UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

JOSEANE VASCONCELOS RONDON BARRIOS

Documentação para a certificação de dispositivos de segurança usando a norma IEC 61508

Trabalho de Graduação

Prof. Dra. Taisy Weber
Orientadora

Porto Alegre, dezembro de 2010.

AGRADECIMENTOS

Dedico este trabalho ao meu esposo, Maurício da Silva Barrios, que sempre esteve ao meu lado, e isso é tudo o que eu preciso.

Agradeço ao meus pais, Paulo e Fátima, que sempre tiveram orgulho de mim, não importando o que eu fizesse.

Agradeço à minha orientadora, Prof. Dra.Taisy Weber, pela generosidade, paciência e por acreditar em mim.

Agradeço ao meus verdadeiros amigos, que souberam cumprir o seu papel nos bons e maus momentos.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

ASIC	Application-specific integrated circuit
E/E/EP	Elétrico/ Eletrônico / Eletrônico programável
EUC	Equipment Under Control
IEC	International Electrotechnical Comission
PES	Programmable Electronic System
PLC	Programmable logic controller
SIL	Safety Integrity Levels
SoC	System on a chip

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

Este trabalho é um estudo sobre a norma IEC 61508 - segurança funcional de sistemas elétricos, eletrônicos e eletrônicos programáveis relacionados à segurança. O trabalho foca no projeto de desenvolvimento do software de segurança, detalhado na norma 61508-3 - segurança funcional de sistemas elétricos, eletrônicos e eletrônicos programáveis relacionados à segurança: requisitos de software; visando o aprofundamento no estudo da documentação exigida para a certificação na norma. Para o entendimento da norma, alguns conceitos relacionados à segurança e de projeto de desenvolvimento de softwares serão apresentados nesse trabalho.

A certificação de dispositivos de segurança crítica é necessária nas mais diversas aplicações, sendo a norma IEC 61508 largamente aplicada na indústria de software para esses dispositivos e aplicações, e ainda, servindo como base para outras normas. Basicamente, a avaliação de dispositivos de segurança é realizada através da documentação apresentada. Este trabalho tem como principal objetivo esclarecer como a documentação deve ser produzida, estruturada, organizada e apresentada.

Palavras-Chave: IEC 61508, documentação, norma, segurança, software.

ABSTRACT

This work is a study about IEC 61508 - Functional safety of electrical, electronic and programmable electronic safety-related systems. The work focuses on the design of software safety development, detailed in the standard 61508-3 - Functional safety of electrical, electronic and programmable electronic safety-related systems: software requirements, aimed at deepening the study of the documentation required for certification in the standard. To understand the standard, some concepts related to safety and software development project will be presented here.

The certification of safety-critical devices is needed in several applications and the IEC 61508 standard is widely applied in the software industry for such devices and applications, being a basis for other standards. Basically, the evaluation of safety devices is accomplished through documentation provided. This work has as main objective to explain how the documentation should be produced, structured, organized and presented.

Keywords: IEC 61508, documentation, standard, safety, software.

1 CONTEXTUALIZAÇÃO

Na década de 80, muitas empresas começaram a desenvolver normas de segurança para softwares direcionadas aos mais diversos setores. A IEC (International Electrotechnical Commission) criou em setembro de 1985 a IEC 61508, uma norma internacional de segurança funcional genérica para dispositivos elétricos, eletrônicos e eletrônicos programáveis (E / E / EPS). Neste trabalho, será abordada com mais profundidade a parte da norma referente aos requisitos de software, a IEC 61508-3, da versão 1.0.

A norma foi criada devido a necessidade de se garantir que os dispositivos eletrônicos relacionados à segurança fossem realmente seguros. Infelizmente, trantando-se de software, tal garantia é impossível, visto que não é viável testar todas as possibilidades de falhas. O que é possível se alcançar é uma taxa aceitável de risco de falhas, que na norma é garantida através do cumprimento de métricas específicas, que englobam diferentes tipos de testes e avaliações. Por exemplo: é possível sentir-se relativamente seguro numa viagem de avião, mas sabe-se que aviões podem ocasionalmente cair. O risco de acidentes de aviões é aceitável, visto que são raros, mas não é possível eliminar por completo o risco de ocorrerem.

Para que seja possível avaliar a segurança de sistemas E / E / EP, deve-se compreender os conceitos que caracterizam um sistema seguro. Neste trabalho, o conceito de segurança é oriundo da palavra *safety,* "que caracteriza a segurança de funcionamento em situações críticas" (SIQUEIRA, 2006). Outro conceito importante é o de perigo, oriundo da palavra *hazard*, que significa uma fonte potencial de dano. Dano, neste caso, se origina da palavavra *harm*, que significa lesão física ou dano à saúde das pessoas, quer direta, ou indiretamente, como resultado de avaria à propriedade ou ao meio ambiente (LADKIN, 2008).

Uma aplicação muito importante da norma diz respeito a sistemas embarcados de segurança crítica. Um sistema embarcado é basicamente um dispositivo que foi desenvolvido para uma aplicação específica, com funções bem definidas e que geralmente é acoplado a um sistema externo. Um sistema embarcado de segurança crítica deve ser considerado altamente seguro, já que sua aplicação envolve grande responsabilidade. Exemplos de sistemas embarcados de segurança crítica:

- aviônicos, como sistemas de controle inercial, controle de vôo e outros sistemas integrados nas aeronaves, como sistemas de orientação de mísseis;
- controladores de tração, motor e antibloqueio em automóveis; freios ABS;
- equipamentos de suporte a vida em hospitais;

- urna eletrônica;
- sistemas distribuídos de controle de tráfego;
- sistemas de proteção e parada de emergência em máquinas e equipamentos.

De acordo com a norma IEC 61508, um sistema é considerado seguro se conseguiu atingir todos os requisitos necessários, seguindo as métricas exigidas conforme o nível de integridade de segurança (SIL) exigido para a aplicação em questão. Essa análise é baseada na documentação apresentada, onde deverão estar especificados todos os resultados atingidos, para que a segurança do sistema possa ser avaliada.

A documentação deve prover as informações necessárias para que todas as fases dos ciclos de vida possam ser efetivadas, além da gerência, verificação e avaliação da segurança funcional. A documentação deve ser correta, concisa, de fácil entendimento pelos envolvidos no projeto, acessível e manutenível. A sua estrutura pode levar em consideração procedimentos das empresas e as práticas de trabalho de setores de aplicação específicos (SIQUEIRA, 2006). Detalhes sobre a documentação serão apresentados no capítulo 5.

2 CONCEITOS PARA O ENTENDIMENTO DA NORMA IEC 61508

Para que se possa entender o funcionamento da norma IEC 61508, é necessário que alguns conceitos sejam compreendidos plenamente. Neste capítulo, serão apresentados os seguintes conceitos:

- segurança;
- segurança funcional;
- abordagem baseada em risco;
- nível de integridade de segurança.

2.1 O que é um sistema seguro?

Para a norma internacional IEC 61508, um sistema seguro é um sistema livre de riscos inaceitáveis, envolvendo prejuízos físicos ou danos à saúde de pessoas, resultantes direta ou indiretamente de danos a propriedade ou ao ambiente.

O termo segurança é usado para descrever sistemas que irão realizar uma função ou funções específicas que vão garantir que os riscos serão mantidos num nível aceitável. Tais funções são chamadas de funções de segurança.

Para Dunn (2003), a segurança falha na prática por três principais motivos - seus criadores ou usuários:

- têm uma compreensão incompleta do que faz um sistema "seguro";
- deixam de considerar o sistema maior em que o conceito implementado será embarcado, ou
- ignoram pontos únicos de falha, o que vai tornar o conceito de segurança não seguro quando posto em prática.

2.2 Segurança funcional

A segurança funcional é realizada pelo conjunto de funções de segurança do sistema. Segurança funcional faz parte da segurança global, que depende de um sistema ou equipamento operando corretamente em resposta às suas entradas.

"Nem segurança nem a segurança funcional podem ser determinadas sem considerar o sistema como um todo e o ambiente com o qual eles interagem" (BELL, 2005).

Dois tipos de requisitos são necessários para alcançar a segurança funcional:

- requisitos da função de segurança (o que faz a função, o que ela deve realizar) e
- requisitos de integridade de segurança (a probabilidade de uma função de segurança estar sendo executada de forma satisfatória, ou seja, grau de certeza necessário de que a função de segurança será cumprida).

Os requisitos da função de segurança são derivadas da análise de perigo e os requisitos de integridade de segurança são derivados da avaliação de riscos. Quanto maior o nível de integridade de segurança, menor é a probabilidade de falha perigosa. Os conceitos de integridade de segurança, avaliação de riscos e análise de perigo serão explicados mais adiante.

Segurança funcional é apenas um dos métodos existentes para lidar com o perigo. Há outros meios para eliminar ou reduzir riscos, como a segurança inerente através do projeto, que é de primordial importância.

Qualquer sistema implementado em qualquer tecnologia que realiza funções de segurança é um sistema relacionado à segurança. O sistema relacionado à segurança pode ser separado de qualquer sistema / equipamento de controle ou pode ser acoplado / embarcado a ele. Níveis mais elevados de integridade de segurança exigem maior rigor na engenharia do sistema de segurança.

2.3 Abordagem baseada em risco

A norma IEC 61508 utiliza análise baseada em risco para determinar o nível de integridade de segurança.

Para sistemas relacionados à segurança, perigo (hazard) é o elemento que conecta uma falha no sistema a um subsequente infortúnio, definido como qualquer condição real ou potencial que possa causar

- lesão, doença ou morte de indivíduos;
- danos ou perdas de um sistema, equipamento ou propriedade, ou
- danos ao meio ambiente.

Exemplos de perigo incluem a perda de controle de vôo, refrigeração nuclear central, ou a presença de material tóxico ou gás natural.

Basicamente, a análise de perigo consiste em identificar como cada componente da aplicação pode falhar, realizando análise dos modos de falha para descobrir todas as possíveis fontes de falhas em cada componente. Estes incluem falhas aleatórias de hardware, defeitos de fabricação, falhas de programação, o stress ambiental, erros de projeto e erros de manutenção. Estas análises fornecem informações para estabelecer

uma ligação entre todos os possíveis modos de falhas em componentes e infortúnios/acidentes (DUNN, 2003).

O risco de acidente ou infortúnio avalia o impacto de um acidente ou infortúnio em termos de duas preocupações principais: potencial gravidade e a probabilidade de sua ocorrência.

Geralmente a população estabelece um risco aceitável para um certo tipo de infortúnio ou acidente, desde que eles ocorram com pouca freqüência. Estatísticas de vários infortúnios comuns e sua freqüência média representam risco aceitável , podendo variar de 10^-2 a 10^-10 incidentes por hora.

A avaliação de risco determina os requisitos de desempenho da função de segurança. O objetivo é garantir que a integridade de segurança da função de segurança é suficiente para garantir que ninguém seja exposto a um risco inaceitável associado a evento perigoso.

As normas de segurança, como por exemplo a norma IEC 61508, tratam de determinar o que constitui um risco aceitável. No caso da IEC, é utilizado o nível de integridade de segurança (SIL).

2.4 Níveis de Integridade da Segurança (SIL)

Nível de integridade de segurança ou SIL é definido como um nível relativo de redução de riscos propiciados por uma função de segurança, ou especifica um nível de meta de redução de riscos. Para o entendimento da norma, SIL é uma medida de desempenho de segurança exigida para a certificação de acordo com o nível de segurança que a aplicação visa atender.

A IEC 61508 especifica quatro níveis de integridade de segurança. O SIL (1) é o mais baixo nível de integridade de segurança e o SIL(4) é o mais alto nível de integridade de segurança. O nível de integridade de segurança aumenta com a gravidade do dano e a frequência de exposição ao perigo. A norma detalha os requisitos necessários para atingir cada nível de integridade de segurança. Estes requisitos são mais rigorosos com maiores níveis de integridade de segurança a fim de alcançar a menor probabilidade exigida de falhas perigosas. Os requisistos são avaliados através da aplicação de métricas, de acordo com o SIL pretendido. A IEC 61508 utiliza três principais classificações para utilização de métricas: altamente recomendada (HR), recomendado (R) ou não recomendado (NR). A não utilização de uma métrica HR deve ser justificada - mas nenhuma idéia do que é válido como justificativa é dada.

A integridade de segurança exigida em sistemas E / E / EP relacionados à segurança, em relação a uma função de segurança específica, deve ser de um tal nível, para garantir que:

- a frequência de falhas em sistemas relacionados à segurança é suficientemente baixa para evitar que a frequência de eventos perigosos seja superior ao risco tolerável, e / ou;

- os sistemas de segurança modifiquem as consequências de falhas num nível necessário para atender ao risco tolerável (BELL, 2005).

Nas tabelas a seguir, seguem exemplos de requisitos de níveis de integridade exigidos para projetos de E/E/EP relacionados à segurança.

Tabela 2.1: Nível de integridade de segurança: medidas de falhas para funções de segurança que operam em modo de funcionamento de baixa demanda (onde a frequência da demanda por operação realizada por um software relacionado à segurança não é superior a um por ano e não superior a duas vezes ao proof-test frequency).

Nível de Integridade de Segurança (SIL)	**Modo de baixa demanda de operação** (Probabilidade média de falhas em executar a função para qual foi projetado)
4	$\geq 10^{-5}$ to $< 10^{-4}$
3	$\geq 10^{-4}$ to $< 10^{-3}$
2	$\geq 10^{-3}$ to $< 10^{-2}$
1	$\geq 10^{-2}$ to $< 10^{-1}$

Fonte: BELL, 2005

Tabela 2.2: Nível de integridade de segurança: medidas de falhas para funções de segurança que operam em modo de funcionamento de alta demanda ou o modo contínuo de operação (quando a freqüência de demandas realizadas por um software relacionado à segurança é superior a um por ano ou superior a duas vezes ao proof-check frequency).

Nível de Integridade de Segurança (SIL)	**Alta demanda de operação** (probabilidade de falha perigosa por hora)
4	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-6}$ to $< 10^{-5}$

Fonte: BELL, 2005

2.5 Sistemas relacionados à segurança e exemplos

Um sistema relacionado à segurança é um sistema que é capaz de executar os requisitos especificados em cada função de segurança e de realizá-los de acordo com a integridade de segurança exigida. Os requisitos da integridade da segurança da função de segurança definem os requisitos de integridade de segurança do sistema relacionado à segurança (BELL, 2005).

Normalmente, virtualmente qualquer sistema de computador, seja ele um controlador fly-by-wire de aeronaves, um robô industrial, uma máquina de radioterapia, ou um sistema automotivo antiderrapante contém cinco componentes primários. De acordo com Dunn (2003), os componentes são o seguintes:

A *aplicação (ou planta ou processo)* é a entidade física que o sistema monitora e controla. Aplicações típicas incluem uma aeronave em vôo, o braço de um robô, um paciente humano, um freio de automóvel.

