereby the software and computer system design attributes are considered and possibly given credit for preventing or minimizing specific failure modes (refer to Section 6.1.2).

6. **Determine the Limiting Software Failure Impact Type** (i.e., for all failure modes). This involves applying the classification criteria outlined in Section 6. of this Guideline as appropriate for the plant system type under consideration. If this step is not (or cannot) be performed, than a Type I failure impact should be assumed.

7. **Determine the Software Category from Table 7.1.** The category is found in the table at the intersection of the row corresponding to the plant system's safety significance and the column corresponding to the limiting failure impact type.

8. **Determine the Limiting Software Category**. This step is necessary only when there is more than one plant system involved (i.e. as identified in Step 1 above), or if a plant system can be classified as more than one type (i.e., as identified in Step 2 above), the software category should be determined for all cases, and the most restrictive category should be used. If the result is considered acceptable, or if further refinements of the analysis cannot be performed, then the categorization process is complete. Alternatively, additional iterations (ie. Step 8.b in Figure 7.1 below) on the analysis may be performed, possibly by taking the assessment in steps 3 through 7 down to further levels of detail.

Table 7.1: Software Category as a Function of Safety Significance and Impact Type

Plant System Safety Significance	SOFTWARE FAILURE IMPACT TYPE		
	IMPACT TYPE I	IMPACT TYPE II	IMPACT TYPE III
High	1	2	4
Medium	2	3	4
Low	3	3	4

The categorization process may be iterative. An initial category may be determined, but further analysis may be requested in order to resolve outstanding issues, which manifest themselves as conservative assumptions made during the categorization process. Through a series of iterations, it may be possible to justify a less stringent software category. The process is illustrated in Figure 4.2.

