AS IEC 61511.2—2004
IEC 61511-2:2003
Reconfirmed 2015

Australian Standard™

Functional safety—Safety instrumented systems for the process industry sector

Part 2: Guidelines for the application of AS IEC 61511.1

This Australian Standard was prepared by Committee IT-006, Information Technology for Industrial Automation and Integration. It was approved on behalf of the Council of Standards Australia on 5 March 2004 and published on 10 May 2004.

The following are represented on Committee IT-006:

Association of Consulting Engineers Australia

Australian Electrical and Electronic Manufacturers Association

CSIRO Centre for Planning and Design

CSIRO Manufacturing & Infrastructure Technology

Department of Defence (Australia)

Institute of Instrumentation, Control and Automation Australia

Institution of Engineers Australia

Monash University

RMIT University

The University of Melbourne

This Standard was issued in draft form for comment as DR 04054.

STANDARDS AUSTRALIA

RECONFIRMATION

OF

AS IEC 61511.2—2004

Functional safety—Safety instrumented systems for the process industry sector
Part 2: Guidelines for the application of AS IEC 61511-1

RECONFIRMATION NOTICE

Technical Committee IT-006 has reviewed the content of this publication and in accordance with Standards Australia procedures for reconfirmation, it has been determined that the publication is still valid and does not require change.

Reconfirmed documents cannot be amended, but may be formally revised and a new edition published.

Certain documents referenced in the publication may have been amended since the original date of publication. Users are advised to ensure that they are using the latest versions of such documents as appropriate, unless advised otherwise in this Reconfirmation Notice.

Approved for reconfirmation in accordance with Standards Australia procedures for reconfirmation on 14 July 2015.

The following are represented on Technical Committee IT-006:

Australia Safety Critical Systems Association
Australian Computer Society
Australian Industry Group
Australian Petroleum Production and Exploration Association
Consult Australia
Engineers Australia
Institute of Chemical Engineers Australia
Institute of Instrumentation, Control & Automation
ISACA
Process Control Society
The University of Queensland
Workplace Health and Safety Queensland

NOTES

Australian Standard™

Functional safety—Safety instrumented systems for the process industry sector

Part 2: Guidelines for the application of AS IEC 61511.1

First published as AS IEC 61511.2—2004.

Published by Standards Australia International Ltd
GPO Box 5420, Sydney, NSW 2001, Australia

ISBN 0 7337 5914 9

PREFACE

This Standard was prepared by the Standards Australia Committee IT-006, Information Technology for Industrial Automation and Integration.

This Standard is identical with, and has been reproduced from IEC 61511-2:2003, *Functional safety—Safety instrumented systems for the process industry sector*—Part 2*: Guidelines for the application of IEC 61511-1*.

The objective of this Standard is to provide guidance on the specification, design, installation, operation and maintenance of Safety Instrumented Functions and related safety instrumented systems as defined in IEC 61511-1.

This Standard is Part 2 of AS IEC 61511—2004, *Functional safety—Safety instrumented systems for the process industry sector*, which is published in parts as follows:

Part 1: Framework, definitions, system, hardware and software requirements
Part 2: Guidelines for the application of AS IEC 61511-1 (this Standard)
Part 3: Guidance for the determination of the required safety integrity levels

The terms ‘normative’ and ‘informative’ are used to define the application of the annex to which they apply. A normative annex is an integral part of a standard, whereas an informative annex is only for information and guidance.

As this Standard is reproduced from an International Standard, the following applies:

(a) Its number does not appear on each page of text and its identity is shown only on the cover and title page.

(b) In the source text ‘this international standard’ should read ‘this Australian Standard’.

(c) A full point should be substituted for a comma when referring to a decimal marker.

CONTENTS

INTRODUCTION

Safety instrumented systems have been used for many years to perform safety instrumented functions in the process industries. If instrumentation is to be effectively used for safety instrumented functions, it is essential that this instrumentation achieves certain minimum standards.

This International Standard addresses the application of safety instrumented systems for the Process Industries. It also deals with the interface between safety instrumented systems and other safety systems in requiring that a process hazard and risk assessment be carried out. The safety instrumented system includes sensors, logic solvers and final elements.

This International Standard has two concepts, which are fundamental to its application; safety lifecycle and safety integrity levels. The safety lifecycle forms the central framework which links together most of the concepts in this International Standard.

The safety instrumented system logic solvers addressed include Electrical (E)/Electronic (E)/ and Programmable Electronic (PE) technology. Where other technologies are used for logic solvers, the basic principles of this standard may also be applied. This standard also addresses the safety instrumented system sensors and final elements regardless of the technology used. This International Standard is process industry specific within the framework of the IEC 61508 series.

This International Standard sets out an approach for safety lifecycle activities to achieve these minimum standards. This approach has been adopted in order that a rational and consistent technical policy is used. The objective of this standard is to provide guidance on how to comply with IEC 61511-1.

To facilitate use of this standard, the clause and subclause numbers provided are identical to the corresponding normative text in 61511-1 (excluding the annexes).

In most situations, safety is best achieved by an inherently safe process design whenever practicable, combined, if necessary, with a number of protective systems which rely on different technologies (for example, chemical, mechanical, hydraulic, pneumatic, electrical, electronic, thermodynamic (for example, flame arrestors), programmable electronic) which manage any residual identified risk. Any safety strategy considers each individual safety instrumented system in the context of the other protective systems. To facilitate this approach, this standard

- requires that a hazard and risk assessment is carried out to identify the overall safety requirements;
- requires that an allocation of the safety requirements to the safety functions and related safety systems, such as the safety instrumented system(s), is carried out;
- works within a framework which is applicable to all instrumented methods of achieving functional safety;
- details the use of certain activities, such as safety management, which may be applicable to all methods of achieving functional safety.

This International Standard on safety instrumented systems for the process industry:

- addresses relevant safety lifecycle stages from initial concept, through design, implementation, operation and maintenance and decommissioning;
- enables existing or new country specific process industry standards to be harmonized with this standard.

This standard is intended to lead to a high level of consistency (for example, of underlying principles, terminology, information) within the process industries. This should have both safety and economic benefits.

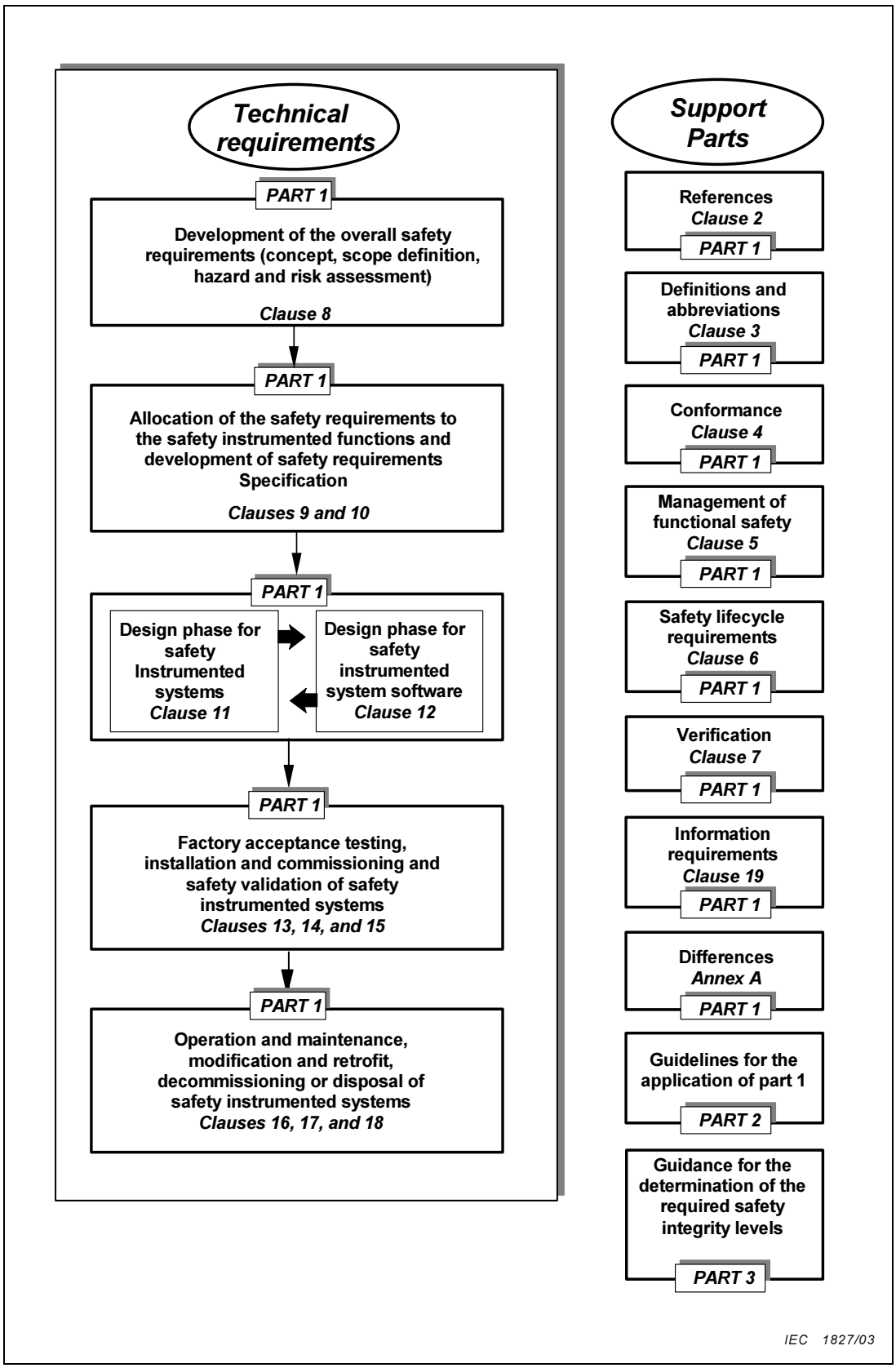


Figure 1 – Overall framework of this standard

STANDARDS AUSTRALIA

Australian Standard

Functional safety—Safety instrumented systems for the process industry sector
Part 2: Guidelines for the application of AS IEC 61511.1

1 Scope

IEC 61511-2 provides guidance on the specification, design, installation, operation and maintenance of Safety Instrumented Functions and related safety instrumented system as defined in IEC 61511-1. This standard has been organized so that each clause and subclause number herein addresses the same clause number in IEC 61511-1 (with the exception of the annexes).

2 Normative references

No further guidance provided.

3 Terms, definitions and abbreviations

No further guidance provided except for 3.2.68 and 3.2.71 of IEC 61511-1.

3.2.68 A safety function should prevent a specified hazardous event. For example, “prevent the pressure in vessel #ABC456 exceeding 100 bar.” A safety function may be achieved by

a) a single safety instrumented system (SIS), or

b) one or more safety instrumented systems and/or other layers of protection.

In case b), each safety instrumented system or other layer of protection has to be capable of achieving the safety function and the overall combination has to achieve the required risk reduction (process safety target).

3.2.71 Safety instrumented functions are derived from the safety function, have an associated safety integrity level (SIL) and are carried out by a specific safety instrumented system (SIS). For example, “close valve #XY123 within 5 s when pressure in vessel #ABC456 reaches 100 bar”. Note that components of a safety instrumented system may be used by more than one safety instrumented function.

4 Conformance to this International Standard

No further guidance provided.

5 Management of functional safety

5.1 Objective

The objective of Clause 5 of IEC 61511-1 is to provide requirements for implementing the management activities that are necessary to ensure that the functional safety objectives are met.

5.2 Requirements

5.2.1 General

5.2.1.1 No further guidance provided.

5.2.1.2 When an organization has responsibility for one or more activities necessary for functional safety and that organization works according to quality assurance procedures, then many of these activities described in this clause will already be carried out for the purposes of quality. Where this is the case, it may be unnecessary to repeat these activities for the purposes of functional safety. In such cases, the quality assurance procedures should be reviewed to establish that they are suitable so that the objectives of functional safety will be achieved.

5.2.2 Organization and resources

5.2.2.1 The organizational structure associated with safety instrumented systems within a Company/Site/Plant/Project should be defined and the roles and responsibilities of each element clearly understood and communicated. Within the structure, individual roles, including their description and purpose should be identified. For each role, unambiguous accountabilities should be identified; and specific responsibilities should be recognised. In addition, whom the individual reports to and who makes the appointment should be identified. The intent is to ensure that everyone in an organization understands their role and responsibilities for safety instrumented systems.

5.2.2.2 The skills and knowledge required to implement any of the activities of the safety life cycle relating to the safety instrumented systems should be identified; and for each skill, the required competency levels should be defined. Resources should be assessed against each skill for competency and also the number of people per skill required. When differences are identified, development plans should be established to enable the required competency levels to be achieved in a timely manner. When shortages of skills arise, suitably qualified and experienced personnel may be recruited or contracted.

5.2.3 Risk evaluation and risk management

The requirement stated in 5.2.3 of IEC 61511 is that hazards are identified, risks evaluated and the necessary risk reduction is determined. It is recognized that there are numerous different methodologies available for conducting these evaluations. IEC 61511-1 does not endorse any particular methodology. Instead, the reader is encouraged to review a number of methodologies on this issue in IEC 61511-3. See 8.2.1 for further guidance.

5.2.4 Planning

The intent of this subclause is to ensure that, within the overall project, adequate safety planning is conducted so that all of the required activities during each phase of the lifecycle (for example, engineering design, plant operation) are addressed. The standard does not require any particular structure for these planning activities, but it does require periodic update or review of them.

5.2.5 Implementing and monitoring

5.2.5.1 The intent of this subclause is to ensure that effective management procedures are in place to

- ensure that all recommendations resulting from hazard analysis, risk assessment, other assessment and auditing activities, verification and validation activities are satisfactorily resolved.
- determine that the SIS is performing in accordance with its safety requirements specification throughout its operational lifetime.

5.2.5.2 Note that, in this context, suppliers could include design contractors and maintenance contractors as well as suppliers of components.

5.2.5.3 A review of the SIS performance should be periodically undertaken to ensure the original assumptions made during the development of the safety requirements specification (SRS) are still adhered to. For example, a periodic review of the assumed failure rate of different components in a SIS should be carried out to ensure that it remains as originally defined. If the failure rates are worse than originally anticipated, a design modification may be necessary. Likewise, the demand rate on the SIS should be reviewed. If the rate is more than that which was originally assumed, then an adjustment in the SIL may be needed.

5.2.6 Assessment, auditing and revision

Assessments and audits are tools targeted at the detection and elimination of errors. The paragraphs below make clear the distinction between these activities

Functional safety assessment aims to evaluate whether provisions made during the assessed lifecycle phases are adequate for the achievement of safety. Judgements are made by assessors on the decisions taken by those responsible for the realisation of functional safety. An assessment would for example be made prior to commissioning as to whether procedures for maintenance are adequate.

Functional safety auditors will determine from project or plant records whether the necessary procedures have been applied at the specified frequency by persons with the necessary competence. Auditors are not required to make judgements on the adequacy of the work they are considering. However, if they became aware that there would be benefits in making changes, then an observation should be included in the report.

It should be noted that in many cases there can be an overlap between the work of the assessor and the auditor. For example an auditor may need to determine not only whether an operator has been given the necessary training but in addition make judgements as to whether the training has resulted in the required competency.

5.2.6.1 Functional safety assessment

5.2.6.1.1 The use of Functional Safety Assessment (FSA) is fundamental in demonstrating that a Safety Instrumented System (SIS) fulfils its requirements regarding safety instrumented function(s) and Safety Integrity Level (SIL). The basic objective of this assessment is to demonstrate compliance with agreed standards and practices through independent assessment of the system's development process. An assessment of a SIS may be needed at different lifecycle stages. In order to conduct an effective assessment, a procedure should be developed that defines the scope of this assessment along with some guidance on the makeup of the assessment team.

The following attributes are considered good practice for Functional Safety Assessment:

- A plan should be generated for each FSA identifying such arrangements as the scope of the assessment, the assessors, the competencies of the assessors and the information to be generated by the assessment.
- The FSA should take into account other standards and practices, which may be contained within external or internal corporate standards, guides, procedures or codes of practice. The FSA plan should define what is to be assessed for the particular assessment/ system/application area.
- The frequency of FSAs may vary across different system developments but as a minimum should always take place before the potential hazards being presented to the system. Some companies also like to conduct an assessment prior to the construction/installation phase to prevent costly rework later in the lifecycle.
- FSA frequency and rigour should be defined taking into account system attributes such as:
    - complexity;
    - safety significance;
    - previous experience of similar systems;
    - standardization of design features.

- Sufficient evidence of design, installation, verification and validation activities should be available prior to the assessment. The availability of sufficient evidence could itself be an assessment criterion. The evidence should represent the current/approved state of system design or installation.
- The independence of the assessor(s) must be appropriate.
- The assessor(s) should have experience and knowledge appropriate to the technology and application area of the system being assessed.
- A systematic and consistent approach to FSA should be maintained throughout the lifecycle and across systems. FSA is a subjective activity therefore detailed guidance, possibly through the use of checklists, as to what is acceptable for an organisation should be defined to remove as much subjectivity as possible.

Records generated from the FSA should be complete and the conclusions agreed with those responsible for the management of functional safety for the SIS prior to commencement of the next lifecycle phase.

5.2.6.1.2 The need for someone independent to the project team is to increase objectivity in the assessment. The need for someone of senior stature (for example, experience, grade level, position) is to ensure their concerns are duly noted and addressed. As the note also suggests, on some large projects or assessment teams, it may be necessary to have more than one senior person on this team that is independent to the original project team.

Depending upon the company organisation and expertise within the company, the requirement for an independent assessor may have to be met by using an external organisation. Conversely, companies that have internal organisations skilled in risk assessment and the application of safety instrumented systems, which are independent to and separate (by ways of management and other resources) from those responsible for the project, may be able to use their own resources to meet the requirements for an independent organisation.

5.2.6.1.3 The amount of assessment depends on the size and complexity of a project. It may be possible to assess the results of different phases at the same time. This is particularly true in the case of small changes in a running plant.

5.2.6.1.4 In some countries, a functional safety assessment undertaken at stage 3 is often referred to as the Pre-Startup-Safety-Review (PSSR).

5.2.6.1.5 No further guidance provided.

5.2.6.1.6 No further guidance provided.

5.2.6.1.7 The assessment team should have access to any information they deem necessary for them to conduct the assessment. This should include information from the hazard and risk assessment, design phase through installation, commissioning and validation.

5.2.6.2 Auditing and revision

5.2.6.2.1 This subclause is intended to give guidance about auditing, using an example illustrating relevant activities.

a) Audit categories

Safety instrumented system audits provide beneficial information to plant management, instrument maintenance engineers and instrument design engineers. This enables management to be proactive and aware of the degree of implementation and effectiveness of their safety instrumented systems. Many types of audits, which can be carried out exist. The actual type, scope, and frequency of the audit of any specific activity should reflect the potential impact of the activity on the safety integrity.

Types of audit include:

1) audits, both independent and self-audit;

2) inspections;

3) safety visits (for example, plant walk about and incident review);

4) safety instrumented systems surveys (via questionnaires).

