IAEA Services Series No. 6

ESRS guidelines for software safety reviews

Reference document for the organization and conduct of Engineering Safety Review Services (ESRS) on software important to safety in nuclear power plants

INTERNATIONAL ATOMIC ENERGY AGENCY

July 2000

The originating Section of this publication in the IAEA was:

Engineering Safety Section
International Atomic Energy Agency
Wagramer Strasse 5
P.O. Box 100
A-1400 Vienna, Austria

ESRS GUIDELINES FOR
SOFTWARE SAFETY REVIEWS:
REFERENCE DOCUMENT FOR THE ORGANIZATION AND CONDUCT OF
ENGINEERING SAFETY REVIEW SERVICES (ESRS) ON
SOFTWARE IMPORTANT TO SAFETY IN NUCLEAR POWER PLANTS
IAEA, VIENNA, 2000
IAEA-SVS-06

Printed by the IAEA in Austria
July 2000

FOREWORD

The IAEA provides safety review services to assist Member States in the application of safety standards and, in particular, to evaluate and facilitate improvements in nuclear power plant safety performance.

Complementary to the Operational Safety Review Team (OSART) and the International Regulatory Review Team (IRRT) services are the Engineering Safety Review Services (ESRS), which include reviews of siting, external events and structural safety, design safety, fire safety, ageing management and software safety.

Software is of increasing importance to safety in nuclear power plants as the use of computer based equipment and systems, controlled by software, is increasing in new and older plants. Computer based devices are used in both safety related applications (such as process control and monitoring) and safety critical applications (such as reactor protection). Their dependability can only be ensured if a systematic, fully documented and reviewable engineering process is used.

The ESRS on software safety are designed to assist a nuclear power plant or a regulatory body of a Member State in the review of documentation relating to the development, application and safety assessment of software embedded in computer based systems important to safety in nuclear power plants. The software safety reviews can be tailored to the specific needs of the requesting organization. Examples of such reviews are: project planning reviews, reviews of specific issues and reviews prior final acceptance.

This report gives information on the possible scope of ESRS software safety reviews and guidance on the organization and conduct of the reviews. It is aimed at Member States considering these reviews and IAEA staff and external experts performing the reviews.

The ESRS software safety reviews evaluate the degree to which software documents show that the development process and the final product conform to international standards, guidelines and current practices. Recommendations are made to help resolve any safety concerns, suggestions indicate useful changes that can enhance safety and good practices are singled out to serve as an example.

The IAEA officer responsible for this publication was J. Pachner of the Division of Nuclear Installation Safety.

EDITORIAL NOTE

The use of particular designations of countries or territories does not imply any judgement by the publisher, the IAEA, as to the legal status of such countries or territories, of their authorities and institutions or of the delimitation of their boundaries.

The mention of names of specific companies or products (whether or not indicated as registered) does not imply any intention to infringe proprietary rights, nor should it be construed as an endorsement or recommendation on the part of the IAEA.

CONTENTS

1. INTRODUCTION

1.1. BACKGROUND

Computer based systems are of increasing importance to safety in nuclear power plants as their use in both new and older plants is rapidly increasing. They are used both in safety related applications such as process control and monitoring functions as well as in safety critical applications such as reactor protection or safety feature actuation. The dependability[1] of computer based systems important to safety is therefore of prime interest and must be ensured.

With current technology, it is possible in principle to develop computer based instrumentation and control systems for systems important to safety that have the potential of improving the level of safety and reliability with sufficient dependability. However, their dependability can only be predicted and demonstrated if a systematic, fully documented and reviewable engineering process is followed. The Safety Guide "Software for computer based systems important to safety in nuclear power plants" [1] provides guidance on the collection of evidence and preparation of documentation that is used in the safety demonstration of software embedded in computer based systems important to safety in nuclear power plants for all phases of system life cycle[2].

1.2. OBJECTIVE

The objective of this publication is to provide information and illustrative examples on how to plan, conduct and document review missions for software important to safety at nuclear power plants. The report is intended for use by Member States, IAEA staff and external experts involved in IAEA software safety review missions.

1.3. SCOPE

This document addresses the organizational matters related to the preparation and conduct of the reviews. The technical aspects of software important to safety review are covered by other IAEA publications and internationally accepted standards (IEC, ISO). A list of these publications is provided in the Annex.

Software is only one portion of a computer based system. The other portions are the computer hardware and the instrumentation (sensors and actuators) to which the computer is connected. The reviews described here focus only on the software portion of the computer system; the complete system and the hardware and instrumentation subsystems are discussed here only as they relate to the software subsystem.

1.4. STRUCTURE

Section 2 explains the technical background, describes a software development process, defines the objective and scope of software safety reviews and identifies documents needed for the review.

Section 3 addresses the organization, planning, scheduling and agreements needed prior to conducting a review.

[1] Dependability – Trustworthiness of the delivered service such that reliance can justifiably be placed on this service. Reliability, availability, safety are attributes of dependability.

[2] System life cycle – All stages through which a system passes from conception to final disposal.

Section 4 describes the process of conducting a review. The technical bases of the evaluation are referenced. The contents and styles of the final report documenting the results of the review are also discussed.

Appendix I contains the list of selected issues to be addressed during the review mission.

Appendix II provides guidance for drafting technical notes.

The Annex gives examples of standards, guides and other documents related to software important to safety.

2. OVERVIEW OF SAFETY REVIEW SERVICES TYPICAL FOR SOFTWARE IMPORTANT TO SAFETY

2.1. TECHNICAL BACKGROUND

From the viewpoint of safety and reliability assessment, computer based systems have two basic properties that make them different from other electronic control systems. They are programmable and their hardware is based on discrete digital logic. As for other systems, hardware faults may be due to errors in design or manufacturing, but typically they result from wear-out, degradation or environmental processes and are of random nature. Software, the programmable part of the computer, does not wear out, but it may be affected by the changes of the operating environment. Software faults may result from either bad or unclear requirements specification (which give rise to errors in the logical design or implementation) or errors introduced during the implementation and maintenance phase.

The programmable nature of computer based systems coupled with the discrete logic means that they can have a number of advantages over non-digital and non-programmable systems. They make easier the implementation of complex functions; in particular, they can provide improved monitoring of plant variables, improved operator interfaces, improved testing, calibration, self-checking and fault diagnosis facilities. They can also provide more accuracy and stability. The use of multiplexed bus[3] structures may lead to reduced cabling requirements. They are easily modified, since it requires less physical disruption of equipment, which can be useful for maintenance.

These advantages are counterbalanced by some disadvantages. First there is a marked tendency to implement substantially more complex functions in digital systems with the attendant complexity in the software. This tendency should be resisted. Secondly, software engineering is a relatively new discipline and its practitioners tend to be overconfident in their ability to produce error-free programs. Also because of this inexperience they have not fully developed the techniques for specifying, designing, implementing and testing software systems important to safety that the other engineering disciplines have. In fact there is a body of evidence that says that typically there is not enough time to completely test a complex software system in order to demonstrate that the code is completely error-free: this means that almost every software system of any substantial complexity will be delivered with errors. As a result the task of the reviewer is to verify that the best software engineering techniques have been used in the construction of the software under review so that the probabilities of an error in the software which would jeopardize safety are made vanishingly small.

[3] Bus – A set of conductors connecting the various functional units in a computer.

The software development process for software important to safety should be organized conceptually as an ordered collection of distinct phases. This can lead to development that gives more evidence of correctness by construction and ease the verification process and ensure that faults are detected early on the design process. Experience tells that errors made in the early phases of the development, as the specification phase, are the ones that are most difficult to detect in the final program and therefore most likely to cause problems during operation. This also implies that early involvement of the regulatory body in the development process will clearly aid the licensing process.