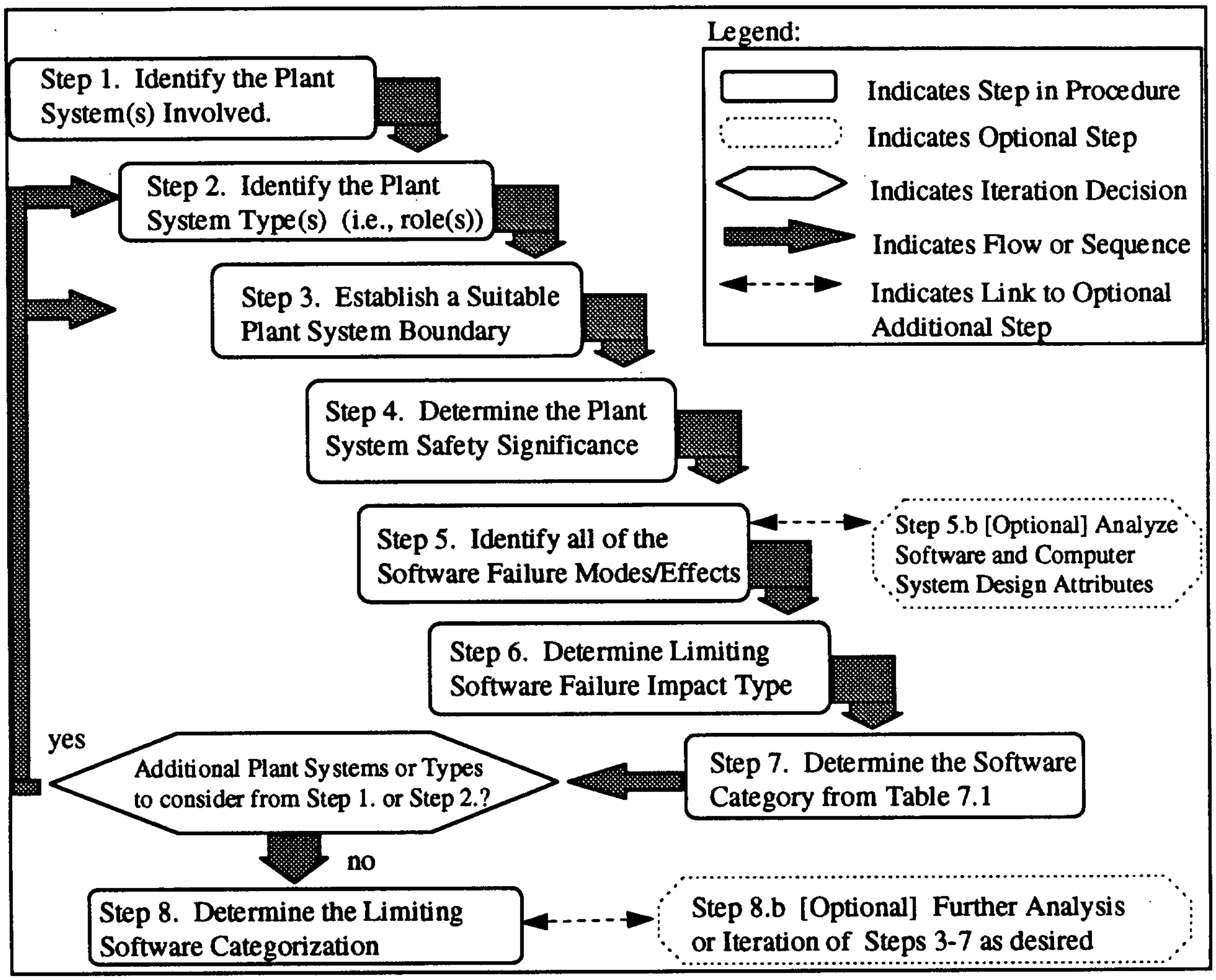

Figure 7.1: Overview of the Categorization Procedure

Based on Table 7.1 and the discussion in Section 4, the following observations can be made. The standard for safety critical software will be applied only to special safety systems and to process systems for which the initiating event frequency is $\leq 10^{-3}$ occ/y. Since there are currently no process systems in this category which contain software whose failure could directly cause a serious process failure, Category 1 has not been applied to process systems.

A reduction in software engineering rigour from Category 1 to Category 2 occurs if either the system safety significance decreases from High to Medium while the software failure impact type remains at Type I, or if the significance stays High but the impact type moves from I to II. This reduction is justified on the basis of a reduction in the safety significance of the system, in the first case, and on the basis that the consequences of software failure are not as severe, in the second case. In both cases, there is a significant reduction in the overall risk associated with software failure.

Category 3 is applied when either the system safety significance is Low and the impact type is I, or the impact type is II for a system of Medium or Low safety significance. In each case, there is a significant reduction in overall risk associated with software failure, so the reduction in software engineering rigour is justified.

If the software failure impact type is III, then by definition there is no safety-related impact of software failure, even though the software exists in a safety-related system. In such cases, the software is Category 4. Based on other considerations (eg., production reliability), it may be desirable to apply the Category 3 standard, but such discussion is beyond the scope of this Guideline.

Failures which may lead to initiating events not explicitly or adequately documented in the licensing safety analysis could be uncovered while categorizing software. In this case, appropriate Nuclear Safety personnel should be consulted for further clarification of the issue.

8. CATEGORIZATION REPORT CONTENT AND FORMAT

The following format is suggested for the categorization report. The format is not considered mandatory, but is presented as an aid to ensuring that the information necessary to support the chosen software category is provided in the categorization report. The intent of a categorization report is to capture the salient arguments and decision criteria (with appropriate explanation and references as required) applied in arriving at a categorization. The process is intended to be repeatable and consistent. A categorization report should include the following basic sections:

- Introduction,
- Plant System and Safety Significance Analysis,
- Failure Impact Type Analysis,
- Software Categorization and Summary Discussion, and
- References and Appendices (or attachments as required).

The following is a brief explanation of what contents are expected in each of these report sections. The introduction should:

- provide background on the plant system in which the software will be implemented,
- identify any new plant systems, or modifications to existing systems or software,
- describe the purpose of the plant system and the role of the software, and
- identify unique features of the plant system or software that are relevant to categorization.

The Plant System and Safety Significance Analysis Section (i.e., the documentation of the steps in Phase I of the categorization process) should:

- identify the plant system(s) of which the software is a part, and describe each plant system boundary (note: each plant system boundary identified should be addressed),
- describe other relevant sub-systems within the plant system(s), especially those that can play a mitigating role in the event of software failure,
- classify the plant system(s) as one or more of a safety or mitigating system, a process system, or a monitoring or testing system,
- identify the safety-related functions of the plant system(s) (i.e., the mitigating role for safety/mitigating systems, initiating events for process systems, and the monitoring/testing functions for monitoring/testing systems), and
- document the assessment of the (each) plant system's safety significance and identify the significance category.