A distinction needs to be made between “surveillance and checking” and audit activities. Surveillance and checking focuses on evaluating the performance of specific lifecycle activities (for example, supervisor checking completion of maintenance activity prior to the component being returned to service.) In contrast, audit activities are more comprehensive and focus on overall implementation of safety instrumented systems concerning the safety lifecycle. An audit would include determination as to whether the surveillance and checking program is carried out.

Audits and inspections may be carried out by a company’s/site’s/plant’s/project’s own staff (for example, self-audit) or by independent persons (for example, corporate auditors, quality assurance department, regulators, customers or third parties).

Management at the various levels may want to apply the relevant type of audit to gain information on the effectiveness of the implementation of their safety instrumented systems. Information from audits could be used to identify the procedures that have not been properly applied, leading to improved implementation.

b) Audit strategy

Site/plant/project implementing audit programmes might consider rolling, independent or self-audit and inspection programmes.

Rolling programmes are updated regularly to reflect previous safety instrumented systems performance and audit results, and current concerns and priorities. These cover all site/plant/project related activities and aspects of the safety instrumented systems in an appropriate time period and to an appropriate depth.

The primary reason for, and the added value from audits comes from acting on the information they provide in a timely manner. The actions aim to strengthen the effectiveness of safety instrumented systems, for example, to help minimize the risk of employees or members of the public being injured or killed, contribute to improving safety culture, contribute to prevent any avoidable release of substance into the environment.

In summary, the audit strategy may have a mix of audits types, driven by management (the customer), and in order to feed back the relevant information up the management chain for timely action.

c) Audit process and protocols

The overall aim is to achieve maximum value from the performance of the audit, which can only be achieved when all parties (including auditors, contact nominee, plant managers and head of departments, etc.) understand the need for and can influence each audit. The following audit process and protocols might help to ensure some consistency in the approach to achieving these aims. They bear on the following five key stages of the audit process:

1) Audit strategy and programme

The purpose of each audit should be clearly defined and the audit groups identified, together with the roles and responsibilities of each audit group.

There should be an auditing strategy.

There should be a programme of audits.

There should be regular reviews of the audit process, programme and strategy implementation.

2) Audit preparation and pre-planning

Prior to commencement of an audit, the senior manager of the site/plant/project and/or the appropriate audit coordinator should identify a contact nominee.

The auditors and contact nominee should at an early stage discuss, understand and agree on:

– the scope of the audit;
– the timing of the audit;
– the people who need to be available;
– the basis for the audit or audit standard;
– putting the extra effort into the preparation stage and involving the plant personnel, thereby increasing the chances of a successful audit.

The following should be used as a guide for time to be spent at each stage:

– audit preparation: 30 %
– conducting the audit: 40 %
– reporting of findings: 20 %
– audit follow-up: 10 %

The auditor should prepare for the audit by gathering information, procedures/ instructions etc., and data and preparing checklists when appropriate.

The auditor should highlight and explain how the possibility of a change to the scope of the audit may occur during the audit, if serious observations/failings are discovered.

3) Conducting the audit

The auditor is to conduct the audit within groups of consecutive days during the set audit period, taking due cognisance of possible disruption to site/plant/project personnel.

The contact nominee should be periodically briefed during the audit of the findings identified, thereby avoiding surprises at the end of the audit.

The auditor should try to involve plant personnel in the audit process in order to impart learning and understanding (of the process and findings) to achieve ownership.

The style of the auditor is crucial to the success of the audit – he should try to be helpful, constructive, courteous, focused and objective.

As a minimum the auditor should try to achieve the agreed scope and timetable - variations will need to be negotiated.

4) Reporting the findings

The auditor should hold a closing meeting either at the end of the audit or later, but before the final report is issued.

The appropriate management should be given the opportunity to comment on the draft report and findings and discuss these at a formal close out meeting if desired.

It is normal practice to request a plan of action from the site/plant/project to address the findings of the report.

5) Audit follow-up

Audit reports normally require a response in the form of an action plan. The auditor might verify satisfactory completion of the action at the due date or at the next audit, whichever is appropriate.

Site/plant/project tracking systems may be used to check the implementation of action plans.

A periodic review/summary of audit findings of each audit group should be considered and its results widely communicated.

The findings/outcome from audits may be used to review the frequency of audits and are input to the management review of safety instrumented systems.

5.2.6.2.2 This subclause reinforces the role that management of change plays in the auditing process.

5.2.7 SIS configuration management

5.2.7.1 Requirements

5.2.7.1.1 To manage and maintain traceability of devices through the lifecycle, a mechanism to identify, control and track the model/versions of each device may be established.

At the earliest possible stage of the safety lifecycle, a unique plant identification should be given to each device. In some cases, earlier models/versions still in use may also be maintained and controlled. This is the first step in the configuration management program which should incorporate the following considerations.

The configuration management system may include:

a) the provision of a procedure for identification of all devices during all phases of the lifecycle;

b) the unique identification, of the model/version and build status of each device including software, including the supplier, date and where applicable, change from the model/ version originally specified;

c) the identification and tracking of all actions and changes resulting from fault observations and audits;

d) control of the issue of a release into service, identifying the status and model/version of the associated devices;

e) safeguards that have been established to assure that unauthorised alterations/ modifications are not made to the SIS while in operation;

f) the identification of the versions of each software item which together constitute a specific version of a complete device;

g) the provision of co-ordination for the updating of multiple SIS in one or more plants;

h) documented authorisation of release into service;

i) an authorised list of signatures for device release into service;

j) the stage/phase devices are brought under configuration control;

k) control of the associated deliverable documentation;

l) identification of the each model/version of a device;

- functional specification;
- technical specification;

m) all departments/organizations involved in the management and maintenance of SIS are identified and responsibilities assigned and understood.

6 Safety lifecycle requirements

6.1 Objectives

The functional safety achieved in any process facility is dependent on a number of activities being carried out in a satisfactory manner. The purpose of adopting a systematic safety lifecycle approach towards a safety instrumented system is to ensure that all the activities necessary to achieve functional safety are carried out and that it can be demonstrated to others that they have been carried out in an appropriate order. IEC 61511-1 sets out a typical lifecycle in Figure 8 and Table 2. Requirements for each lifecycle phase are given in Clauses 8 through 16 of IEC 61511-1.

The standard recognizes that the specified activities might be structured in different ways, provided that all the requirements are complied with. This restructuring can be beneficial if it allows safety activities to be better integrated into normal project procedures. The purpose of Clause 6 of IEC 61511-1 is to ensure that if a different safety lifecycle is used, the inputs and output of each phase of the lifecycle are defined and all essential requirements are incorporated.

6.2 Requirements

6.2.1 The key consideration is to define in advance the safety lifecycle of the SIS that is going to be used. Experience has shown that problems are likely to occur, unless this activity is planned well in advance and agreements are reached with all persons, departments and organizations taking responsibility. At best, some work will be delayed or have to be redone; at worst, safety can be compromised.

6.2.2 Although it is not a requirement, it is generally beneficial at an early stage to map the proposed safety lifecycle of the SIS on to the project lifecycle of the process including which of the boxes in IEC 61511-1 Figure 8 apply to the project. When doing this, the information needed to begin a safety lifecycle activity should be considered together with who is likely to be able to provide it. In some cases it may not be possible to determine accurate information on a particular issue until late in the design phase. In such cases, it may be necessary to make an estimate based on previous experience and then confirm the data at a later date. Where this is the case, it is important to note this on the safety lifecycle.

6.2.3 Another important part of safety lifecycle planning is to identify the techniques that will be used during each phase. The identification of such techniques is important since it is often necessary to use a specific technique that requires persons or departments with unique skills and experiences. For instance, consequences in a particular application may be dependent on the maximum pressure developed after a failure event; and the only way this can be determined is to develop a dynamic model of the process. The information requirements for dynamic modelling will then have an important impact on the design process.

7 Verification

7.1 Objective

The purpose of verification is to ensure that the activities for each safety lifecycle phase, as determined by verification planning, have, in fact, been carried out and that the required outputs of the phase, whether they be in the form of documentation, hardware or software, have been produced and are suitable for their purpose.

7.1.1 Requirements

7.1.1.1 IEC 61511-1 recognizes that organizations will have their own procedures for verification and do not always require them to be carried out in the same way. Instead, the intent of this subclause is that all verification activities are planned in advance, along with any procedures, measures and techniques that are to be used.

7.1.1.2 No further guidance provided.

7.1.1.3 It is important that the results of verification are available so that it can be demonstrated that effective verification has taken place at all phases of the safety lifecycle.

8 Process hazard and risk assessment

8.1 Objectives

The overall objective here is to establish the need for safety functions (for example, protection layers) together with associated levels of performance (risk reduction) that are needed to ensure a safe process. It is normal in the process sector to have multiple safety layers so that failure of a single layer will not lead to or allow a harmful consequence. Typical safety layers are represented in Figure 9 of IEC 61511-1.

8.2 Requirements

8.2.1 The requirements for hazard and risk assessment are specified only in terms of the results of the task. This means that an organization may use any technique that it considers to be effective, provided it results in a clear description of safety functions and associated levels of performance.

A hazard and risk assessment should identify and address the hazards and hazardous events that could occur under all reasonably foreseeable circumstances (including fault conditions and reasonably foreseeable misuse).

On a typical project in the process sector, a preliminary hazard and risk assessment needs to be carried out early during the basic process design. An assumption at this stage is that hazards have been eliminated or reduced as far as is reasonably practicable, by the application of inherent safety principles and the application of good engineering practice (this activity of hazard reduction is not within the scope of IEC 61511). For the SIS, this preliminary hazard and risk assessment is important because establishing, designing and implementing an SIS are complex tasks and can take a considerable length of time. Another reason for undertaking this work early is that information on system architecture will be needed before the process and instrumentation diagrams are finalized. There will usually be sufficient information enabling preliminary hazard and risk assessment to proceed once a process flow diagram has been completed and all of the initial process data is available. It should be recognised that additional hazards may be introduced as detailed design proceeds. A final hazard and risk assessment may therefore be necessary once the process and instrumentation diagram has been finalized. This final analysis generally uses a formal and fully documented procedure such as hazard and operability study (HAZOP). It should confirm that the safety layers as designed are adequate to ensure the safety of the plant. During this final analysis it is necessary to consider whether failures in the safety systems introduce any new hazards or demands. If any new hazards are established at this stage, it may be necessary to define new safety functions. Another more likely outcome is that additional events are identified that lead to the hazards that were already identified at the preliminary stage. It will then be necessary to consider if any revision of the safety functions and performance requirements that were determined in the original analysis is needed.

The approach used to identify hazards will depend on the application being considered. For certain simple processes where there is extensive operating experience of a standard design, such as simple off-shore wellhead towers, it may be sufficient to use industry developed check lists (for example, the safety analysis checklists in ISO 10418 and API RP 14C). Where the design is more complex or a new process is being considered, a more structured approach may be necessary (for example, IEC 60300-3-9:1995).

NOTE Further information on selection of appropriate techniques is given in ISO 17776.

When considering the consequences of a particular failure event, all possible outcomes, and the frequency of the failure event as it contributes to each outcome, should be analysed. No credible outcome should be ignored or discarded from a risk analysis. Exposing piping or vessels to pressures above design will not always result in catastrophic loss of containment. In many cases, equipment will have been subjected to test pressure greater than design and the only consequence may be leakage of flammable substances leading to the possibility of fire. In evaluating consequences, persons responsible for the mechanical integrity of the plant will need to be consulted. They will need to take into account the original test pressure but also whether the original design included corrosion allowances and whether a corrosion management programme is in place. Where consequences are based on such assumptions, it is important that this is clearly stated so that relevant procedures can be incorporated into the safety management system. A further issue when considering consequences will be the number of persons likely to be effected by a particular hazard. In many cases, operational and maintenance staff will only be present in the hazardous zone on an infrequent basis and this should be taken into account when predicting consequences. Care is needed when using this statistical approach since it will not be valid in all cases, such as where the hazard only occurs during start-up and staff are always present. Also considerations should be given to the potential increased number of people being in the vicinity of the hazardous event as a result of investigating the symptoms during the build-up to the event.

When assessing the potential sources of demand on the SIS, the assessment should include the following situations: start-up, continuous operation, shutdown, maintenance errors, manual interventions (for example, controllers on manual) loss of services (for example, air, cooling water, nitrogen, power, steam, trace heating, etc.).

When considering the frequency of demands, it may be necessary in some complex cases to undertake a fault tree analysis. This is often necessary where severe consequences only result from simultaneous failure of more than one event (for example, where relief headers are not designed for worst case relief from all sources). Judgement will need to be made on when operator errors are to be included in the list of events that can cause the hazard and the frequency to be used for such events. Operator error could often be excluded if the action is subject to permit procedures or lock-off facilities are provided to prevent inadvertent action. Care is also needed where credit is taken for reduction in demand frequency due to operator action. The credit that can be taken will need to be limited by human factor issues such as how quickly action needs to be taken and the complexity of the tasks involved. Where an operator, as a result of an alarm, takes action and the risk reduction claimed is greater than a factor of 10, then the overall system will need to be designed according to IEC 61511-1. The system that undertakes the safety function would then comprise the sensor detecting the hazardous condition, the alarm presentation, the human response and the equipment used by the operator to terminate any hazard. It should be noted that a risk reduction of up to a factor of 10 might be claimed without the need to comply with IEC 61511. Where such claims are made, the human factor issues will need to be carefully considered. Any claims for risk reduction from an alarm should be supported by a documented description of the necessary response for the alarm and that there is sufficient time for the operator to take the corrective action and assurance that the operator will be trained to take the preventive actions.

An alarm system can be used as a method of risk reduction by reducing the demand rate on the SIS providing:

- the sensor used for the alarm system is not used for control purposes where loss of control would lead to a demand on the SIF;
- the sensor used for the alarm system is not used as part of the SIS;
- limitations have been taken into account with respect to risk reduction that can be claimed for the BPCS and common cause issues.

Examples of techniques that can be used to establish the SIL of safety instrumented systems are given in IEC 61511-3 which also contains guidance on what to consider when selecting the method to use for a specific application.

When establishing whether risk reduction is required it is necessary to have some process safety and environmental targets. These may be specific to the particular site or operating company and will be compared with the level of risk without additional safety functions. After establishing the need for risk reduction, it will be necessary to consider what functions are required to be carried out to return the process to a safe state. In theory, the functions may be described in general terms without a reference to a particular technology. In the case of over-pressure protection for instance, the function may be described as prevention of pressure rise above a specified value. Either a relief valve or a safety instrumented system could then carry out this function. If the function is described as above, the selection of the type of technology to use would be decided in the next lifecycle step (allocation of safety instrumented functions to protection layers). In practice, the functional requirements would be different depending on the type of system selected; and this stage, and the next, may in some cases be combined.

In summary, the hazard and risk analysis should consider the following:

- each determined hazardous event and the event sequences that contribute to it;
- the consequences and likelihood of the event sequences with which each hazardous event is associated; these may be expressed quantitatively or qualitatively;
- the necessary risk reduction for each hazardous event;
- the measures taken to reduce or remove hazards and risks;

- the assumptions made during the analysis of the risks, including the estimated demand rates and equipment failure rates; any credit taken for operational constraints or human intervention should be detailed;
- references to key information which relates to the safety-related systems at each SIS lifecycle phase (for example verification and validation activities).

The information and results which constitute the hazard and risk analysis should be documented.

It may be necessary for the hazard and risk assessment to be repeated at different stages in the overall SIS safety lifecycle, as decisions are taken and available information becomes more refined.

8.2.2 In the process industry, an important cause of demands that will need to be considered in many applications is the BPCS failure. It should be noted that failure of the BPCS may be caused by the sensor, valve or control system.

Sometimes, control systems used in the process industry have redundant processors but sensors and valves are usually non-redundant. When assigning a failure rate to the BPCS, there is an important limitation that needs to be recognised. IEC 61511-1 limits the dangerous failure rate, in relation to a particular hazard, that can be claimed to 10^{-5} per hour unless the system is implemented according to the requirements of this standard. The reason for the limit is that if a lower dangerous failure rate is claimed, it would be in the range of failure rates within Table 4 of IEC 61511-1. The limit ensures that high levels of confidence are not placed on systems that do not meet the requirements of IEC 61511-1.

8.2.3 No further guidance provided.

9 Allocation of safety functions to protection layers

9.1 Objective

In order to determine the need for a SIS and its associated SIL, it is important to consider what other protection layers exist (or need to exist) and how much protection they provide. After considering the other protection layers, a determination should then be made on the need for a SIS protection layer. If a SIS protection layer is needed, a determination should then be made on the SIL for the safety instrumented function(s) of this SIS.

9.2 Requirements of the allocation process

9.2.1 The requirement here is to agree on the safety layers to be used and to allocate performance targets for the safety instrumented functions. In practice, safety functions are in many cases only allocated to safety instrumented systems where there are problems in using inherently safe designs or other technology systems.

Examples of such problems include limitations on flare capacity or protection against exothermic reactions. Any decision to use instrument based systems rather than more traditional approaches such as relief valves will need to be supported by sound reasons that will stand up to regulatory authority challenge.

As stated above, the hazard and risk assessment and allocation may be concurrent activities or allocation may in some circumstances take place prior to hazard and risk assessment. Decisions on the allocation of safety functions to safety layers are often taken on the basis of what has been found to be practicable by the user organization. Established industry good practice should also be taken into account. Decisions will then be taken on the safety instrumented systems, assuming credit for the other safety layers. For example, where relief valves have been installed and these have been designed and installed according to industry codes, it may then be decided that these are adequate on their own to achieve adequate risk reduction. Safety instrumented systems would then only limit pressure where size or performance of the relief valve(s) was insufficient for the application or release to the atmosphere is to be prevented.

9.2.2 No further guidance provided.

9.2.3 When a safety function is allocated to a safety instrumented function, it will be necessary to consider whether the application is in demand or in continuous mode. The majority of applications in the process sector operate in demand mode where demands are infrequent. In such cases, Table 3 in IEC 61511-1 is the appropriate measure to use. There are some applications where demands are frequent (for example, greater than one per year) and it is more appropriate to consider the application as continuous mode because the probability of dangerous failure will be primarily determined by the failure rate of the SIS. In such cases, Table 4 in IEC 61511-1 is the appropriate measure to apply. Continuous mode applications where failure would result in an immediate hazard are rare. Burner or turbine speed control may be continuous mode applications if protection systems are insufficient for all failure modes of the control system.

Table 3 of IEC 61511-1defines SIL in terms of PFD_{avg}. The target PFD_{avg} will be determined by the required risk reduction. The required risk reduction can be determined by comparing the process risk without the SIS with the tolerable risk. This can be determined on a quantitative or qualitative basis using the techniques in IEC 61511-3.

Table 4 of IEC 61511-1defines SIL in terms of the target frequency of dangerous failures to perform the SIF. This will be determined by the tolerable failure rate of the SIS, taking into account the consequence of failure in a particular application. When Table 4 of IEC 61511-1 is used to determine the required SIL, the target is based on the frequency of dangerous failure for the safety instrumented system. In using Table 4 of IEC 61511-1, it is incorrect to convert the frequency of dangerous failure into a probability of dangerous failure on demand using the proof test interval or the demand rate. While the units may appear to be correct, this results in an inappropriate conversion of Table 4 of IEC 61511-1 and may result in under-specification of the safety function SIL requirements.