Each phase uses information developed in earlier phases and provides input information for later phases. It should be noted, however, that the development process is by nature iterative. As the design progresses, faults and omissions made in the earlier stages become visible and require iterations on the earlier stages. It is in this connection important that accurate and easily reviewable documentation is produced for all stages of the design process. The documentation used to demonstrate adequacy to the regulator should be the same as the documentation actually used in the design. Typical phases of the development process, which also includes implementation and maintenance, are shown in Fig. 1 [1]. The boxes show the development activities to be performed and the arrows show the intended order and primary information flow. Figure 2 shows verification and validation activities which should be done to show that the outcome of one phase is in agreement with the requirement in a phase previous of the development process [1].

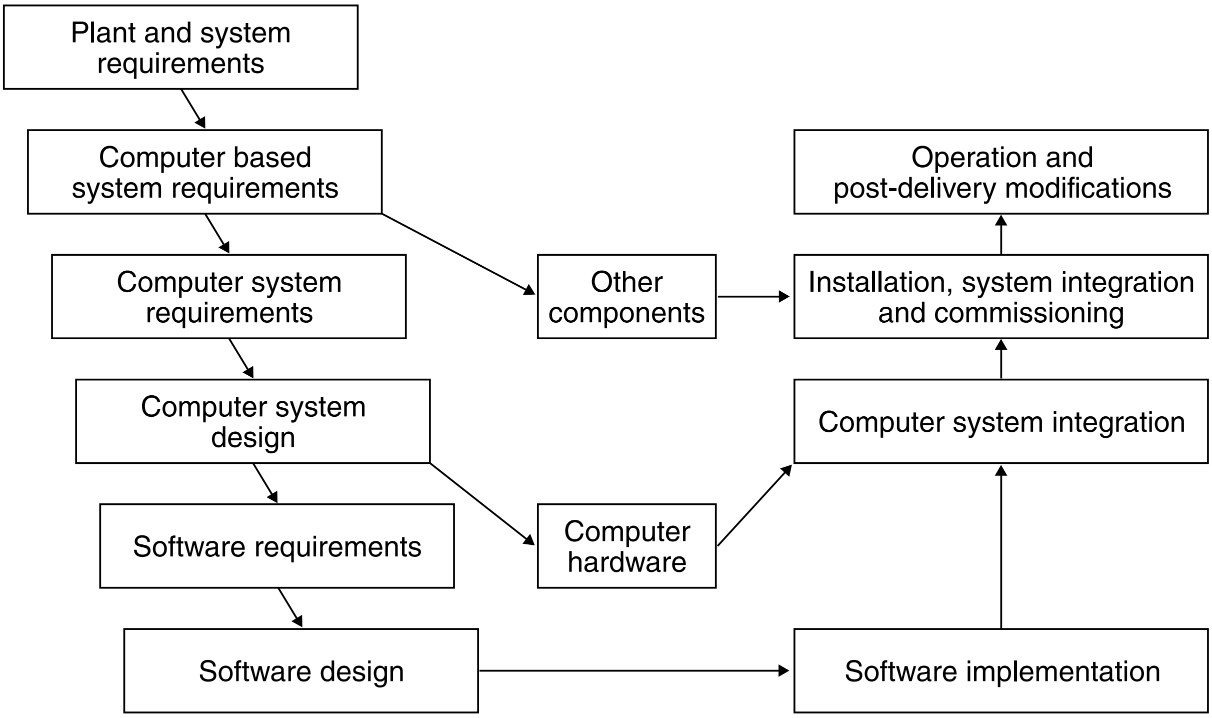
Plant and system requirements
Computer based system requirements
Computer system requirements
Computer system design
Software requirements
Software design
Other components
Computer hardware
Operation and post-delivery modifications
Installation, system integration and commissioning
Computer system integration
Software implementation

FIG. 1. Development of software for computer based systems important to safety (see Ref. [1]).

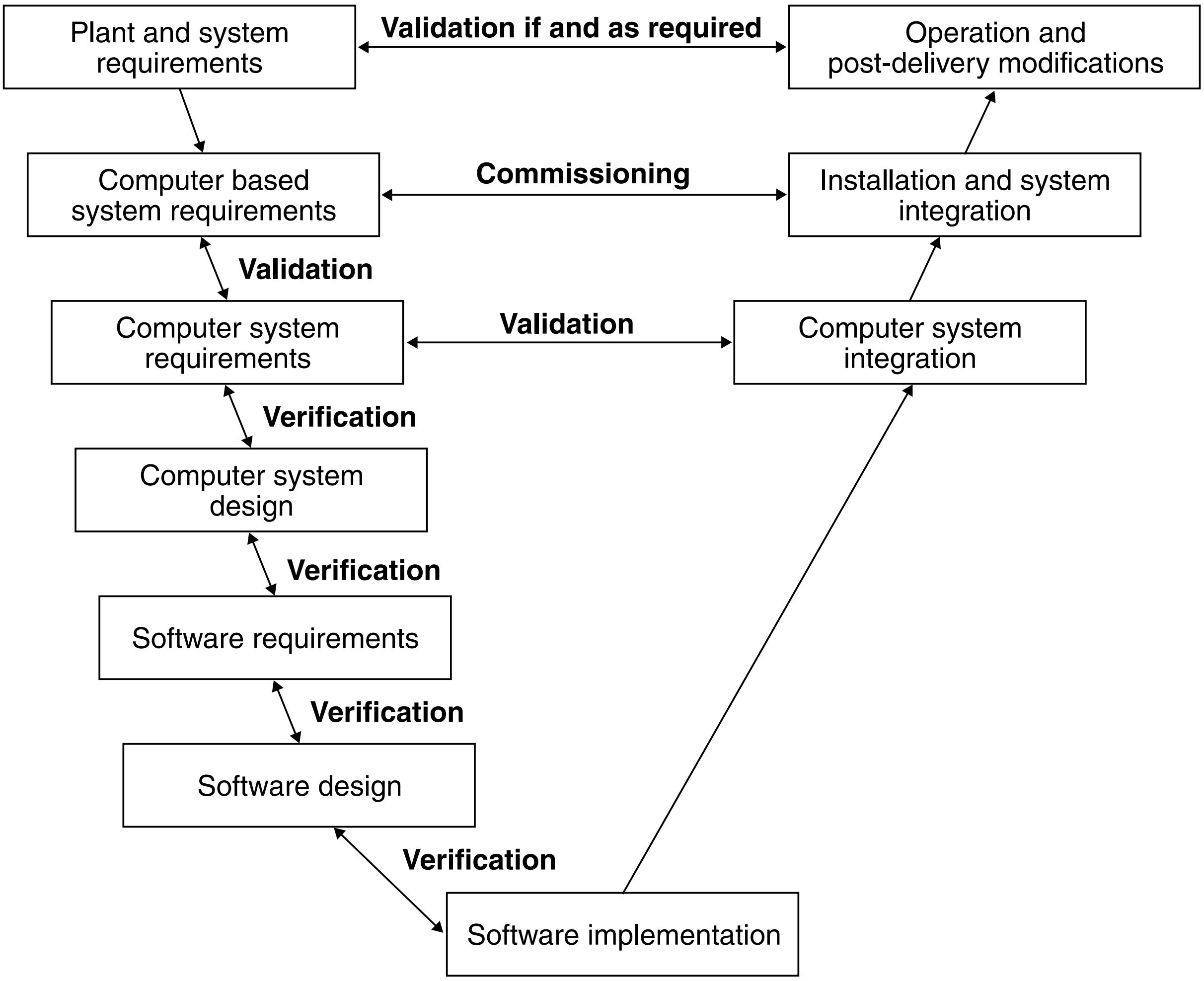
Plant and system requirements
Validation if and as required
Operation and post-delivery modifications
Computer based system requirements
Commissioning
Installation and system integration
Validation
Computer system requirements
Validation
Computer system integration
Verification
Computer system design
Verification
Software requirements
Verification
Software design
Verification
Software implementation

FIG. 2. Verification, validation and commissioning of software for computer based systems important to safety (see Ref. [1]).