The Software Failure Impact Analysis Section (i.e., the documentation of the steps in Phase II of the categorization process) should:

- identify the effects of software failure relevant to the safety role of the plant system,
- for each software failure and effect, identify the software failure impact type, taking into account and documenting credited mitigating provisions (this should include documentation of any design attributes and failure handling mechanisms which may be credited in the failure impact type determination) (note: it is important that this be documented for each software failure effect, as opposed to just listing the limiting software failure, so that designers are aware of all possible failure modes and their impact), and
- summarize the findings by identifying the limiting software failure impact types and explain their basis.

The Software Categorization and Summary Discussion Section should:

- identify the software category from Table 7-1, and explain its basis,
- if the (any) plant system has more than one role, explain which is limiting and hence is the basis for categorization,
- discuss any uncertainties associated with the outcome,
- discuss the limiting cases for categorization, and identify design alternatives that would reduce the criticality level,
- discuss the tradeoffs associated with the alternative arguments, and
- state all assumptions which, if changed, could affect the software category (e.g., assumptions about test intervals for testing software categorization, or assumptions about the degree of independence among sub-systems when independent mitigating provisions are credited)

Finally, the References Section should list all supporting documentation cited in the report. Appendices and attachments should be included and referenced as is needed to improve the understandability and completeness of the document.

9. REFERENCES

[1.] A.K Lee, Guideline for Categorization of Software in Ontario Hydro's Nuclear Facilities with Respect to Nuclear Safety, Rev. 0, June 1991.

[2.] Software for Computers in the Application of Industrial Safety Related Systems, Secretariat IEC TC 65A, ISO/IEC JTC1/SC7 N917, 65A(Secretariat)122, Section 6, December 12, 1991.

[3.] Electrical/Electronic/Programable Electronic Safety-Related Systems: General Aspects, Version 4 SC65A/WG10/216(A), 65(Secretariat)123, 1992.

[4.] Classification and Graded Recommendations for Digital Systems and Software in Instrumentation and Control Systems of Nuclear Power Plants, Source Unknown (Looks like EPRI/Utility V&V Working Group).

[5.] Nuclear Power Plants - Instrumentation and Control Systems Important for Safety - Classification, IEC1226, First Edition, 1993-05.

[6.] The Classification of Instrumentation and Control Systems Important to Safety for Nuclear Power Plants, 45A(central office)128, 1992-03-13, Project number 45A.20.1.

[7.] Information Technology - Classification and Assignment: Software Integrity Levels, ISO/IEC JTC1/SC7/WG9, Project 7.30, Working Draft 1.0, June 16, 1994.

[8.] Categorization of Software, ISO/IEC JTC1.7.22, Version 1.0, December 19, 1991.

[9.] Application Criteria for Programmable Digital Computer Systems in Safety Systems of Nuclear Power Generating Stations, ANSI/IEEE-ANS-7-4.3.2-1982.

[10.] Software Considerations in Airborne Systems and Equipment Certification, RTCA Inc., DO178 B/ED/12B, SC-167/Eurocae/WG12.

[11.] Standard for Software Engineering of Safety Critical Software, Ontario Hydro Reference 982 C-H-69002-0001, Rev. 00, 90/12/21.

[12.] Software Engineering of Category II Software, Ontario Hydro Reference 907-C-H-69002-0100, Rev. 00, 1993 05.

[13.] Software Engineering of Category III Software, Ontario Hydro Reference 907-C-H-69002-0200, Rev. 00, 1993 05.

[14.] Atomic Energy Control Board Consultative Document C-6, Requirements for the Safety Analysis of CANDU Nuclear Power Plants, June 1, 1980.

APPENDIX A: Considerations for Computer System Failure Assessment

The following three tables identify some of the more common technical issues to be considered in evaluating causes, effects, prevention, detection, localization, or recovery of software errors. It is intended as an aid to identify possible issues when considering system design attributes and their effect (if any) on the failure impact type.

