
Organizer: H. Yoshikawa (Kyoto University)

- Objectives of the Workshop
- Introduction of IAEA Guideline
- Investigation items of this workshop study
Objectives of the Workshop

Should we proceed to:

- Review and Scrutinize Our Present Status of the Designing, Operation and Maintenance of Computer-based Safety System of Nuclear Power Plant for Already Operated or Under Planning both in Japan and Korea, from the Standpoint of the IAEA Guideline, or

- Directed how the IAEA Guideline would be Extended or Upgraded, in the Light of Further Application of Rapidly Progressing Information Technology to the I&C Systems of NPP.

- This Time, the Activity will be Limited to the Short Introduction to the Contents of the IAEA Guideline Followed by the Discussion on How We Proceed to the Matters in Concern.
Introduction of The IAEA Guideline

Software for Computer Based System Important to Safety in Nuclear Power Plants

IAEA Dr6_SWSG.doc 20-08-98
1 Objectives, Scope and Structure (1)

Provide Guidance for All Phases of System Life-Cycle (Conception, Development and Operation) on the Collection of Evidence and Preparation of Document That is Used in the Safety Demonstration of Software Embedded in Computer Based Systems Important to Safety in Nuclear Power Plant.

Recommendation Covers Both Control of the Development Process and the Objective Evidence Required to Demonstrate the Safety of the Product.
.2 Objectives, Scope and Structure (2)

Provides Guidance for All Involved in the Production, Assessment and Licensing of Computer Based Systems

(Plant System Designers, Software Designers, Programmers, Verifiers, Validators, Certifiers, Regulators, Plant Operators)

Computer Based System Design May Result from a Mapping of System Requirement onto Various Types of Software

- Software to be Developed
- Software to be Produced by Configuring Pre-Developed Software
- Purchased Software or Firmware
.2 Objectives, Scope and Structure (3)

- Technical Background
- Application of General Safety Principle to Computer Based Systems
- Project Planning
- “Recommendations” and “Documents” to Individual Phases of Development Life Cycle.

- Computer System Requirements
- Computer System Design
- Software Requirements
- Software Design
- Software Implementation
- Verification and Analysis

- Computer System Integration
- Validation
- Installation and Commissioning
- Operation
- Post-Delivery Modification
3.2 Technical Background

A. Characteristics of Computer Based Systems

B. Process of Computer Based System Development

C. Safety and Reliability Issues of Computer Based Systems

D. Organizational and Legal Issues
.2 .A Characteristics of Computer Based Systems

Two Basic Properties:

1. Programmable
2. Hardware is based on discrete digital logic

Software fault is more human error nature

| Hardware          | Error in Designing or Manufacturing
|                  | Random nature - Wear out, Degradation, Environmental Process |
| Software         | Does not wear out
|                  | Affected by Change of Operating Environment |
|                  | Bad or unclear requirement specification |
|                  | Error introduced in the implementation phase |
Balance Sheet of Computer Based Systems

<table>
<thead>
<tr>
<th>Advantage</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Easier implementation of complex function</td>
<td>• Software implementation tends to be more complex, more prone to design error</td>
</tr>
<tr>
<td>• Improved monitoring of plant variables, operator interface, testing, calibration, self-checking, fault diagnosis</td>
<td>• Discrete logic models of real world gives rise to:</td>
</tr>
<tr>
<td>• More accurate and stability</td>
<td>- more sensitive (less tolerant) to small errors</td>
</tr>
<tr>
<td>• Multiplexed bus structure reducing cabling requirement</td>
<td>- more difficult to test</td>
</tr>
<tr>
<td>• Easily modified</td>
<td></td>
</tr>
</tbody>
</table>
.2 .B Process of Computer Based System Development

- Stepwise Controlled Process
  - An ordered collection of distinct phases
  - Extended information at earlier phase becomes input to the later phase
  - Development of safety-important systems becomes iterative process

- Validation Process is carried out
  - To the product of each development phase verified against the requirements of the previous phase
  - To confirm compliance of the product with all functional and non-functional requirements
Development of Software for Computer Based Systems Important to Safety

Kyoto University, Hidekazu Yoshikawa
Safety and Reliability Issues of Computer Based systems

- Reliability evaluation of Software-based System is more difficult than non-programmable system, hence the claim for high software reliability is unbelievable at the present stage.

- Plant design of digital protection system attaining $10^{-4}$ per demand should be carefully made.

- Problem of common mode failure for the redundant system with the copies of identical software is difficult to cope with, because software failure is not random nature but systematic one.

- Computer based I&C system with enough dependability can be made with “independence” and “diversity”.