A short list describing the phases of the software development process is given in the following. The description of these phases is provided in detail in Ref. [1].

1. *Plant safety analysis phase.* In this phase a plant-wide safety assessment is performed and documented. It covers the entire plant in general and is not specific to any computer systems which are planned. It uses existing plant descriptions and the safety philosophy of the Member State.
2. *Safety system requirements phase.* In this phase the requirements imposed on the various plant safety systems by the analysis performed in paragraph 1 above are documented.
3. *Safety system design phase.* In this phase design requirements are produced and documented for each safety subsystem which implements the requirements produced in paragraph 2 above.
4. *Computer system requirements phase.* In this phase the requirements imposed on the computer systems to be used in the plant safety systems are documented. Three aspects are of particular importance:
   - The definition of the system, with a clear boundaries defined between the target system, the plant and the environment.
   - The functional specification which defines what the target system is intended to do.

- The specifications of particular safety defence in depth[4] and diversity features which shall be included to enhance safety and protect against undesirable consequences of failures in the target system.

5. *Computer systems design phase.* In this phase systematic and structured or formal means are used to map the requirements onto a design and the results documented. Hardware items to be purchased or custom built are identified, software and firmware items to be purchased or custom built are identified and software to be re-used is identified.
6. *Software requirements phase.* In this phase functional software requirements are documented which will drive the implementation of the software.
7. *Integration requirements phase.* In this phase any requirements on the interfaces between purchased software and firmware, the hardware, re-used software and the software to be developed are documented.
8. *Software design.* In this phase the division of the software into a set of interacting modules and the detailed description of those modules is documented.
9. *Software implementation.* In this phase the source code and the executable code are produced, based on the specifications for the internal design and for the interfaces of the software modules. Unit tests and module interface tests are performed.
10. *Computer system integration.* This phase consists of three parts, such as loading of all software into the hardware system, testing that the software-hardware interface requirements are satisfied and testing that all the software can operate in the integrated software-hardware environment.
11. *Validation phase.* In this phase the integrated computer system is validated against the computer system requirements.
12. *Installation* consists of delivery of the integrated computer system to the site and the physical installation in the plant. This phase also includes acceptance test at the site and commissioning.
13. *Operation* follows installation, commissioning and any regulatory approval for use. The system becomes part of the plant and is operated by the licensee. The operation phase of a particular system continues until it is removed or replaced.
14. *Post-delivery modification* during which it is necessary to ensure that the functional safety of the computer based system is appropriate both during and after modification or change activities have taken place.

Although information from all phases of software development are important for the assessment of safe application of the software, the actual software review process focuses on the above phase 6 (software requirements) through phase 14 (post-delivery modification).

2.2. OBJECTIVE OF SOFTWARE SAFETY REVIEWS

The objective of software safety reviews is to assist a plant operator or a regulatory body of a Member State in ensuring the safety of software based instrumentation and control systems important to safety by advising on the review of documentation and other information relevant to the development, application and safety assessment of such software. The relevant documents may relate to appropriate phases in the software development process as shown in Fig. 1.

[4] Defence in depth – A hierarchical deployment of different levels of equipment and procedures in order to maintain the effectiveness of physical barriers placed between a radiation source or radioactive materials and workers, members of the public or the environment, in operational states and, for some barriers, in accident conditions.

The software safety review team shall consist of experts in the field and should be able to judge to which degree the documents show that the development process and the final product conforms to international standards and guidelines (in particular to Ref. [1]) and to current practice.

2.3. SCOPE OF SOFTWARE SAFETY REVIEWS

The scope of a software safety reviews may be different from case to case. It is understood that actual review will be tailored to the specific needs of the requesting organization and to the actual stage of the software development project to be reviewed. The exact scope should be determined in discussion between the IAEA team leader and the requesting organization in the preparatory phase. The scope of the review dictates all elements of the review execution: the discipline composition of the review team, the reference material to be provided prior to and during the review, the discipline composition of the in-country counterpart specialist and so on. The scope should be clearly stated and unambiguous and should be reflected in the review agenda.

For the purposes of this publication, it is assumed that a software review will consist either of a comprehensive evaluation of the whole software life cycle or a more detailed assessment of one or more of the process phases listed in Section 2.1. There may be a variety of such reviews, depending on the specific needs of requesting organizations. In addition, the in-country counterpart can be either a regulatory body or a nuclear power plant operator. However, at the request of one of these, the review can be focused on evaluation of preparedness of other organizations involved in the project on development of software important to safety (software vendors, engineering organizations, etc.). Some examples of possible software safety reviews are shown in the following.

A project planning review, which can be a review of planning phase documentation for the purpose of evaluation of its completeness and adequacy. The scope of such a review should basically include all planning phases of a software life cycle.

A review of specific issues, which could be a review of one or a combination of issues related to individual phases of a software life cycle (see Section 2.1). The level of detail to which the individual elements of the review will be investigated would be deeper than in other cases, where the scope of the mission is wider. The review can be carried out in any phase of a software life cycle.

A review prior to final acceptance, which would be basically a review of all phases of a software life cycle before the final acceptance of a project at the utility and regulatory level. This review should include also a review of the evidence that all requirements were met during the development process.

2.4. DOCUMENTATION NEEDED TO CONDUCT SAFETY REVIEWS FOR SOFTWARE IMPORTANT TO SAFETY

Typically, software is developed in various phases. Each phase has documentation which is specific to the phase for which it is created. A particular set of phases is listed in Section 2.1. A set of documents which are identified by the individual phases in which they are produced is set forth in the following.

The list of documents is representative of a typical safety related software project. The reviewers may find that not all of the documents listed are available from the host organization. Further, some documents provided by the host organization may not have names identical to those of the list. These problems must be explored so that a comprehensive list of documents needed for the review can be provided by the host organization.

Note that the review may be limited to only one or a few phases of the project. Further, the phases submitted to the review may not be completed and some relevant documentation may therefore be lacking. This must be taken into account when developing the list of documents required for the review.

Traceability is the property which allows a particular item of software to be traced back to the requirement(s) which it satisfies. A design element can be traced to one or more requirements, and a software module can be traced to one or more design elements. A traceability matrix is one way to document traceability. Two-way traceability is preferable to one-way traceability.

Note that two-way traceability is mentioned specifically in the software planning phase (see under configuration management in Section 2.4.1).

Additionally, the review team may be involved with assessment of undocumented items. For example, the review team may be requested to assess the preparedness of the software organization for a specific task. This will require the establishment of criteria for judging the preparedness of the organization and interviewing project personnel.

The following section provides an overview of typical documents. Brief descriptions given are not intended to be a specification of documents' contents but are only a brief and partial outline of the contents.

2.4.1. Software planning phase

Management plan. The software management plan, which is subordinate to the computer system management plan, contains a description of the software organization with the responsibilities of each of the subdivisions of the software organization. Further, it lists all of the plans which must be produced. Approval chains are established for documentation. Teams responsible for software quality assurance and the software verification and validation are expected to be independent from those responsible for the software development — this fact must be noted and the appropriate reporting chain indicated.

Development plan. The software development plan outlines the method to be used for software development. It shows the life cycle model to be used, if one is chosen and identifies the phases of software development together with the activities required in each phase. Documentation required is listed with an outline of the contents of each. software tools to be used for development are to be identified. Project schedules are to be included.