The targets for average probability of failure on demand or frequency of dangerous failures per hour apply to the safety instrumented function, not to individual components or subsystems. A component or subsystem (for example, sensor, logic solver, final element) cannot have a SIL assigned to it outside its use in a specific SIF. However, it can have an independent maximum SIL capability claim.

The outcome of the hazard and risk assessment and allocation process should be a clear description of the functions to be carried out by the safety systems, including potential safety instrumented systems together with safety integrity level requirements (along with mode of operation, continuous or demand) for any safety instrumented function. This forms the basis for the SIS safety requirements specification. The description of the functions should be clear as to what needs to be done to ensure that safety is maintained.

At this stage of the implementation, it is unnecessary to specify architectural details for sensors and valves. Decisions on architectures are complex and whether a particular system requires 2oo3 sensors and 1oo2 valves will depend on many factors.

9.2.4 The implications of Tables 3 and 4 of IEC 61511-1 need to be fully understood. In particular, the PFD_{avg} that can be claimed for a single safety instrumented function is limited to 10^{-5}, corresponding to a risk reduction of 10^5 (SIL 4). Reliability analysis may indicate that it is possible to achieve a PFD_{avg} due to random hardware failures of less than 10^{-5}, but IEC 61511-1 presumes that systematic failures and common mode failures will limit the actual performance that can be achieved. It is strongly recommended that where risk analysis shows such a high risk reduction to be necessary, the difficulty of achieving a SIL 4 safety instrumented function in the process sector should be noted. Consideration should be given to using multiple independent SISs, of lower integrity.

With reference to Note 4:

Multiple SISs may be utilized in order to achieve higher levels of risk reduction (for example, greater than 10^3). When using multiple SISs to achieve higher risk reduction, it is important that each of the SISs is independently able to carry out the safety function and that there is sufficient independence between the SISs. For example, it might not be advisable to combine a SIL 2 pressure sensing loop with a SIL 1 level sensing loop to achieve an over pressure safety function having a risk reduction requirement of 10^3 because by the time the level sensor detected a high level, the vessel might have already exceeded its pressure constraints.

Furthermore, where multiple SISs are used, one should take into account common cause failures. In addition, all of the other requirements defined in IEC 61511-1 should be satisfied, including the minimum fault tolerance requirements defined in Table 5.

To illustrate how combining multiple SISs might be used to achieve higher levels of risk reduction, consider the following example:

A 2oo3 transmitter set, a 2oo3 logic solver and a 1oo2 final element set which yields a SIS with a PFD_{avg} of $3{,}05 \times 10^{-4}$. This SIS achieves a risk reduction of approx. $3{,}3 \times 10^3$.

It would be incorrect to assume that using two such systems together would result in a risk reduction of 10×10^6 ($3{,}3 \times 10^3 \times 3{,}3 \times 10^3$). Common cause factors, such as using similar technologies, designing both systems from the same functional specification, human factors (for example, programming, installation, maintenance), external factors (for example, corrosion, plugging, freezing of air lines, lightning) will limit the system improvement. It would also be necessary to take into account any components shared between the two systems.

A more feasible solution may be to utilize a non-redundant second system using components as diverse as possible (in order to minimize potential common cause problems).

For example consider a SIS comprising a single switch, relay logic and a single final element which yields a system with a PFD_{avg} of $7{,}7 \times 10^{-3}$. This system achieves a risk reduction of approx. $1{,}3 \times 10^2$.

Combining the software based SIS with the simplex relay SIS results in an overall theoretical risk reduction of $4{,}3 \times 10^5$ ($3{,}3 \times 10^3 \times 1{,}3 \times 10^2$). While combining the performance as shown above appears to be theoretically possible (since either SIS could shut the process unit down), once again, common cause factors have to be taken into account, and the achieved risk reduction will be somewhat less due to these factors.

9.3 Additional requirements for safety integrity level 4

9.3.1 No further guidance provided.

9.3.2 No further guidance provided.

9.4 Requirement on the basic process control system as a layer of protection

9.4.1 The basic process control system may be identified as a protection layer subject to certain conditions. If functions are implemented in the BPCS for the purpose of reducing the process risk, the BPCS can be allocated a risk reduction for the identified risks it is intended to reduce.

9.4.2 Risk reduction of less than 10 may be claimed from instrumented systems without the need to comply with IEC 61511-1. This allows the BPCS to be used for some risk reduction without the need to implement such systems to the requirements of IEC 61511-1. Any claim made should be justified by consideration of the integrity of the BPCS (determined by reliability analysis or performance data) and the procedures used for configuration, modification and operation and maintenance. When allocating risk reduction to functions in the BPCS, it is important to ensure that access security and change management are provided. The risk reduction that can be claimed for a BPCS function is also determined by the degree of independence between the BPCS function and the initiating cause. Figure 2 illustrates independence of the BPCS function and the initiating cause.

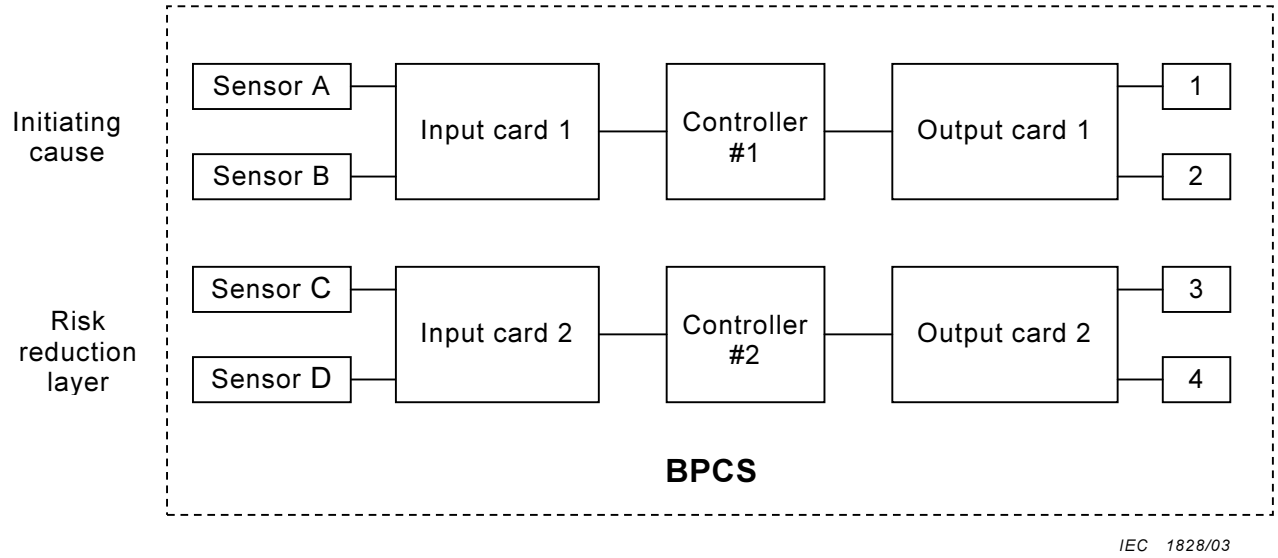


Figure 2 – BPCS function and initiating cause independence illustration

For example, consider the case where a flow control loop is the initiating cause. This initiating cause includes a flow transmitter, a controller, and a control valve. In order to allocate risk reduction to a pressure control loop in the BPCS, the pressure transmitter should be wired to an independent controller, modulating an independent final element (for example, vent valve to flare system).

9.4.3 No further guidance provided.

9.5 Requirements for preventing common cause, common mode and dependent failures

9.5.1 An important issue to be considered at an early stage is whether there are any common cause failures between redundant parts within each layer (for example, between 2 pressure relief valves on the same vessel), between safety layers or between safety layers and the BPCS. An example of this could be where failure of a basic process control system measurement could cause a demand on the safety instrumented system and a device with the same characteristics is used within the safety instrumented system. In such cases it will be necessary to establish if there are credible failure modes that could cause failure of both devices at the same time. Where a common cause of failure is identified then the following actions can be taken.

a) The common cause can be reduced by changing the design of the safety instrumented system or the basic process control system. Diversity of design and physical separation are two effective methods of reducing the likelihood of common cause failures. This is usually the preferred approach.

b) The likelihood of the common cause event should be taken into account when determining whether the overall risk reduction is adequate. This may require a fault tree analysis to be constructed that includes demand causes as well as protection system failures. Common cause failures can be represented on such fault trees and their effect on overall risk can be quantified through appropriate modelling methods.

It should be noted that any sensors or actuators which are shared by the BPCS and SIS are very likely to introduce common cause failures and that the approach to such sharing of devices should be as discussed in this **subclause**.

9.5.2 The considerations listed below apply when an assessment is carried out on the likelihood of common cause, common mode and dependent failures. The extent, formality and depth of the assessment will depend on the safety integrity level of the intended function. The effect of common cause, common mode and dependent failures may be dominant for safety integrity levels of 3 or higher. The following should be considered:

- independence between protection layers – a failure mode effects analysis should be carried out to establish if a single event can cause failure of more than one protection layer or failure of the BPCS and a protection layer. The depth and rigor of the analysis will depend on the risk.
- diversity between protection layers - the aim should be diversity between protection layers and the BPCS but this is not always achievable. An example could be over pressure protection where a failure of the BPCS pressure control loop would cause a demand. The BPCS and the SIS will both require pressure measurement and there will be a limit on the suitable equipment available. Some diversity can be achieved by using equipment from different manufacturers but if SIS and BPCS sensors are connected to the process using the same type of hook up, then the diversity may be of limited value.
- physical separation between different protection layers – physical separation will reduce the impact of common cause failures due to physical causes. Measurement connection locations for BPCS and SIS should be given maximum physical separation subject to functional needs such as accuracy and response time.

10 SIS safety requirements specification

10.1 Objective

The development of the SIS safety requirements specification is one of the more important activities of the whole safety lifecycle. It is through this specification that the user is able to define how he wants the Safety Instrumented Functions (SIF) to be designed and integrated into a SIS.

Final validation of the SIS is carried out using this specification.

10.2 General requirements

10.2.1 The SIS safety requirements specification may be a single document or a collection of several documents including procedures, drawings or corporate standard practices. These requirements may be developed by the Hazard and Risk Assessment team and/or the project team itself.

10.3 SIS safety requirements

10.3.1 As described in IEC 61511-1, there are a number of design requirements that need to be defined early in a project to ensure the Safety Instrumented Functions provide the desired protection.

Safety requirements specifications for individual subsystems may also be derived from this overall specification.

Some considerations with respect to the safety requirements specifications are as follows:

a) The first items that will need to be defined is the safety instrumented function along with its Safety Integrity Level (SIL). An example of a Safety Instrumented Function is “protect the reactor from overpressure by shutting down the inlet valves on high pressure”. Typically the function description will comprise the following elements.

- Which measurements need to be taken to detect the onset of the hazardous conditions. A simple example could be that a pressure rise above a specified value needs to be detected. The value of the parameter at which action should be taken will need to be outside the normal operating range and less than the value that will result in the hazardous condition. An allowance will need to be made for the response of the system and the accuracy of measurement. In setting the limit, there will therefore need to be a discussion with those responsible for the safety instrumentation system design and implementation.
- The actions that need to be taken that will prevent the hazardous condition. A simple example could be to reduce the flow of steam to a reboiler within a specified time. It should be noted that it is not usually sufficient to state that steam flow to the reboiler should be shut-off. The designer will need to know what is necessary for successful operation. In heating duties it may for example be sufficient to reduce flow to less than 10 % of flow within one minute. In other examples it may be necessary to have tight shut-off within a few seconds.
- The actions not needed to prevent the hazardous condition that may be of benefit for operational reasons. Such actions may include presentation of alarms, shut down of upstream or downstream units to reduce demands on other protection systems or actions that will enable fast start up once the cause of the hazard has been eliminated. It is important to separate these actions from the actions necessary to prevent the hazardous condition so as to minimize costs and restrict the boundary of the safety instrumented system to what is necessary. The wider the boundary is set, the more difficult it will be to show that the overall probability of failure on demand meets the requirements associated with the specified integrity level.
- Any identified process states or sequences of the SIS operation which should be prevented because they will result in hazardous situations.

b) This specification should define the safe state of the process for each identified function in terms of which flows should be started or stopped, which process valves should be opened or closed and the state of operation of any rotating equipment (pumps, compressors, agitators). If bringing the process to a safe state involves sequencing, the sequencing should also be identified.

NOTE In defining the final elements, consideration should be given to the benefits of diversity, for example, shutting off the product stream and shutting off the steam flow to reduce high pressure.

c) The requirement for a desired proof test interval should be defined at the beginning so the design of the SIS can take it into consideration. For example, if proof testing is to be performed during planned shutdowns (for example, every 3 years), the design might require more redundancy than if the proof test interval is to be annual.

d) Requirements for being able to manually bring the process to a safe state should be defined. For example, if there is a requirement for the operator to be able to manually shutdown a piece of equipment from either the control room or from a field location, then this needs to be specified. Any requirement for independence of manual shutdown switches from the SIS logic solver also needs to be defined.

e) All requirements for restarting the process after a shutdown need to be specified. For example, some users have electronic reset switches on the main control panel or in the field and others like to use solenoids with latching handles. If there is a specific requirement like this reset action, it should be part of the safety requirements specification.

f) If there is a target frequency for nuisance trips, this also should be specified as part of the safety requirements specification. This will be a factor in the design of the SIS.

g) The interfaces between the SIS and the operator need to be fully described, including alarms (pre-shutdown alarms, shutdown alarms, bypass alarms, diagnostic alarms), graphics, sequence of events recording.

h) There may be a need for bypasses to allow the SIS to be tested or maintained while the process is running. If there are specific requirements for bypassing such devices as key lock or passwords, these also need to be specified as part of the safety requirements specification.

i) The failure modes and response of the SIS on the detection of faults should be defined. For example, a transmitter can be designed to fail toward a trip condition or away from a trip condition. If it is designed to fail away from the trip condition, then it is important that the operator gets an alarm on the transmitter failure and is trained to take the necessary corrective action to get the transmitter repaired as quickly as possible. See also IEC 61511-1, 11.3 relating to requirements on detection of a fault.

10.3.2 No further guidance provided.

11 SIS design and engineering

11.1 Objective

The objective of this subclause is to provide guidance in the design of the SIS. Each SIF has its own SIL. A component of a SIS, for example, a logic solver, may be used by several SIFs with different SILs.

11.2 General requirements

11.2.1 No further guidance provided.

11.2.2 No further guidance provided.

11.2.3 No further guidance provided.

11.2.4 IEC 61511–1, Clause 11, has a number of design requirements for a SIS. One item of concern is independence between the SIS and the BPCS.

A SIS is normally separated from the BPCS for the following reasons:

a) To reduce the effects of the BPCS on the SIS, especially when they share common equipment. For example if the BPCS and SIS share a common valve for shutdown and control, then in the event of a dangerous failure of that valve, it would not be available to perform a SIS shutdown function.

b) To retain flexibility for changes, maintenance , testing and documentation relating to the BPCS.

NOTE 1 The SIS normally has more robust requirements than the BPCS and the intent is not to subject the BPCS to the same robust requirements that are required for the SIS. However it should be noted that uncontrolled BPCS modifications can be a cause of increased demand on the SIS.

c) To facilitate the validation and functional safety assessment of the SIS.

d) Access to the programming or configuration functions of the BPCS may need to be limited to meet the modification management arrangements if the BPCS is combined with the SIS.

Where a failure of the common equipment can cause a demand on the SIS, then an analysis should be conducted to ensure the overall hazard rates satisfies the expectations. The overall hazard rate will be the sum of the dangerous failure rate of the common elements and the hazard rate from other sources of demand (including dangerous failure of the independent parts of the SIS).

Separation between the SIS and the BPCS may use identical or diverse separation. Identical separation would mean using the same technology for both the BPCS and SIS whereas diverse separation would mean using different technologies from the same or different manufacturer.

Compared with identical separation, which helps against random failures, diverse separation offers the additional benefit of reducing the probability of systematic faults and of reducing common cause failures.

Identical separation between the SIS and BPCS may have some advantages in design and maintenance because it reduces the likelihood of maintenance errors. This is particularly the case if diverse components are to be selected which have not been used before within the user's organisation.

Identical separation between SIS and BPCS may be acceptable for SIL 1, SIL 2 and SIL 3 applications although the sources and effects of common cause failures should be considered and their likelihood reduced. Some examples of common cause failures are:

a) plugging of instrument connections and impulse lead lines;

b) corrosion and erosion;

c) hardware faults due to environmental causes;

d) software errors;

e) power supplies and power sources;

f) human errors.

Diverse separation offers the additional benefit of reducing the probability of systematic failures (a factor especially important in SIL 3 and SIL 4 applications) and reducing common cause failures.

There are four areas where separation between the SIS and BPCS is generally provided:

1) field sensors;

2) final elements;

3) logic solver;

4) wiring.

Physical separation between BPCS and SIS may not be necessary provided independence is maintained, and the equipment arrangements and the procedures applied ensure the SIS will not be dangerously affected by

- failures of the BPCS;
- work carried out on the BPCS for example, maintenance, operation or modification.

Where procedures are necessary to ensure the SIS is not dangerously affected, the SIS designer will then need to specify the procedures to be applied.

a) Field sensors

Using a single sensor for both the BPCS and SIS requires further review and analysis. The additional review and analysis is necessary because a failure of this single sensor could result in a hazardous situation. For example, a single level sensor used for both the BPCS and a SIS high level trip could create a demand if the sensor fails low (i.e., below the set point of the level controller). As a result of the sensor failing low, the controller would drive the valve open. Since the same sensor is used for the SIS, then it will not detect the resultant high level condition.

Where a single sensor is used for both a BPCS and SIS function, the requirements of IEC 61511-1 will normally only be satisfied if the sensor diagnostics can reduce the dangerous failure rate sufficiently and the SIS is capable of placing the process in a safe state within the required time. In practice this is difficult to achieve even for SIL 1 applications. For a SIL 2, SIL 3 or SIL 4 safety instrumented function, separate SIS sensors with identical or diverse redundancy will normally be needed to meet the required safety integrity.

NOTE 2 When a single separate SIS sensor is used, there may be advantages to repeating the signal to the BPCS through suitable isolators. Such an arrangement can lead to improved diagnostic coverage by allowing signal comparison between BPCS and SIS sensors.

When redundant SIS sensors are used, the sensors may also be connected to the BPCS through suitable isolators. Suitable algorithms in the BPCS such as “middle of three” may increase safety by reducing the demand rate on the SIS.

b) Final Element

In the same way as for the sensors, using a single valve for both the BPCS and SIS requires further review and analysis. In general, a single valve used for both the SIS and BPCS in not recommended if a failure of the valve would place a demand on the SIS.

Where a single valve is used by both the BPCS and SIS, the requirements of IEC 61511-1 will normally only be satisfied if the valve diagnostics can reduce the dangerous failure rate sufficiently and the SIS is capable of placing the process in a safe state within the required time.

In practice, this is difficult to achieve even for SIL 1 applications. For a SIL 2, SIL 3 or SIL 4 safety instrumented function, separate SIS valves with identical or diverse redundancy will normally be needed to meet the required safety integrity.