O *sensor* converte uma propriedade física da aplicação avaliada ou medida em um sinal elétrico correspondente para uma entrada no computador. Sensores típicos incluem acelerômetros, transdutores de pressão e medidores de tensão.

O a*tuador* (ou efetor ou elemento final) converte um sinal elétrico a partir da saída do computador para uma ação física correspondente que controla uma função da aplicação. Atuadores típicos incluem motores, válvulas, mecanismos de freio e bombas.

O *operador* é o ser humano que acompanha e ativa o sistema de computador em tempo real. Operadores típicos incluem piloto de avião, operador de usina, técnico de saúde.

O *computador* é composto de hardware e software que utilizam sensores e atuadores para monitorar e controlar a aplicação em tempo real. O computador vem em muitas formas, tais como um controlador de placa única, controlador lógico programável, computador de vôo ou um sistema em um chip. Muitos sistemas de computadores, tais como aqueles usados na indústria para controle de supervisão e aquisição de dados, são constituídos por redes complexas construídas a partir destes componentes básicos.

São exemplos de sistemas E / E / EPs relacionados à segurança:

- de parada de emergência do sistema em uma fábrica de processos químicos perigosos;
- indicador de carga de segurança em guindastes;
- sistema de sinalização ferroviária;

- sistema de bloqueio de proteção e sistemas de parada de emergência para uma máquina;
- unidade de velocidade variável num motor usado para restringir a velocidade como um meio de proteção;
- sistema de bloqueio e controle da dose de exposição de uma máquina de radioterapia em medicina;
- de posicionamento dinâmico (controle de movimento de um navio quando em proximidade com uma instalação offshore);
- operação fly-by-wire de controle de vôo de aeronaves em superfícies;
- luzes indicadoras de automóveis, anti-bloqueio de trava e de sistemas de engenharia do motor;
- monitoramento remoto, operação ou programação de uma rede habilitada em processos de fábricas ;
- uma ferramenta de apoio à decisão baseada em informações onde resultados errôneos afetam a segurança.

3 A NORMA INTERNACIONAL IEC 61508

A norma internacional IEC 61508, segurança funcional para sistemas de segurança de elétricos / eletrônicos / eletrônicos programáveis (E /E/ EP), tem por finalidade:

- aprimorar o potencial de tecnologias de E/E/EP a fim de melhorar o desempenho em relação à segurança e economia;
- permitir a evolução tecnológica para que esta tenha destaque num quadro global de segurança;
- proporcionar um sistema de abordagem tecnicamente sólido, com flexibilidade suficiente para o futuro;
- proporcionar uma abordagem baseada no risco para determinar o desempenho exigido dos sistemas relacionados à segurança;
- fornecer um padrão genérico que pode ser usado diretamente pela indústria, mas que também possa ajudar com o desenvolvimento de normas de outros setores (por exemplo, máquinas, processos químicos, médicos ou ferroviário) ou normas de produtos (por exemplo, sistemas de fornecimento de energia);
- proporcionar um meio para que os usuários e operadores possam ganhar confiança quando utilizarem tecnologias baseadas em computador;
- estabelecer requisitos com base em princípios comuns para facilitar:

  - uma maior eficiência na cadeia de abastecimento para os fornecedores de subsistemas e componentes para vários setores;

  - melhorias na comunicação e nos requisitos (ou seja, para aumentar a clareza do que precisa ser especificado);

  - desenvolvimento de técnicas e medidas que poderiam ser usados em todos os setores, aumentando os recursos disponíveis;

  - o desenvolvimento de serviços de avaliação da conformidade, se necessário.

3.1 Características gerais da norma IEC 61508:

- usa uma abordagem baseada no risco para determinar os requisitos de integridade de segurança de sistemas E / E / EP baseados em segurança e inclui uma série de exemplos de como isso pode ser feito;
- usa um modelo de ciclo de vida total como um framework técnico para as que as atividades necessárias para garantir a segurança funcional sejam alcançadas pelos sistemas E / E / EP relacionados à segurança;
- abrange todas as atividades de segurança do ciclo de vida desde o conceito inicial, através de análise de perigos e avaliação de riscos, desenvolvimento das prescrições de segurança, especificação, projeto e implementação, operação e manutenção, modificação, até o final da desativaçao e / ou eliminação;
- engloba aspectos do sistema (abrangendo todos os subsistemas que executam as funções de segurança, incluindo hardware e software) e mecanismos de falha (hardware aleatórios e sistemáticos);
- contém os requisitos para a prevenção de falhas (evitando a introdução de faltas) e os requisitos para as falhas de controle (garantir a segurança, mesmo quando há presença de falhas);
- especifica as técnicas e medidas que são necessárias para atingir a integridade de segurança exigidos.

3.2 Partes do framework da IEC 61508

IEC 61508, intitulado segurança funcional dos sistemas elétricos / eletrônicos / eletrônicos programáveis relativos à segurança, consiste em 7 partes (ver figura 3.1):

- IEC 61508-1, requisitos gerais;
- IEC 61508-2, os requisitos para sistemas elétricos / eletrônicos / eletrônicos programáveis relacionados à segurança;
- IEC 61508-3, os requisitos de software;
- IEC 61508-4, definições e abreviaturas;
- IEC 61508-5, exemplos de métodos para determinação dos níveis de integridade de segurança;
- IEC 61508-6, orientações sobre a aplicação da IEC 61508-2 e IEC 61508-3;
- IEC 61508-7, revisão de medidas e técnicas.

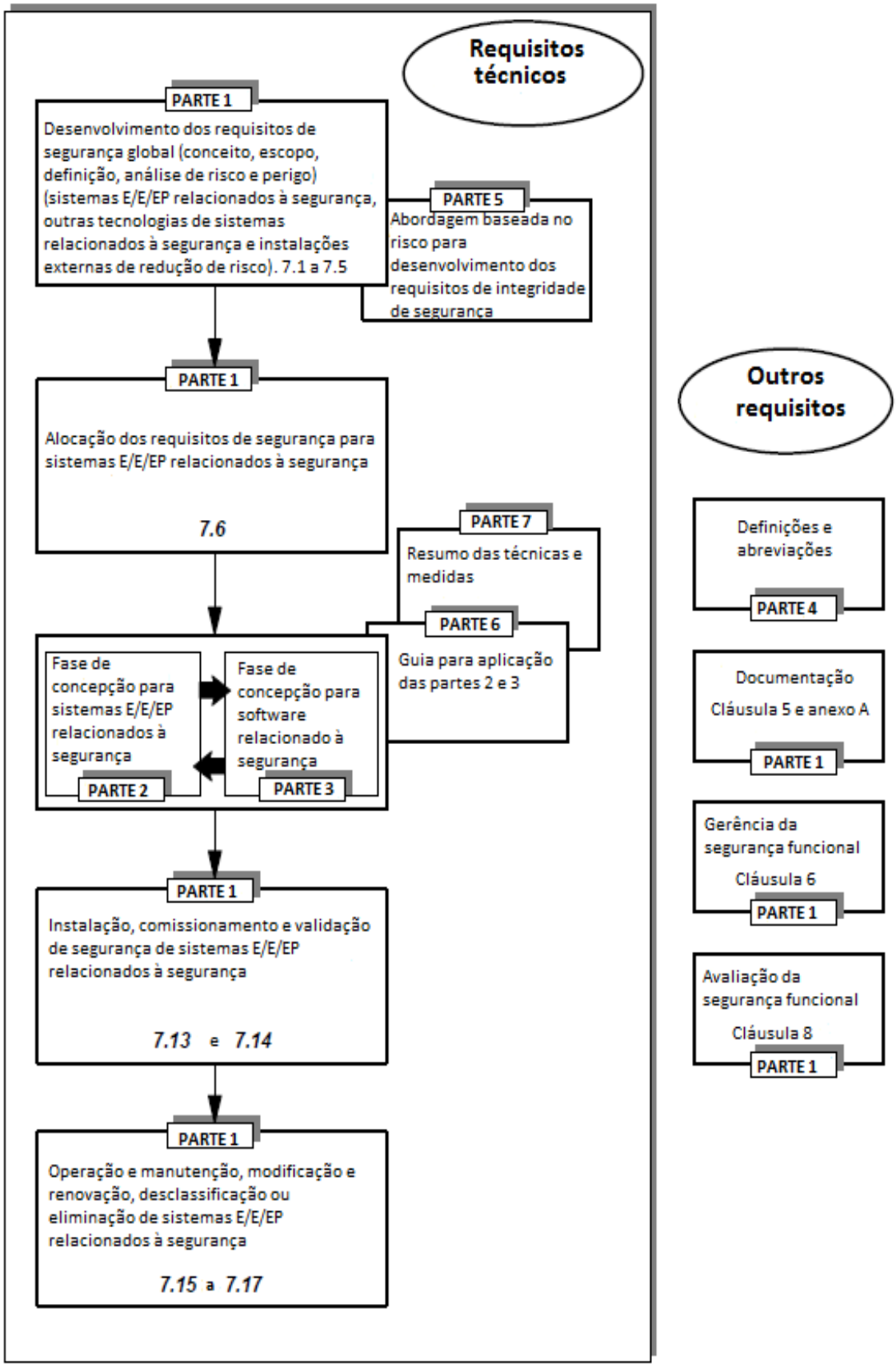


Figura 3.1: Mapa de requisitos para as partes 1 a 7 da IEC 61508

3.3 Críticas à norma

Existem diversos trabalhos que questionam a aplicação da norma, no que diz respeito a segurança e confiabilidade da aplicação do SIL como medida confiável. A norma também é muito ampla e de difícil compreensão, levando a interpretações ambíguas em muitos casos. Outro problema é a restrição de métricas largamente difundidas na engenharia de software. A norma não recomenda a utilização de programação orientada a objetos assim como uso inteligência artificial. A norma também recomenda o uso de linguagens fortemente tipadas, restringindo muito as opções do desenvolvedor. Ao estudar as métricas, nota-se uma forte ênfase na parte formal, o que pode trazer muitas dificuldades aos desenvolvedores e projetistas, caso não possuam uma boa formação em informática teórica.

É possível a aplicação de práticas que não estejam recomendadas na norma, mediante detalhada justificativa baseada em engenharia de software. O grande problema é que não se tem conhecimento sobre o tipo de justificativa válida perante os avaliadores, já que a norma é extremamente vaga nesse aspecto.

As normas variam em seus detalhes, mas a maioria adota uma abordagem semelhante para lidar com a segurança - por recomendar ou prescrever processos de desenvolvimento e métodos. As normas contêm uma série de conselhos sensatos e orientações (embora algumas partes sejam questionáveis), embora não realmente sobre segurança. As normas são efetivamente preocupados com a qualidade e repetibilidade – ambos objetivos louváveis. No entanto, ao tentar ler a norma DS 00-55 e a Parte 3 da norma IEC 61508, e tentar identificar as técnicas que se centram sobre segurança - ou mesmo a palavra segurança - parece que há baixa correlação entre as normas e as taxas de falhas perigosas observadas, simplesmente porque as normas não contemplam as questões de segurança. Isto, naturalmente, levanta a questão de como produzir normas que sejam mais preocupados com a segurança (MCDERMID, 2001).

Há também vários problemas inerentes à utilização de níveis de integridade de segurança. Estes podem ser resumidos da seguinte forma:

- pobre harmonização da definição nas diferentes normas que utilizam SIL;
- métricas orientadas a processo para a definição do SIL;
- estimativa do SIL com base em estimativas de confiabilidade;
- a complexidade dos sistemas, especialmente em sistemas de software, permitindo estimativas de SIL difíceis, quase impossíveis.

Estes problemas conduzem à afirmações errôneas como, “este sistema é um sistema SIL N porque o processo da norma usado durante o desenvolvimento foi o processo para desenvolvimento de sistemas SIL N”, ou o uso do conceito de SIL fora de contexto como, “este é um dispositivo SIL 3”. De acordo com a IEC 61508, o conceito de SIL está relacionado à taxa de falhas perigosas, e não somenta a taxa de falhas. A definição de modo de falha perigosa por análise de segurança é intrínsica a determinação correta de taxa de falhas.

Em geral, a IEC 61508 e as normas derivadas são volumosas e bem difíceis de se ler e interpretar. Muitos requisitos não são atribuídos a uma determinada faixa de níveis de integridade de segurança ou à complexidade do projeto. Isso torna difícil se adequar a projetos menores e faz a gestão de segurança funcional ser muito cara para as pequenas e médias empresas. Na tabela 3.1 estão listados os preços de algumas normas da IEC, consultados no site da IEC.

Tabela 3.1: Valores de algumas normas da IEC

Norma da IEC	**Preço em Reais (R$)**
IEC 61508-1 ed 2.0 (2010-04) : Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 1: General requirements	452,05
IEC 61508-2 ed 2.0 (2010-04): Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems	481,16
IEC 61508-3 ed 2.0 (2010-04): Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 3: Software requirements	489,72
IEC 61508-4 ed 2.0 (2010-04): Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 4: Definitions and abbreviations	282,53
IEC 61508-5 ed 2.0 (2010-04): Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 5: Examples of methods for the determination of safety integrity levels	349,31
IEC 61508-6 ed 2.0 (2010-04): Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3	489,72
IEC 61508-7 ed 2.0 (2010-04): Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 7: Overview of techniques and measures	508,55
IEC 61506 ed 1.0 (1997 -02): Industrial-process measurement and control - Documentation of application software	405,82
IEC 61511-1 ed1.0 (2003-01) : Functional safety - Safety instrumented systems for the process industry sector - Part 1: Framework, definitions, system, hardware and software requirements	470,88
ISO/IEC 1476 ed1.0 (2009-01): Information technology - Functional safety requirements for home and building electronic systems (HBES)	210,61
IEC 61355-1 ed 2.0 (2008-04): Classification and designation of documents for plants, systems and equipment - Part 1: Rules and classification tables	349,31
S+ IEC 61508 ed2.0 (2010-04): Functional safety of electrical/electronic/programmable electronic safety-related systems - ALL PARTS together with a commented Redline version	4366,37

A possível ambiguidade na interpretação encoraja muitos a usar o padrão como uma toolbox onde podem tirar, requisitar ou implementar o que eles bem entenderem ou preferirem. Considerando que na América do Norte, os usuários parecem estar sobretudo preocupados com a integridade de segurança do hardware (taxas de falhas

perigosas e fração de falhas seguras), na Europa muitos usuários parecem estar mais preocupados com a integridade da segurança do software. A norma também deixa muito espaço para interpretação. Na Europa e na Alemanha, a maioria dos especialistas interpreta os requisitos das restrições de arquitetura de hardware e diagnóstico da parte 2 da norma diferente do que os especialistas americanos o fazem, levando a situações descritas no parágrafo acima. Normas do setor da indústria européia, tais como a EN 954-1 e EN 298 não aceitam sistemas onde o modo de falha de um único componente pode levar a um estado inseguro enquanto IEC 61508 aceita (FALLER, 2001).