Quality assurance programme description. The software quality assurance programme description typically identifies the elements of software quality which must be included in the development of the project software. Reviews, walkthroughs[5], etc., are identified with the points in the process where they take place. quality assurance audits and their timing and

[5] Walkthrough – A review process in which a designer or programmer leads one or more members of a review team through a computer programme or its design, or parts thereof.

frequency are identified. Standards to be used are to be listed. The quality assurance programme description clearly shows that the quality assurance team is independent of the software development team and that it reports at the same level as the software development team manager.

Configuration management plan. The software configuration management plan concerns itself with the control of all software documentation produced. It identifies what documents are configuration items, the procedure for putting a configuration item under control, how such an item is protected from unauthorized modification, version control and numbering, change control, timing of software backups and methods of protection from loss, etc. It also requires that each item produced be traceable to an item which requires it and be further traceable to each item it uses. Configuration management assures that the software which is running is the software that is supposed to be running.

Verification and validation plan. The software verification and validation plan describes a method to show that the documentation produced in one phase of the software development cycle which is tapped for the next phase of software development is properly carried out in that phase (verification). It further requires an end-to-end check of the integrated computer system (hardware and software) to ensure compliance with the functional, performance and interface requirements (validation). Reports are produced at the end of each phase in the software development cycle attesting to the successful completion of the required verification and validation activity. Initial verification and validation may detect discrepancies which are documented and corrected, but the final verification and validation report indicates that all problems have been addressed. The plan clearly shows that the verification and validation activities are organized independent of the software development team and that it reports at the same level as the software development team manager.

Pre-integration test plan. This plan does not identify all tests that are to be run, but it does identify the minimum set of tests and the documentation which is required for software units and modules up to integration testing.

Acceptance test plans. The acceptance tests are run for the organization that is the customer for the software. There may be more than one acceptance test since the customer may want to see tests run at the factory prior to moving to the site and there may be other tests the customer wishes. The number and scope of such tests is a matter for negotiation between the vendor and the customer. The tests are planned with the co-operation of the customer so that when the tests are successfully completed the customer is satisfied with the product. The Plan describes the tests to be run and the documentation to be produced.

Integration plan. The software integration plan outlines the methods to be employed in integrating all of the software modules into a complete system and the tests which are to be run at each stage of the integration process.

Installation and commissioning plan. This plan outlines the general procedures for installing and testing the complete system (hardware and software) on the operational site.

Safety plan. At each phase in the software development cycle the results of that phase must be examined to determine whether products of that phase have preserved the safety of the system. If items of documentation are vague or unclear and show a potential for misinterpretation, those must be identified and rectified. If unsafe coding practices or indeterminate methods are used, these must be corrected. This plan identifies the method for examining the necessary documentation in order to identify what the problems, if any, may be. A report is issued at the end of each safety analysis stating the results. The safety plan should also identify particular safety enhancements embedded in the system which enhance safety (such as error correction code, safety checks, redundancy and diversity, etc.).

Maintenance plan. As concerns the software, the maintenance plan outlines the process of correcting faults in the software that led to failures during operation. The plan should basically comprise a failure reporting procedure, a fault correction procedure and a procedure for release of corrected software (new versions). The content of the maintenance plan may fall completely under the scope of the configuration management plan.

Training plan. The training plan outlines the process used for education and training of the computer system operators. The training programme may also extend to managers and maintenance personnel.

2.4.2. Configuration control activities

During the process of software development from the first step of putting the requirements document under configuration control until the final commissioning, configuration control is a continuing activity. Changes are made on a regular basis and from time to time activity reports on configuration control are issued indicating the software configuration baseline[6], change control actions undertaken, revision number of the software, revision numbers of various software elements, etc.

2.4.3. Software requirements phase

Requirements documents. The software requirements document must flow from the higher level computer system requirements. Each requirement stated in this document must come directly from one or more requirements in the computer system requirements document. Each requirement should be open to testing as far as possible.

Requirements verification report. This report follows the analysis of the requirements to see if each requirement in the requirements document is derived from one or more requirements in the computer system requirements. It identifies any problems found in the software requirements, corrective action taken and the final state of the document. Note that this description applies to all subsequent verification reports and will not be repeated.

Requirements safety analysis report. Following the software safety plan this report contains the results of the safety analysis stating either that safety has been preserved or that some safety problems have been identified in the requirements document, that the required corrective actions have been taken and that they have been finally resolved. Note that this description applies to all subsequent verification reports and will not be repeated.

Requirements quality assurance report. This report contains the results of any reviews held and any audits held during this phase. Note that this description applies to all subsequent verification reports and will not be repeated.

[6] Configuration baseline – Configuration of a product, formally established at a specific point in time, which serves as reference for further activities.

2.4.4. Commercial off the shelf software and software tools

The present section does not refer to a phase of software development as the other sections (such as 2.4.1–2.4.3) do. The reason behind the present section is that decisions on commercial off the shelf software and tools are often made during or shortly after the requirements phase of a software project (see Section 2.4.3).

Commercial off the shelf software embedded in the final safety system application requires safety qualification commensurate with the safety grade of the application for which it is used. Ultimately the user of the commercial off the shelf software should issue a report for each piece of commercial off the shelf software in which the safety justification is provided that the quality of the software is adequate for the application. This report cites all the evidence considered and why this evidence was considered adequate.

Tools used to support software development activities during the software life cycle should be also analysed as to the degree to which they impact the safety of the end product.

Some of the types of evidence which can be cited as justification in such a report are:

- Software development documentation such as system requirements documents, design documentation, code listings, etc.;
- Documents showing the independence of the verification and validation activity and verification and validation reports;
- Documents showing the software quality assurance activity and quality assurance reports of independent reviews, walkthroughs, etc.;
- System historical documents showing a steady decrease in the number of fault reports from users per unit time (fault detection rate), the number of versions produced over time, etc. The aim here is to show that the potential number of problems remaining in the system is small.

Other evidence of qualification may include consideration of:

- User experience;
- Number of installations;
- Number of machine-years of operation.

Some intangibles which may be considered are:

- The experience of the vendor software organization in the area of related software;
- The commitment of the vendor software organization to the production of high quality error-free software;
- The safety culture[7] of the vendor organization.

2.4.5. Software design phase

The architecture document shows the various software modules and units of the software system, how they interconnect and how they interact with the environment. It is an overview of how the system is put together. One likely form of the architecture document is a drawing.

[7] Safety culture – The assembly of characteristics and attitudes in organizations and individuals which establishes that, as an overriding priority, protection and safety issues receive the attention warranted by their significance.

Design documents for each software module down to the unit level. Each design document contains interface information describing input and output variables and their relationship. A design document contains sufficient information to allow the writing of a code module which takes the input variables and operates upon them to produce the output values specified. Other code modules may be utilized by the module specified to perform its function.

Verification report for each design element.

Safety analysis report for each design element.

Design quality assurance reports for each design element.

2.4.6. Software implementation phase

Software style manual. This manual, which is written early in the software cycle and put under configuration control, describes how a software module or unit is to look, what text is contained in a module, the approximate maximum size of a module, the actions to be taken, the actions to avoid, standard coding practice, national and international standards employed, etc. The attempt is to make the code modules readable and of sufficiently small size in order that they can be reviewed and maintained successfully.

Code listings or equivalent source language for each coded design element. It is expected that each code module will be written according to the software style manual.

Test procedures and reports for each code unit. Each procedure contains an orderly and systematic process for testing a particular code module or unit. It shall describe the test data selection strategy, how the test is performed (manual testing, automatic testing with a test harness and an oracle[8], etc.).

Verification report for each code element.

A safety analysis report for each coded design element.

Implementation quality assurance reports for each coded design element.