Table A.1: ***Common Inter-dependency Issues to be Considered*** **(note this is not intended to be a comprehensive list)**

a) for Common Interface Dependencies, consider failures due to:

- non-determinant timing delays
- invalid interface states
- invalid data across interfaces
- invalid interface functions
- loss of interface functionality
- initialization or re-start timing, state, or sequence errors

b) for Shared Resource Dependencies, consider common mode failures associated with the:

- operating system
- shared RAM
- CPU registers
- peripheral device hardware
- input/output ports or peripheral device drivers
- common block data

Table A.2: ***Common Failure Handling Mechanisms to be Considered*** **(note this is not intended to be a comprehensive list)**

- process failure detection
- data integrity checks
- state integrity checks
- hardware watchdog timers
- software watchdog timers
- input/output data integrity checks
- error prevention mechanisms (e.g. limited write space)
- manual recovery (human intervention)
- automatic recovery or re-start
- operator annunciation of failures
- redundancy with automatic fail-over (i.e. hot standby systems)

Table A.3: ***Specific Issues to Consider in Software Failure Mode and Impact Analysis*** **(note this is not intended to be a comprehensive list)**

- minimum or maximum time to detect failure or degradation
- maximum allowable duration of failure, unsafe state, or unavailability
- minimum or maximum safe response or recovery time
- requirements to recover to normal or safe computer system state
- dependency on and reliability of operator notification and/or intervention
- dependencies of plant state and operator control actions or sequences
- dependencies on external computer system actions in failure conditions

APPENDIX B: Consideration of Operator Interaction with the Computer System

Historically, the reliability of computer systems in the nuclear industry has been centred around hardware and software. In recent years, with software-controlled systems playing an increasingly important role in all types of systems, it has been recognized that the human component (i.e., the operator) in the system is a significant contributor to the reliability equation.

Thus, the overall system reliability is some combination of hardware, software, and human factors. To adequately assess the nature of human reliability, one must have a good understanding of the specific human tasks involved, and the nature of the operational goals in question.

For the purposes of categorization, operator interactions with the computer system should be considered in the determination of failure impact type by applying the following basic steps:

1. Characterize the operator involvement in the system (e.g., maintenance, testing and direct operation, including reliance on any operational procedures, etc.). This involves examining the human role in interacting with the software to achieve system goals and the identification of interaction tasks. This may include describing the way they are performed and the environment or conditions in which they are performed.

2. Identify the types and possibilities for error during human interaction with the software. More specifically, identify the failure modes and proposed mitigating provisions with operator involvement that may affect the ability of the system to meet its safety functions. Potential errors can be identified by person(s) knowledgeable in the tasks associated with the plant system containing the software.

3. Examine the possible consequences of software induced operator error. The consequences identified can then be evaluated as to their failure impact type.

Note that in cases where new or changed functionality is being introduced into the control room (e.g., as part of an upgrade, retrofit, or an add-on system), the role of the system and the decision-making process of the operator, as well as specific plant operating procedures, may change. It is possible that this change may alter or invalidate some of the basic assumptions used in previous safety analyses (i.e., how the safety significance of the system was determined). As a result, it may be necessary to question the assumptions in existing plant system reliability models, and previous categorizations that were influenced by human reliability concerns. In such cases, the system safety significance, taking operator reliability into account, may have to be re-evaluated. It is important to identify any changes in previously credited operator reliability that may result from any such changes or additions to the computer system functionality. These should be documented and a nuclear safety analyst should be consulted. To determine operator reliability, human factors issues must be considered.

DISTRIBUTION LIST FOR COG-95-264

G.H. Archinoff AECL - SP1 F1	G.J. Hinton AECL - SP2 F3	J. Pauksens AECL - SP2 F4
S. Basu OH - Bruce A NGS	R.J. Hohendorf OH - H12 F27	G. Raiskums AECL - SP1 F3
W.C. Bowman OH - H12 A26	D.R. Huget OH - H12 B25	C. Royce OH - Bruce B NGS
A. Campbell OH - Pickering NGS	E. Hung OH - Darlington NGS	G.R. Schneider OH - Pickering NGS
D. Chan AECL - SP1 F2	N. Ichiyen AECL - SP2 F3	R.R. Shah AECL CRL - Stn 30
A.T. Chen OH - Pickering NGS	P.K. Joannou OH - H12 F27	T.E. Smart OH - Pickering NGS
G.D. Cleghorn OH - Darlington NGS	R.A. Judd AECL CRL - Stn 91	H. Storey NBEPC - Pt Lepreau GS
J. de Grosbois AECL CRL - Stn 91	A.M. Kozak OH - Bruce B	A. Stretch AECL Saskatoon
E.G. Echlin AECL CRL - Stn 30	D.K. Lau OH - H12 A26	N.J. Webb OH - Bruce A NGS
N. Gour HQ - Gentilly-2	L.R. Lupton AECL CRL - Stn 91	R. Zemdegs OH - H12 F21
J. Harauz AECL - SP1 F3 OH - H12 D27	M.J. MacBeth AECL Saskatoon	