A IEC 61508 também não abrange precauções que poderiam ser necessárias para impedir que pessoas não autorizadas cometam danos e/ou de prejudiquem a segurança funcional alcançada pelo sistema E / E / EP relacionado à segurança.

3.3.1 Observações sobre a documentação

As informações sobre a documentação contidas na norma são relativamente vagas. Dentro da parte IEC 61508-1, no anexo C, encontram-se referências à duas normas diferentes da IEC, que são as seguintes:

- IEC 61506, Industrial-process measurement and control – Documentation of application software (Medição e controle de processo industrial - Documentação de software de aplicação);
- IEC 61355, Classification and designation of documentation for plants, systems and equipament (Classificação e designação de documentação para fábricas, sistemas e equipamentos).

Logo na introdução da norma IEC 61506, é explicado que esta norma não se preocupa com a documentação do software do sistema de computador, exceto em sua interface com a função de controle de processo de software. A norma IEC 61506 diz claramente que está preocupada com um nível mais alto na estrutura de software; o nível de aplicação do software.

Com essas afirmações, pode-se concluir que a norma IEC 61508 não é autocontida e que não oferece suporte à documentação em todas as fases do ciclo de vida de software. Na norma também não é possível encontrar informações relativas ao tipo de justificativas que seriam realmente satisfatórias aos examinadores, no caso da não utilização de técnicas recomendadas ou altamente recomendadas.

3.4 Revisão da norma IEC 61508

Uma nova versão da norma era esperada para maio de 2008, de acordo com dados divulgados na página da IEC.

As equipes de manutenção estariam considerando várias questões, de acordo com Bell (2005), incluindo:

- *Clareza dos requisitos*: A necessidade de fazer mais claras as exigências relacionadas com o cumprimento elementos. O conceito de "capacidade de SIL" será proposto para tratar dos aspectos sistemáticos. Espera-se que essa alteração beneficie os fabricantes dos subsistemas.
- *Os dispositivos programáveis como ASICS:* Propostas que abrangem ASICS serão incluídos no projeto.
- *Segurança*: Atualmente, a norma não cobre explicitamente as considerações de segurança (safety).
- *Prova-em-uso*: A norma abrange este conceito, mas está sendo revista e mais desenvolvimento está sendo considerado.
- *Comunicação digital*: As atuais exigências do padrão serão clarificadas e mais elaboradas.

Como parte de ciclo de vida normal de manutenção da IEC, a norma foi revisada. As alterações foram baseadas em comentários recebidos pelos Comitês Nacionais. Dois times internacionais de manutenção foram responsáveis pela revisão da IEC 61508.

A versão 2.0 acabou sendo lançada em abril de 2010 e não será lançada uma nova edição antes de 2014.

4 CRITÉRIOS PARA SEGURANÇA DE SOFTWARE

4.1 Ciclo de vida de segurança de software

O ciclo de vida de segurança de software tem como objetivo estruturar o desenvolvimento de software em fases e atividades bem definidas.

O ciclo de vida demonstrado neste trabalho corresponde à figura 3 da norma IEC 61508-3. Outros modelos de ciclo de vida podem ser utilizados, desde que seja justificada a necessidade das alterações. A figura 4.1 ilustra o ciclo de vida de segurança de software.

A IEC 61508-1 considera os resultados das fases do ciclo de vida de segurança. No desenvolvimento de alguns sistemas E / E / EP relacionados a segurança, a saída de algumas fases do ciclo de vida pode ser um documento distinto, enquanto as saídas documentadas de várias fases podem ser fundidas. O requisito essencial é que a saída de cada fase do ciclo de vida de segurança esteja apta para a sua finalidade. Em projetos simples, algumas fases do ciclo de vida de segurança podem ser mescladas, de acordo com a cláusula 7.4.5.

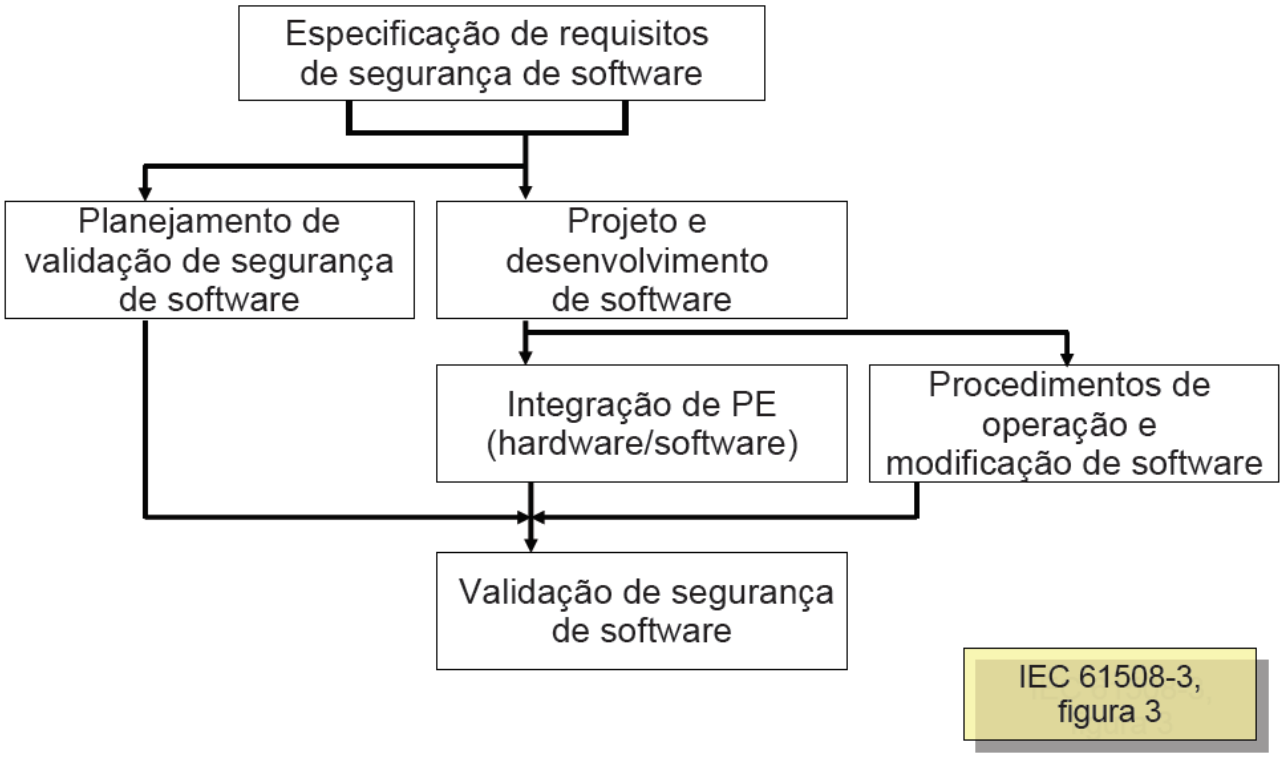


Figura 4.1: Ciclo de vida de segurança de software (WEBER).

4.1.1 Especificação dos requisitos de segurança de software – cláusula 7.2.2

Essa fase do ciclo de vida tem como objetivos:

- a especificação dos requisitos para a segurança de software em função dos requisitos para as funções de segurança do software e dos requisitos de integridade de segurança do software;
- a especificação dos requisitos para as funções do software de segurança para cada sistema E / E / EP de segurança necessárias para implementar as funções de segurança exigidos;
- a especificação dos requisitos para a integridade do software de segurança para cada sistema E / E / EP de segurança necessários para atingir o nível de integridade de segurança especificado para cada função de segurança atribuída a esse sistema E / E / EP relacionado a segurança.

4.1.2 Plano de validação de segurança de software – cláusula 7.3.2

O objetivo é desenvolver um plano de validação à segurança do software. O planejamento deve ser realizado para especificar as etapas processuais e técnicas, que serão utilizadas para demonstrar que o software satisfaz seus requisitos de segurança.

4.1.3 Projeto e desenvolvimento de software

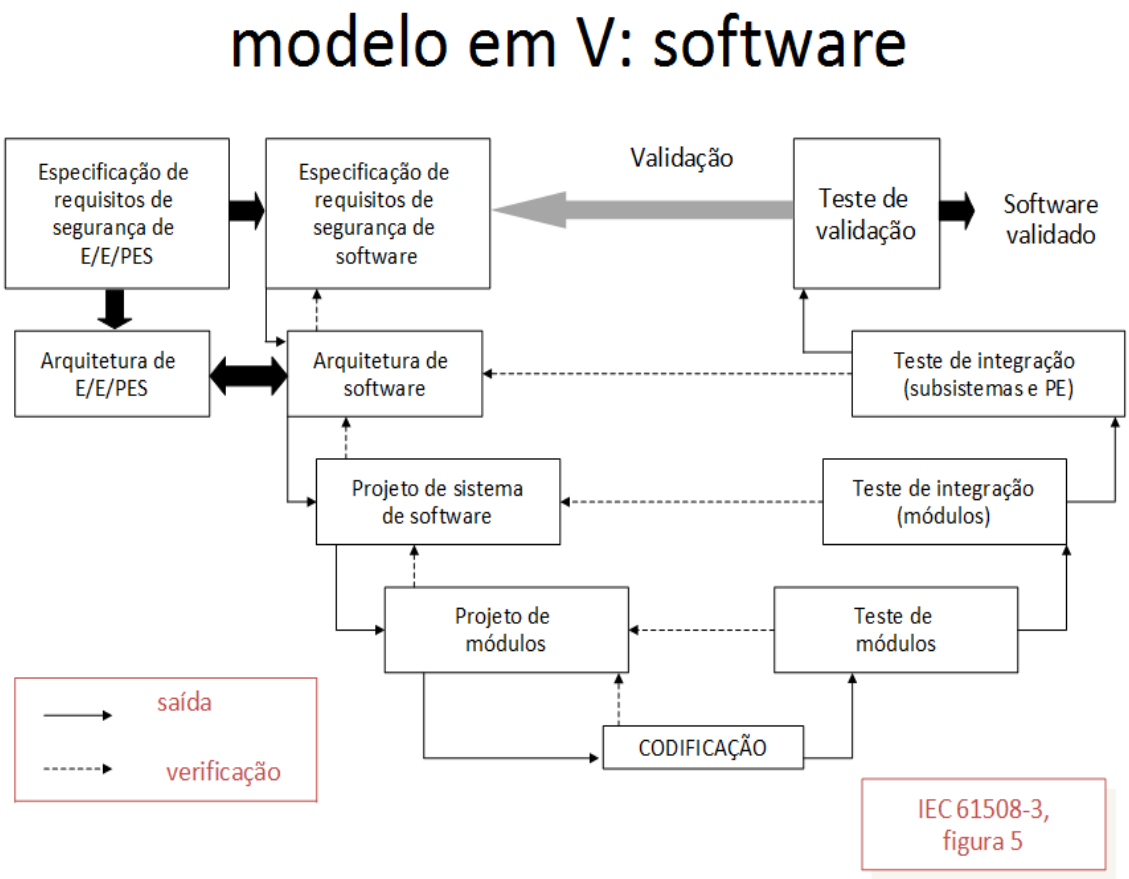


Figura 4.2: Projeto e desenvolvimento de software (WEBER).

4.1.3.1 Arquitetura – cláusula 7.4.3

Essa fase tem como finalidade criar uma arquitetura de software que satisfaça os requisitos especificados para a segurança de software (ver 7.2) com respeito ao nível de integridade de segurança exigido.

Também tem como objetivo analisar e avaliar os requisitos colocados no software pela arquitetura de hardware do sistema E / E / PE relacionado à segurança, incluindo a importância de interações entre hardware/software e E/E/PE para segurança de equipamentos sob controle (EUC).

4.1.3.2 Suporte a ferramentas e linguagens de programação – cláusula 7.4.4

Essa fase tem como finalidade selecionar um conjunto adequado de ferramentas, incluindo linguagens e compiladores, para o nível de integridade de segurança pretendido. Essas ferramentas deverão, ao longo de todo o ciclo de vida de segurança de software, auxiliar na verificação, validação, avaliação e modificação.

A seleção de ferramentas de desenvolvimento dependerá da natureza das

atividades de desenvolvimento de software e da arquitetura de software, de acordo com a cláusula 7.4.3.

4.1.3.3 Projeto e desenvolvimento detalhado (projeto do sistema de software) – cláusula 7.4.5

O objetivo desta fase é projetar e implementar um software que satisfaça os requisitos especificados para a segurança de software, de acordo com o nível de integridade de segurança exigido, o qual é analisável e verificável e que é capaz de ser modificado de forma segura.

O projeto detalhado é aqui definido no sentido de concepção do sistema de software - o particionamento de componentes maiores em uma arquitetura de sistema de módulos de software, projeto de módulo individual de software e programação.

4.1.3.4 Projeto e desenvolvimento detalhado (módulos individuais do projeto de software) – cláusula 7.4.5

Para cada subsistema na descrição do projeto de arquitetura de software, o refinamento do projeto deve ser baseado em uma divisão em módulos de software (isto é, a especificação do projeto do sistema de software). O projeto de cada módulo de software e os testes a serem aplicados a cada módulo do software deve ser especificado.

4.1.3.5 Implementação do código detalhada – cláusula 7.4.6

Essa fase visa garantir que o código-fonte:
- seja legível, compreensível e testável;
- satisfaça os requisitos especificados no projeto de módulo de software;
- satisfaça os requisitos dos padrões de codificação;
- satisfaça todas as exigências pertinentes previstas durante o planejamento de segurança.

4.1.3.6 Teste de módulos do software – cláusula 7.4.7

Essa etapa tem como objetivo verificar se os requisitos de segurança de software (em termos de funções de segurança de software exigidos e da integridade de segurança do software)- foram alcançados – a fim de mostrar que cada módulo de software executa sua função pretendida, não executando funções involuntárias. Cada módulo de software deve ser testado conforme especificado no projeto de software.

4.1.3.7 Teste de integração do software – cláusula 7.4.8

Essa etapa tem como objetivo verificar se os requisitos de segurança de software (em termos defunções de segurança de software exigidos e da integridade de segurança do software) – foram alcançados – a fim de mostrar que todos os módulos de software, componentes e subsistemas interagem corretamente para exercer a função pretendida e não desempenham funções involuntárias. Os testes de integração de software devem ser especificados simultaneamente durante o projeto e fase de desenvolvimento.