Where a single valve is used for both BPCS and SIS functions, the design should ensure that the SIS action overrides the BPCS action. This is normally achieved by having the SIS directly connected to a solenoid valve that removes the power source directly at the actuator, for example, between the valve positioner and the actuator.

When redundant SIS valves are used, the valves may be connected to both the SIS and BPCS.

NOTE 3 Even with redundant valves, it is important to consider common cause failures between the BPCS and SIS valves.

Additional considerations for determining the valve requirements are:

– shutoff requirements;
– reliability experience with the valve in similar process applications;
– unsafe failure modes of the valve;
– operating procedures that make the valve less effective (for example, bypass valves being opened);
– proof testing requirements.

c) Wiring

On energize to trip systems, the BPCS and relevant field device wiring is normally separated from wiring to the SIS and its relevant field devices because of the possibility of accidentally deactivating the safety function without noticing it. Typical guidelines for these types of systems include installing separate multi-conductor cables and junction boxes dedicated to the SIS and BPCS. Where the wiring is not separated, the use of good labelling and maintenance procedures to minimize the potential of errors caused during maintenance resulting in deactivation of the SIS are suggested.

NOTE Energize to trip refers to SIF circuits where the outputs and devices are de-energized under normal operation. Application of power (for example, electricity, air) causes a trip action.

The cable support system (for example, cable trays, conduit), may be common for both de-energize to trip and energize to trip systems, unless separation is required for other reasons (for example, electromagnetic interference). On energize to trip systems, consideration may be given to adding fire protection to the cable trays in fire risk areas.

11.2.5 No further guidance provided.

11.2.6 See 11.8 of this standard for guidance as well as following guidance relating to the Note in 11.2.5 of IEC 61511-1.

The operators, maintenance staff, supervisors and managers all have roles in safe plant operation. However, humans can make errors or be unable to perform a task, just as instruments and equipment are subject to malfunction or failure.

Human performance is therefore a system design element. The human machine interface (HMI) is particularly important in communicating the status of the SIS to operating and maintenance personnel.

Human Reliability Analysis (HRA) identifies conditions that cause people to err and provides estimates of error rates based on past statistics and behavioural studies. Some examples of human error contributing to chemical process safety risk include:

- undetected errors in design;
- errors in operations (for example, wrong set point);
- improper maintenance (for example, replacing a valve with one having the incorrect failure action);
- errors in calibrating, testing or interpreting output from control systems;
- failure to respond properly to an emergency.

NOTE See the following references for additional guidance:

CCPS/AIChE *Guidelines for Improved Human Performance in Process Safety*, New York: American Institute of Chemical Engineers (1994).

CCPS/AIChE *Guidelines for Chemical Process Quantitative Risk Analysis* (second edition), New York: American Institute of Chemical Engineers (2000).

HSE *Reducing error and influencing behaviour*, HSG48, Health and Safety Executive, London (1999), ISBN 0 7176 2452 8.

11.2.7 This subclause addresses the potential hazard that may be created if a SIS automatically restarts the process immediately after the trip condition is corrected. Each SIF should be analysed to determine how it should be reset once the trip condition is corrected. Normally restarting should only be possible after a manual action of the operator.

11.2.8 Manual means that are independent of both the SIS logic solver and the BPCS control system may be provided to allow the operator to initiate a shutdown in an emergency. The requirements for manual shutdown are normally defined in the SRS.

The emergency stop may be connected to the SIS PE logic solver (for example, when a sequenced shut down is required) provided that it is necessary and deemed appropriate by the H and RA team.

11.2.9 This subclause indicates the need for analysis of independence between the SIS and other protection layers, not just between the SIS and BPCS (see IEC 61511-1, Figure 9).

Under some circumstances it may be acceptable that there is incomplete separation between BPCS and the SIS. This is particularly the case where a failure of the common equipment will not cause a demand on the SIS. In such cases, it is necessary to implement the common or shared equipment in accordance with IEC 61511-1.

Where a failure of the common equipment can cause a demand on the SIS, then an analysis should be conducted to ensure the overall hazard rate satisfies the expectations. The overall hazard rate will be the sum of the dangerous failure rate of the common elements and the hazard rate from other sources of demand (including dangerous failure of the independent parts of the SIS). To establish the hazards associated with dangerous failures of the common equipment, the following cases should be considered:

a) Where one element of the redundant configuration is used as a BPCS, consider the hazards arising from dangerous failures of common equipment taking into consideration the performance of the SIS which has been degraded by the failed instruments;

b) Where the shared instruments are not redundant, consider the hazards arising from dangerous failures of the common equipment assuming the SIS did not respond.

11.2.10 Provides cautionary guidelines on using a common element for both the BPCS and the SIS. “Sufficiently low” in the Note means the dangerous failure rate of the shared equipment multiplied by the PFD of the other independent layers (other than the SIF) meets your corporate risk criteria.

11.2.11 In the case of final elements which on loss of power do not fail to the safe state (for example, energize to trip systems) consideration should be given to the provision of local manual means to achieve the safe state.

11.3 Requirements for system behaviour on detection of a fault

11.3.1 No further guidance provided.

11.3.2 No further guidance provided.

11.3.3 No further guidance provided.

11.4 Requirements for hardware fault tolerance

11.4.1 The traditional approach to safety system design was to ensure that no single fault would result in loss of intended function. System architectures such as 1oo2 or 2oo3 have a fault tolerance of 1 because they are able to function on demand even in the presence of one dangerous fault. Such systems were employed as a standard approach for safety systems to ensure they were sufficiently robust to be able to withstand random hardware failures. Fault tolerance architectures also gave protection to a wide range of systematic faults (mainly in hardware) because such faults do not necessarily arise at the same instant of time.

This standard recognizes that the process industry needs more than one level of performance from safety systems and has adopted the concept of safety integrity levels with increasing performance depending on the need for risk reduction in the specific application involved. Because of the different levels of performance it is no longer appropriate to expect all safety integrity levels to be fault tolerant. In selecting the architecture to use for a specified integrity level it is however important to ensure that it is sufficiently robust for both random hardware faults and systematic faults. To ensure robustness against random hardware faults this standard requires that a reliability analysis be carried out.

The requirements of this part of the standard are targeted at ensuring that architectures have the necessary fault tolerance for random hardware faults and some systematic faults. In deciding the extent of fault tolerance needed there are a number of factors that should be taken into consideration as follows:

- The complexity of the devices used within the subsystem. A device will be less likely to be subject to systematic faults if the failure modes are well defined, the behaviour under fault conditions can be determined and there is sufficient failure data from field experience;
- The extent to which faults lead to a safe condition or can be detected by diagnostics so that a specified action can be taken. This capability is termed the safe failure fraction of the device;
- The safety integrity level requirement for the application involved.

The international working group that prepared IEC 61508 considered the above factors and specified the extent of fault tolerance required in IEC 61508-2. In preparing this sector-specific standard for the process sector it was considered that the requirements for fault tolerance of field devices and non PE logic solver could be simplified and the requirements in IEC 61511-1 could be applied as an alternative. It should be noted that subsystem designs may require more component redundancy than what is stated in Tables 5 and 6 in order to satisfy availability requirements.

The requirements for hardware fault tolerance can apply to individual components or subsystems required to perform a SIF. For example, in the case of a sensor subsystem comprising a number of redundant sensors, the fault tolerance requirement applies to the sensor subsystem in total, not to individual sensors.

11.4.2 Table 5 of IEC 61511-1 defines the minimum fault tolerance for PE logic solvers. The fault tolerance requirement depends on the required SIL of the SIS and the subsystem safe failure fraction. Information on safe failure fraction of logic solvers can normally be obtained from the PE logic solver vendor. If the PE logic solver is not used according to the assumptions made in the calculation of the SFF then the claims made for safe failure fraction should be carefully considered. In particular, the assumptions made should be examined to ensure that the boundary and environment assumed in the SFF calculations are valid for the application being considered. This is because the SFF will depend on a number of issues such as whether the subsystem is energize or de-energize to trip. Data sources and assumptions made during a calculation of SFF should be documented. The SFF is related to random hardware failures only. In establishing the SFF it is acceptable to assume that the subsystem has been properly selected for the application and is adequately installed, commissioned and maintained such that early life failures and age related failure may be excluded from the assessment. Human factors do not need to be considered when determining SFF.

11.4.3 Table 6 of IEC 61511-1 defines the basic level of fault tolerance for sensors, final elements, and non-PE logic solvers having the required SIL claim limit in the first column. The requirements in Table 6 are based on the requirements in IEC 61508-2 for PE devices with a SFF between 60 and 90 %. The requirements are based on the assumption that the dominant failure mode is to the safe state or that dangerous failures are detected.

11.4.4 This subclause allows the hardware fault tolerance of all subsystems except PE logic solvers to be reduced by one on certain conditions. These conditions will apply to devices such as valves or smart transmitters and reduce the likelihood of systematic failures such that the requirements are aligned to the requirements of IEC 61508-2 for non PE devices.

11.4.5 In some cases it may be possible to reduce the fault tolerance by following the fault tolerance requirements of IEC 61508-2. This may be achieved by introducing additional diagnostics such as signal comparison or regularly scheduled partial stroke testing such that the SFF of the subsystems is higher than 90 %.

11.5 Requirements for selection of components and subsystems

11.5.1 Objectives

No further guidance provided.

11.5.2 General requirements

11.5.2.1 There are some considerations for selecting components and sub–systems to be used in a SIS. The first option is that the components be designed in accordance with IEC 61508–2 (requirements for electrical/electronic/programmable electronic safety-related systems) and IEC 61508-3 (software requirements). The second option is to use components and sub–systems that are known to be reliable through extensive use in similar service and in a similar environment.

Whichever option is chosen, it has to be demonstrated that the component or subsystem

a) is reliable enough to achieve the overall target PFD or target dangerous failure rate of the safety instrumented function,

b) meets the architectural constraint requirement, and

c) has a sufficiently low likelihood of systematic faults.

The requirement of c) can be satisfied either by compliance with IEC 61508-2 and IEC 61508-3 or by the prior use requirements in 11.5 of this standard.

11.5.2.2 No further guidance provided.

11.5.2.3 No further guidance provided.

11.5.2.4 No further guidance provided.

11.5.3 Requirements for the selection of components and subsystems based on prior use

11.5.3.1 There are very few field devices (sensors and valves) that are designed per IEC 61508-2 and IEC 61508-3. Users and designers will therefore have to depend more heavily on using field devices that have been “proven-in-use”.

Many users have a list of instruments that are approved or recommended for use in their facility. These lists have been established by extensive successful operating experience on their BPCS. Sensors and valves that have had a history of not performing as desired have been eliminated.

Normally the sensors and valves that are on these approved or recommended lists for the BPCS could also be considered as proven-in-use for SISs subject to the assessment required by 61511-1. This list of instruments should include the version of the device and be supported by documented monitoring of field returns at the user and at the manufacturer. In addition the manufacturer should have a modification process which evaluates the impact of reported failures and modifications.

If such a list does not exist, then users and designers need to conduct an assessment on the sensors and valves to ensure that they are satisfied the instrument will perform as desired. This may require discussions with other users or designers to see what they are using for similar applications.

11.5.3.2 It should be noted that for more complex devices, it may become more difficult to show that the experience gained in an application is relevant. As an example, experience gained by using a PLC in an application involving the use of simple ladder logic may not be relevant if the equipment was to be used for complex calculations or sequences.

In general, the relevant aspects of the operating profile of field devices are different from those of a logic solver.

For field devices the following points contribute to the operating profile:

- functionality (for example, measurement, action);
- operating range;
- process properties (for example, properties of chemicals, temperature, pressure);
- process connection.

For logic solvers, the following points contribute to the operating profile:

- version and architecture of hardware;
- version and configuration of system software;
- application software;
- I/O configuration;
- response time;
- process demand rate.

For all devices, the following points contribute to the operating profile:

- EMC;
- environmental conditions.

11.5.4 Requirements for selection of FPL programmable components and subsystems (for example, field devices) based on prior use

11.5.4.1 No further guidance provided.

11.5.4.2 No further guidance provided.

11.5.4.3 No further guidance provided.

11.5.4.4 This subclause explains additional requirements when trying to qualify a FPL programmable device to a SIL 3 capability.

11.5.4.5 This subclause mandates a Safety Manual for a FPL programmable device with a SIL 3 capability.

11.5.5 Requirements for the selection of LVL programmable components and subsystems (for example, logic solvers) based on prior use

11.5.5.1 This subclause lists additional requirements for LVL PE logic solvers having SIL 1 or SIL 2 capability. LVL PE logic solver with SIL 3 or 4 capability should be in accordance with IEC 61508-2 and IEC 61508-3.

11.5.5.2 No further guidance provided.

11.5.5.3 No further guidance provided.

11.5.5.4 No further guidance provided.

11.5.5.5 This subclause lists additional requirements to achieve SIL 1 and SIL 2 capability for a safety configured PE logic solver. For additional considerations, see Annex D.

11.5.5.6 This subclause lists additional requirements to achieve SIL 2 capability for a safety configured PE logic solver.

11.5.5.7 This subclause mandates a Safety Manual for a LVL programmable device with a SIL 2 capability.

11.5.6 Requirements for the selection of FVL programmable components and subsystems (for example, logic solvers)

11.5.6.1 No further guidance provided.

11.6 Field devices

11.6.1 No further guidance provided.

11.6.2 No further guidance provided.

11.6.3 No further guidance provided.

11.6.4 No further guidance provided.

11.7 Interfaces

User interfaces to a SIS are operator interfaces and maintenance/engineering interfaces. The information or data which is communicated between the SIS and the operator displays can be either SIS related or informative.

If an operator action is part of the safety instrumented function, everything needed to perform this action should be considered as part of the SIF. This would include, for example, an alarm indicating that the operator has to shutdown the process. In this example, the shutdown switch (the means of implementing the shutdown action) should be considered as part of the SIF.

Data communication which is not part of the SIF (for example, display of the actual value of a SIF sensor if the trip function is realised within the SIF) may be displayed in the BPCS if it can be shown that the safety instrumented functions are not compromised (for example, read-only-access in the BPCS).

11.7.1 Operator interface requirements

The operator interfaces used to communicate information between the operator and the SIS may include:

- video displays;
- panels containing lamps, push buttons, and switches;
- annunciator (visual and audible);
- printers (should not be the sole method of communication);
- any combination of these.

a) video displays

BPCS video displays may share SIS and BPCS functions provided the displayed data is for information only. Safety critical information is additionally displayed via the SIS (for example, if the operator is part of the safety function).

When operator action is needed during emergency conditions, the update and refresh rates of the operator display should be carried out in accordance with the safety requirements specification.

Video displays relating to the SIS should be clearly identified as such, avoiding ambiguity or potential for operator confusion in an emergency situation.

The BPCS operator interface may be used to provide automatic event logging of safety instrumented functions and BPCS alarming functions.

Conditions to be logged might include the following:

– SIS events (such as trip and pre-trip occurrences);

– whenever the SIS is accessed for program changes;

– diagnostics (for example, discrepancies, etc).

It is important that the operator be alerted to the bypass of any portion of the SIS via an alarm and/or operating procedure. For example, bypassing the final element in a SIS (for example, shutoff valve) could be detected via limit switches on the bypass valve that turn on an alarm on the panel board or by installing seals or mechanical locks on the bypass valve that are managed via operating procedures. It is generally suggested to keep these bypass alarms separate from the BPCS.

b) panels

Panels should be located to give operators easy access.

Panels should be arranged to ensure that the layout of the push buttons, lamps, gauges, and other information is not confusing to the operator. Shutdown switches for different process units or equipment, which look the same and are grouped together, may result in the wrong equipment being shut down by an operator under stress in an emergency situation. The shutdown switches should be physically separated and their function labelled. Means should be provided to test all lamps.

c) printers and logging

Printers connected to the SIS should not compromise the safety instrumented function if the printer fails, is turned off, is disconnected, runs out of paper or behaves abnormally.

Printers are useful to record the sequence in which events occur, diagnostics and other events and alarms, with time and date stamping and identification by tag number. Report formatting utilities should be provided.

If printing is a buffered function (information is stored, then printed on demand or on a timed schedule), then the buffer should be sized so that information is not lost, and under no circumstances should SIS functionality be compromised due to filled buffer memory space.

11.7.1.1 The operator should be given enough information on one display to rapidly convey critical information. Display consistency is important and the methods, alarm conventions and display components used should be consistent with the BPCS displays.

Display layout is also important. Layouts with a large amount of information on one display should be avoided since they may lead to operators misreading data and taking wrong actions. Colours, flashing indicators, and judicious data spacing should be used to guide the operator to important information so as to reduce the possibility of confusion. Messages should be clear, concise and unambiguous.

The display should be designed such that data can be recognized by operators who may be colour blind. For example, conditions shown by red or green colours could also be shown by filled or unfilled graphics.

11.7.1.2 No further guidance provided.

11.7.1.3 No further guidance provided.

11.7.1.4 No further guidance provided.

11.7.1.5 No further guidance provided.

11.7.2 Maintenance/engineering interface requirements

11.7.2.1 No further guidance provided.

11.7.2.2 Maintenance/engineering interfaces consist of means to program, test and maintain the SIS. Interfaces are devices which are used for functions such as:

a) system hardware configuration;

b) application software development, documentation, and downloading to the SIS logic solver;

c) access to application software for changes, testing, and monitoring;

d) viewing SIS system resource and diagnostic information;

e) changing SIS security levels and access to application software variables.

Maintenance/engineering interfaces should be capable of displaying the operating and diagnostic status of all SIS components (for example, as input modules, processors) including the communication between them.

Maintenance/engineering should provide means for copying application programs to storage backup media.

A personal computer connected to a SIS for maintenance/engineering purposes, should not compromise safety functions if the personal computer fails, is turned off or is disconnected.

11.7.2.3 No further guidance provided.

11.7.2.4 No further guidance provided.

11.7.3 Communication interface requirements

11.7.3.1 No further guidance provided.

11.7.3.2 No further guidance provided.

11.7.3.3 No further guidance provided.

11.7.3.4 No further guidance provided.

11.8 Maintenance or testing design requirements

11.8.1 The design of the SIS should take into consideration, how the system is going to be maintained and tested. If the SIS is to be tested while the process is running, the design should not require the disconnection of wires, applying jumpers or forcing software registers since using these techniques may jeopardize the integrity of the SIS. The system design should provide technical and procedural requirements of the SIS in order to accomplish full system testing of sensors, logic solver and final elements safely.

It is important to define how a SIS is going to be maintained while the process is running. For example, if a transmitter or valve needs to be worked on, consideration needs to be given on how the maintenance department will work on these instruments without causing a nuisance trip while maintaining the safety of the process.

It should be noted that any limit on the testing period of final elements should be taken into account in the calculation of the PFD_{avg} of the SIF.

11.8.2 No further guidance provided.

11.8.3 The installation of bypasses may reduce the level of security in a SIS. This reduction in security may be overcome by:

a) Using passwords and/or key locked switches. Some designs may incorporate locked cabinets containing the appropriate bypasses.

b) Clear identification of piping bypasses may be accomplished by either sealing valve positions or installing safety signs indicating importance of the appropriate position.