2.4.7. Software integration phase

Integration and test procedures, test results documentation (note that results may in fact come from an oracle). Integration consists of the activities that are required in order to combine the hardware and software portions of the final system. The integration activities are governed by the integration plan (see Section 2.4.1). A test procedure is written for each set of elements which are to be integrated, culminating in a test procedure for the complete system. Each procedure specifies those elements which must be available for integration, the tests to be run and the results expected. Integration test reports contain a comparison of the expected results with the results obtained and the resolution of any discrepancies.

Integration verification report.

Integration safety analysis reports.

Integration quality assurance reports.

[8] Oracle – A procedure to state whether the result of a program execution is correct.

2.4.8. Software validation phase

Validation procedures, test results documentation (note that results may in fact come from an oracle). Validation is the set of activities that ensure that the final system, as actually implemented, satisfies the system requirements. The validation is carried out according the verification and validation plan (see Section 2.4.1). The individual procedures describe an orderly and systematic method for completing this function. The results specified are compared with the results obtained and any discrepancies resolved.

Validation safety analysis reports.

Final validation report. This report contains a narrative of the procedures followed in the validation test, the test results, corrective action taken and a statement that the system meets all of the system requirements.

Validation quality assurance reports.

2.4.9. Software installation and commissioning phase

Installation and commissioning and test procedures, test results documentation (note that results may in fact come from an oracle). Software installation and commissioning is the process of installing the final version of software in target environment. Installation and commissioning procedures describe an orderly and systematic method for installing and testing the software to show that it operates according to the original requirements. The expected results from the testing are compared with the actual test results and any discrepancies resolved. A test report is issued describing the testing, the results and how discrepancies were resolved.

Acceptance test procedures, test results documentation (note that results may in fact come from an oracle). The acceptance test is run for the organization that is the customer for the software. The test is planned with the co-operation of the customer so that when the test is successfully run the customer is satisfied with the product. The plan describes the tests to be run, the procedure involved if a test fails and the documentation to be produced. The actual results of the tests are compared with the expected test results and any discrepancies resolved. A report is written describing the running of the test and how discrepancies were resolved.

Installation and commissioning safety analysis report.

Installation and commissioning quality assurance report.

2.4.10. Operation and modification documentation

System operators manual. The software system will require operators to be able to diagnose failures, update and re-load the system when approved changes are made and to re-start the system if a major problem occurs. This manual describes operator interfaces to the software to facilitate these actions.

Maintenance manual. This manual is a variation on the configuration control system used during development. It contains change control procedures, version control procedures, etc.

If the review is called to review a system which has been in operation for some time, such as due to modifications in the system, changes in the plant, upgrading of safety requirements, etc., there are additional documents of interest to the review mission. These include:

Change reports. This documentation describes all changes made to the system since it first was set into operation. If there has been changes to the plant which are relevant for the reviewed system, this should also be documented.

Experience reports. This documentation describes the experiences the users of the system have had since the system first was set into operation. This includes minor problems (major faults should presumably have been corrected upon detection) such as slow responses, bad user interface, etc. Such experience may not be documented, but rather based on interviews with operating staff.

3. REVIEW ORGANIZATION, PLANNING AND SCHEDULING

3.1. PROTOCOL

The review will be organized according to a written agreement between the relevant authorities of the Member State concerned and the IAEA. This agreement will specify the nature of the review and the related legal and financial conditions.

The review, including a preparatory meeting, is normally conducted in English. The requesting organization should therefore provide any necessary interpretation and translation services to facilitate the review.

The report of the review will be confidential and initially will be made available solely to the Member State visited. However, unless the Member State explicitly requests otherwise, this restriction will be lifted following a period of 60 days following acceptance of the final report by the Member State. The decision to implement any recommendations contained in the report will lie entirely with the relevant authorities of the Member State concerned. No legal responsibility is borne by the IAEA nor by any of the individual team members with regard to the advice provided by the review.

3.2. PREPARATORY MEETING

Once agreement has been reached between the Member State and the IAEA on the broad nature of the review, an early preparatory meeting is essential to achieving a well co-ordinated programme of preparations. Good preparation is an essential prerequisite for an efficiently conducted, successful review and this cannot be too strongly emphasized. The meeting should take place some six months before the review. A venue at the IAEA Headquarters in Vienna might be beneficial insofar as Agency staff and management will be easily available.

However, a preferable location is usually the site of the requesting organization. This location is desirable for a number of reasons. The IAEA representative would then have the opportunity, at first hand, to see documentation and to meet involved personnel. A frank and open discussion between the involved parties can then provide a basis for determining not only the phases of a software development process including verification and validation to be included in the scope of the review but also the level of detail desired within each of review elements. By taking some time to determine the extent of documentation available and to learn of the specific concerns of the requesting organization, the IAEA representative may be able to suggest some elements of the software review which can be eliminated or emphasized (as applicable) to increase the effectiveness of the review.

Participants in the preparatory meeting should include:

IAEA

- The IAEA team leader assigned for the review;
- A software expert assisting the IAEA, if necessary;
- The IAEA officer responsible for the area of software relevant to safety or for computer based systems important to safety in broader scope (if available).

Requesting organization

- The requesting organization representative assigned to co-ordinate and to organize all technical, practical and administrative arrangements before and during the review;
- The requesting organization manager or his deputy (if available);
- The software development project co-ordinator or manager (line organization).

The software development project co-ordinator should be experienced in the respective field of review and should be familiar with the organizational and technical aspects of the specific project to be reviewed.

The objectives of the preparatory meeting are as follows:

(1) to define the specific phases (elements) of the software life cycle to be included in the scope of the review;
(2) to define the level of detail to which each element will be reviewed;
(3) to determine the number of team members needed and their respective technical qualifications; to establish the duration of the review;
(4) to determine the technical preparations required to be made by the requesting organization including the records and documentation to be provided by the requesting organization;
(5) to make arrangements for the co-ordination of the planning and preparatory activities of the requesting organization and the IAEA;
(6) to agree to the administrative and domestic arrangements necessary for the review.

3.3. DEFINITION OF THE SCOPE OF THE SOFTWARE SAFETY REVIEW

As described in Section 2.3, the scope of a software safety review should be tailored to meet the particular requirements of the requesting organization. It is important that the requesting organization give considerable thought to its own objectives and expectations for the review. Based on these expectations the IAEA will assist, at the preparatory meeting, in defining the scope for the proposed review.

As mentioned in Section 2.3, the scope of a software review may be different for every case as the problems, conditions and needs of each requesting organization are expected to be unique. In addition, the stage of a project under which the software important to safety is developed will be different.

3.3.1. Elements of the review

It is recommended to use the software life cycle phases (including verification and validation) described in Section 2.1 while defining the individual elements of the review.

3.3.2. Level of detail of the review

After the individual elements for the software safety review have been defined, it is necessary to establish the level of detail to which the experts will investigate each element. Assuming, for example, that the requirements phase of the software development process is an element of the review, the review may cover only the requirements specifications and requirements verification report, or it may also include the computer system specification (so that the reviewer can evaluate the conformance of the software specifications with the computer system specifications). Consideration of questions such as this will greatly assist in the planning of the review and will help to focus the review on those issues of most benefit to the requesting organization.

3.3.3. Applicable guidelines and standards

The guidelines and criteria for performing the review are mainly those established in the IAEA documents, in terms of reference prepared specifically for the project, in guides of the Member State, in well documented international practice and other relevant documents. The IAEA Safety Standards are the main basis for the review. In addition to that, international codes and standards (ISO, IEC) are likely to be used in software important to safety project. Codes and standards of the country requesting the review may be used and, additionally, the codes and standards of the country for which the computer system has been designed and built may be also be included. Examples of such standards, guidelines and other relevant documents are given in the Annex.

3.4. REVIEW PLANNING AND SCHEDULING

Once the scope for the software safety review has been defined, the number of required experts must be determined and the duration of the review agreed. Factors affecting these issues include the finances available and the availability of experts.