4.1.4 Integração de eletrônicos programáveis (hardware e software) – cláusula 7.5.2

Um dos objetivos dessa fase é integrar o software ao hardware eletrônico programável. Outro objetivo é unir o software e o hardware eletrônico programável para garantir sua compatibilidade e atender às exigências do nível desejado de integridade de segurança.

Os testes de integração de software devem ser especificados simultaneamente durante as fases de projeto e fase de desenvolvimento.

4.1.5 Procedimentos de operação e modificação do software - cláusula 7.6.2

Essa fase tem como objetivo fornecer informações e procedimentos relativos ao software necessários para garantir que a segurança funcional do sistema E / E / PE de segurança seja mantida durante a operação e modificação.

4.1.6 Validação de segurança em software – cláusula 7.7.2

O objetivo desta seção é garantir que o sistema integrado esteja de acordo com os requisitos especificados de segurança de software no que diz respeito ao nível de integridade de segurança.

Se a conformidade com os requisitos de segurança do software já foi estabelecida como parte do sistema E / E / PE de segurança (de acordo com cláusula 7.7 da IEC 61508-2), a validação não precisa ser repetida.

Os resultados da validação de segurança de software devem ser documentados.

4.1.7 Modificação de software – cláusula 7.8.2

Essa fase tem como objetivo fazer correções, melhorias ou adaptações para a validação do software, assegurando que nível de integridade de segurança pretendido para o software seja mantido. Na norma IEC 61508 software, ao contrário de hardware, não sofre manutenção porém, sofre modificação.

4.1.8 Verificação de software – cláusula 7.9.2

Essa etapa visa testar e avaliar os resultados de cada fase do ciclo de vida de segurança de software, para garantir a exatidão e consistência em relação aos padrões e saídas fornecidos como entrada para cada fase.

4.1.9 Avaliação do plano de segurança funcional de software - cláusula 8

A avaliação do plano de segurança funcional tem o intuito de investigar e chegar a um julgamento sobre a segurança funcional alcançada pelo sistema E / E / PE relacionado à segurança.

A seleção de técnicas de anexos A e B não garante por si só que a integridade de segurança exigida será alcançada (ver 7.1.2.6).

O avaliador deve também considerar:

- a coerência e a complementaridade dos métodos escolhidos, linguagens e ferramentas para o desenvolvimento integral do ciclo;
- se os desenvolvedores utilizam métodos, linguagens e ferramentas que compreendem;
- se os métodos, linguagens e ferramentas são bem adaptados aos problemas específicos encontrados durante o desenvolvimento.

4.2 Exemplo de aplicação de tabelas de Integridade de Segurança de Software contidas na norma IEC 61508-3

O exemplo que será descrito está contido na norma IEC 61508-6 – guia de aplicação da IEC 61508-2 e IEC 61508-3. Este exemplo fornece orientações para aplicação das tabelas de integridade de segurança de software, especificados no anexo A da norma IEC 61508-3. Nas tabelas do exemplo, pode-se observar que há uma descrição da interpretação de cada técnica de acordo com aplicação.

O primeiro exemplo é um sistema eletrônico programável com nível de integridade de segurança 2, necessário para um processo em uma usina química. O sistema eletrônico programável relacionado à segurança usa lógica ladder para o programa de aplicação, como uma ilustração de programação de aplicativo em uma linguagem de variabilidade limitada. Para um sistema real, todas as entradas nas tabelas devem ser suportadas pela justificativa documentada, confirmando que os comentários feitos estão corretos e que representaram uma resposta apropriada para o sistema e a aplicação.

O aplicativo consiste em vários tanques de reatores ligados a tanques intermediários de armazenamento, os quais são preenchidas com gás inerte em determinados pontos do ciclo da reação para suprimir combustões e explosões. As funções do sistema eletrônico programável relacionado à segurança incluem: receber entradas dos sensores; alimentação e fechamento de válvulas, bombas e atuadores; a detecção de situações perigosas e ativação de alarmes; interface com um sistema de controle distribuído, conforme exigido pela especificação dos requisitos de segurança.

Suposições:

- o sistema eletrônico programável de segurança é um controlador PLC;
- análise de perigos e riscos demostrou que um sistema eletrônico programável relacionado à segurança era necessário, e que SIL 2 seria exigido para esta aplicação (pela aplicação da norma IEC 61508-1 e IEC 61508-2);

- embora o controlador opere em tempo real, é necessário um tempo de resposta relativamente lento;
- há interfaces com operador humano e com o sistema de controle distribuído;
- o código-fonte do sistema de software e o projeto dos sistemas eletrônicos programáveis do PLC não estão disponíveis para avaliação, mas eles foram qualificados para o SIL 2 com base na IEC 61508;
- a linguagem utilizada para programação do aplicativo foi lógica ladder, produzida utilizando o sistema de desenvolvimento do fornecedor de PLC;
- o código do aplicativo é necessário para executar apenas um tipo de PLC;
- o desenvolvimento do software foi revisado por uma pessoa independente da equipe de desenvolvimento;
- uma pessoa independente da equipe de software testemunhou e aprovou os testes de validação;
- alterações (se existirem) necessitam de autorização de uma pessoa independente da equipe de software.

As tabelas a seguir mostram como as tabelas do Anexo A da norma IEC 61508-3 podem ser interpretadas para esta aplicação. A coluna intitulada como Ref., quando mostra subcláusulas (como B.2.4, C.3.1), refere-se á norma IEC 61508-7 – revisão de técnicas e medidas. Quando na coluna Ref. são citadas tabelas (como table B.7), refere-se à norma 61508-3.

Tabela 4.1 – Especificação dos requisitos de segurança de software
(ver cláusula 7.2 da IEC 61508-3)

	Technique/measure	Ref	SIL2	Interpretation in this application
1	Computer-aided specification tools	B.2.4	R	Development tools supplied by the PLC manufacturer
2a	Semi-formal methods	Table B.7	R	Cause-effect diagrams, sequence diagrams, function blocks. Typically used for PLC application software requirements specification
2b	Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	C.2.4	R	Not used for limited variability programming
NOTE The software safety requirements were specified in natural language.				

Tabela 4.2 – Projeto e desenvolvimento de software:
projeto de arquitetura de software (ver cláusula 7.4.3 da IEC 61508-3)

	Technique/measure	Ref	SIL2	Interpretation in this application
1	Fault detection and diagnosis	C.3.1	R	Checking of data range, watch-dog timer, I/O, communication. Raise an alarm if errors (see 3a)
2	Error detecting and correcting codes	C.3.2	R	Embedded with user options - careful selection required
3a	Failure assertion programming	C.3.3	R	Dedicate some PLC program ladder logic to test certain essential safety conditions (see 1)
3b	Safety bag techniques	C.3.4	R	Check legal I/O combinations in an independent hardware safety monitor
3c	Diverse programming	C.3.5	R	Required by the application (see 3b)
3d	Recovery block	C.3.6	R	Embedded with user options – careful selection required
3e	Backward recovery	C.3.7	R	Embedded with user options - careful selection required
3f	Forward recovery	C.3.8	R	Embedded with user options - careful selection required
3g	Re-try fault recovery mechanisms	C.3.9	R	Used as required by the application (see 2 and 3b)
3h	Memorizing executed cases	C.3.10	R	Not used for limited variability programming
4	Graceful degradation	C.3.11	R	Not used for limited variability programming
5	Artificial intelligence fault correction	C.3.12	NR	Not used for limited variability programming
6	Dynamic reconfiguration	C.3.13	NR	Not used for limited variability programming
7a	Structured methods including for example, JSD, MASCOT, SADT and Yourdon.	C.2.1	HR	Data flow methods and data logic tables may be used for representing at least the design architecture
7b	Semi-formal methods	Table B.7	R	May be used for DCS interface
7c	Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	C.2.4	R	Rarely used for limited variability programming
8	Computer-aided specification tools	B.2.4	R	Development tools supplied by the PLC manufacturer
NOTE It is impractical to implement some of the above techniques in limited variability programming.				

Tabela 4.3 – Projeto e desenvolvimento de software:
ferramentas de apoio e linguagem de programação
(ver cláusula 7.4.4 da IEC 61508-3)

	Technique/measure	Ref	SIL2	Interpretation in this application
1	Probabilistic testing	C.5.1	R	Not used for limited variability programming
2	Dynamic analysis and testing	B.6.5 Table B.2	HR	Used
3	Data recording and analysis	C.5.2	HR	Records of test cases and results
4	Functional and black box testing	B.5.1 B.5.2 Table B.3	HR	Input data is selected to exercise all specified functional cases, including error handling. Test cases from cause consequence diagrams, boundary value analysis, and input partitioning
5	Performance modelling	C.5.20 Table B.6	R	Not used for limited variability programming
6	Interface testing	C.5.3	R	Included in functional and black-box testing

Tabela 4.4 – Projeto e desenvolvimento de software:
projeto detalhado (ver cláusula 7.4.5 e 7.4.6 da IEC 61508-3)
(está incluído projeto de sistema de software, projeto de módulo de software e codificação)

	Technique/measure	Ref	SIL2	Interpretation in this application
1	Suitable programming language	C.4.6	HR	Usually ladder, and often the proprietary variety of the PLC supplier
2	Strongly typed programming language	C.4.1	HR	IEC 61131-3 structured text
3	Language subset	C.4.2	---	Beware of complex "macro" instructions, interrupts which alter PLC scan cycle, etc.
4a	Certified tools	C.4.3	HR	Available from some PLC manufacturers
4b	Tools: increased confidence from use	C.4.4	HR	PLC supplier's development kit; in-house tools developed over several projects
5a	Certified translator	C.4.3	HR	Available from some PLC manufacturers
5b	Translator: increased confidence from use	C.4.4	HR	Not used for limited variability programming
6	Library of trusted/verified software modules and components	C.4.5	HR	Function blocks, part programs

Tabela 4.5 – Projeto e desenvolvimento de software:
teste de módulo de software e integração
(ver cláusula 7.4.7 e 7.4.8 da IEC 61508-3)

	Technique/measure	Ref	SIL2	Interpretation in this application
1a	Structured methods including for example, JSD, MASCOT, SADT and Yourdon	C.2.1	HR	Not used for limited variability programming
1b	Semi-formal methods	Table B.7	HR	Cause-effect diagrams, sequence diagrams, function blocks. Typical for limited variability programming
1c	Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	C.2.4	R	Not used for limited variability programming
2	Computer-aided design tools	B.3.5	R	Development tools supplied by the PLC manufacturer
3	Defensive programming	C.2.5	R	Included in the system software
4	Modular approach	Table B.9	HR	Order and group the PLC program ladder logic to maximize its modularity with respect to the functions required
5	Design and coding standards	Table B.1	HR	In-house conventions for documentation and maintainability
6	Structured programming	C.2.7	HR	Similar to modularity in this context
7	Use of trusted/verified software modules and components (if available)	C.4.5	HR	Used

Tabela 4.6 – Integração de dispositivos eletrônicos programáveis (hardware e software) (ver cláusula 7.5 da IEC 61508-3)

	Technique/measure	Ref	SIL2	Interpretation in this application
1	Functional and black-box testing	B.5.1 B.5.2 Table B.3	HR	Input data is selected to exercise all specified functional cases, including error handling. Test cases from cause consequence diagrams, boundary value analysis, and input partitioning
2	Performance testing	C.5.20 Table B.6	R	When the PLC system is assembled for factory acceptance test

Tabela 4.7 – Validação de segurança de software (ver cláusula 7.7 da IEC 61508-3)

	Technique/measure	Ref	SIL2	Interpretation in this application
1	Probabilistic testing	C.5.1	R	Not used for limited variability programming
2	Simulation/modelling	Table B.5	R	Not used for limited variability programming, but becoming more commonly used in PLC systems development
3	Functional and black-box testing	B.5.1 B.5.2 Table B.3	HR	Input data is selected to exercise all specified functional cases, including error handling. Test cases from cause consequence diagrams, boundary value analysis, and input partitioning

Tabela 4.8 – Modificação de software (ver cláusula 7.8 da IEC 61508-3)

	Technique/measure	Ref	SIL2	Interpretation in this application
1	Impact analysis	C.5.23	HR	An impact analysis is carried out to consider how the effect of the proposed changes is limited by the modularity of the overall system
2	Reverify changed software module	C.5.23	HR	Repeat earlier tests
3	Reverify affected software modules	C.5.23	HR	Repeat earlier tests
4	Revalidate complete system	C.5.23	R	Impact analysis showed that the modification is necessary, so revalidation is done as required
5	Software configuration management	C.5.24	HR	Baselines, records of changes, impact on other system requirements
6	Data recording and analysis	C.5.2	HR	Records of test cases and results

Tabela 4.9 – Verificação de software (ver cláusula 7.9 IEC 61508-3)

	Technique/measure	Ref	SIL2	Interpretation in this application
1	Formal proof	C.5.13	R	Not used for limited variability programming
2	Probabilistic testing	C.5.1	R	Replaced by operating experience of existing parts
3	Static analysis	B.6.4 Table B.8	HR	Clerical cross-referencing of usage of variables, conditions, etc.
4	Dynamic analysis and testing	B.6.5 Table B.2	HR	Automatic test harness to facilitate regression testing
5	Software complexity metrics	C.5.14	R	Not used for limited variability programming
Software module testing and integration		See table E.5		
Programmable electronics integration testing		See table E.6		
Software system testing (validation)		See table E.7		

Tabela 4.10 – Avaliação de segurança funcional (ver cláusula 8 da IEC 61508-3)

	Technique/measure	Ref	SIL2	Interpretation in this application
1	Checklists	B.2.5	R	Used
2	Decision/truth tables	C.6.1	R	Used to a limited degree
3	Software complexity metrics	C.5.14	R	Not used for limited variability programming
4	Failure analysis	Table B.4	R	Cause-consequence diagrams at system level, but otherwise, failure analysis is not used for limited variability programming
5	Common cause failure analysis of diverse software (if diverse software is actually used)	C.6.3	R	Not used for limited variability programming
6	Reliability block diagram	C.6.5	R	Not used for limited variability programming

5 DOCUMENTAÇÃO

De acordo com cláusula B.1.2 do anexo B da norma IEC 61508-7, o objetivo geral da documentação é evitar falhas e facilitar a avaliação da segurança do sistema, documentando cada fase do desenvolvimento. A capacidade e segurança operacional, assim como os cuidados tomados no desenvolvimento por todas as partes envolvidas, devem ser demonstrados na avaliação. Com a finalidade de demonstrar a atenção dada ao desenvolvimento e, para garantir a verificação das evidências de segurança em qualquer tempo, uma importância especial é dada à documentação.