For example, for a 1oo2 sensor configuration, some users like to bypass both sensors at one time but others like to have a separate bypass for each sensor. If both sensors are bypassed, it will be necessary to put measures in place to ensure that risk remains tolerable. Either can be possible, but this should be addressed early in the design.

Likewise, some process operations do not support the valve being moved while the process is running or installing a bypass around the valve may be impractical. In these cases, the design should allow for testing the SIS as far as practical, i.e., at least through the solenoid valve. In this case, some type of bypass around the solenoid can be included in the design with the usual alarming or procedural controls for this bypass.

11.8.4 No further guidance provided.

11.9 SIF probability of failure

11.9.1 Users and designers should refer to Annex A of this standard for guidance in techniques available to ensure SIS design satisfies performance relating to random hardware failures.

11.9.2 Most of the techniques in Annex A of this standard require some quantification of the diagnostic coverage of the SIS. Diagnostics are tests performed automatically to detect faults in the SIS that may result in safe or dangerous failures.

A particular diagnostic technique cannot usually detect all possible faults. An estimate of the effectiveness of the diagnostics used may be provided for the set of faults being addressed. Subclauses 7.4.4.5 and 7.4.4.6 of IEC 61508-2 provide requirements for how diagnostics could be determined (see also Annex C of IEC 61508-6 for an example of how diagnostic coverage is calculated).

Improving the diagnostic coverage of the SIS may assist in satisfying the SIL requirements. In this case, both the diagnostic coverage and the period between diagnostic tests (the diagnostic test interval) should be taken into account when calculating the probability of failure (demand mode) or frequency of failure (continuous mode) of the SIS. For further guidance, refer to IEC 61508-6, Annex B or ISA TR84.00.02.

In situations where the SIS is the only layer of protection and is used for a safety function operating in the continuous mode of operation, then the diagnostic test interval will need to be such that faults in the SIS are detected in time to ensure the integrity of the SIS and to allow action to be taken to ensure a safe state in the event of a failure occurring in the process or the basic process control system.

To achieve this, the sum of the diagnostic test interval and the reaction time to achieve a safe state should be less than the “process safety time”. The process safety time is defined as the time period between a failure occurring in the process or the basic process control system (with the potential to give rise to a hazardous event) and the occurrence of the hazardous event if the safety instrumented function is not performed.

Critical and potentially critical faults to common components (such as faults to CPU/RAM/ROM) typically inhibit nearly the entire processing of data and are therefore more far reaching than a fault of a single output point. Failure modes that carry a high failure probability have to be detected with more confidence. Furthermore, the detectability of failure modes should be taken into account.

For each diagnostic implemented, testing interval and resulting action on fault detection should meet the safety requirements specification.

Where these diagnostics are not “built in” the vendor supplied equipment, externally configured diagnostics may be implemented at the system or application level in order to meet the SIL for the SIF.

Diagnostics may not be capable of detecting systematic errors (such as software bugs). However, appropriate precautionary measures to detect possible systematic faults may be implemented.

Diagnostics may be accomplished using a variety or combination of methods, including:

a) Sensors

1) Diagnostic alarms could be provided to detect a sensor that has completely failed upscale or downscale. One way this can be accomplished is by use of an out of range alarm. For example, in a high temperature trip application with redundant temperature transmitters, a low out of range alarm could be added to diagnose a transmitter failure or loss of transmitter signal.

2) If redundant transmitters are used, comparison of the analogue values detects anomalies that may occur during normal operation. If three transmitters are used, the middle of the three readings can be used (mid-value selection). Mid value selection is advantageous over comparison to the average because the average is skewed by the device that is not functioning properly. Significant deviations between readings may be created by

- plugging or freezing in the impulse leads;
- reduction in purge supply pressure;
- process coating of thermowells;
- grounding or power supply problems;
- non-response of a transmitter that has an output value that is no longer changing.

3) Time delays may be provided to prevent nuisance alarms due to variations in sensor response to process changes caused by sensor location or sensor technology. For example, some redundant flow sensors may have 1 to 2 s delays. There are a number of software packages available from vendors to monitor redundant sensor readings and calculate the standard deviation in order to initiate the diagnostic alarms.

4) Another method of sensor diagnostics is comparison of related variables (for example, flow totalizers versus tank level changes or pressure and temperature relationship).

b) Final elements

1) Comparison of the feedback from the final element (such as limit switches or position transmitters) to the requested state may be performed to verify that the expected actions have been taken. Sufficient time delays should be used to filter the alarm for valves in transition (for example, from fully opened to fully closed). This comparison of the feedback from the final element to the requested state can only be considered to be a diagnostic if the valve periodically changes to the safe state as part of normal operation (for example, batching operation).

2) Some valves, actuators, solenoids, and/or positioners may provide diagnostic capability.

c) Logic Solvers

Safety-configured or IEC 61508 series compliant PE logic solvers typically include diagnostics which detect various faults. The types and diagnostic coverage will generally be described in the Safety Manual.

d) Externally configured diagnostics

Examples of these include watchdog timers and end-of-line monitors.

With reference to the Note in 11.9.2.c) of IEC 61511-1 regarding confidence in reliability data, mean time to failure (MTTF) is typically determined by recording the number of failures (n) which occur in a sample of components during an accumulated number of operating hours (T). A confidence level in the resulting MTTF can be derived using the 'Chi-square' test (see '*Reliability, maintainability and risk*, D J Smith' ISBN 0 7506 5168 7). This means that the value of MTTF to be used in the reliability calculations for a SIS will, in general, be lower than the value of MTTF calculated as T/n. This reduction factor will be greater for a higher required confidence level and for lower numbers of observed failures. However, in general, it is reasonable to assume that at a 70 % confidence level the reduction factor is not significant compared to other sources of uncertainty associated with reliability modelling.

12 Requirements for application software, including selection criteria for utility software

Clause 12 of IEC 61511-1 does not differentiate between SIL 3 and lower SIL application software design methods because experience shows that there is little difference in the methods when using:

- either FPLs or LVLs; and
- IEC 61511-1 compliant logic solver; and
- the corresponding Safety Manual.

There may be differences for test and verification for different SILs. See 12.7.2.3 of this part for guidance.

12.1 Application software safety lifecycle requirements

12.1.1 Objective

12.1.1.1 No further guidance provided.

12.1.2 Requirements

12.1.2.1 No further guidance provided.

12.1.2.2 Notes 1 and 2: When limited variability languages such as IEC 61131-3 ladder diagram or function block diagram are used for the design, implementation, verification and validation of application software, then only two levels of the standard software "V" model shown in Figure 3 need apply. In this case, it is assumed that the used function blocks conform to IEC 61508-3, then:

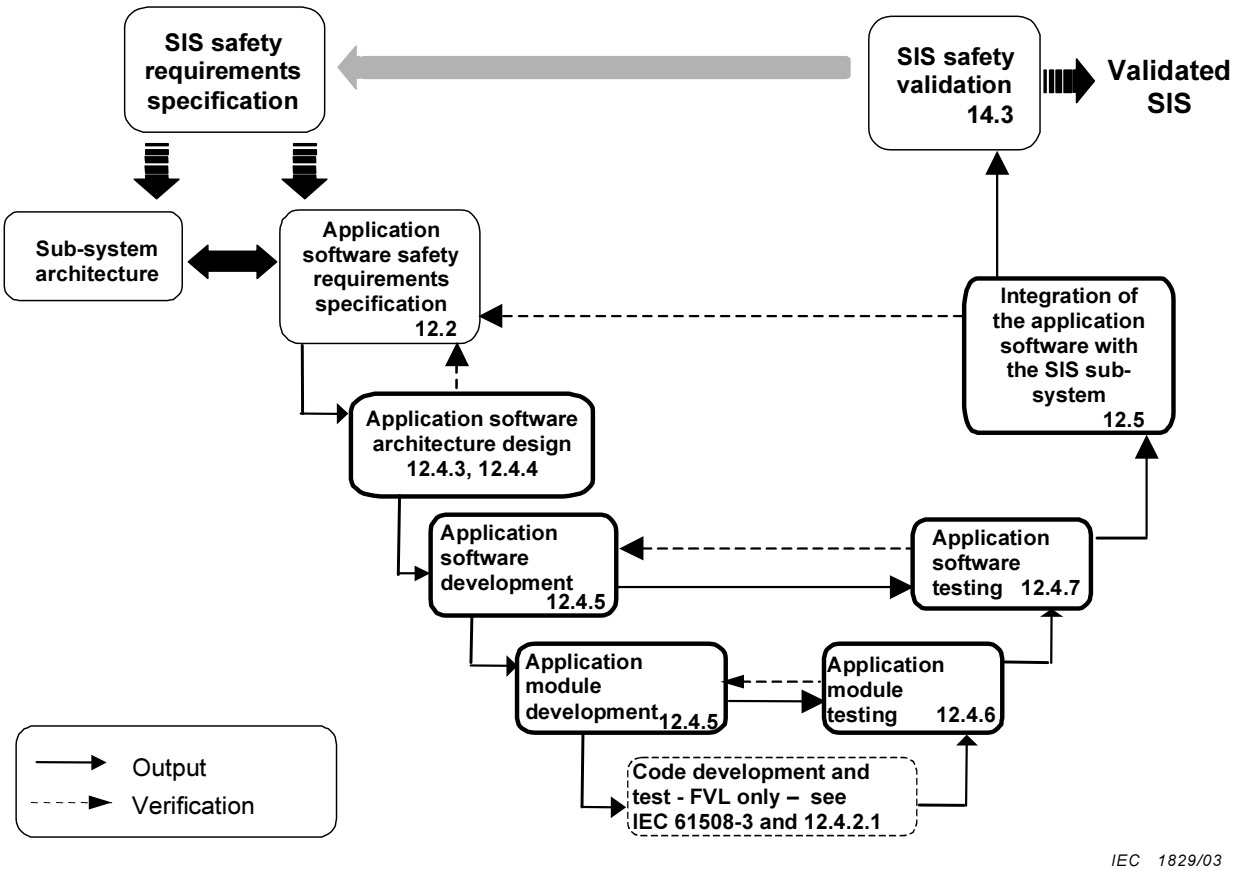
- "application software architecture design" is applied to the software for each SIF in a way that ensures the software design is consistent with the hardware architecture;
- "application software development" is interpreted as the design and implementation of the safety logic using the IEC 61508-3 and IEC 61508-4 compliant limited variability language;
- "application software testing" is interpreted as the verification and test of the application software; and
- "Integration of the application software with the SIS subsystem" is interpreted as the integration and verification of each process safety function implemented in the limited variability language.

An example of an application software development lifecycle using an IEC 61508 series SIL 3 compliant PLC is given in Annex D.

Where a new "function" or "function block" is to be implemented using elements of the IEC 61508 series compliant Limited Variability Language (for example, implementation of a common burner interlock sequence or pump interlock sequence) then:

- "Application Module Development" in the "V" model is interpreted as the design and implementation of the new function; and
- "Application Module Testing" is interpreted as the verification and testing of the new function.

In the case where a new function is to be written in a full variability language and therefore software code development is needed, then, as the "V" model (Figure 3) indicates, the developer should follow all of the lifecycle phases and procedures defined in IEC 61508-3.



NOTE Unless otherwise indicated, subclause numbers in this figure refer to IEC 61511-1.

Figure 3 – Software development lifecycle (the V-model)

12.1.2.3 No further guidance provided.

12.1.2.4 The following are considerations for the selection of methods, techniques and tools:

To select methods, techniques and tools that may contribute towards the software having the required quality, consider the following key quality parameters for the application software:

- simplicity;
- suitable commentary and natural language support;
- compartmentalization to reflect the application;
- test coverage;
- understandability by personnel involved in the support process;
- commonality of style with other related application software.

Approaches to identifying the important parameters include

- discussions with stakeholders including operations and maintenance;
- review of current practice and industry standards;
- review of manufacturer's recommendations;
- analysis of previous experience;
- discussions with peers.

Select the methods, techniques and tools to optimise the important quality parameters taking into account the considerations below.

Methods and techniques should be selected to minimize the risk of introducing faults into the application software during development. This may include the consideration of

- well-defined syntax and semantics;
- suitability for the application;
- understandability by the application developers;
- guarantee of properties important to the SIF (for example, worst case execution time);
- evidence of successful use in similar applications;
- rules and constraints aimed at restricting the use of “unsafe” features of the method.

Tools should be selected to implement the methods and techniques so as to reduce human error in their practical application. This may include the consideration of

- familiarity with tools by the appropriate members of the development team;
- evidence of successful use of the tools in similar applications;
- rules and constraints aimed at restricting the use of “unsafe” features of the tools;
- documented list of the precise version of all tools and the SIS;
- compatibility between the different tools and with the SIS;
- ability to generate application software documentation.

Typical examples of tools used during the lifecycle phases include:

- application code generators;
- configuration management;
- static analysers (for example, tag name checker, scan time checker);
- simulators;
- test harnesses including software test programs;
- engineering workstation.

Other methods, techniques and tools that could be considered include metrics measurements (for example, test coverage) and use of different tools to enhance verification of a function(s) (for example, back-to-back tools).

In order to reveal and remove faults that already exist in the software, verification is recommended throughout the development lifecycle. Typical approaches are described in 12.7.2.3.

To ensure that the faults remaining in the software will not lead to unacceptable results, the following could be considered:

- on-line checking techniques and exception handling;
- use of vendor offsite databases and global fault reporting;
- monitoring of SIS failure reports and of process issues and their impact on the SIS;
- mirroring of key SIS functionality in other systems;
- use of a duplicate of the SIS application software during the training process.

To ensure that the software can be maintained throughout the lifetime of the SIS, the following could be considered:

- program for management of change (see Clause 17 of IEC 61511-1);
- ongoing management support and maintenance training;
- availability of support tools and development platform throughout the lifetime of the SIS;
- well-documented and preferably widely used methods to facilitate adequate human resources and skills throughout the life of the SIS;
- use of development and documentation rules aiming at facilitating understanding and limiting the effects of changes in software;
- 'as-built' and up-to-date documentation;
- ability to develop and test off-line.

12.1.2.5 No further guidance provided.

12.1.2.6 No further guidance provided.

12.1.2.7 No further guidance provided.

12.1.2.8 No further guidance provided.

12.2 Application software safety requirements specification

12.2.1 Objective

12.2.1.1 No further guidance provided.

12.2.2 Requirements

The overall SIS architecture may impose additional functional software requirements to the specified safety instrumented functions. A typical example is the 1oo2 selection logic for redundant sensors as well as a specified safe action on detection of a dangerous failure by sensor self-diagnostics. Examples given in Annex B list those requirements originated from the applied architecture.

The application software should also take into consideration the diagnostics provided by the PES and be developed to take the appropriate actions defined in the logic solver Safety Manual.

The detailed safety requirements for each safety instrumented function can typically be defined by use of logic diagrams or cause and effect drawings. In many cases, the programming languages provided by the logic solver vendor can be used to define the requirements. Typical languages that can be used are function block diagram or cause effect matrix. The vendor supplied language selected should be suitable for the application. The use of the vendor supplied languages to define the detailed requirements can often avoid errors that occur in the translation of the requirements from other forms of documentation. Liberal use of comments should be provided to define safety and non-safety functions and the SIL requirements of all safety functions.

The detailed functional safety requirements specification should include all necessary functions during all modes of operation of the process being protected. Additionally, the periodic testing of all the safety instrumented functions should be provided. This typically requires the definition of maintenance override capabilities so the sensors and final elements can be tested without shutting down the process. The same methodology described in the paragraph above can be used to document these requirements.

If multiple SIS are used to implement safety instrumented functions, documentation should be provided to explain which functions are to be implemented in each SIS. If multiple SIS are used to implement the same safety instrumented function then the interaction and independence of each SIS should be documented. This documentation should include the expected SIL that should be provided by each SIS.

For additional guidance, refer to 10.2.1 and 10.3.1 of this standard.

12.2.2.1 No further guidance provided.

12.2.2.2 Prior to development of the application software, the user provides a process risk and hazard assessment which is used to identify the software safety requirements in terms of the safety instrumented functions and their SIL. Once the decision to implement the safety instrumented functions in software is made, any conflicts, discrepancies and omissions in the safety requirements specification which come to the attention of the software designers should be addressed. One example might be the effect of the order of execution of the safety instrumented functions within the software. Another example would be the response of the application software as it relates to energy outages.

12.2.2.3 The application software safety requirements should be developed as a traceable response to the SIF safety requirements specification. Factors to be addressed include:

- functionality and timing requirements needed to implement the user-defined SIF;
- software system's interface with the process and people;
- relationship between the process hazards and the functionality provided by the application software;
- boundaries of behaviour of the application software which are permitted in order to remain within the safety envelope of the process (for example, inability to deal with erroneous input conditions);
- allowable functionality of the utility software provided within the logic solver, (for example, prioritisation of the safety logic and I/O over communications, error handling and system diagnostics;
- hardware platform and system software on which the application software executes and the configuration of the hardware and system software;
- hazards which could arise in the process as a result of the functionality of the system of which the software is a part (for example, inappropriate hardware failure modes on removal of power);
- constraints on the methods and procedures which could be used by the designers as a result of the Safety Manual for the supporting logic solver.

In order to avoid difficulties at later stages of the development process, it is also important to consider the strategy by which it was intended to show that the application software requirements had been achieved.

Where application software is used in the safety instrumented system, the functional safety assessment may include:

- inspection techniques to show that the application software functions achieve the process hazard requirements;
- functional testing to show that the application software executed the required functions and, as far as possible, that any extra functionality in the software would not result in hazardous conditions;
- structural testing to show that the application software executed the required functions in the necessary timing;
- functional failure analysis and "what if" analysis to show that application software functions would not result in hazardous conditions;
- audit to show that a controlled process of development and verification is in place and the correct software version is in use.

12.2.2.4 No further guidance provided.

12.2.2.5 No further guidance provided.

12.2.2.6 No further guidance provided.

12.3 Application software safety validation planning

For additional guidance, see 14.3.

12.3.1 Objective

12.3.2 Requirements

12.3.2.1 No further guidance provided.

12.4 Application software design and development

12.4.1 Objectives

12.4.1.1 No further guidance provided.

12.4.1.2 No further guidance provided.

12.4.1.3 No further guidance provided.

12.4.1.4 No further guidance provided.

12.4.1.5 No further guidance provided.

12.4.2 General requirements

There are a number of approaches to providing safe application software in SISs. However, regardless of the approach used to achieve safe application software, it is assumed that the safety life cycle steps prior to application software development have been executed properly (for example, hazard and risk assessment, functional description development, equipment (hardware and software) selection).

When the facility has no experience, support, or troubleshooting capability, then prior to implementing the following approach, training and operating experience (preferably in a non safety application) is recommended. To enhance this effort, a liaison with other PE logic solver users of the same equipment in the same environment should be established. The degree of confidence in this approach is a major factor in determining the application of the PE logic solver in the SIS application.

Following is a list of items to consider when developing application software for SISs.