Kyoto University, Hidekazu Yoshikawa
Organizational and Legal Issues

Issues here are:

- (1) Existence of suitable legal and administrative framework for licensing important to safety
- (2) Sufficient competence and resources within organization involved in system development process
- Quantification of software reliability is unresolved issue, hence the regulatory position for the demonstration of safety and the reliability of software to be required should be clarified at the initial stage of project.
- Legal, regulatory, procedural and administrative organizational framework is necessary for processing the licensing of safety critical system for the related industry.
- Both competence and resource are necessary in regulatory organization, licensee’s design team, regulatory body’s technical support and suppliers.
- Licensees should release the proprietary information at licensing phase.
- It is necessary to establish proper structures for treating operational and maintenance issues (ex. Post-delivery modification of software.)
3 Application of General Safety Principles to Computer Base Systems

Four Categories of Related Principles

A. Safety Philosophy
B. Design and Development
C. Management and Quality Assurance
D. Documentation
### 3. A Safety Philosophy

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Simplicity in the Design</td>
</tr>
<tr>
<td>2</td>
<td>Safety Culture</td>
</tr>
<tr>
<td>3</td>
<td>Safety Categorization</td>
</tr>
<tr>
<td>4</td>
<td>Balance between Risk Reduction and Development Effort</td>
</tr>
<tr>
<td>5</td>
<td>Defense in Depth</td>
</tr>
<tr>
<td>6</td>
<td>Redundancy</td>
</tr>
<tr>
<td>7</td>
<td>Single Failure Criterion</td>
</tr>
<tr>
<td>8</td>
<td>Diversity</td>
</tr>
<tr>
<td>9</td>
<td>Fail-safe Design</td>
</tr>
<tr>
<td>10</td>
<td>Robustness</td>
</tr>
<tr>
<td>11</td>
<td>Security</td>
</tr>
<tr>
<td>12</td>
<td>Maintainability</td>
</tr>
<tr>
<td>13</td>
<td>Full Representation of Operating Mode</td>
</tr>
<tr>
<td>14</td>
<td>Human Interface/Anticipation of Human Limitation</td>
</tr>
<tr>
<td>15</td>
<td>Demonstrable Dependability</td>
</tr>
<tr>
<td>16</td>
<td>Testability</td>
</tr>
</tbody>
</table>

*Kyoto University, Hidekazu Yoshikawa*
(1) Simplicity in the Design

- Unnecessary complexity be avoided in the design
- Functional requirement of safety system be limited to safety functions (function not essential to safety be separated out into other systems)
- Top down decomposition, level of abstract and modular structure for coping with the problem of unavoidable complexity
- Logic behind system modularization and definition of interface be as simple as possible (ex. information hiding)
- Simple algorithms in the design of system modules

Kyoto University, Hidekazu Yoshikawa
(2) Safety Culture

Personnel on software development project for high safety importance should include application specialists, computer software/hardware specialists, in order that the combination of expertise helps to ensure safety requirement and each specialist understand how their jobs related to achieving safety requirements.
(3) Safety Categorization

Use of safety oriented categorization scheme (to define safety significance of I&C system functions, ex. IEC 1226) be helpful (to direct appropriate degree of attention by plant designers, operators, regulatory authorities) to the specification, design, qualification, manufacturing, installation, maintenance and testing of the system and equipment to ensure safety.
(4) Balance between Risk Reduction and Development Effort

- Top-down design/development process facilitates the trade-off assessment to the conflicting design objectives.

- Graded design and qualification requirements be derived from safety categorization scheme.

- Appropriate measures to warrant level of confidence be associated with each category.

- Hardware be assessed by quantitative technique while software only qualitative.
(5) Defense in Depth

Software based system consists main line of safety, while defense in depth (e.g. some of effective back-up ) be provided.
(6) Redundancy

- Multiple redundant instrument channels with voting used in analogue application does not prevent system faults that could lead to impairment of all redundant channels.
(7) Single Failure Criterion

- Single failure criterion is difficult to apply for software, since a fault which comes a software failure may present in all replicas of this software. This is unlike to hardware random failure.
(8) Diversity

- Reliability of computer based systems can be enhanced by applying ‘diversity’ to reduce software common cause failure. Diversity functions be applied in all phases: system components at different levels of design, methods, tools, personnel.

- Although diversity software may provide improved protection against common mode software errors, it does not guarantee the absence of coincidental error.
(9) Fail-safe Design

- Fail-safe features, supervision and fault tolerance should be added into software, but only to the extent that additional complexity is justified by demonstrated global increase of safety.

- Use of external devices such as “watchdog timers” makes more dependable.
Achievement of completely specifying the desired and safe response to all combinations of input is difficult, and since errors are always possible, defensive design and coding should be used to ensure safe response under all circumstances as far as achievable.
(11) Security

Demonstrated measures should be taken to protect the computer based system during whole life time against physical attacks, intentional and non-intentional intrusion, frauds, viruses, etc.
(12) Maintainability

- Should be designed to detect, locate and diagnose failures so that the system be repaired or replaced efficiently. Software with “modular structure” be easier to repair, review, analyze.