Medidas comuns e importantes são a introdução de um guia de orientações e de auxílio digital, ou seja,

- guia de orientações que
  - especifique um plano de grupo;
  - contenha checklists para os conteúdo, e
  - determine o formato do documento.
- administração da documentação através de um projeto de biblioteca organizada por computador;
- as medidas individuais: separação da documentação
  - dos requisitos;
  - do sistema (documentação do usuário) e
  - do desenvolvimento (incluindo a inspeção interna);
- agrupamento da documentação de desenvolvimento de acordo com o ciclo de vida de segurança;
- definição dos módulos de documentação padronizada, a partir do qual os documentos podem ser compilados;
- clara identificação dos elementos constitutivos da documentação;
- seleção da forma de descrição clara e inteligível:
  - notação formal para as determinações;
  - linguagem natural para as apresentações, justificativas e representação dos objetivos;
  - representação gráfica para os exemplos;

- definição semântica de elementos gráficos, e
- diretórios de palavras especializadas.

As informações a seguir estão contidas na norma 61508-1, requisitos gerais.

5.1 Objetivos

Os objetivos da documentação:

- especificar informações necessárias que devem ser documentadas de modo que todas as fases de todo o processo, E / E / EPS e ciclo de vida de segurança de software possam ser efetivamente realizadas.
- especificar informações necessárias que devem ser documentadas de modo que a gestão da segurança funcional (cláusula 6), atividades de verificação (cláusula 7.18) e a avaliação da segurança funcional (cláusula 8) possam ser efetivamente realizadas.

Em linhas gerais, a norma não exige que a documentação seja apresentada como documentos físicos, a menos que seja explicitamente declarado nas subseções relevantes. A documentação pode ser apresentada em diferentes formas, como papel, filme ou qualquer forma que possa ser apresentada em telas ou displays.

5.2 Requisitos

a) A documentação deverá conter informações suficientes, para cada fase de todo o processo, E / E / EPS e ciclo de vida de software de segurança completo, necessárias para o desempenho eficaz das fases subsequentes e atividades de verificação. O que caracterizará informação suficiente dependerá de uma série de fatores, incluindo a complexidade e tamanho do sistema E / E / EP relacionado à segurança e as exigências relativas à aplicação específica.

b) A documentação deverá conter informações suficientes exigidas para a gestão de segurança funcional (cláusula 6).

c) A documentação deverá conter informações suficientes necessárias para a execução de uma avaliação de segurança funcional, juntamente com as informações e resultados obtidos a partir de qualquer valiação de segurança funcional.

d) A menos que justificado no planejamento da segurança funcional ou especificado na norma de aplicação do setor, as informações que deverão ser documentadas serão as estipuladas nas diversas cláusulas da norma.

e) A disponibilidade de documentação deve ser suficiente para as tarefas a serem realizadas em respeito das cláusulas da norma.

f) A documentação deve:

- Ser precisa e concisa;
- Ser de fácil compreensão para as pessoas que farão uso dela;
- Atender as finalidades para as quais se destina;
- Ser acessível e sustentável.

g) A documentação ou o conjunto de informações devem ter os títulos ou nomes indicando o escopo dos conteúdos e alguma forma de índice, de modo a permitir o acesso imediato às informações exigidas pela norma.

h) A estrutura da documentação pode levar em conta os procedimentos da empresa e as práticas de trabalho dos setores específicos da aplicação.

i) Os documentos ou conjunto de informações devem possuir um índice de revisão (números de versão) para tornar possível identificar diferentes versões do documento.

j) Os documentos ou conjunto de informações devem ser estruturados de forma a tornar possível a busca de informações relevantes. Deverá ser possível identificar a última revisão (versão) de um documento ou conjunto de informações.

k) A estrutura física da documentação varia de acordo com uma série de fatores como o tamanho do sistema, sua complexidade e as exigências organizacionais.

l) Todos os documentos devem ser corrigidos, alterados, revistos, aprovados e submetidos a um esquema apropriado de controle de documentos.

n) Se forem usadas ferramentas automáticas ou semi-automáticas para a produção de documentação, procedimentos específicos podem ser necessários para assegurar o eficaz cumprimento das medidas em vigor para a gestão de versões ou de outras aspectos relativos ao controle de documentos.

5.3 Exemplo de estrutura de documentação

Aqui será apresentado um exemplo de estrutura de documentação e método de especificação de documentos para estruturar as informações, a fim de satisfazer os requisitos da cláusula 5 da norma IEC 61508-1: requisitos gerais. Este exemplo foi retirado desta mesma norma.

A documentação deve conter informações suficientes necessárias para executar de forma eficaz:

- Cada fase de uma forma geral, E / E / EPS e ciclos de vida de software de segurança;
- A gestão da segurança funcional (cláusula 6);
- Avaliação de segurança funcional (cláusula 8);

O que constitui informação suficiente dependerá de uma série de fatores, incluindo a complexidade e tamanho do sistema E / E / EP relacionado à segurança e às exigências relativas à aplicação específica. A documentação necessária pode ser especificada nas aplicações específicas das normas internacionais.

A quantidade de informação em cada documento pode variar de algumas linhas a muitas páginas, e o conjunto completo de informações pode ser dividido e apresentado em vários documentos físicos ou um documento físico somente. A estrutura da documentação física penderá novamente do tamanho e complexidade do sistema E/E/PE relacionado à segurança, e levará em conta os procedimentos da empresa as práticas de trabalho do setor da aplicação específica.

A estrutura da documentação indicada neste exemplo serve para ilustrar uma forma particular de estruturação possível da informação e uma forma de como os documentos poderiam ser intitulados. Essa estrutura de documentação encontra-se mais detalhada na norma 61506, medição e controle de processo industrial - documentação de software de aplicação.

Um documento é uma quantidade de informações estruturadas destinadas à percepção humana, que podem ser intercambiáveis como unidade entre os usuários e/ou sistemas, de acordo com a norma ISO 8613-1. O termo se aplica não só aos documentos no sentido tradicional, mas também a conceitos como arquivos de dados e informações de banco de dados.

Para a norma, o termo documento é normalmente entendido no sentido de informação ao invés de documento físico, a menos que seja explicitamente declarado, ou entendido no contexto da cláusula ou subseção em que é afirmado. Os documentos podem estar disponíveis em diferentes formas para apresentação (por exemplo, papel, filme ou qualquer outro meio de dados a serem apresentados em telas ou displays).

A estrutura da documentação deste exemplo especifica os documentos em duas partes:

- Tipo de documento;
- Atividade ou objeto.

O tipo de documento é definido na IEC 61355, classificação e designação de documentação para fábricas, sistemas e equipamentos, e caracteriza o conteúdo do documento. Por exemplo, descrição de função ou diagrama de circuito. A atividade ou objeto descreve o escopo do conteúdo.

Os tipos de documentos básicos especificados neste exemplo são:

- **Especificação** - especifica a função requerida de desempenho ou atividade (por exemplo: especificação de requisitos);
- **Descrição** - especifica uma função prevista ou real, projeto, desempenho ou atividade (por exemplo: a descrição da função);
- **Instrução** - especifica em detalhes como as instruções como quando e como realizar certas tarefas (por exemplo: a instrução do operador);
- **Plano** - especifica o plano de quando, como e por quem atividades específicas devem ser realizados (por exemplo: plano de manutenção);
- **Diagrama** - especifica a função por meio de um diagrama (símbolos e linhas) representando sinais entre os símbolos;
- **Lista** - fornece informações em um formulário de listas (lista de códigos de exemplo, lista de sinais);
- **Log** - fornece informações sobre eventos em um formulário de registro cronológico;
- **Relatório** - descreve os resultados das atividades, tais como investigações, avaliações, testes, etc (por exemplo: relatório de manutenção);
- **Requisição** - fornece uma descrição das medidas solicitadas, que devem ser aprovados e mais especificadas (por exemplo, requisição de manutenção).

O tipo de documento básico pode ter um prefixo, como especificações de requisitos ou especificações de teste, o que caracterizar melhor o conteúdo.

5.4 Estrutura de documentação de acordo com o ciclo de vida de software

As tabelas 5.1, 5.2 e 5.3, retiradas da norma 61508-1, fornecem uma estrutura de documentação de exemplo para a estruturação de informações a fim de cumprir os requisitos especificados na cláusula 5. As tabelas indicam a fase do ciclo de vida de segurança (ver figura 5.1) associada aos documentos (normalmente a fase em que eles são desenvolvidos). Os nomes dados aos documentos nos quadros estão de acordo com o esquema descrito no exemplo de estrutura de documentação, explicado no item anterior.

Além dos documentos listados nas tabelas 5.1, 5.2 e 5.3, podem ser necessários documentos complementares com informações detalhadas ou informações estruturadas para um determinado propósito. Por exemplo, listas de peças, listas de sinais, listas de cabos, tabelas de fiação, diagramas de loop, a lista de variáveis. Exemplos de tais variáveis são valores para reguladores, valores de alarme para as variáveis, prioridades na execução de tarefas no computador. Alguns dos valores das variáveis poderiam ser dados antes da entrega do sistema, outros poderiam ser dados durante o comissionamento e manutenção.

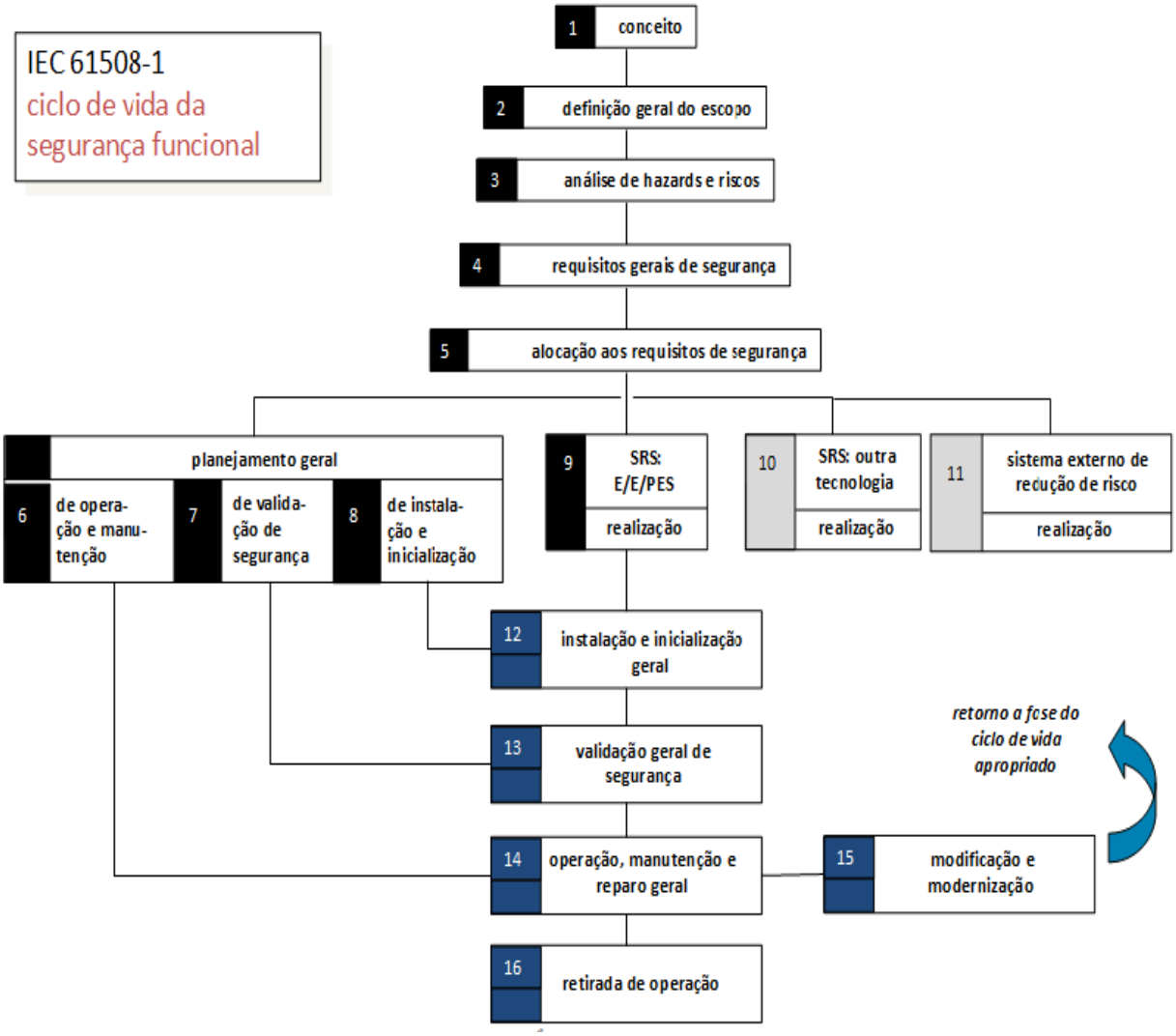


Figura 5.1: ciclo de vida da segurança funcional

Tabela 5.1: Exemplo de estrutura de documentação para informações relacionadas ao ciclo de vida de segurança geral

Fase do ciclo de vida de segurança geral	**Informação**
Conceito	Descrição (conceito geral)
Definição do escopo geral	Descrição (definição do escopo geral)
Análise de risco e perigo	Descrição (análise de risco e perigo)
Requisitos de segurança geral	Especificação (requisitos de segurança, abrangência, funções de segurança geral, integridade de segurança geral)
Alocação de requisitos de segurança	Descrição (alocação de requisitos de segurança)
Planejamento de operação e manutenção gerais	Plano (operação e manutenção gerais)
Planejamento de validação de segurança geral	Plano (validação de segurança geral)
Planejamento de instalação e comissionamento geral	Plano (instalação geral); Plano (comissionamento geral)
Realização	Realização de sistemas E/E/EP relacionados à segurança (veja IEC 61508-2 e IEC 61508-3)
Instalação e comissionamento gerais	Relatório (instalação geral); Relatório (comissionamento geral)
Validação de segurança geral	Relatório(validação de segurança geral)
Operação e manutenção gerais	Log (operação e manutenção gerais)
Modificação e renovação/substituição gerais	Requisição (modificação geral); Relatório (análise de impacto de modificação e renovação/substituição gerais); Log (modificação e renovação/substituição gerais)
Desclassificação ou eliminação	Relatório (análise de impacto de desclassificação ou eliminação gerais); Plano (desclassificação e eliminação gerais); Log (desclassificação e eliminação gerais)
Relativo a todas as fases	Plano (segurança); Plano (verificação); Relatório (verificação); Plano (avaliação de segurança funcional); Relatório (avaliação de segurança funcional);

Tabela 5.2: Exemplo de estrutura de documentação para informações relacionadas ao ciclo de vida de segurança de E/E/EP