- break the application software into discrete SIF with a SIL for each SIF;
- understand the hardware architecture of each SIF and duplicate this hardware in each SIF application software;
- do not optimise the application software if this leads to excessive complexity (this often requires an advanced programmer to interpret the application software);
- use application software development techniques from the vendor instructions (for example, Safety Manual);
- do not combine application software from one SIF with any other SIF;
- use application software language (for example, type, function) in which the facility is trained, capable of understanding and troubleshooting;

- provide a written description of the application software consistent with the functional description, located with the application software documentation;
- modularise the application software consistent with the process flow (for example, the first module is common application software which is not SIF related but which is required in the SIS, the second module is the first SIF located at the process inlet, the last module is the last SIF located at the process outlet);
- thoroughly test (for example, simulate, inspect, review) each application software module and obtain second independent analysis (include the operating and maintenance department here and in all subsequent steps); thoroughly test the combination of modules that make up a process subsystem and obtain second independent analysis;
- thoroughly test the SIS application software;
- obtain second independent analysis;
- utilize application software when checking out the hardware (for example, confirming I/O connected to correct sensor/final element);
- include testing of the application software in the run-in (for example, process operation without hazardous material) of the process;
- application software support team members are to be available during process turnover to facility (for example, commissioning).

The application software documentation will be used to determine the suitability of the application software to each SIF SIL. An independent analysis should be made to determine that the application software meets the SIL.

IEC 61508-3 and IEC 61508-6 provide alternate approaches and further guidance in this matter.

12.4.2.1 No further guidance provided.

12.4.2.2 With regard to guidance on selection of application software design methods and techniques, systems with a safety requirement up to SIL 3 should be designed in accordance with the instructions given in the supplier's Safety Manual as part of a system conforming with IEC 61508. For SIL 4 systems, the developer should additionally confirm that the selected methods do conform with the requirements of IEC 61508-3.

With regard to guidance on selection of application software test and verification methods and techniques, systems with a safety requirement up to SIL 3 should be verified in accordance with the guidance given in 12.7. For SIL 4 systems, the verifier should also confirm that the selected methods do conform with the requirements of IEC 61508-3.

12.4.2.3 No further guidance provided.

12.4.2.4 In general, in order to ensure testability, it is recommended that the application software integration test specifications are considered during the design and development phase.

12.4.2.5 Where the application software in a SIS is to implement safety instrumented functions of different SILs, they should be clearly separated and labelled. This allows the software of each safety instrumented function to be traceable to the proper sensor and final element redundancy. It also allows the functional and validation testing of the functions to be commensurate with the SIL. The labelling should identify the SIF and the SIL.

Separate areas of the software should be used for non-safety and safety instrumented functions. One way to demonstrate adequate independence could be to comply with all of the following:

a) safety instrumented functions in the application software are clearly labelled as SIF application code;

b) non safety instrumented functions in the application software are clearly separated;

c) all variables used in the implementation of safety instrumented functions are labelled;

d) all application code implementing non-safety-instrumented functions are labelled as non-safety instrumented function code;

e) all application code using non safety variables and SIF variables meet the following conditions:

   – the non safety application code (programs, functions and function blocks) do not write into any SIF variables used in the safety application code,

   – the safety application code does not depend on any non safety variables in the implementation of safety instrumented functions;

f) all safety application software (i.e., code and variables) is protected against any non-safety software changes;

g) if safety and non-safety application software share the same resources (for example, CPU, operating system resources, memory, buses), then the safety instrumented function (for example, response time) of the safety application software is never compromised.

Ideally, the interactions between the application code (SIF and non safety) and all variables (SIF and non safety) should be checked automatically by the application development software. If this feature is not available, the application software developer and other persons performing verification and validation of the application software should check all application code and associated variables for conformance to the separation rules given above.

12.4.2.6 No further guidance provided.

12.4.2.7 No further guidance provided.

12.4.3 Requirements for application software architecture

The software architectural variations possible in a typical SIS logic solver are very limited and are best understood by looking at the major steps in the development of the application programs. The developer will typically perform the following major steps in the development and testing of the application programs.

a) Configure the I/O modules and memory variable data areas.

b) Develop the tag names for all the I/O and memory variables. The tag naming should follow a consistent convention.

c) Define the technique for maintenance override. Some users will require switches wired through digital input points to initiate maintenance override. Others will use controlled data input to the SIS from a display station. In any case, secure handling has to be ensured to avoid unintended overrides. Maintenance overrides should be announced.

d) Define the sensor and final element diagnostics and the periodic testing philosophy. This will be dependent on the sensor and final element redundancy. The philosophy needs to be defined carefully and should include the appropriate alarming during the test period.

e) Define the communication variables to other systems peripheral to the SIS. If the variables are memory variables they will have to be assigned to appropriate data areas so they can be accessed by the communication subsystem. Variables that can be modified by other systems peripheral to the SIS should be carefully defined and are typically placed in a special read/write area of memory.

f) Define where and how the sequence of events is recorded and understand its impact on the SIS.

g) Develop custom functions and function blocks. This customisation is very desirable since repetitive operations can be programmed, tested and used repeatedly in the application programs.

NOTE Functions, function blocks and programs are defined in IEC 61131-3.

h) Decide what safety instrumented functions and other functions should be included within a given program. It is desirable to separate the safety and non-safety functions into separate programs so that the emphasis can be placed on the safety critical programs. It is also desirable to limit the size of the programs to a few functions.

i) Develop the application programs. The application program structure should be consistent with the structure of the process. (for example, in a chemical plant the application software for each process unit should be grouped together. Within each process unit separation is provided between equipment for ease of understanding and maintenance).

j) Determine the proper execution order of the networks and logic, within each program and the execution sequence and desired execution rates of all the application programs. Confirm that the execution rates of the application programs are consistent with the required process response times from the software safety requirements specification.

k) Test the application software using the monitoring capability of the development environment (where available).

l) Download the application software into the logic solver.

m) Test all the logic solver inputs, outputs, application software and the interface to the other systems peripheral to the SIS.

12.4.3.1 No further guidance provided.

12.4.3.2 No further guidance provided.

12.4.3.3 No further guidance provided.

12.4.3.4 No further guidance provided.

12.4.3.5 Examples of safety data integrity verification include

- out of range I/O data checks;
- validation of communicated application data;
- tag naming consistency checks for example, multiple use of same tag name checks;
- override validity checks for example, maintenance and start-up override validity checks;
- alarm and set point validity check.

12.4.4 Requirements for support tools, user manual and application languages

A development environment is a set of tools which supports the coding of the application software, the configuration of application parameters and interfaces and the testing/ monitoring of the application software execution. The environment typically provides the following capabilities.

a) **Configuration editor.** This editor is used to configure the I/O subsystem, the I/O memory variables, and communication functions.

b) **Language editors.** These editors are used by the application programmer to develop the programs that perform all the functions needed by the system (safety and non-safety).

c) **Libraries of certified functions and function blocks.** These functions and function blocks can be used in the application programs.

d) **Custom function and function block development capability.** Some suppliers provide a development environment that allows the user to develop custom functions and function blocks that can be used by the supported application languages. These custom functions and function blocks should be thoroughly tested prior to use in the application program.

e) **Application program scheduling facility.** These scheduling facilities support the setting of the order of desired execution sequence and their scan rates.

f) **Downloading capability.** This allows the developer to download the application software, function block libraries, variable data and other configuration information into the logic solver hardware for execution.

g) **Emulation capability.** Some suppliers provide a development environment with the capability to emulate all of the application programs on the computer that supports the development environment. This allows thorough off-line testing of the application programs before they are downloaded into the logic solver.

h) **Program monitoring capability.** The monitoring capability allows the user to view data from the executing program on user-defined screens or on the actual function block or ladder diagram program screens. The development environment may also provide the capability to monitor the execution of the emulator. In addition, the programs executing in the logic solver can be monitored.

i) **Diagnostic displays of the logic solver.** These displays show the status of the main processor modules, communication modules, and the I/O modules in the system. Typically, the pass, fail, active status of each module is shown; and in many cases, more detailed information about faults in the system is available.

12.4.4.1 No further guidance provided.

12.4.4.2 No further guidance provided.

12.4.4.3 No further guidance provided.

12.4.4.4 Application language translators that are proven in use and/or have been certified to accepted industry standards are preferred.

12.4.4.5 No further guidance provided.

12.4.4.6 No further guidance provided.

12.4.4.7 Safety Manual example

Components and devices used in SIF applications that comply with this standard should be provided with documentation that details all known aspects of installation, maintenance, configuration, programming and operation that should be observed if the component or device is to meet the safety requirements specification of the application.

This standard is frequently titled the “Safety Manual” of the component or device. It may, however, be comprised of the suppliers standard Installation, Maintenance and User’s Manuals with an additional document specifying those aspects relating to its use in SIF applications, the limitations of use in these applications, the actions that should be taken on diagnostic alarms and the known failure modes. It should also define those features, configurations and/or program statement types that should not be used when the component or device is used in a SIF application.

Limited variability programming allows the use of global data; therefore, the Safety Manual should provide guidance to the programmer on how to use the programming tools to scrutinise and check the correct use of data variables. Other features to address may include memory mapping, checks on status flags and validity checks on input values.

Instructions and examples to enable a group of programmers to produce programs of similar format and style may also be provided either as part of the Safety Manual or as an application specific document. These instructions should include details of specific algorithms or functions that are not to be used in the programs, since the algorithms or functions may result in unexpected behaviour which might affect safety.

The programmer should be warned not to make any assumptions beyond those defined in the Safety Manual, for example, not to use compiler capabilities which are omitted from the Safety Manual. Ideally, the compiler would have been configured to enforce these restrictions.

Example of a typical Safety Manual organization and contents

The following example of a manual organization diagram with contents example is for a typical logic solver that conforms to IEC 61511.

The example shows each individual chapter with the primary contents headings for each chapter shown.

Table 1 – Typical Safety Manual organisation and contents

Chapters	Principal contents
Introduction	General information, equipment requirements, manual organization, conventions, related documentation, release history, terminology, product overview.
Installation	Site planning environment, process connections, start-up procedures, shut-down procedures, application modifications, implementation of functions in systems already operating.
Configuration and application building	Design considerations[a], capacity and performance, tutorial
Runtime operation	Product operation, operating overview, operating instructions
Maintenance	Preventive maintenance, hardware indicators, error messages, application and system alarms, fault finding and user repair
Appendices	System messages, check list, application solutions
Index	Safety message index

[a] Design considerations specify all aspects of configuration and application programming that are relevant to the safe configuration and programming of the PE logic solver. These will include but not be limited to:
- logic solver processing times, I/O update rates, communication rates, sequence of logic solver operations;
- system alarm handling requirements;
- constraints of configuration and programming.

12.4.4.8 No further guidance provided.

12.4.5 Requirements for application software development

Before proceeding with the development of the application software, the following items should be checked:

- the SIS logic solver and its associated I/O modules should be in compliance with IEC 61511-1;
- all restrictions and operating procedures necessary for compliance with IEC 61511-1 should be provided in user documentation or documents issued by the logic solver vendor. These documents are commonly referred to as the Safety Manual;
- sensors and final elements utilising programmable electronics should be in compliance with IEC 61511-1;
- when periodic on-line testing is performed, a maintenance override capability may be provided to allow testing of sensors and final elements.

The application software is typically written in the programming languages provided by the logic solver supplier or the smart field device suppliers. The application can be written using a full variability language (FVL) such as instruction list or C, a limited variability language (LVL) such as function block diagram or ladder diagram, or a fixed program language (FPL) where the user only enters data needed by the fixed program.

If the application software is written in a FVL, the developer should follow the requirements and guidelines in IEC 61508-3. If the application software is written in LVL or FPL, the developer may follow the IEC 61511-1 requirements and guidelines. The developer should follow the restrictions and procedures provided by the logic solver vendor in the Safety Manual. Programming guidelines and coding/configuration rules should also be developed and used if needed.

12.4.5.1 No further guidance provided.

12.4.5.2 No further guidance provided.

12.4.5.3 An example of an application global variable would be a safety alarm such as a high temperature alarm that is changed depending on the batch constituents under process.

An example of an application global constant would be the high combustible gas alarm limit used in fire and gas protection systems, for example, 20 % LEL (Lower Explosion Limit).

12.4.5.4 No further guidance provided.

12.4.5.5 No further guidance provided.

12.4.5.6 No further guidance provided.

12.4.6 Requirements for application software module testing

Application software testing may take place initially on a simulator and then on the logic solver hardware against the specifications produced in the design and requirements specification stages. The purpose of the initial testing phases (simulation and testing against the design specifications) is:

- to demonstrate that the software modules provided the necessary functionality and are incapable of any prohibited behaviour;
- to subject the software to a wide range of conditions and sequences to show that it is resilient to unexpected behaviour.

The purpose of subsequent stages of testing (integration test and factory acceptance test) are to show that the application software achieved its requirements on the specified hardware and within the defined time relationships.

The final stage of testing, i.e., demonstration that the integrated system worked correctly in its intended environment, with the intended physical devices and interfaces and with the defined operating procedures, can only be fully completed during the whole system installation and commissioning.

From the start of the formal testing, all changes to software functions and configuration data should be implemented strictly in accordance with a defined modification procedure.

12.4.6.1 No further guidance provided.

12.4.6.2 No further guidance provided.

12.4.6.3 No further guidance provided.

12.4.7 Requirements for application software integration testing

12.4.7.1 No further guidance provided.

12.4.7.2 No further guidance provided.

12.4.7.3 No further guidance provided.

12.5 Integration of the application software with the SIS subsystem

12.5.1 Objective

12.5.1.1 No further guidance provided.

12.5.2 Requirements

12.5.2.1 The integration test may be implemented at any phase up to the SIS validation.

12.5.2.2 No further guidance provided.

12.5.2.3 No further guidance provided.

12.6 FPL and LVL software modification procedures

12.6.1 Objective

12.6.1.1 No further guidance provided.

12.6.2 Modification requirements

Wherever possible, on-line modifications to a safety instrumented system should be avoided. If on-line modifications are required, the complete procedure should be documented and approved according to the safety planning.

The following process is recommended for all changes to programmable safety instrumented systems:

a) Planning and resources

A program to modify a programmable safety instrumented system should be managed, planned and resourced to the appropriate level to ensure the safe implementation of the change.

b) Impact analysis

The required modification may require a full hazard and risk assessment including all possible effects on the unchanged parts of the system (safety impact analysis).

c) Design

The modification design should follow the full lifecycle process as described in IEC 61511–1.

d) Verification

Full offline verification for hardware and application software should be completed prior to the installation of the change.

Where the boundary of the software changes can be clearly delineated and controlled, only the delineated application software needs to be verified before commissioning.

e) Installation and commissioning

The installation and commissioning of the change should follow the procedures defined in IEC 61511–1 for installation and commissioning of safety instrumented systems.

f) Acceptance test validation

A system validation (cause and effect test) will be implemented for the modified parts of the systems prior to bringing the modified parts of the system online.

g) Personnel

Only identified personnel who are competent to implement modifications based on their training and expertise should be authorised to carry out modifications.

h) Off-line modifications

When implementing off-line modifications of the application software, it should be verified that the correct versions of the application software, including operational parameters, are used.

12.6.2.1 No further guidance provided.

12.7 Application software verification

12.7.1 Objectives

12.7.1.1 No further guidance provided.

12.7.1.2 No further guidance provided.

12.7.2 Requirements

The application software safety requirements specification will include:

- the safety instrumented function requirements (for example, SIL's of safety instrumented functions; logic flow diagrams/cause and effect diagrams);
- timing constraints (for example, input to output minimum response times);
- architectural constraints (for example, redundancy requirements, communication interfaces and functional segregation).

Verification ensures that the specified requirements are being met at each phase of the application software development.

Data verification includes confirmation that data used within the application software is correct and where appropriate unique (for example that TAG names are uniquely assigned, that data is not misused by subsequent functions and that constants such as alarm set points are valid and correct).

Verification for protection against unauthorised change, would include verification that the mechanisms exist (for example, password protection with levels of access) and that these mechanisms have been adequately utilised.

12.7.2.1 No further guidance provided.

12.7.2.2 No further guidance provided.

12.7.2.3 At each distinct phase of the application software development cycle (including testing), verification confirms that the phase has been successfully completed. Verification is, in general, completed by a verification team that consists of one or more persons.

To reduce errors due to preconceived mindsets, the verification should include:

- for SIL 1, a peer review by another member of the application development team;
- for SIL 2, a peer review by a person who is not a member of the application development team;
- for SIL 3, a peer review by a person who is a member of an independent department.

Where the software development tools include some automatic verification operations (for example, checking for double use of tags (named variables)) then the verification team should confirm that the tools have been properly used and the correct results obtained.

For all SILs, it is recommended that the test coverage encompasses all application software SIFs and SIS failure responses (for example, power supply failures, processor failure, input hardware failure, output hardware failure and communication failures). However to further

reduce any errors remaining in the software, for higher SILs it is recommended that the following additional testing is carried out:

- for SIL 2 and SIL 3, testing based on the internal structure (for example, internal algorithms, internal states);
- for SIL 3, stress testing (for example, abnormal range conditions of input variables and internal variables, abnormal combinations of inputs, abnormal sequences and loading).

For all SILs it is recommended that the verification and test documentation is sufficient to show that the verification and tests have been carried out and were successful. However, for higher SILs, it is also recommended that:

- for SIL 2 and SIL 3, the documentation is sufficient to allow an assessment of the adequacy of the verification and testing;
- for SIL 3, the documentation should be sufficient to allow an independent person to repeat the tests and review the coverage achieved.

12.7.2.4 No further guidance provided.

13 Factory acceptance testing (FAT)

13.1 Objectives

13.1.1 No further guidance provided.

13.2 Recommendations

13.2.1 Although conducting a Factory Acceptance Test (FAT) is not a requirement, a FAT is recommended for those logic solvers implementing safety instrumented functions having fairly complex application logic or redundancy arrangements (for example, 1oo2, 1oo2D, 2oo3 etc.).

13.2.2 The most important part of the FAT is to have a well defined, well written and well structured test procedure that defines how to test the application logic and what to look for after each step.

Personnel that will be operating the process should attend the FAT since it will give them some early training on the operation of their SIS. Often, they can also provide good suggestions or enhancements to the test procedure that were not foreseen during the design.

13.2.3 No further guidance provided.

13.2.4 No further guidance provided.

13.2.5 During the FAT, interfaces should be tested (for example, communications between the BPCS and SIS).

13.2.6 No further guidance provided.

13.2.7 No further guidance provided.

14 SIS installation and commissioning

14.1 Objectives

14.1.1 No further guidance provided.

14.2 Requirements

14.2.1 No further guidance provided.

14.2.2 The SIS should be installed per the design and installation plan. Any deviations from the design should be properly reviewed with the project team to ensure all of the design requirements are still satisfied. After the SIS has been properly installed, it should be fully commissioned and validation activities should be initiated.

14.2.3 While IEC 61511-1 has addressed commissioning as a single phase, it is recognized that the application, the experiences of the project team, and project needs may require commissioning to be accomplished in several phases.

14.2.4 No further guidance provided.

14.2.5 No further guidance provided.

15 SIS safety validation

15.1 Objective

15.1.1 The objective of the SIS safety validation is to validate that the SIS achieves the requirements stated in the safety requirements specification. Validation activities should be completed prior to the placing of the SIS into operation.

15.2 Requirements

15.2.1 No further guidance provided.

15.2.2 No further guidance provided.

15.2.3 No further guidance provided.

15.2.4 If the SIS has already been through a Factory Acceptance Test (FAT), this may be taken into consideration during the validation. The validation team should review the results of the FAT to ensure that all of the application software was successfully tested and all problems found during the FAT have been corrected.