- Anticipated changes for functionality of software be limited to a small portion of the design.
Full Representation of Operating Mode

The requirement and design of the software important to safety must explicitly define the outputs for full range of all inputs, for each of the operating modes.
(14) Human Interfaces/Anticipation of Human Limitation

- Operator interfaces should be designed to provide the operator with sufficient but not overwhelming amount of information, and to provide sufficient time for reacting (ex. thirty minutes rule).

- All operator inputs should be checked for validity as a defense against impairment by operator error.

- System interface should be designed to facilitate the human involvement after reactor trip like manual backup, intervention, manual resetting.
For VDU utilization,

- Enough response time, navigation, help function
- Each display screen should be restricted to one purpose of specific mode and context, unification of naming convention.

Demonstration of the adequateness of manual operation by task analysis, ease of in-service inspection, etc.
Demonstrable Dependability

The system must not only be dependable but also be possible to demonstrate to the regulator that it is dependable. This should be made through the production of adequate documents by licensees.
Testability

Each requirement both functional and non-functional, and each design feature should be expressed so that a test can be done to determine whether that feature has been implemented correctly.
3. B Design and Development Activities

(1) Stepwise Control Process
(2) Review Ability
(3) Comprehensive Testing
(4) Use of Automated Tools
(5) Traceability
(6) Compliance with Standards
Stepwise Control Process

- This gives development more evidence of correctness as well as ease of verification process to ensure that errors are detected easily on the design process.
(2) Review Ability

- Accurate and easily reviewable documentation should be produced for all stages of design process.
(3) Comprehensive Testing

- Test plan should be established and agreed with the regulator at each stage.
- Demonstration of test coverage including tracing of test cases to source document should be provided for development, verification and validation.
- Test results, test coverage demonstration and other test records should be available for third party audit.
(4) Use of Automated Tools

- Two kinds of tools, software development tools (code generators, compilers, and linkers) and software verification tools (static analysis tools and test coverage monitors, etc.) should be qualified to a level commensurate with their role in the software development and safety demonstration.

- Techniques used to gain confidence in the tool should be defined and documented.
(5) Traceability

- Traceabilities, from requirements to design, from design to code, from requirements, design and code to test, should be maintained.
(6) Compliance with Standards

- Design principles, safety requirement and technical standards used in the design should be identified and compliance analysis should be prepared for key standard with respect to computer system design specification and implementation.
3. C Management and Quality Assurance

(1) Clearly Defined Personnel Functions and Qualifications

(2) Acceptable Practices

(3) Quality Assurance

(4) Assignment of Responsibilities

(5) Independent Assessment
3. D Documentation

Confidence in the reliability of a software product is largely based on evidence of the soundness of the process of constructing it.

Documentation plays a crucial role in providing the “transparency” and “traceability”.

Appropriate documents should be produced at each step of the development process, and documentation be updated throughout the iterative development and ongoing maintenance process.

The documents available to the regulator be identical to those used by the designers.
3. D Documentation: continued

The documents should have the following attributes:

- **Understandability**: Should be understandable by people with a variety of backgrounds and expertise.

- **Process**: Requirements and descriptions of designs should be stated formally with explanations given in natural language.

- **Traceability and Completeness**: The purpose of tracing is to demonstrate that the implementation is complete with respect to the computer system requirements and design, and that there is no unsafe implementations, features.

- **Consistency**: The document should not contain contradictory or inconsistent statement. The notation, terminology, comments and techniques should be used in a uniform way throughout the documentation.

- **Verifiability**: Verifiability is improved when documents are understandable, unambiguous and traceable.

- **Modifiability**: The structure and style of documentation must be such that any necessary changes can be made easily, completely and consistently.

*Kyoto University, Hidekazu Yoshikawa*
4.4 Project Planning

Licensing approval of systems important to safety will be facilitated if the development process is carefully planned and if there is clear evidence that the plans have been followed.

Project planning can be documented in a comprehensive safety plan, or as a set of plans that cover all aspects of the project.

A development plan should define a set of development activities and the essential characteristics of each activity. Other aspects of the project which must be planned are quality assurance, verification and validation, configuration management, commissioning and installation.
Relationship of Verification and Validation to Requirement, Design and Implementation

*Validation: Testing conducted on a completed, integrated computer system (hardware and software) to ensure compliance with requirement specification.