Fase do ciclo de vida de segurança de E/E/EP	**Informação**
Requisitos de segurança de E/E/EPs	Especificação (Requisitos de segurança de E/E/EPs, abordagem: funções de segurança de E/E/EPs e integridade de segurança de E/E/EPs)
Planejamento de validação de segurança de E/E/EPs	Plano (validação de segurança de E/E/EPs)
Projeto e desenvolvimento de E/E/EPs	
Arquitetura de E/E/EPs	Descrição (projeto de arquitetura de E/E/EPs, abrangência: arquitetura de hardware e arquitetura de software); Especificação (testes de integração de eletrônicos programáveis); Especificação (testes de integração de hardware eletrônico programável e não eletrônico programável);
Arquitetura de hardware	Descrição (projeto de arquitetura de hardware); Especificação (testes de integração de arquitetura de hardware
Projeto de módulo de hardware	Especificação (projeto de arquitetura de hardware); Especificação (teste de módulo de hardware)
Construção e/ou aquisição de componente	Módulo de hardware; Relatório (teste de módulo de hardware)
Integração de componente eletrônico programável	Relatório (teste de integração de componente eletrônico programável e software) (veja tabela A3)
Integração de componente E/E/EPs	Relatório (teste de integração de componente eletrônico programável e hardware)
Procedimentos de operação e manutenção de E/E/EPs	Instrução (usuário); Instrução (operação e manutenção)
Validação de segurança de E/E/EPs	Relatório (validação de segurança de E/E/EPs)
Modificação de E/E/EPs	Instrução (procedimentos de modificação de E/E/EPs); Requisição (modificação de E/E/EPs); Relatório (análise de impacto de modificação de E/E/EPs); Log (modificação de E/E/EPs);
Relativo a todas as fases	Plano (segurança de E/E/EPs); Plano (verificação de E/E/EPs); Relatório (verificação de E/E/EPs); Plano (avaliação de segurança funcional de E/E/EPs); Relatório (avaliação de segurança funcional de E/E/EPs)

Tabela 5.3: Exemplo de estrutura de documentação para informação relacionada ao ciclo de vida de segurança de software

Fase do ciclo de vida de segurança de software	**Informação**
Requisitos de segurança de software	Especificação (requisitos de segurança de software, abrangência: funções de segurança de software e integridade de segurança de software)
Planejamento de validação de software	Plano (validação de segurança de software)
Projeto e desenvolvimento de software	
Arquitetura de software	Descrição (projeto de arquitetura de software) (ver tabela A2 para descrição de projeto de arquitetura de hardware); Especificação (teste de integração de arquitetura de software); Especificação (teste de integração de eletrônico programável e software); Instrução (ferramentas de desenvolvimento e codificação manual)
Projeto de sistema de software	Descrição (projeto de sistema de software); Especificação (teste de integração de sistema de software)
Projeto de módulo de software	Especificação (projeto de módulo de software); Especificação (teste de módulo de software)
Codificação	Lista (código fonte); Relatório (teste de módulo de software); Relatório (revisão de código)
Teste de módulo de software	Relatório (teste de módulo de software)
Integração de software	Relatório (teste de integração de módulo de software); Relatório (teste de integração de sistema de software); Relatório (teste de integração de arquitetura de software)
Integração de componente eletrônico programável	Relatório (teste de integração de componente eletrônico programável e software)
Procedimentos de operação e manutenção de software	Instrução (usuário); Instrução (operação e manutenção)
Validação de segurança de software	Relatório (validação de segurança de software)
Modificação de software	Instrução (procedimentos de modificação de software); Requisição (modificação de software); Relatório (análise de impacto de modificação de software); Log (modificação de software)
Relativo a todas as fases	Plano (segurança de software); Plano (verificação de software); Relatório (verificação de software); Plano (análise de segurança funcional de software); Relatório (análise de segurança funcional de software)

5.5 Estrutura física de documentos

A estrutura física da documentação é a forma pela qual os diferentes documentos são combinados em documentos, conjuntos de documentos, pastas e grupos de arquivos. A figura 5.2 mostra exemplos de tais conjuntos de arquivos estruturados de acordo com grupos de usuários. O mesmo documento pode ocorrer em diferentes conjuntos.

Para um sistema grande e complexo, muitos documentos físicos são passíveis de serem divididos em várias pastas. Para um pequeno sistema de baixa complexidade, com um número limitado de documentos físicos, que podem ser combinados em uma pasta com guias diferentes para os diferentes conjuntos de documentos (ver figura 5.3).

A estrutura física proporciona um meio de seleção para documentação necessária de atividades específicas por uma pessoa ou grupo de pessoas que executam as atividades. Consequentemente, alguns dos documentos físicos podem ocorrer em vários conjuntos de pastas ou outros meios de comunicação (por exemplo, CDs ou DVDs).

As informações exigidas pelos documentos na tabela 5.1 podem estar contidas dentro dos diferentes grupos de documentos mostrados nas figuras 5.2 e 5.3. Por exemplo, dentro do conjunto de engenharia, documentos, tais como a descrição de análise de risco e de perigo e / ou especificação de requisitos gerais de segurança estariam contidos nele.

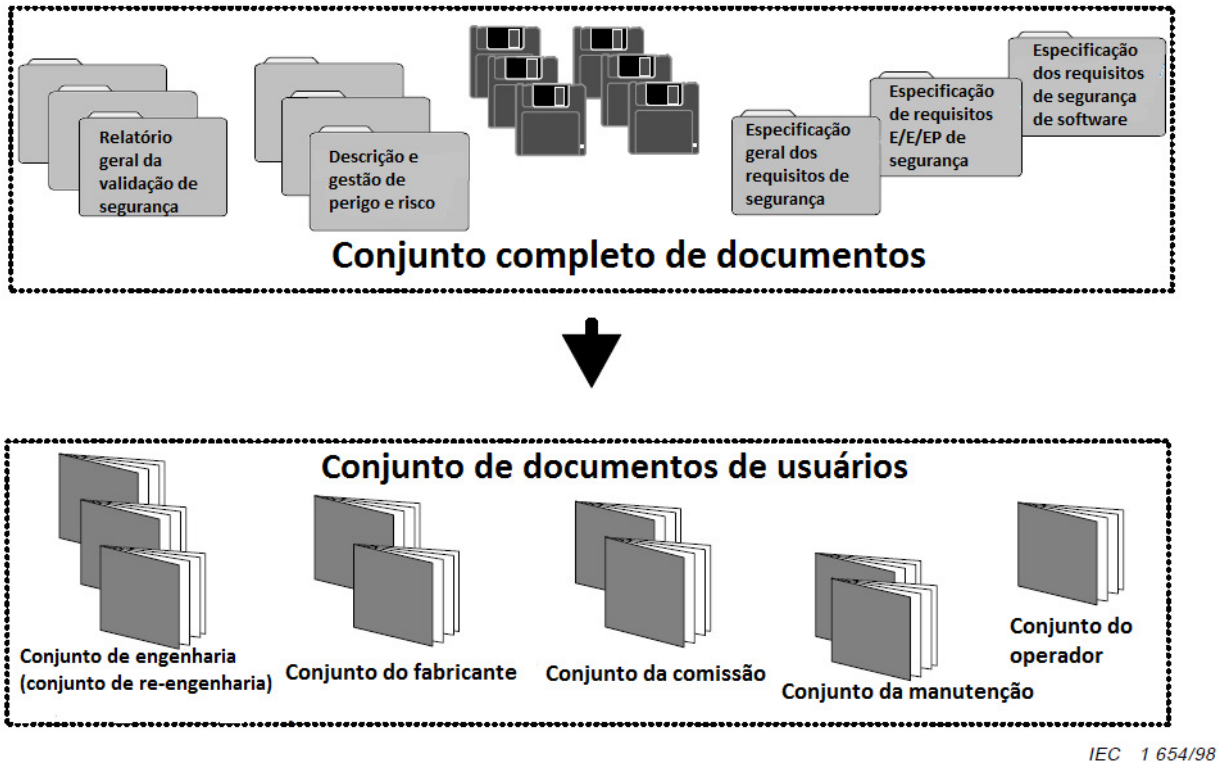


Figura 5.2: Estrutura da informação em conjunto de documentos para grupos de usuários

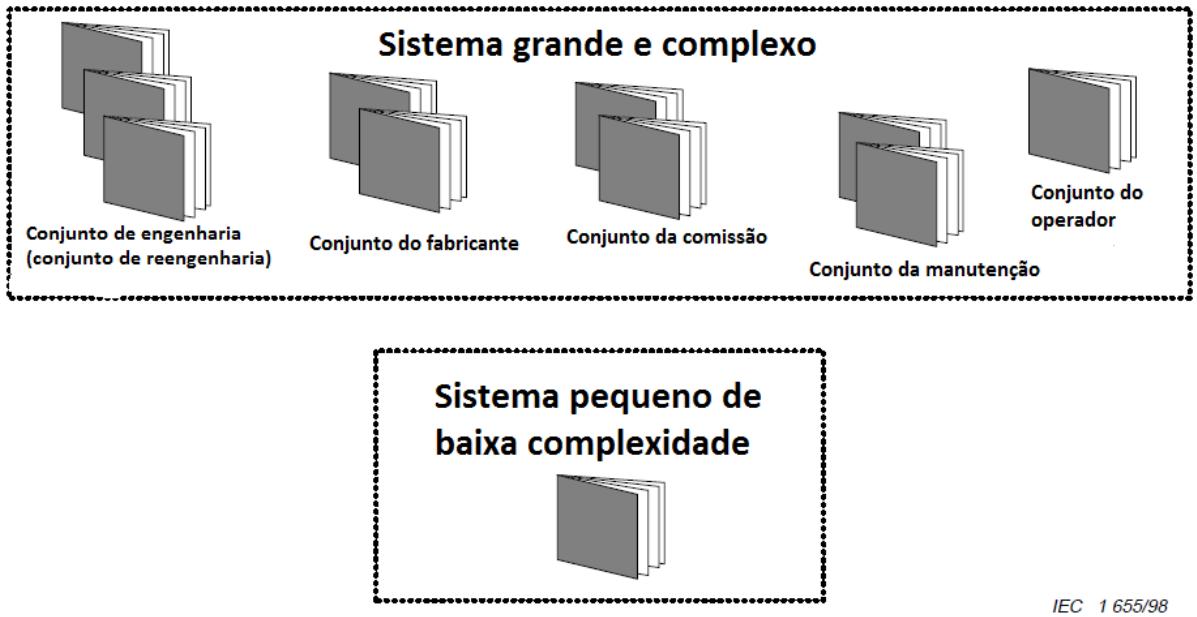


Figura 5.3: Estrutura da informação para grandes sistemas complexos e sistemas pequenos de baixa complexidade

5.6 Lista de documentos

A lista de documentos geralmente inclui as seguintes informações:

- Número do diagrama ou do documento;
- Índice de revisão;
- Código de designação do documento;
- Título;
- Data de revisão;
- Suporte de dados (*data carrier*).

Esta lista pode aparecer em diferentes formas, por exemplo, em um banco de dados capaz de ser classificado de acordo com o número do documento ou código de designação do documento. O código de designação do documento pode conter a designação de referência para a função, localização ou produto descrito no documento, tornando-se uma poderosa ferramenta na busca de informações.

6 O SOFTWARE PROPOSTO: SAFETY INTEGRITY DETERMINATION (SID)

Neste capítulo será descrito o funcionamento do software Safety Integrity Determination (SID), que poderá auxiliar no processo de certificação de dispositivos de segurança, de acordo com a norma IEC 61508-3. Num trabalho anterior (SCHMIDT, 2009), um protótipo de software para auxiliar na certificação foi construído, mas este não contemplava auxílio na documentação do processo. Também não havia nenhuma forma de gerenciamento do processo de certificação. Neste trabalho foram implementadas melhorias nesse protótipo, acrescentando suporte à documentação e gerenciamento do processo de certificação.

O software SID pretende servir como uma ferramenta de apoio à certificação, possibilitando que o projetista tenha uma visão mais ampla do funcionamento da norma, dos requisitos a serem seguidos e da documentação a ser produzida.

O software SID não tem pretensões comerciais, sendo seu único intuito servir como alternativa às ferramentas existentes no mercado, possibilitando o estudo mais dinâmico da norma IEC 61508. A utilização correta do software, porém, não garante a certificação, já que não existe um algoritmo a ser seguido, devido às diferenças nos projetos de software de segurança, fazendo com que cada avaliação seja única. A certificação também depende da qualidade da documentação, o que é impossível de se mensurar através de algoritmos, ficando a cargo dos certificadores o poder de avaliação.

6.1 Estruturação do software SID

O software SID para auxílio à certificação foi baseado nas tabelas do anexo A da norma IEC 61508-3 (que se encontram no Anexo A deste trabalho). Estas tabelas listam as técnicas e medidas (na coluna Technique/Measure) que serão aplicadas ao sistema relacionado à segurança, de acordo com o nível de integridade de segurança pretendido (nas colunas SIL 1, SIL 2, SIL 3 e SIL4. Nas tabelas, cada técnica possui uma classificação de recomendação em relação ao SIL, como segue:

- HR: High Recommended (altamente recomendado);
- R: Recommended (recomendado);
- ---: Não há recomendação contra ou a favor;

- NR: Not Recommended (não recomendado).

Uma técnica altamente recomendada (HR) sobrepõe em importância uma técnica recomendada (R). Uma técnica - - não tem significativa importância para a certificação, enquanto uma técnica não recomendada (NR) não tem seu uso aconselhado e pode interferir na certificação.

A lista de técnicas também compreende técnicas alternativas, listadas com o número da técnica juntamente com uma letra (como 1a, 1b, 1c, etc.). Neste caso, as técnicas são equivalentes para a certificação, necessitando que somente uma delas seja realizada.

Cada técnica possui uma descrição do seu funcionamento, que na tabela do Anexo A encontra-se na coluna Ref. Essa descrição encontra-se na norma IEC 61508-7, revisão de medidas e técnicas.

6.2 O funcionamento do software SID

O software SID apresenta duas áreas distintas: a área de administração e a área do usuário. A área de administração compreende cadastro de usuários. Os usuários têm permissões de acesso diferentes, atribuídas pelo gerente de projeto. A área de usuário é a aplicação propriamente dita.

A tela inicial apresenta logo no topo o SIL do projeto em que se está trabalhando. Na tela do usuário com permissão de gerente, aparece um botão *Change* ao lado SIL, que permite que o SIL do projeto possa ser alterado, o que consequentemente altera a classificação das técnicas a serem aplicadas nesse projeto. Ao lado do botão *Change*, há o botão *Verify Validity*, que será explicado mais adiante. Ainda na tela inicial, apresenta-se a lista de títulos das tabelas do anexo A da norma IEC 61508-3, o que corresponde às fases do ciclo de vida de segurança de software, como mostra a figura 6.1. Ao lado de cada técnica, há uma imagem de uma seta direcionada para baixo.