It may be unnecessary to repeat application software testing at the final validation. This is applicable when

- this approach was anticipated and included in the validation planning,
- the application software has been verified to meet the safety requirements specification during the FAT, and
- the application software version is verified to be the identical version tested at the FAT.

However, it will be very important to ensure that there has been no shipping/storage/handling damage, that all sensors and final elements are correctly connected to the logic solver, that the safety instrumented functions perform properly and that the operator interface provides the necessary information. The equivalent of a proof test is strongly recommended in order to claim SIS validation, because a separate test of the logic solver and the field elements does not equal a complete end-to-end proof test.

15.2.5 No further guidance provided.

15.2.6 No further guidance provided.

15.2.7 No further guidance provided.

15.2.8 No further guidance provided.

16 SIS operation and maintenance

16.1 Objectives

No further guidance provided.

16.2 Requirements

16.2.1 No further guidance provided.

16.2.2 No further guidance provided.

16.2.3 No further guidance provided.

16.2.4 No further guidance provided.

16.2.5 No further guidance provided.

16.2.6 No further guidance provided.

16.2.7 No further guidance provided.

16.2.8 No further guidance provided.

16.3 Proof testing and inspection

16.3.1 Proof testing

16.3.1.1 The proof test interval should be selected to achieve the average probability of failure on demand as required in the safety requirements specification.

16.3.1.2 No further guidance provided.

16.3.1.3 The frequency of proof tests should be consistent with applicable manufacturer's recommendations and good engineering practices, and more frequently, if determined to be necessary by prior operating experience.

There are a number of strategies being used to select the proof test interval for a SIF.

For example, some users like to make this proof test interval as long as possible to minimize maintenance cost and the potential impact of testing. In this case, the SIS design may include more redundancy in equipment, increased diagnostic coverage and robust components. After completion of the design, a calculation may then be performed on the design to determine the maximum test interval allowed to achieve the SIL performance defined for the SIF. The negatives to this design philosophy are that each system in a plant will have a different test interval and may require more rigorous compliance tracking. It also may encourage designing the performance toward the low end of the performance curve (for example, $PFD_{avg} = 10^{-1}$ for SIL 1 systems and $PFD_{avg} = 10^{-2}$ for SIL 2 systems).

Other users may wish to standardize on the basis of a defined test interval and test all systems in a manufacturing plant at the same test interval. For example, they may wish to test each SIF annually so they design each SIS accordingly. By pre-selecting a proof test interval prior to beginning the design, user companies can then pre-select architectures, components and diagnostic coverage that will satisfy the SIL for most applications. By having these features already defined in their corporate standards, it reduces the design engineering cost for most applications. In this case, a calculation should be performed on the SIS to ensure the required SIL performance is satisfied with the pre-selected proof test interval.

In the choice of a proof test interval, considerations should be given to the demand rate for Demand Mode systems, the failure rate of each component being tested, and the overall system performance requirements.

NOTE For those applications where exercising the final trip element may not be practical, the procedure should be written to include:

a) testing the final element during unit shut down;

b) testing the SIS by exercising the output(s) as far as practical (for example, output trip relay, shut down solenoid, partial valve movement) during on-line testing;

c) any limitation of the testing period of the final elements should be taken into account in the calculation of the PFD_{avg} of the SIF.

16.3.1.4 No further guidance provided.

16.3.1.5 No further guidance provided.

16.3.1.6 No further guidance provided.

16.3.2 Inspection

As stated in IEC 61511-1, inspecting the SIS is different from proof testing. Whereas a proof test is ensuring the SIS will operate properly, a visual inspection is required to validate the mechanical integrity of the installation.

Normally, the inspection is done at the same time as the proof test but it may be done at a more frequent interval if desired.

16.3.3 Documentation of proof tests and inspection

It is important to document the results of the proof test and inspection for a record of what was found. There are no specific requirements for how long these results should be retained but generally a sufficient number are retained to allow for re-examination of previous results to see if there is a history of component failure.

For example, if a sensor failed a proof test, it is good practice to review the results of previous proof tests to see if this sensor had failed a similar proof test within the past few tests. If the history indicates repeating failures, consideration should be given to redesigning the SIS using a different type of sensor.

17 SIS modification

17.1 Objective

No further guidance provided.

17.2 Requirements

17.2.1 No further guidance provided.

17.2.2 No further guidance provided.

17.2.3 No further guidance provided.

17.2.4 No further guidance provided.

17.2.5 No further guidance provided.

17.2.6 No further guidance provided.

18 SIS decommissioning

18.1 Objectives

No further guidance provided.

18.2 Requirements

18.2.1 No further guidance provided.

18.2.2 No further guidance provided.

18.2.3 No further guidance provided.

18.2.4 No further guidance provided.

18.2.5 No further guidance provided.

19 Information and documentation requirements

19.1 Objectives

19.1.1 No further guidance provided.

19.2 Requirements

19.2.1 The list of the information and documentation that may be used to implement a SIS, includes:

a) results of the hazard and risk assessment;
b) assumptions used when determining the safety integrity levels;
c) safety requirements specifications;
d) application logic;
e) design documentation;
f) modification information and/or documentation;
g) records of verification and validation;
h) commissioning and SIS validation procedure(s);
i) SIS operating procedures;
j) SIS maintenance procedures;
k) proof test procedures;
l) results of assessments and audits.

19.2.2 No further guidance provided.

19.2.3 No further guidance provided.

19.2.4 No further guidance provided.

19.2.5 No further guidance provided.

19.2.6 No further guidance provided.

19.2.7 No further guidance provided.

19.2.8 No further guidance provided.

19.2.9 No further guidance provided.

Annex A
(informative)

Example of techniques for calculating the probability of failure on demand for a safety instrumented function

A.1 General

This annex references a number of techniques for calculating the probabilities of failure for a safety instrumented system designed and installed in accordance with IEC 61511-1. This information is informative in nature and should not be interpreted as the only evaluation techniques that might be used.

The methodologies referenced are from Annex B of IEC 61508-6, IEC 61078, IEC 61025, IEC 61165, and the ISA TR 84.00.02 series.

A.2 Reliability block diagram technique

IEC 61078 and Annex B of IEC 61508-6 illustrate the reliability block diagram technique for calculating the probabilities of failure for safety instrumented functions designed in accordance with IEC 61511-1 and this standard.

A.3 Simplified equations technique

ISA TR 84.00.02-2 illustrates a simplified equation technique for calculating the probabilities of failure for safety instrumented functions designed in accordance with IEC 61511-1 and this standard.

A.4 Fault tree analysis technique

IEC 61025 and ISA TR 84.00.02-3 illustrate the fault tree analysis technique for calculating the probabilities of failure for safety instrumented functions designed in accordance with IEC 61511-1 and this standard.

A.5 Markov modelling technique

IEC 61165 and ISA TR 84.00.02-4 illustrate the Markov modelling technique for calculating the probabilities of failures for safety instrumented functions designed in accordance with IEC 61511-1 and this standard.

Annex B
(informative)

Typical SIS architecture development

B.1 Background

B.1.1 Introduction

The following is provided as an example to illustrate the various steps performed to develop a SIS architecture, which satisfies the requirements of IEC 61511-1. SIS engineering follows guidelines and practices and uses standardized equipment as outlined below.

B.1.2 Guidelines and practices

In the past, safety applications were called "critical instrument systems". Engineering rules, typical examples and best practices as well as test procedures were developed.

Guidelines to determine the required safety instrumented function and SIL with Layer of Protection Analysis (LOPA, as in Annex F of IEC 61511-3), as well as instrument redundancy and design practices exist.

B.1.3 Instrumentation

Instrumentation in safety applications (SIS) utilises vendor information on diagnostics and safe failure fraction (SFF) as well as performance information collected from the applications to calculate the probability of failure on demand (PFD).

B.1.4 Logic solver

The hardware, system software and development system of the logic solver is IEC 61508 SIL 3 compliant and has a limited variability language for its application program.

The system Safety Manual gives detailed guidance on the system application and application software development.

Standard user definable safety functions (for example, transmitter fault detection, redundancy selection such as 1oo2, 2oo3, and output safety override) are available as application program templates. Templates are user developed.

B.2 Work process

B.2.1 Introduction

All engineering activities follow a predefined overall project work process. The development of a SIS has its own process. Individual steps are mapped into the overall process. Functional safety assessment is carried out at the appropriate stages.

B.2.2 Typical SIS lifecycle steps

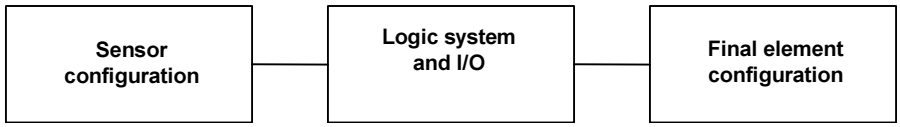
Developing a SIS application requires the following typical steps. In the following we will only discuss step 3, 4 and those parts of step 5 which are related to the system architecture.

Step	Title	Activity
1	Application scope	Define process equipment
2	Functional safety requirements of the process equipment	Define hazard potential , perform Level Of Protection Analysis (LOPA)
3	System safety requirement assignment	Design SIS structure
4	Safety requirement assignment within the SIS	Identify SIS hardware
5	Application software development	Design SIS software
6	Application software testing and validation	Test SIS
7	Installation	Field installation
8	Commissioning	Overall acceptance
9	Operation	Run process

B.2.3 Safety requirement assignment

Available information from LOPA: safety requirements specification and SIL for the SIS application (for example, SIL for each SIF)**.**

Model used to achieve SIL:



IEC 1830/03

Determination PFD: the overall PFD (see above) stays within the SIL limits**.**

Abridged method: standard instrumentation configurations including redundancy types (for example,1oo2), available diagnostics and test intervals can be provided in Tables related to the SIL requirements. These Tables should be based on experience data and proven design of various process applications within the facility. Combining alternative system configurations with known element data to block diagrams enables the selection of the most appropriate choice**.**

SIS component specification: all system components have proven characteristics (for example, PFD, SFF, fault tolerance, systematic requirements for the specified SIL) as mandated in IEC 61511-1**.**

- **sensors and final elements** are appropriately selected for the process application and various type features are standardized by the engineering department according to operating experience.
- **logic systems:** I/O is specified according to sensor and final element requirements. The logic solver, application language, development tools and communication interface is part of the approved safety system. The operator interface can be tailored to application requirements.

B.2.4 Safety requirement assignment within the SIS

In this step, all functions of the safety requirements specification are allocated to system components, functions or software. Safety integrity requirements will determine the appropriate SIS components and the possible SIS architecture.

B.2.5 Architecture related application software requirements

After selection of the SIS architecture, application software may have to be specified for implementation of redundancy (for example, 1oo2) and/or diagnostics, as required for sensors, logic solver, and final elements.

B.2.6 Application software development

The programming language is function block diagram (a limited variability language). Code development and testing is a well known process. Additionally, there are several restrictions for safety function programming which are described in the system Safety Manual in detail.

B.3 Example 1

B.3.1 Introduction

The example used below is not from a real scenario, and excludes consideration of common cause failures with other safety layers. It is specially composed to demonstrate how to apply the previous described SIS design process.

B.3.2 Hazardous scenario

Temperature control of a steam heated reactor fails and opens the steam control valve fully.

B.3.3 SRS and SIL

Safety requirements specification: if reactor pressure exceeds 10 bar, close off steam to the reactor jacket within 20 seconds to avoid exothermic reaction. There is no operator action necessary. The required SIL is 3.

B.3.4 System architecture

System components: pressure sensor configuration, logic solver configuration, final element configuration. Proven in use smart sensors are directly connected to inputs of the logic system. Emergency block valve has solenoid valve integrated and is directly connected to outputs of the logic system. All MTTF data come from actual operating experience.

Available instrumentation:

- pressure sensors comply with 11.4.4 of IEC 61511-1: MTTF 10^5 h, DC = 70 %, SFF = 90 %, proof test interval every year, MTTR = 8 h.
- emergency block valve complies with 11.4.4 of IEC 61511-1: MTTF 8×10^4 h, DC = 0 %, SFF = 60 %, proof test every 6 months, MTTR = 8 h.

Single element PFD:

- sensor: $2{,}2 \times 10^{-3}$ (see Clause A.1) – not acceptable.
- logic solver (redundant): $1{,}3 \times 10^{-4}$ including I/O interface (from certificate).
- valve: $2{,}41 \times 10^{-3}$ (see Clause A.1) – not acceptable.

Find acceptable sensor architecture: select 1oo2 redundancy.
Common cause = 10 %, DC = 90 % (see Clause A.1).
New PFD for 1oo2 sensor architecture: $2{,}3 \times 10^{-4}$.
Check Table 6 of IEC 61511-1 and 11.4.4 of IEC 61511-1, actual fault tolerance = 1 $\rightarrow$ SIL 3 – acceptable.

Find acceptable final element architecture: select 1oo2 redundancy.
Common cause = 10 %, (see Clause A.1).
New PFD for 1oo2 final element architecture: $4{,}65 \times 10^{-4}$.
Check Table 6 and 11.4.4 of IEC 61511-1, actual fault tolerance = 1 $\rightarrow$ SIL 3 – acceptable.

PFD check: sensor + logic solver + final element.
$(2,3 + 1,3 + 4,7) \times 10^{-4} = 8,3 \times 10^{-4} < 10^{-3}$

B.3.5 Additional architecture related safety software

Sensor configuration software: for the above 1oo2 sensor signal selection software is programmed (existing function block) to close the steam valve if:

- one of the two sensors reads a condition exceeding the specified process value;
- the diagnostic reveals a dangerous failure.

Final element configuration software: both steam valve outputs are de-energized in the case that a safe output action is commanded by the safety program.

B.4 Example 2

B.4.1 Introduction

Similar example with consequences resulting in a lower SIL.

B.4.2 Hazardous scenario

Temperature control of a steam heated reactor fails and opens the steam control valve fully.

B.4.3 SRS and SIL

Safety requirements specification: if batch reactor pressure exceeds 10 bar, close off feed of reactant "A" to the reactor within 20 seconds to avoid exothermic reaction. There is no operator action necessary. The required SIL is 2.

B.4.4 System architecture

System components: pressure sensor configuration, logic solver configuration, final element configuration. Proven in use smart sensors are directly connected to inputs of the logic system. Emergency block valve has solenoid valve integrated and is directly connected to outputs of the logic system. All MTTF data are actual operating experience.

Available instrumentation:

- Pressure sensors comply with 11.4.4 of IEC 61511-1: MTTF 10^5 h, DC = 70 %, SFF = 90 %, proof test interval every year, MTTR = 8 h.
- Emergency block valve complies with 11.4.4 of IEC 61511-1: MTTF $2,5 \times 10^4$ h, DC = 0 %, SFF = 60 %, proof test every week (168 h), MTTR = 8 h.

Single element PFD:

- Sensor: $2,2 \times 10^{-3}$ (see Clause A.1) – acceptable.
- Logic solver (redundant): $1,3 \times 10^{-4}$ including I/O interface (from certificate).
- Valve: see below (see Clause A.1 for the formula).

Single sensor PFD:

PFD for 1oo1 sensor architecture: $2,2 \times 10^{-3}$.
Check Table 6 and 11.4.4 of IEC 61511-1, actual fault tolerance = 0 $\rightarrow$ SIL 2 – acceptable.

Single final element PFD: (see Clause A.1 for the formula).
PFD = $\lambda_D \times t_{CE}$, λ_D = 1/(25 000 x 2), t_{CE} = 168/2 + 8
PFD for 1oo1 final element architecture: $1{,}84 \times 10^{-3}$.
Check IEC 61511-1 Table 6 and 11.4.4, actual fault tolerance = 0 $\rightarrow$ SIL 2 - acceptable.

B.4.4.1 PFD check: sensor + logic solver + final element.
$(2{,}2 + 0{,}1 + 1{,}8) \times 10^{-3} = 4{,}1 \times 10^{-3} < 10^{-2}$

B.4.5 Additional architecture related safety software

Final element configuration software: The steam valve output is de-energized when a safe output action is commanded by the safety program**.**

Additionally, monitoring software which proves that the safe state of the valve is reached each time the valve is operated (once per batch, typically every 8 hours) is written. In case of a test failure or if more than 168 hours have elapsed since the last test, the logic solver output stays in the safe state (emergency block valve closed) and the condition is alarmed. This automatic test allows setting the proof test interval in the PFD calculation to 168 hours.

Annex C
(informative)

Application features of a safety PLC

The following is an outline of some key steps an integrator considers when utilizing a small (for example, less than 150 I/O) safety PLC in a SIS application. It is presented to assist the reader during initial design planning.

The safety PLC is a certified SIS logic solver per the IEC 61508 series. For a specific safety application, sensors and final elements are connected to the SIS logic solver I/O terminals and the application program is implemented. All safety functionalities referring to failures of the SIS logic solver (for example, online checks, time control) are part of the embedded system. Necessary checks of sensors and final elements are implemented within the application software; for some functions, approved function blocks exist.

Safety integrity data (for example, PFD, SIL claim limit, etc) of all devices exist. Safety integrity data of the logic solver is given in the manual of the logic solver.

C.1 System

The SIS logic solver is a PLC, which is specifically designed for safety applications. It is type approved to comply with the IEC 61508 series up to SIL 3. It has input and output interfaces for safety-related process signals and communication with other safety PLC's. It also has interfaces for signals and communication which are not safety-related. The system consists of:

- CPU with special hardware features for functional safety, a special operating system and embedded functions for control of failures (for application programming and software integration the integrated redundancy is covered by the development system. The programmer sees only one CPU);
- development system for limited variability language (for example, function block diagram);
- library with approved function blocks;
- special configuration tool for safety instrumented function parameters;
- tool to confirm that the downloaded run-time application software is identical to the source application software;
- Safety Manual.

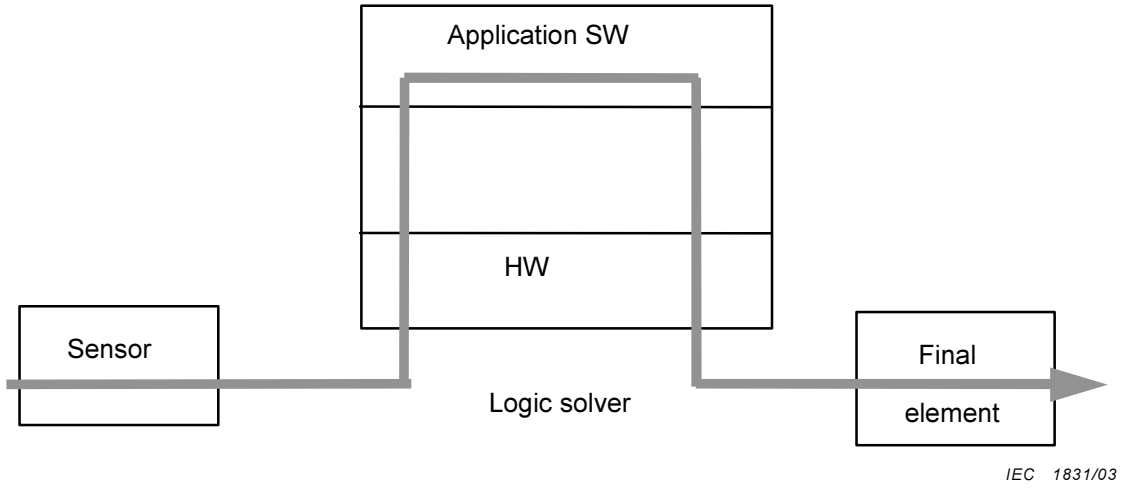


Figure C.1 – Logic solver

C.2 Work process

a) Safety requirements specification will conform to this standard: the following are some key considerations:

   1) specification of all safety instrumented functions;

   2) the range of analogue inputs;

   3) definition of online diagnostics of sensors and final elements;

   4) description of system reactions in case of detected failure modes;

   5) definition of safety instrumented function parameters (for example, maximum cycle time, maximum allowed time of discrepancy of compared inputs);

   6) restrictions in the Safety Manual.

b) Application software safety requirements specification should be derived from a).