It is difficult to establish the required duration of a typical software safety review with any precision. Many variables affect the number of days needed to conduct an effective review. However, general bounding conditions may be suggested which, by necessity, are affected by the specific scope of the review, the elements of software development process to be reviewed, the level of detail of the review of each element and the number and expertise of the review team members.

It should be recognized that a software safety review mission may take a considerable time. A very limited review of only a few software development process elements would probably require only one or two experts for a one week period. A more ambitious scope of work covering several phases of a software development process could require as many as two or three experts for a two week period. A comprehensive review covering all phases of the software development process could require several experts for a period of several weeks. In this case, several reviews might be organized in the course of the entire project.

The overall duration of the review will be affected by many project specific factors such as the availability and accessibility of records and documentation, etc. Of course, not all work of the experts would need to be performed onsite. Where appropriate documents are available, they can be sent to the experts for detailed review before the mission to the requesting organization site actually starts (such as at least four weeks in advance). This process would greatly improve the effectiveness of the time spent at the site during the review.

The dates of the review should be determined at the preparatory meeting. A period of at least six months is recommended between the preparatory meeting and the review to allow for the timely completion of the preparatory work.

3.5. RESPONSIBILITIES OF THE IAEA

3.5.1. Preparation of the working schedule

Based on the agreed scope of the review and the available resources, the IAEA team leader will prepare a working schedule for the review. The working schedule will indicate all the activities necessary for the successful completion of the review, broken down to individual task areas allocated to selected experts. The activities covered will include entrance and exit meetings, review of documentation, personnel interviews, team daily progress reviews and the drafting of the notes.

The IAEA will submit the working schedule to the requesting organization for agreement.

NOTE:
The final schedule must consider the scope of the review already agreed to (see Section 2.3). If the scope and schedule are incompatible, the scope must be re-negotiated. The schedule negotiated with the requesting organization shall be relative to the release of the technical documentation (see Section 3.6.1).

3.5.2. Request for technical documentation

Based on the agreed scope of the review, the IAEA team leader will determine, after discussion with the plant representative, the documentation which should be made available for the review. The documentation falls into three groups:

(1) documents to be distributed to the experts prior to the review mission as briefing and familiarization material (normally, this material is in English and may be specifically prepared for the review);
(2) documents to be reviewed during the review mission (normally, English version should be available);
(3) documents to be available for reference during the review mission (these documents need not be translated in full, but translation services must be available during the review mission to enable effective reference).

The typical documentation related to each phase of a software life cycle (including verification and validation and other project associated activities) is listed in Section 2.4.

The requesting organization shall release all needed documentation for the software safety review. The documentation can be used only for purposes of the review. The reviewers shall be aware on the confidentiality of any of the documentation and are obliged to maintain this confidentiality. Release of any documentation to a third party shall be allowed only with the written consent of the requesting organization.

3.5.3. Team composition

Based on the agreed scope for the review, the IAEA team leader will suggest a tentative team composition. Functional qualifications will be determined, although it is unlikely that specific individuals can be named at the preparatory meeting. The IAEA team leader is responsible for recruiting the experts who will form the team carrying out the review. Experts will be sought whose knowledge and experience matches the defined scope of the review (the IAEA will consider any constraints or preferences the requesting organization expresses in respect of team composition).

At its discretion, the requesting organization can decline some of the proposed IAEA experts bearing in mind protection of industrial property.

For example, if the review scope is to evaluate the whole development process prior to final acceptance of software important to safety, the team members should be specifically qualified by education and practical experience to evaluate the adequacy of the evidence and that all the requirements for the process were met. The combined experience of the team of technical specialists should include general knowledge of all the elements of the software development process (including verification and validation and other project associated activities) listed in Section 2.1. On the other hand, if the review is to be a detailed exercise in verification and validation, the review team should contain technical specialists who are experienced in different types of verification and validation techniques.

3.5.4. Request for support services

The IAEA team leader should present a list of support services required to be provided by the requesting organization in order to ensure the successful completion of the review. He should discuss each item to ensure that the review needs are understood and seek assurance from the requesting organization that they can be provided. The list should also include the need of accommodation for meetings, conferences and working space, overhead projector equipment, international fax and telephone facilities, word processing and copying facilities, interpretation and translation services, etc.

3.6. RESPONSIBILITIES OF THE REQUESTING ORGANIZATION

3.6.1. Technical documentation

After agreeing on the specific documentation required, it is the responsibility of the requesting organization to ensure that the required documents are made available in the form requested in a timely fashion. The requesting organization representative should be in contact with the IAEA team leader to keep him informed of the progress made in document preparation and to give early warning of any difficulties encountered. They must jointly decide if the difficulty is such as to necessitate postponing the review.

3.6.2. Selection of counterparts

It is the responsibility of the requesting organization to select technically qualified persons to act as counterparts to the experts carrying out the review. These should be knowledgeable in the topics to be reviewed and be capable of speaking authoritatively on behalf of the requesting organization. Arrangements should be made for the counterparts to be available throughout the period in which their specialty is reviewed and for the experts to be able to call upon other specialists for support, as needed, during the course of the review.

3.6.3. Support services

The requesting organization is responsible for providing the review with the support services required by the IAEA (Section 3.5.4) to ensure the successful completion of the review. The requesting organization should give careful consideration to the list of support services presented by the IAEA at the preparatory meeting and indicate any problems that they might foresee.

3.6.4. Administrative requirements

The requesting organization should inform the IAEA of the information required of each of the team members to allow admission to its premises. Visa requirements should also be made known well ahead of schedule.

3.7. MEETINGS

The working schedule should specify the entrance meeting, daily meetings and exit meeting that will take place during the review.

3.7.1. Entrance meeting

This meeting, normally chaired by the requesting organization management, allows the requesting organization manager to state his expectations of the review, to briefly outline the situation of his organization, to welcome the IAEA team and to introduce the requesting organization specialist. The IAEA team leader will usually respond by introducing the experts and briefly outlining the method and logistics they will follow.

3.7.2. Daily meetings

Meetings should be held daily between all team members to review progress, raise issues and revise the work programme to overcome any difficulties encountered. It may prove useful to admit the counterparts as observers to these meetings.

A daily meeting between the IAEA team leader and requesting organization management should be held to discuss any findings or problem issues identified during the day's work. These meetings are primarily intended to report progress but can be used to seek assistance in overcoming problems experienced by the experts in their work.

3.7.3. Exit meeting

The exit meeting is a formal meeting at which the experts present a brief verbal account of their work and the findings of the review. The closing remarks from the requesting organization manager mark the formal end of the onsite phase of the review.

4. CONDUCT OF THE REVIEW

4.1. THE SOFTWARE SAFETY REVIEW

The review team uses two methods to acquire the information needed to develop recommendations. These are:

- review of documents and written material, and;
- interviews with personnel.

The integration of these sources of information should be arranged in such a way as to allow the team to first gain a general understanding of the overall software design, followed by the software safety philosophy and finally the details of the documentation and implementation of the software system.

Recommendations and suggestions should be formulated on the basis of weaknesses identified. Similarly, good practices encountered in the review should be documented for the benefit of other utilities and described in the technical notes in sufficient detail to be readily understandable.

4.1.1. Review of documentation

The list of documents to be reviewed should be negotiated with the counterparts of the requesting organization and a list of the agreed upon documents prepared during the pre-review preparatory meeting.

The documentation needed for the software safety review will vary depending on the specific scope of the review and the level of detail to be provided. For additional guidance, refer to Section 2.4.

4.1.2. Interviews with personnel

After the review of written material, interviews with personnel can be used to:

- obtain additional information not covered in the documentation;
- review issues arising out of the documentation reviews.