Ao clicar-se nessa seta, logo abaixo aparecerá uma descrição da documentação necessária nesta fase e, ao lado da descrição, um link *Documents* (este item será desenvolvido mais adiante). Nesta mesma tela, ilustrada pela figura 6.2, também serão listadas as técnicas referentes ao ciclo de vida. Para cada técnica há:

- um *checkbox*, que deverá ser marcado quando a técnica estiver sendo utilizada;
- *Associeted Docs*: ao se clicar no link *Documents*, abre uma janela pop-up para upload, produção e alteração de documentos (este item será aprofundado mais adiante);

- *Status*: informação do status da documentação (empty, working ou complete);
- *SIL*: informação sobre o SIL que está sendo aplicado, descrevendo se a técnica é HR, R, --- ou NR;
- *Reference*: referência da técnica na norma IEC 61508-7, que é um link que ao ser clicado, abre uma janela pop-up com nome, referência, objetivo e descrição da técnica, como mostra a figura 6.3.

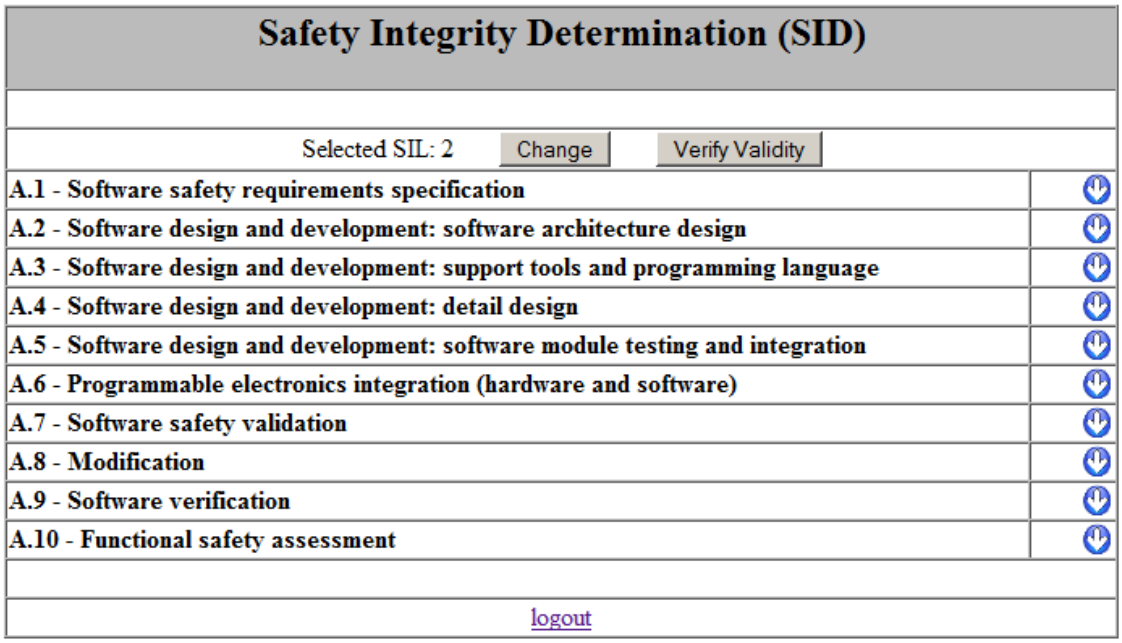
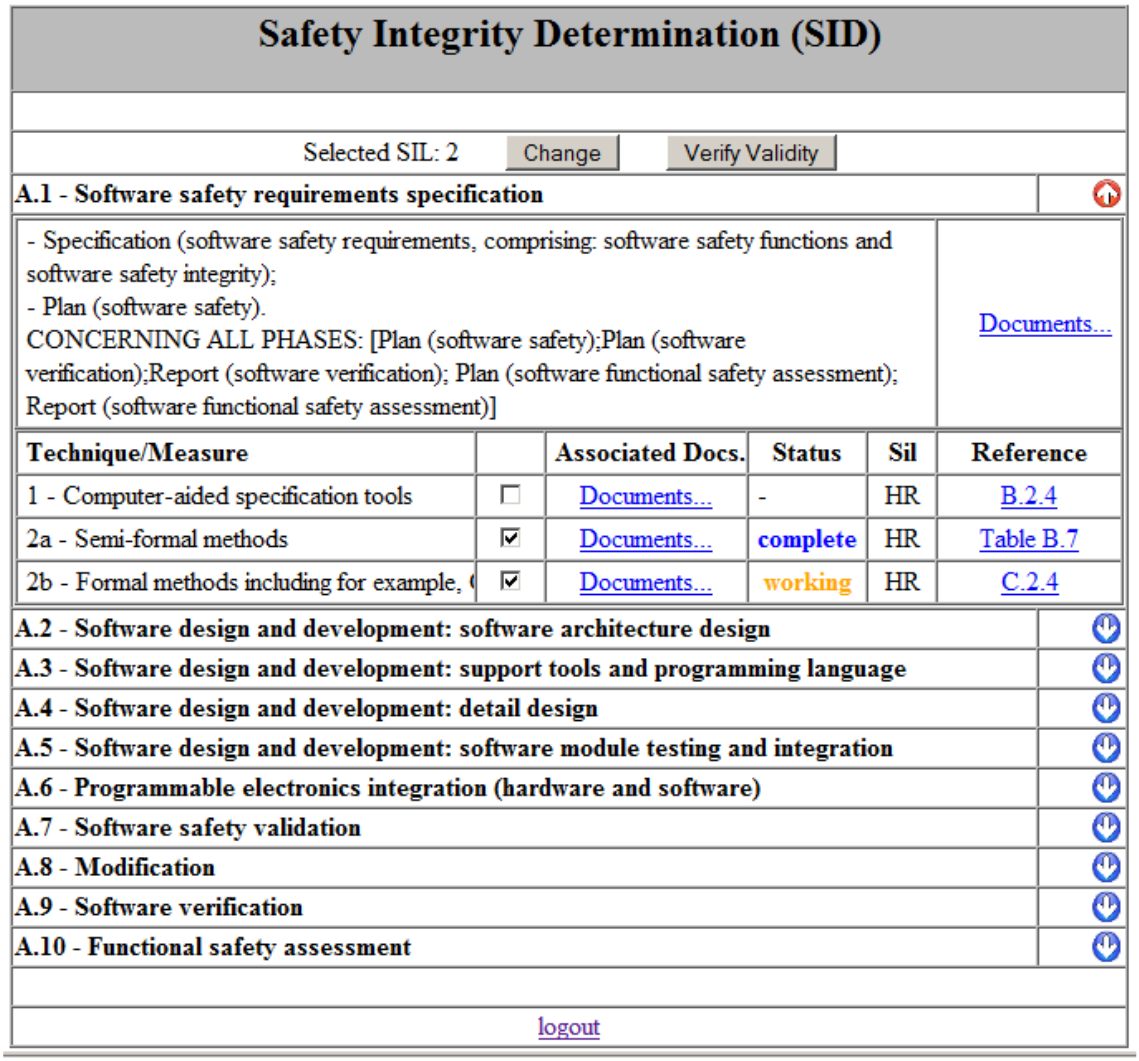


Figura 6.1: Tela inicial



Safety Integrity Determination (SID)

Selected SIL: 2 | Change | Verify Validity

A.1 - Software safety requirements specification

- Specification (software safety requirements, comprising: software safety functions and software safety integrity);
- Plan (software safety).
CONCERNING ALL PHASES: [Plan (software safety);Plan (software verification);Report (software verification); Plan (software functional safety assessment); Report (software functional safety assessment)]

Documents...

Technique/Measure		Associated Docs.	Status	Sil	Reference
1 - Computer-aided specification tools	☐	Documents...	-	HR	B.2.4
2a - Semi-formal methods	☑	Documents...	complete	HR	Table B.7
2b - Formal methods including for example, (	☑	Documents...	working	HR	C.2.4

A.2 - Software design and development: software architecture design

A.3 - Software design and development: support tools and programming language

A.4 - Software design and development: detail design

A.5 - Software design and development: software module testing and integration

A.6 - Programmable electronics integration (hardware and software)

A.7 - Software safety validation

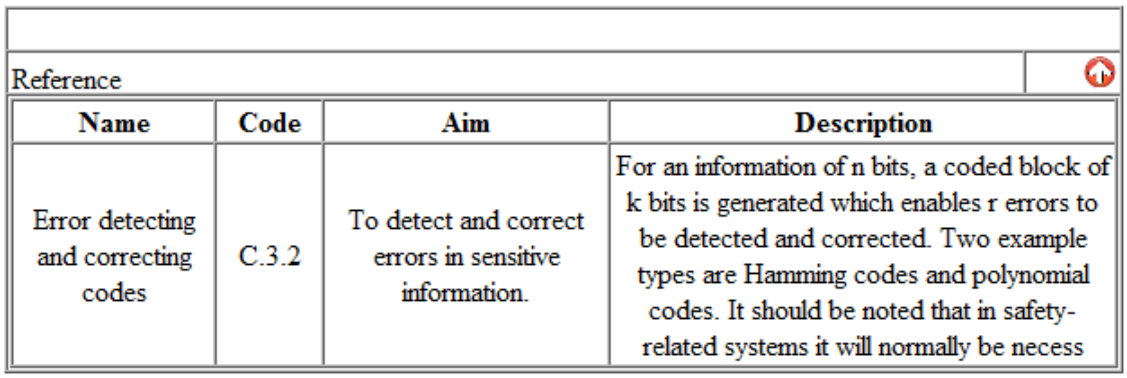
A.8 - Modification

A.9 - Software verification

A.10 - Functional safety assessment

logout

Figura 6.2: Lista de técnicas



Reference

Name	Code	Aim	Description
Error detecting and correcting codes	C.3.2	To detect and correct errors in sensitive information.	For an information of n bits, a coded block of k bits is generated which enables r errors to be detected and corrected. Two example types are Hamming codes and polynomial codes. It should be noted that in safety-related systems it will normally be necess

Figura 6.3: Referência de técnica

Na tela descrita anteriormente, aparece um link *Documents* ao lado da descrição dos documentos da fase do ciclo de vida, assim como na linha de cada técnica. Ao clicar-se no link *Documents*, abrirá uma janela pop-up (figura 6.4), onde será permitido:

- fazer upload de arquivos;
- criar arquivos;
- editar arquivos;
- remover arquivos;
- visualizar arquivos;
- colocar status dos documentos (somente em documentos referentes às técnicas).

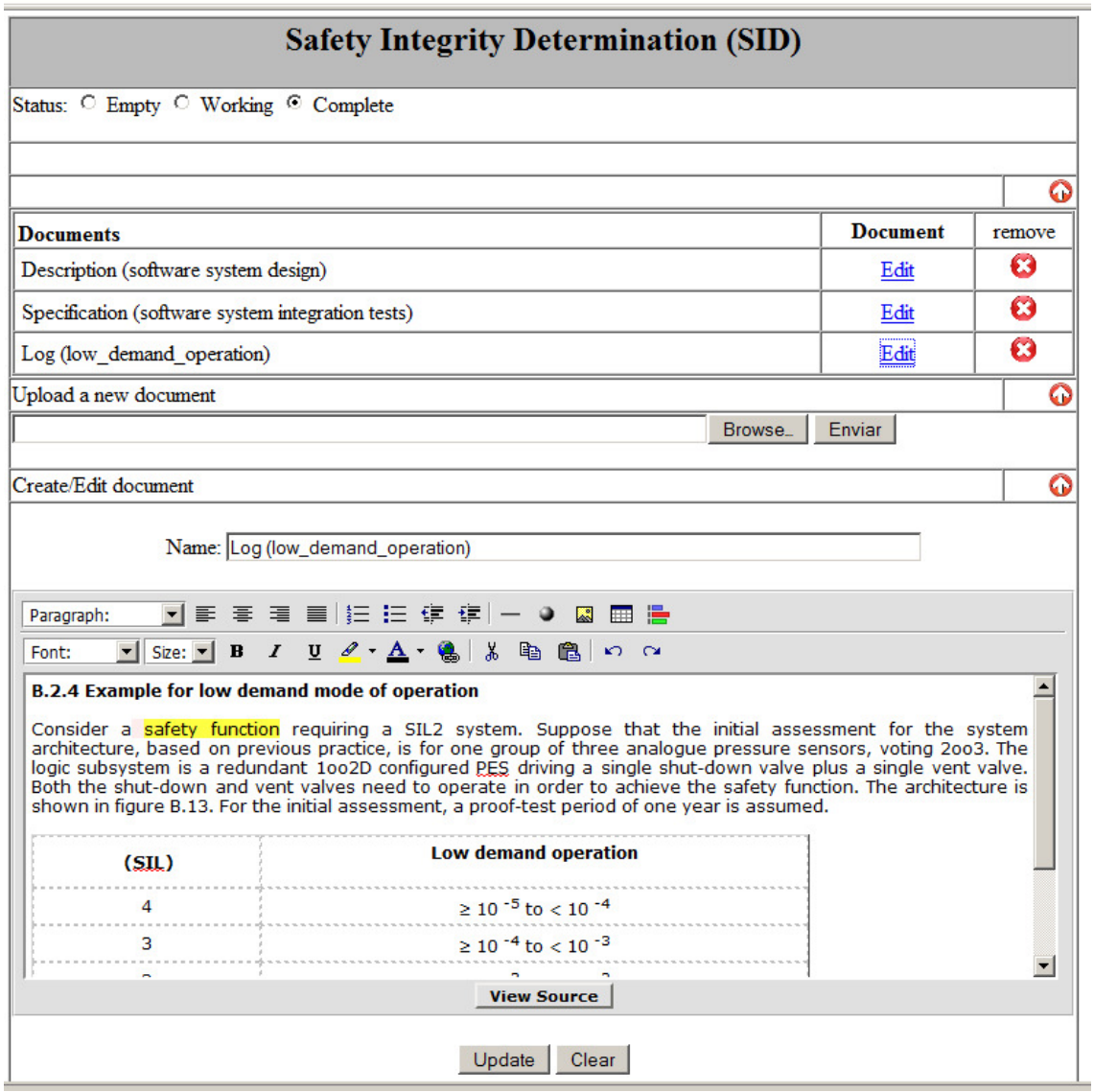


Figura 6.4: Tela de administração de documentos

Na tela inicial, há um botão *Verify Validity*, ao lado do botão *Change*, que de acordo com as técnicas que foram assinaladas, fará uma estimativa sobre a possibilidade da certificação. Essa estimativa se dá de acordo com o SIL do projeto. Caso uma técnica altamente recomendada (HR) não esteja assinalada, o sistema listará na saída a técnica em questão, avisando que essa técnica é HR. O mesmo ocorre para técnicas recomendadas (R). No caso de estar assinalada uma técnica não recomendada (NR), o sistema mostrará a técnica na saída do programa, com um aviso de que a técnica em questão não é recomendada e sugerindo uma técnica alternativa. Na saída do programa, também é mostrada a lista das técnicas assinaladas. A seguir, a figura 6.5 ilustra uma possível saída do programa.