*Verification: The process of ensuring that a phase in the software development meets the requirements imposed on it by the previous phase.
4. Project Planning: Continued

A. Development Plan

B. Quality Assurance Plan

C. Verification and Validation Plan

D. Configuration Management Plan

E. Installation and Commission Plan
## 4. A Development Plan

- Needed Coverage of Development Plan -

<table>
<thead>
<tr>
<th>Phases</th>
<th>All phases of development process should be identified. Each phase consists of requirements, design and implementation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods</td>
<td>The methods to be used in the development should be identified and the selection should be related to the quality assurance plan where standard and procedures are established.</td>
</tr>
<tr>
<td>Tools</td>
<td>Tools to be used should be identified in the QA plan, and they must be selected to facilitate the proper application of the selected methods, standards and procedures. The use of tools is recommended as they relieve staff from clerical, error-prone programming and manual verification tasks.</td>
</tr>
<tr>
<td>Documents</td>
<td>It should be indicated when all requirements, quality attributes and performance characteristic will be found, and what acceptance criteria will be used for the whole project.</td>
</tr>
<tr>
<td>Schedule and Milestones</td>
<td>The schedule for documents and identification of period when project review is conducted should be established based on: assessing resource availability, training needs, adequacy of facilities and tools available, estimating the duration of each phase, time necessary for project review at key points.</td>
</tr>
<tr>
<td>Personnel</td>
<td>To ensure those people in the software development and maintenance activity are competent.</td>
</tr>
</tbody>
</table>
4. B  Quality Assurance Plan

Should be prepared and implemented by the licensee, and should be available for regulatory review/approval before the project begins, with the following items included:

- Identification of governing standards, procedures and tools.
- QA plan should indicate who must review and approve before official release for each document.
- Describe organizational structure that includes assurance of the independence of QA auditors.
- Mechanism for identifying, reporting and disposing of non-conformance to standards and procedures.
- Identification of all necessary plans (development plan, configuration management plan, commissioning and installation plan).
- Identification of the number and scope of QA audits.
- Procedure for qualifying the tools.
- A mechanism for checking the quality of components from external suppliers.
4. C Verification and Validation Plan

- **Validation**: Demonstrate the computer system achieves overall safety and functional requirements by two distinct steps:
  - (i) Validation of computer system requirements against plant and system requirements
  - (ii) Validation of computer system implementation against computer system requirements.

- **Verification**: Required for each of the documents produced by the following development phases:
  - (i) Computer System Design
  - (ii) Software Requirements
  - (iii) Software Design
  - (iv) Software Implementation

Techniques to be used to verify the software should be stated in the V&V plan. It is expected that they include a combination of static examination of document and dynamic execution of the implementation.
The records of V&V result should be determined to be maintained for the system lifetime.

The team or teams performing V&V should be identified in the V&V plan. The V&V task should be allocated between the teams in order to maintain “independence” which includes:

- Technical independence
- Management independence
- Financial independence

The V&V plan should include a mechanism for recording all errors found during the analysis and making sure that they are properly resolved via change control process.
4.D Configuration Management Plan

Version Control - All items of software development should be under configuration management control. All identifiable items (i.e., documents, components of software, data structure, etc.) should be uniquely identified including a version number.

Change Control - The change control procedure should maintain records of the problems that were identified to necessitate the changes, how the problems were analyzed, what items were affected, what specific changes were made to fix the problem, and what versions and/or baseline were produced to resolve the problem.
4. E Installation and Commissioning Plan

After a system has been constructed and validated in isolation, it is integrated with other plant systems and tested within the real plant environment.

The process of installation and commissioning must be carefully planned to co-ordinate the proper transition from development to use, and the hand over from the developers and verifiers to the users and maintainers.
Installation and Commission Plan: Continued

The related plan includes:

- Sequence of steps for proper integration of the system into the plant and the corresponding plant status needed for safe introduction of the new or changed system.

- The required interactions with the regulators.

- The commissioning test cases and sequences and the corresponding plant status needed to confirm proper functioning of the system in the plant environment.

- Description of the records and reports that will be generated to describe the results of commissioning.

- Initiation of the process to instruct and inform the users and maintainers

- Transfer of the configuration management and change control process from developers to maintainers.
Ⅲ．Investigation Items of

Workshop Study - Open Questions

1. Compatibility of the Present Practice to IAEA Guideline
   How far the real practices both in Japan and Korea would satisfy the guideline?

2. Evaluation of IAEA Guideline from Specific Standpoints of Practitioners
   Is the guideline really useful for system developers, regulators and plant operators?

3. Deficient Issues in the Present Guideline
   Could this guideline cover the issues of further introduction of rapidly progressing IT to plant I&C system designing, backfitting and maintenance?

4. Direction Setting of Future I&C System Development
   What would be the lessons learned from this guideline for establishing practical design, developing, regulating and operating, for further introduction of advanced software technology for plant I&C systems?

Kyoto University, Hidekazu Yoshikawa