The interviews also provide an opportunity for important information to be exchanged between reviewers and counterparts. Interviews should be give-and-take discussions; not interrogation of counterparts by reviewers. When properly conducted, such interviews form a critical part of the overall reviews.

4.2. REPORTING

4.2.1. General description

A suggested table of contents for the final report is given in Appendix II. The main body of the report is located in the evaluation sections of the final report. In these sections, the report should discuss each element of the software process that has been included in the scope of the review.

4.2.2. Recommendations, suggestions and good practice

The software safety review may result in recommendations or suggestions or identifications of good practices in accordance with the following definitions:

Recommendation: a recommendation is advice on how improvements in the software development practice or product can be made in areas that have been reviewed. Such advice is based on proven international practices and should address the root causes rather than the symptoms of concerns raised. A recommendation should be specific, realistic and designed to result in tangible improvements.

Suggestion: a suggestion is either an additional proposal in conjunction with a recommendation, or may stand on its own. It may indirectly contribute to improvements in the software development practice or product, but it is primarily intended to make the practice or product performance more effective, to indicate enhancements of existing practices and to point out alternatives to current practices.

Good practice: a good practice is an indication of an outstanding performance, programme, activity or equipment markedly superior to that observed elsewhere not just the fulfillment of current requirements and expectations. It should be superior enough to be brought to the attention of other software development groups as a model in the general drive for excellence.

4.2.3. Technical notes

During the course of the review, after each co-ordination meeting, team members will write detailed technical notes on their observations and conclusions, including any recommendations, suggestions or good practices. These form the basis of oral presentations at the exit meeting. One or more copies of the technical notes are given to the requesting organization representative prior to the exit meeting.

The technical notes are the field notes of the individual team members and are not intended for distribution beyond the requesting organization. Guidance for drafting technical notes is given in Appendix II [2].

4.3. THE REVIEW TEAM REPORT

On completion of the review, the team leader will prepare the team report on the basis of the technical notes. This is an official team document that summarizes the team's main findings and conclusions including all recommendations, suggestions and good practices. Before the text is finalized, the management of the requesting organization whose software has been reviewed will be given the opportunity to comment. The final report will be submitted through official channels to the Member State concerned. The IAEA will restrict distribution to the utility concerned, the contributors to the report and responsible IAEA staff. Further distribution is at the discretion of the Member State.

Appendix I

LIST OF REVIEW TOPICS

This list identifies series of topics relevant to the review of the different phases in the software development and the corresponding verification and validation activities. The list is correlated with the IAEA Safety Guide [1] which covers each of the topics in more detail. Please note, however, that this list is not necessarily complete and the review team members should review and modify it based on their own expertise.

Project planning and management

— Project definition

Documentation of all phases in the project plan, with adequate level of detail to identify any deviations in time to take corrective action;
Project organization, including lines of communication, lines of responsibilities and lines of authority;
Circulation of the project to all participants and participating departments concerned with the plan;
Approval of the project plan by management.

— Quality assurance

A quality assurance programme;
An independent quality assurance team.

— Qualification, certification and approval

Specified applicable standards and guidelines;
Qualification, certification and approval plan.

— Configuration management

Organization;
Identification of items;
Procedures for change control;
Revision control.

— Verification and validation

A formalized documented verification and validation plan;
Independent verification and validation requirement;
Internationally accepted verification and validation standards and guidelines.

— Documentation

Content, style and level of depth for each document;
Documentation standards and guidelines specified;
Formal document control mechanism associated with the documentation and document revisions.

Verification and safety analysis

This surveys verification and validation and dependability analysis activities which can be applied in the different project phases.

— Safety aspects
 Hazard analysis;
 Preservation of safety through all phases;
 Search for possible common cause failures.

— Correctness aspects
 Coverage of all requirements;
 Static analysis;
 Document inspections;
 Formal verification;
 Reverse engineering.

— Testing
 Testing strategy and coverage;
 Testing preparation and conduct;
 Test data selection strategies (such as debug testing, random testing, simulation testing, etc.);
 Test results.

— Auxiliary aspects
 Assessment of auxiliary programs;
 Assessment of development tools.

Software requirements

— Functionality
 Specification of the functionality required from the software;
 Timing requirements;
 Accuracy;
 Division of functionality between different design elements.

— Structural requirements.
 Traceability requirements;
 Requirements on the software structure;
 Requirements on interfaces;
 - to the external world,
 - between modules.

— Dependability aspects
 Reliability requirements;
 Fault tolerance (redundancy, diversity, defence in depth, etc.);
 Safety checks;
 Requirements to prevent failure propagation;
 Security requirements.

— Operability aspects
 Specifications concerning maintenance or future modifications;
 User interfaces.

— Requirement reviews
 Structured walkthroughs;
 Review reports.

Software design

— Structural attributes

Software architecture;
Implementation constraints;
Reviewability;
Simplicity;
Conformance to design standards and guidelines;
Traceability;
Verifiability.

— Commercial off the shelf software

Vendor qualification (vendor's pedigree, quality assurance procedures, etc.);
Available system documentation:

- Product description (even program listing if possible),
- System user manuals,
- Development documents,
- Verification documents:

  Version history,
  Failure reports,
  User experience.

— Design facilities

Design language;
Development tools;
Dependability aspects;
Preservation of safety;
Fault tolerance (redundancy, diversity, etc.);
Safety checks;
Techniques used to prevent failure propagation;
Security requirements.

— Design reviews

Structured walkthroughs;
Review reports.

Software implementation

— Structural attributes

Module decomposition;
Reviewability;
Simplicity;
Conformance to coding rules;
Traceability;
Verifiability;
Change and version control.

— Software details

Program structure;
Data structures;

Programming languages;
Code listings;
Use of tools.

— Dependability aspects
Preservation of safety;
Achievement of fault tolerance;
Implemented safety checks (self supervision and self checking).

— Code reviews
Structured walkthroughs;
Code inspection;
Review reports.

System integration

— Topics to investigate
Traceability to system requirements;
Only verified hardware and software modules;
Integration verification and testing;
Justification of performed testing.

— Documents
Traceability reports;
Test reports;
Problem reports.

Installation

— Topics to investigate
Installation procedure;
Onsite testing;
Independent commissioning team;
Configuration control.

— Documents
Installation reports;
Test reports;
Commissioning reports.

Validation

— Topics for validation
Conformance with validation plan;
Independent validation team;
Coverage of all software requirements;
Adequate timing performance.

- Documents
  - Test reports;
  - Fault detection and correction reports.

Operation

- General topics
  - In-service testing;
  - Maintenance;
  - Corrective actions;
  - Anomalies.

- Documents
  - Documentation of planned tests during operation;
  - Operation problem. (May not be documented. May be based on interviews with operating staff.)

Post-delivery modifications

- General topics
  - Conformance with configuration management requirements;
  - Justification of software changes;
  - Safety classifications changed;
  - Safety preservation;
  - Reapplication of verification and validation procedures.

- Documents
  - Error reports;
  - Software system change reports, such as detailed description of the modifications;
  - Reports on testing of modifications;
  - Plant change reports.

Appendix II
GUIDANCE FOR DRAFTING TECHNICAL NOTES

II.1. INTRODUCTION

Writing the technical notes is an important task for reviewers. The team members will collect a large amount of information that must be recorded. These facts, impressions and conclusions must be written clearly and concisely since, once the team leaves, the requesting organization will have to work from the technical notes.

In writing the technical notes, the following should be taken into account:

- Emphasis should be given to the reviewers' observations, with clear conclusions and the minimum of description;
- The language should be clear, concise, objective and impersonal;
- Short, direct sentences aid understanding;
- The official names (or official translation) should be used to designate organizational units, positions and systems, and;
- If abbreviations or acronyms are used, they should be defined upon their first use.