   Safety requirements referring to the logic solver hardware (PLC) are described in the Safety Manual. The constraints refer mainly to such items as performance limits, memory size, response time.

   Constraints for software architecture and code implementation are described in the Safety Manual. They refer to the development system of the PLC. Most of the constraints are implicitly given by limited variability language.

c) **Application software architectural design:** the application architectural design should closely reflect the safety instrumented functions and modes of operation specified for the process.

d) **Application software development:** application software development is facilitated by the use of existing function blocks.

e) **Integration:** integration involves the downloading of the configuration data (for example, I/O Tables) and application software and the setting of all parameters, which are different from the default settings.

f) **Verification:** application software is verified before system integration or after system integration. Verification is supported by the development environment.

Annex D
(informative)

Example of SIS logic solver application software development methodology

This example illustrates how one particular SIS logic solver integrator develops safety application software for its customers. This software is typically developed as a part of an overall system integration process that is discussed in the section below.

Since the emphasis is on the safety application software development methodology, it is important to discuss the application software development tools, programming languages and coding standards that were used to develop the application programs. The purpose of this discussion is to provide an example of the typical features of the software development tools, the programming languages and associated language translators that are provided in a SIS logic.

The SIS logic solver has application programming software development tools that support a number of IEC 61131-3 languages. The IEC 61131-3 standard defines a number of languages for the general purpose programming of Programmable Logic Controllers. Since the IEC 61311-3 standard does not address safety applications, it was decided to:

- use limited variability languages common to the process sector;
- eliminate language constructs that are not appropriate for safety applications;
- use a coding standard to further restrict the use of language constructs for critical applications;
- incorporate access security and file protection features;
- supply certified libraries of IEC 61131-3 functions, function blocks, and process related functions (for example, analogue data processing, fire and gas sensors);
- provide third-party certification of the application programming software development tools, libraries, and language translators.

These decisions are discussed in more detail in Clause D.2 on application development software.

An example of a coding standard used by the SIS logic solver programmers is also discussed in Clause D.3. Clause D.4 discusses additional requirements that should be considered for the software development tools.

D.1 Summary of the overall system integration process

The major safety instrumented system integration services provided with the SIS logic solver consisted of a number of activities including.

a) Hardware integration

   This consists of the installation of the SIS logic solver into cabinets with the appropriate termination panels for connecting the process signals to the logic solver I/O modules. Power supplies and power distribution for the logic solver and field devices are also normally included.

b) Application logic definition

The SIS logic solver integration services may also define the detailed logic by working closely with customer engineers. The application logic for each safety instrumented function is defined taking into account the sensor and final element redundancy. The interface for testing and maintenance of the SIS while the process is in operation is also defined to meet the customer's operational requirements. Additional non-safety critical logic may also be included, but is strictly segregated and designed to the same standard as the safety function.

c) Application software implementation and hardware configuration

The SIS logic solver safety certified application software development package is used to configure the SIS logic solver I/O and communication hardware. The application software for each safety instrumented function as well as non-critical application software are also implemented and tested.

d) Factory acceptance testing

Many customers conduct a factory acceptance test to check the correct operation of the hardware and application software before it is shipped to the plant. The hardware and application software are thoroughly tested by the customer's engineers and other operating personnel.

e) Installation of SIS at customer site

Either supplier installation or installation supervision is provided at plant site.

f) Site acceptance testing

Each sensor and final element interface into the SIS logic solvers is checked for proper operation and calibration. Such items as the overall application software, bypass functions for maintenance, are re-tested.

g) Application software and hardware modifications

After initial installation and operation, application software and hardware modifications are implemented using strict plant-approved modification procedures.

D.2 SIS logic solver application development software

As mentioned earlier, the SIS logic solver utilized an application software development package based upon the IEC 61131-3 languages. The software supports three of the IEC 61131-3 languages: structured text, ladder diagram and function block. Separate coding standards are necessary for each language. Instruction List was not included since it is similar to assembly language and is not suited for application programmers. This is consistent with Table C.1 in IEC 61508-7.

A number of additional restrictions were placed upon the IEC 61131-3 language definitions consistent with the requirements outlined in IEC 61508-3 (7.4.4 and Table A.3) and IEC 61508-7 (Clause C.4). These include the following.

a) The IEC 61131-3 standard defines twenty data types (BOOL, SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL, LREAL, TIME, DATE, TOD, DT, STRING, BYTE, WORD, DWORD, LWORD). It should be noted that there are 8 integer data types alone. The support of all these data types also necessitates the support of dozens of conversion and truncation functions. For safety applications many of these data types are not necessary. The number of data types supported was limited to eleven (11). For the particular language the chosen data types provided were BOOL, INT, DINT, DWORD, REAL, LREAL, STRING, TIME, DATE, TOD, and DT. This decision is consistent with the IEC 61508 recommendations to limit the language subset (see Table A.3 in IEC 61508-3).

b) The use of IEC 61131-3 graphic execution control elements (for example, unconditional jumps, conditional jumps, unconditional returns and conditional returns) were not supported since they can lead to looping and unintended bypassing of elements that should be executed (see C.4.6 in IEC 61508-7).

c) A number of structured text language statements were not supported since they can cause looping (for example FOR…END_FOR, WHILE…END_WHILE and REPEAT…END_REPEAT).

d) A limitation was imposed so that the language does not allow multiple programs to write into the same global variable. Many programs can read a global variable but in order to prevent conflicts and overwriting only one program can write into a global variable. In addition, the application programming software provides a warning if multiple writes are programmed accidentally.

e) The programming software should unambiguously define the execution order of all elements in a program. The languages have an algorithm that determines the execution order and displays the execution order on each executable element.

f) The programming software should provide for the separation of safety critical and non-safety critical software. The software provides the programmer with the capability to define safety programs and non-safety programs. It also provides the capability to define safety and non-safety variables. Non-safety programs cannot write into safety variables.

g) The use of VAR_IN_OUT variables has been found to be very confusing to most application users. The use of the VAR_IN_OUT variables needs to be very thoroughly documented, or the programming language should not support them.

D.3 Coding standards for the application programmer

In order to ensure the development of safe application software, coding standards should be established for the application programmer. Following are a number of guidelines for use by application programmers when developing application software with this particular development software:

a) The application programmer should use the limited variability languages (function block diagram or ladder diagram) to implement the safety instrumented functions. Even these languages should be restricted (see Clause D.2 above on language subset).

b) Structured text (ST) is a full variability language, and its use should be limited. The usage should be limited to the implementation of functions and function blocks wherever possible. This restriction was implemented so that operational personnel not proficient in programming would understand the safety program.

c) The size of the programs should be restricted to a reasonable size. Safety instrumented functions for different process units should be in separate programs. Ideally a program should only contain a small number of safety instrumented functions for one process unit.

d) Aliasing should be avoided. For example, if the programming software supports arrays, the programs using the arrays should check the array pointers to make sure they are in the valid range.

e) When the application includes non-safety critical logic as well as the safety critical logic, the non-safety critical logic should be in separate programs and utilise the separation rules incorporated in the program.

D.4 Other requirements for configuration/programming and run-time systems for safety applications

The application programming software provides a number of features that allow user access to SIS logic solver information. However, it is necessary to ensure the security of the developed software and to allow the user to check the software for proper operation. A few of these features are outlined below:

a) The programming software provides a security system that restricts all users to only those functions that are commensurate with their duties (for example, corporate manager, site manager, project manager, project engineer, senior programmer, programmer, operator). Each user logs into the system with a name and password and can then work at their assigned functional level. The security system also provides a user level for safety programming and another for non-safety programming since the user companies may want to restrict the changing of safety programs to a few persons at the site.

b) Protected or locked functions and libraries are provided and the programmer cannot access or change them. This ensures that libraries that have been certified or thoroughly tested cannot be modified unless approved by a formal modification request. The security system allows the user to define a high level person that can access and change the libraries (typically a corporate or site manager).

c) The programming software also provides a version number on all elements in the project being developed. Any change of the system configuration, function, function block, or program results in the version number being changed for that element. This allows the user to quickly know if their documentation is out of date and allows them to concentrate the testing on those items that have been modified. Version comparison functions are included so users can check all changes, including unintentional changes. These comparison functions should include any changes in the global tag name database and the program execution list.

d) The software provides file security by computing and checking the cyclic redundancy checks on all data streams stored in the compound file structure of the application project.

e) The SIS logic solver provides access to its diagnostic information and hence the programmer can take appropriate actions based upon the status of the logic solver.

f) The SIS logic solver provides a run-time environment that provides arithmetic exceptions so the programmer can check for proper arithmetic operations.

g) The programming software provides the ability to emulate all of the programs developed on the programming workstation. This allows the programmer to check all of the developed software off-line before it is loaded into the SIS logic solver. This feature should be mandatory for cases where a change is made to the on-line program while the system is in operation.

h) The software supports DDE (dynamic data exchange) which can be used to interface to simulation software. This provides the capability for additional off-line testing of the application software before it is loaded into the safety controller.

D.5 Assumptions

This clause discusses the assumptions associated with the hardware and software used to develop the application software. Documentation and procedures are also discussed.

1) The SIS logic solver and its associated I/O modules have been assessed by a third party and found to be compliant to the IEC 61508 series. The scope of the IEC 61508 series certification awarded by the third party is for use as a component in SIL 3 safety instrumented functions.

2) The languages are a limited variability subset of the IEC 61131-3 function block diagram (FBD), ladder diagram (LD), and structured text (ST) languages. All functions and function blocks provided in the application libraries have an attribute that identifies whether the function can be used for safety or is restricted to non-safety only. Only functions and function blocks with the safety attribute can be used to implement safety instrumented functions in application programs designated with the safety attribute. Application programs designated with the non-safety attribute can use functions and function blocks with the non-safety attribute and the safety attribute.

3) All of the supported IEC 61131-3 programming languages and libraries of functions and function blocks with the safety attribute have been certified for compliance to the IEC 61508 series.

4) All certifying organization restrictions and operating procedures are provided in the user documentation.

5) For periodic testing of all elements of the SIS, a methodology for maintenance override is typically necessary to allow on-line testing without shutting down the process under control.

6) All system integration functions are performed using ISO 9000 or equivalent procedures.

Annex E
(informative)

Example of development of externally configured diagnostics for a safety-configured PE logic solver

Proven-in-use PE logic solvers should demonstrate sufficient diagnostics in the PE logic solver design. The diagnostics can be software or hardware based and should cover the entire logic solver, including input modules, main processor, output modules, and communications.

Following is a scheme that may be used to provide diagnostics for safety configured PE logic solvers.

E.1 Internally configured diagnostics

Industrial process sector PE logic solvers have internally configured diagnostics. They are referred to as internal watchdog timers (IWDT) in this annex. IWDTs include software, hardware, and communication diagnostic subsystems provided by the manufacturer, within the PE logic solver.

PE logic solvers for SIF applications should provide diagnostics for all elements of the PE logic solver. An IWDT system may provide user selectable options ranging from the shutdown of an input or output card to total shutdown of the system. IWDT diagnostics check items the logic solver manufacturer considers most important. The limitations of an IWDT may include:

- potential common mode failure in which the IWDT fails due to the same cause as the logic solver, resulting in the inability of the IWDT to perform its diagnostic functions;
- implementation may not provide the user with diagnostic information related to the logic solver fault status;
- inability to monitor the entire PE logic solver, including I/O, main processors, and communications;
- inability to monitor the application software modules and execution.

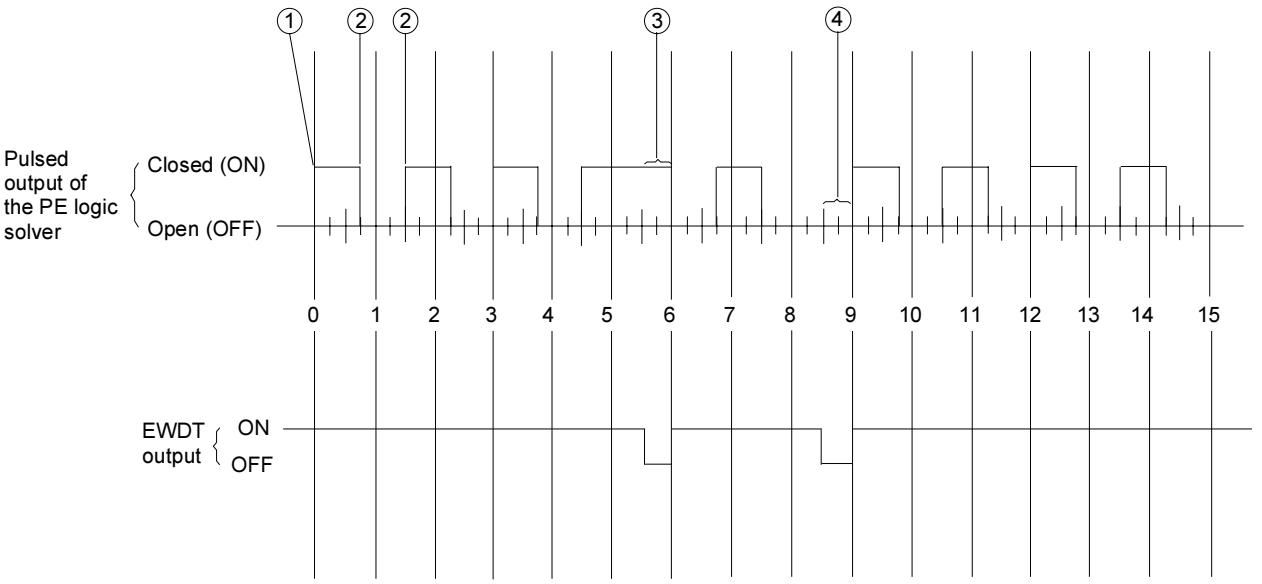
E.2 Externally configured diagnostics

- The limitations inherent in IWDTs may require the addition of external watchdog timers (EWDTs) for PE logic solvers performing safety instrumented functions. The use of EWDTs in no way eliminates the need for IWDTs for safety instrumented functions.
- Examples of EWDT devices frequently used are a rotopulsor monitor or an electronic timing monitor. In its most basic form, the EWDT is continuously pulsed by application logic located in the PE logic solver application software. The concept generally employed is to program several groups of instructions (that are widely separated in key memory locations) to generate a square wave with a desired period. This square wave is used as the input to the EWDT. Figure E.1 is a timing diagram that shows the pulsed output of the PE logic solver and the output of the EWDT.
- This square wave drives a PE logic solver output on and off in the correct timing sequence that keeps the EWDT output energized. Note that the EWDT typically has built-in adjustable ON-delay and OFF-delay timer functions. The ON-delay and OFF-delay timer settings of the EWDT are set so that neither delay should time out. If the EWDT times out, the EWDT output drops out, and the SIF may be shut down and/or alarmed. The pulses in this square wave can be varied by changing the application program in the square wave generator.

- Additional design features to be considered when implementing EWDT diagnostics include:
  - PE logic solver square wave generation for the EWDT utilizes the same instruction set used in the SIF application software;
  - Dedicated PE logic solver inputs to monitor the state of the logic solver input(s) buses to detect abnormal operation;
  - Distribution of the EWDT program across various memory locations of the PE logic solver that will best monitor total memory functionality.
  - Transmission of the generated square wave throughout the PE logic solver communication system to improve PE logic solver communication diagnostics.
- The possible need for reset buttons. A reset button will be required if the EWDT is interlocked down at start-up or upon shutdown. Consider both the EWDT and IWDT when developing the reset circuit;
- The possible need for test buttons. A test button may be desirable to verify EWDT functionality;
- Dedicated PE logic solver outputs to monitor the state of the PE logic solver output(s) buses to detect abnormal operation;
- A surge suppressor to dampen the inductive interaction to the electronics from any electro-mechanical relay contact. Review the application for additional power line conditioning requirements such as:
  - undervoltage protection;
  - electrical noise suppression;
  - lightning protection;
  - alarm development so that either EWDT and IWDT initiation can be determined.

E.3 Reference

CCPS, "*Guidelines for Safe Automation of Chemical Processes*", AIChE, 345 East 47th Street, New York, New York 10017, ISBN 0-8169-0554-1, 1993.



IEC 1832/03

Key

① Closing the control circuit energizes the output.

② Opening and reclosing the control circuit before the set time interval (assume set at 1 second) is complete keeps the EWDT output energized. The output remains energized as long as the monitored pulsing continues to provide at least 1 transition per set time interval.

③ If the monitored control stays on longer than the preset time (③), the EWDT output de-energizes.

④ If the monitored control stays off longer than the preset time (④), the EWDT output de-energizes.

Figure E.1 – EWDT timing diagram

NOTES

Standards Australia

Standards Australia is an independent company, limited by guarantee, which prepares and publishes most of the voluntary technical and commercial standards used in Australia. These standards are developed through an open process of consultation and consensus, in which all interested parties are invited to participate. Through a Memorandum of Understanding with the Commonwealth government, Standards Australia is recognized as Australia's peak national standards body. For further information on Standards Australia visit us at

www.standards.org.au

Australian Standards

Australian Standards are prepared by committees of experts from industry, governments, consumers and other relevant sectors. The requirements or recommendations contained in published Standards are a consensus of the views of representative interests and also take account of comments received from other sources. They reflect the latest scientific and industry experience. Australian Standards are kept under continuous review after publication and are updated regularly to take account of changing technology.

International Involvement

Standards Australia is responsible for ensuring that the Australian viewpoint is considered in the formulation of international Standards and that the latest international experience is incorporated in national Standards. This role is vital in assisting local industry to compete in international markets. Standards Australia represents Australia at both ISO (The International Organization for Standardization) and the International Electrotechnical Commission (IEC).

Electronic Standards

All Australian Standards are available in electronic editions, either downloaded individually from our web site, or via On-Line and DVD subscription services. For more information phone 1300 65 46 46 or visit Standards Web Shop at

www.standards.com.au

GPO Box 5420 Sydney NSW 2001
Administration Phone (02) 8206 6000 Fax (02) 8206 6001 Email mail@standards.com.au
Customer Service Phone 1300 65 46 46 Fax 1300 65 49 49 Email sales@standards.com.au
Internet www.standards.org.au

ISBN 0 7337 5914 9

Printed in Australia

This page has been left intentionally blank.