The technical notes should be written, in English, daily from the first day of review and modified and supplemented, as necessary, throughout the entire period of the review.

II.2. FORMAT

A suggested table of contents is provided in the following:

Summary. A one to two page summary is the first section. This summary should contain the important aspects of the review with emphasis on its findings, conclusions and recommendations.

Introduction. The introduction should present the background material for the review including previous reviews related to software at this site and facility. The objectives of the review should be clearly stated.

Conduct of the review. This section provides the details of the review including schedule, locations, participants, detailed scope of the review and a summary of the reference material provided.

Evaluation sections. The evaluation sections contains recommendations or suggestions of concerns raised by reviewers for which actions are required, the suggested action for each concern and the evidence that will be required to show that the problem no longer exists.

For each issue raised there should be:

- a statement of the specific concern;
- background including a summary of previous reviews (open items, recommended actions, actions taken, etc.);
- a list of reference material used during the review;

- a summary of the documentation presented to the IAEA review team during the review;
- a summary of discussions and remarks including the positions of the in-country counterparts as they relate to the issue under consideration.

Also, the evaluation sections may contain good practices encountered during the review.

Conclusions. Conclusions state the observations of the IAEA review team with regard to the scope of the review and the information provided by the in-country counterparts.

II.3. NUMBERING SYSTEM

Recommendations, suggestions and good practices should be identified by a four digit number. The first three digits give the area and sub-area of the review and the fourth digit is (1) for a recommendations and suggestions or (2) for good practices.

Each recommendation and suggestion must be preceded and supported by a basis which is a statement of the concern giving rise to a recommendation or suggestion. It should briefly identify the issue but not introduce new material or thoughts.

If there are several recommendations and suggestions related to one basis, these should be itemized and each individual item identified (a), (b), (c), etc.

If a relevant sub-area does not call for a recommendation nor a suggestion then a suitable phrase to this effect should be entered in the technical notes, such as “in the reviewed area the performance corresponds with normal proven and effective international practices”. If good practices are identified, then the sub-area number need not be included.

SAMPLE LAYOUT

X.X. SECTION

X.X.X. Paragraph

Preamble: highlights of good performance and problem areas, if any

X.X.X.1. Recommendations and suggestions

(1) Basis:

(a) Recommendation

(b) Recommendation

(c) Suggestion

If there is another basis and subsequent recommendation(s) and suggestion(s):

(2) Basis:

Etc.

If a good practice is identified, then X.X.X.2 would be inserted in a similar format**:**

X.X.X.2. Good practices

Good practice:

REFERENCES

[1] INTERNATIONAL ATOMIC ENERGY AGENCY, Software for Computer Based Systems Important to Safety in Nuclear Power Plants, Safety Standard Series No. NS-G-1.1, Vienna (2000).

[2] INTERNATIONAL ATOMIC ENERGY AGENCY, Guidelines for IAEA International Regulatory Review Teams (IRRTs), IAEA-TECDOC-703, IAEA, Vienna (1993).

Annex

EXAMPLES OF STANDARDS, GUIDES AND OTHER DOCUMENTS RELATED TO SOFTWARE IMPORTANT TO SAFETY

INTERNATIONAL ATOMIC ENERGY AGENCY, Safety of Nuclear Power Plants: Design, Safety Standards Series No. NS-R-1, Vienna (2000).

INTERNATIONAL ATOMIC ENERGY AGENCY, Software for Computer Based Systems Important to Safety in Nuclear Power Plants, Safety Standard Series No. NS-G-1.1, Vienna (2000).

INTERNATIONAL ATOMIC ENERGY AGENCY, Protection System and Related Features in Nuclear Power Plants: A Safety Guide, Safety Series No. 50-SG-D3, IAEA, Vienna (1980).

INTERNATIONAL ATOMIC ENERGY AGENCY, Safety Related Instrumentation and Control Systems for Nuclear Power Plants: A Safety Guide, Safety Series No. 50-SG-D8, IAEA, Vienna (1984).

INTERNATIONAL ATOMIC ENERGY AGENCY, Quality Assurance for Computer Software Related to Safety, Technical Reports Series No. 282, IAEA, Vienna (1988).

INTERNATIONAL ATOMIC ENERGY AGENCY, Software Important to Safety in Nuclear power plants, Technical Reports Series No. 367, IAEA, Vienna (1994).

INTERNATIONAL ATOMIC ENERGY AGENCY, Safety Assessment of Computerized Control and Protection Systems, IAEA-TECDOC-780, Vienna (1994).

INTERNATIONAL ATOMIC ENERGY AGENCY, Reliability of Computerized Safety Systems at Nuclear Power Plants, IAEA-TECDOC-790, Vienna (1995).

INTERNATIONAL ATOMIC ENERGY AGENCY, Verification and Validation of Software Related to Nuclear Power Plant Instrumentation and Control, Technical Reports Series No. 384, Vienna (1999).

INTERNATIONAL ELECTROTECHNICAL COMMISSION, Guide to the Application of Digital Computers to Nuclear Reactor Instrumentation and Control, IEC Standard No. 643, IEC, Geneva (1979).

INTERNATIONAL ELECTROTECHNICAL COMMISSION, Analysis Technique for System Reliability Procedure for Failure Mode and Effect Analysis, IEC Standard No. 812, IEC, Geneva (1985).

INTERNATIONAL ELECTROTECHNICAL COMMISSION, Software for Computers in the Safety Systems of Nuclear Power Stations, IEC Standard No. 880, IEC, Geneva (1986).

INTERNATIONAL ELECTROTECHNICAL COMMISSION, Programmed Digital Computers Important to Safety for Nuclear Power Stations, IEC Standard No. 987, IEC, Geneva (1989).

INTERNATIONAL ELECTROTECHNICAL COMMISSION, Analysis Techniques for Dependability — Reliability Block Diagram Method, IEC Standard No. 1078, IEC, Geneva (1991).

INTERNATIONAL ELECTROTECHNICAL COMMISSION, The Classification of Instrumentation and Control Systems Important for Safety for Nuclear power plants, IEC Standard No.1226, IEC, Geneva (1993).

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, ISO 9000 Standard series: Quality Management and Quality Assurance Standards, Part 3: Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software, ISO, Geneva (1990).

GLOSSARY

The following definitions apply for the purposes of the present publication.

bus A set of conductors connecting the various functional units in a computer.

configuration baseline Configuration of a product, formally established at a specific point in time, which serves as reference for further activities.

defence in depth A hierarchical deployment of different levels of equipment and procedures in order to maintain the effectiveness of physical barriers placed between a radiation source or radioactive materials and workers, members of the public or the environment, in operational states and, for some barriers, in accident conditions.

dependability Trustworthiness of the delivered service such that reliance can justifiably be placed on this service Reliability, availability, safety are attributes of dependability.

oracle A procedure to state whether the result of a program execution is correct.

safety culture The assembly of characteristics and attitudes in organizations and individuals which establishes that, as an overriding priority, protection and safety issues receive the attention warranted by their significance.

system life cycle All stages through which a system passes from conception to final disposal.

walkthroughs A review process in which a designer or programmer leads one or more members of a review team through a computer programme or its design, or parts thereof.

CONTRIBUTORS TO DRAFTING AND REVIEW

Dahll, G.	OECD Halden Reactor Project, Norway
Duong, M.	International Atomic Energy Agency
Krs, P.	State Office for Nuclear Safety, Czech Republic
Kulig, M.	International Atomic Energy Agency
Pachner, J.	International Atomic Energy Agency
Reznik, V.	International Atomic Energy Agency
Wyman, R. H.	United States of America

Consultants Meetings

Vienna, Austria: 29 September–3 October 1997; 9–13 March 1998