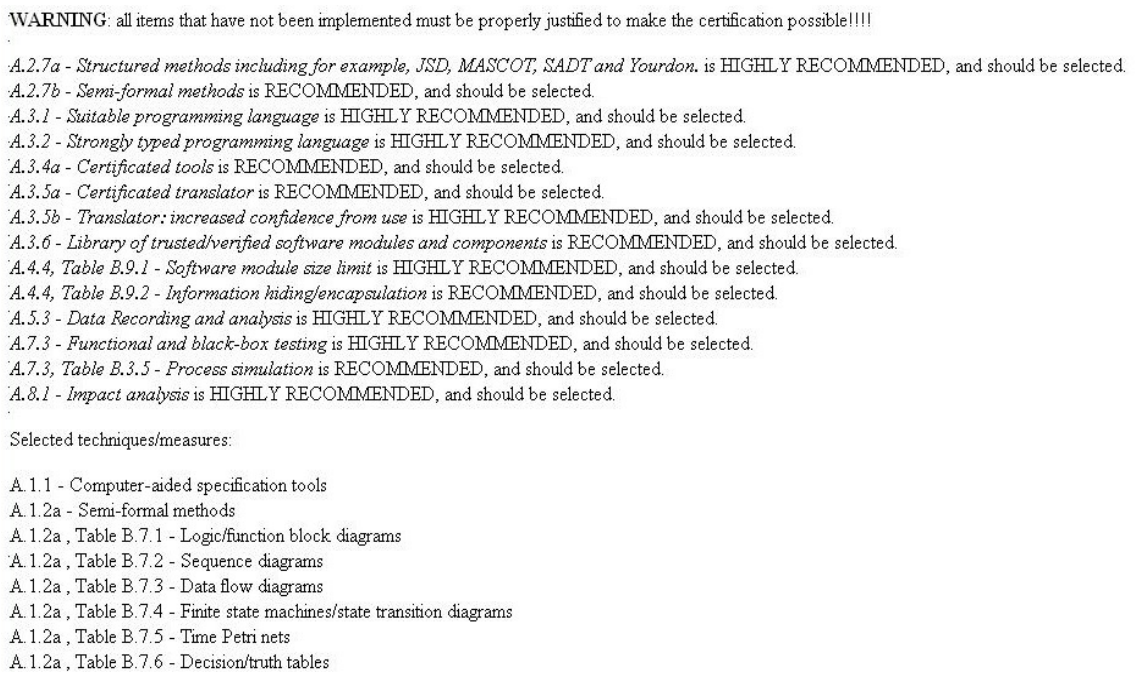


Figura 6.5: Exemplo de saída do programa

Na área de administração (figura 6.6), é possível fazer:

- manutenção de usuários (menu usuario);
- manutenção do ciclo de vida de software (menu technique group);
- manutenção de técnicas (menu technique);
- manutenção de referências de técnicas (menu reference);
- manutenção de SIL (menu certification).

Os usuários são classificados em 3 tipos: gerente, administrador e técnico. O tipo de usuário define o tipo de permissão de acesso ao sistema. O usuário precisa cadastrar user, senha, nome, e-mail e tipo. É possível fazer a visualização da lista de usuários cadastrados, assim como editar usuários e removê-los do sistema, de acordo com a figura 6.6.

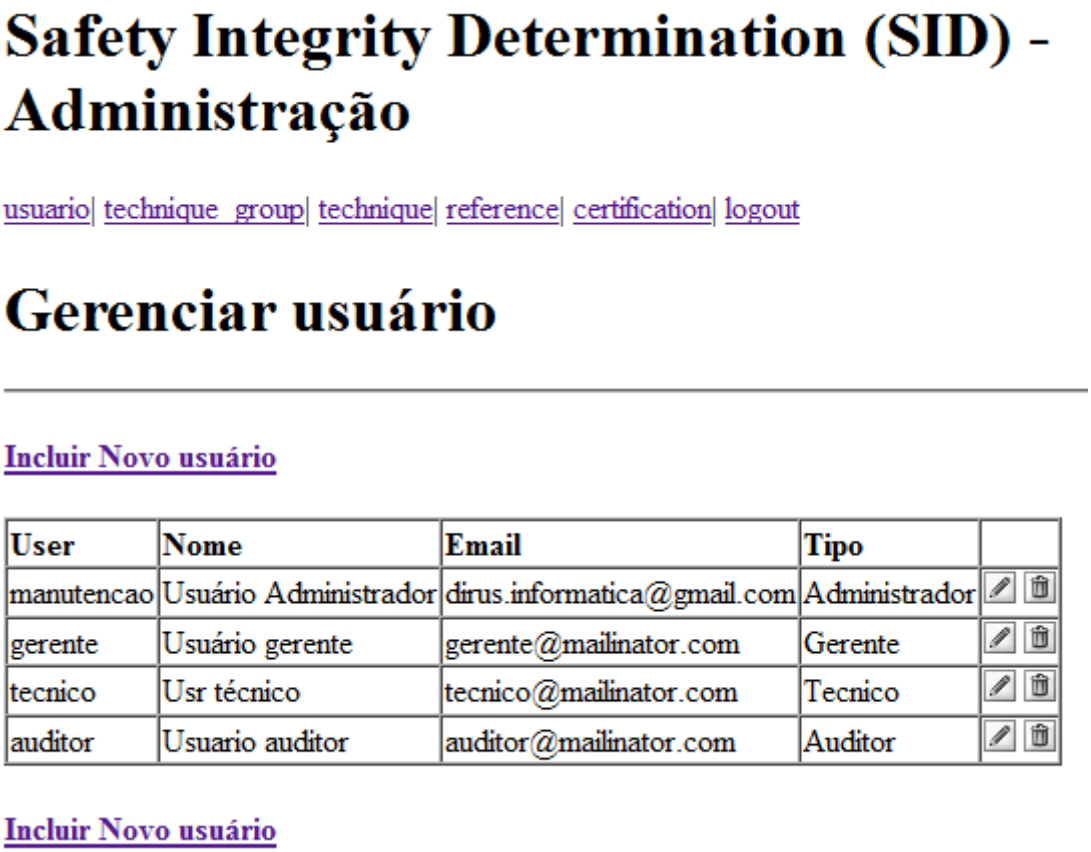


Figura 6.6: Tela de administração

A manutenção dos demais itens do menu (technique group, technique, reference e certification) dizem respeito ao funcionamento da norma. É possível incluir, modificar e remover fases do ciclo de vida, técnicas, referências e SIL. A possibilidade de manutenção desses itens é muito interessante, pois viabiliza a atualização dos dados, caso haja modificações na norma.

6.3 Detalhes de implementação

O trabalho foi desenvolvido para web devido à sua natureza multiusuário, onde os diferentes papéis dos usuários demonstram que o software pode ser acessado de diferentes localizações físicas. Foi utilizada a linguagem PHP, com a ferramenta TemplatePower para facilitar a manipulação dos HTMLs e a ferramenta MDB::QueryTool para acessar o banco de dados. Também foi feito uso de AJAX para agilizar algumas partes da interação do usuário com o sistema.

6.4 Trabalhos futuros

A norma é bastante ampla, possibilitando várias linhas de trabalho. Uma possibilidade de trabalho seria ampliar o software SID, dando suporte também ao ciclo de vida de segurança completo e aos requisitos de hardware, detalhados nas normas IEC 61508-1 e 61508-2. Outra possibilidade de trabalho seria acrescentar ao software SID a gerência de vários projetos simultâneos. Também seria interessante que o projetista pudesse ter mais informações sobre o andamento do projeto, podendo estipular prazos através do software, que emitiria mensagens automáticas alertando atrasos.

CONCLUSÃO

Neste trabalho vários aspectos da norma IEC 61508 foram abordados, com intuito de esclarecer a complexidade da certificação de dispositivos de segurança crítica. A certificação é um processo longo, que exige organização, disciplina e muito conhecimento. Ao longo deste trabalho, pôde-se observar que a norma é bastante abrangente e aplicável em muitos setores, mas que possui alguns problemas.

A norma não é autocontida, exigindo que o conhecimento necessário para a certificação seja buscado também em outras normas. A norma também é vaga em relação à documentação, não oferecendo exemplos concretos de modelos de documentos, nem de justificativas para substituição de técnicas.

Alcançar a certificação é um processo difícil, mas ainda assim, é extremamente importante que as empresas brasileiras de desenvolvimento de software se atentem para esse ramo, pois o mercado nesse nicho é bastante amplo. Diversas empresas brasileiras e multinacionais utilizam dispositivos de segurança certificados, mas geralmente esses dispositivos são produzidos fora do Brasil.

O software SID, produzido nesse trabalho, visa clarificar a aplicação da norma e contribuir para o conhecimento da mesma. Além disso, as ferramentas existentes no mercado são proprietárias, o que torna o software SID uma alternativa para quem deseja aprender mais sobre a norma, ou aplicá-la num processo de certificação. É interessante que o conhecimento sobre normas de segurança seja disseminado no meio acadêmico, a fim de formar profissionais preparados e com visão mais abrangente em relação ao mercado de softwares de segurança.

REFERÊNCIAS

SMITH D.J.; SIMPSON, K.G.L.; **Functional Safety: a straightforward guide to applying IEC 61508 and related standards**, Elsevier, Butterworth-Heinemann, U.K. 2ªedição, 2004.

BARRY, R. **Compiler Verification for safety-critical applications** Embedded System Design Europe (www.embedded.com/europe), junho-julho 2007.

BELL, R. **Introduction to IEC 61508** ACS Workshop on Tools and Standards, Conference in Research and Practice in Information Technology, Vol. Nº 55, 2005.

BOWEN, J.; STAVRIDOU, V. **Safety-critical methods and systems, formal standards** Software Engineering Journal, julho 1993.

BROWN, S. **Overview of IEC 61508 Design of electrical/electronic/programmable electronic safety-related systems** IEEE Computing and Control, Engineering Journal, fevereiro 2000.

BLACK, W.S. **IEC 61508 – what it doesn't tell you** IEEE Computing and Control, Engineering Journal, fevereiro 2000.

DUNN, W. R. (2003). **Designing safety-critical computer systems**. IEEE Computer, 36(11):40 – 46. ISSN 0018-9162, novembro 2003.

FALLER, R. **Project Experience with IEC 61508 and Its Consequences** U. Voges (Ed.): SAFECOMP 2001, LNCS 2187, v. 2187 p. 200–214, 2001.

FENTON, N. E.; NEIL, M. **A Strategy for Improving Safety Related Software Engineering Standards,** IEEE Press – Transactions on Software Engineering, v. 24, n. 11, novembro 1998.

INTERNATIONAL ELECTROTECHNICAL COMISSION. **IEC 61508**: functional safety of eletrical/eletronic/programmable eletronic safety-related systems. Geneva, 2000-05.

JUNG, C. R.; OSÓRIO, F. S.; KELBER, C. R.; HEINEN, F. J. **Computação Embarcada: Projeto e Implementação de Veículos Autônomos Inteligentes** XXV Congresso da Sociedade Brasileira de Computação**,** julho 2005.

LADKIN, P. B.; **An Overview of IEC 61508 on E/E/PE Functional Safety,** Bielefeld**,** Germany**,** 2008, disponível em: **<**http://www.causalis.com/IEC61508FunctionalSafety.pdf **>**, acesso em: nov. 2010.

MCDERMID, J.A. **Software Safety: Where's the Evidence?** Proc. 6th Australian Workshop on Industrial Experience, v.3, 2001.

SIQUEIRA T. F.; MENEGOTTO, C.C.; WEBER, T. S.; NETTO, J.C.; WAGNER, F.R. **Desenvolvimento de Sistemas Embarcados para Aplicações Críticas** IV Escola Regional de Redes de Computadores, Passo Fundo, setembro 2006.

SCHMIDT R. **Certificação de Software para Aplicações Críticas utilizando a norma IEC 61508.** 2009**.** Projeto de Diplomação (Bacharelado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.

THOMAS, M. **Engineering Judgement** Proc. 9th Australian Workshop on Safety Related Programmable Systems (SCS'04), v.47, p.43-47, 2004.

WEBER, T.S. **Segurança Funcional Critica: Conceitos, Padrões e Medidas**, slides de curso ministrado para a empresa Altus.

Páginas na internet:[1]

Wikipedia: The Free Encyclopedia

http://en.wikipedia.org/wiki/Embedded_system/

IEC 61508 Functional Safety Zone (2005). Functional safety and IEC 61508.
http://www.iec.ch/zone/fsafety/

ISO, Internacional Organizations for Standardization

http://www.iso.org/

CSL - Critical Systems Labs

http://www.criticalsystemslabs.com/

IEEE Standard Association

http://standards.ieee.org/reading/ieee/std_public/description/se/1228-1994_desc.html

SEPT Software Engineering Process Technology

http://www.12207.com/safety.htm

1 Todas as páginas utilizadas nesta monografia foram consultadas entre agosto e novembro de 2010.

GLOSSÁRIO

ASICS: circuito integrado de aplicação específica. É um tipo de implementação de sistema embarcado denominado SoC (sistema em um chip).

Comissionamento: processo que visa assegurar que sistemas e componentes de uma unidade industrial sejam projetados, instalados, testados, operados e mantidos de acordo com as necessidades e requisitos operacionais do proprietário.

Dano: é a lesão física ou dano à saúde das pessoas, quer direta ou indiretamente, resultado de avaria à propriedade ou ao meio ambiente.

Evento perigoso: uma situação perigosa que resulta em dano.

Perigo: potencial fonte de dano.

Risco: combinação da probabilidade de ocorrência de danos e da gravidade destes danos.

Risco tolerável: risco que é aceitável em um determinado contexto, com base nos valores atuais da sociedade.

Segurança: livre de risco inaceitável.

ANEXO A – TABELAS DO ANEXO A DA IEC 61508-3

Table A.1 – Software safety requirements specification (see 7.2)

	Technique/Measure*	Ref.	SIL1	SIL2	SIL3	SIL4
1	Computer-aided specification tools	B.2.4	R	R	HR	HR
2a	Semi-formal methods	Table B.7	R	R	HR	HR
2b	Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	C.2.4	---	R	R	HR
NOTE 1 – The software safety requirements specification will always require a description of the problem in natural language and any necessary mathematical notation that reflects the application. NOTE 2 – The table reflects additional requirements for specifying the software safety requirements clearly and precisely.						
* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. Only one of the alternate or equivalent techniques/measures has to be satisfied.						

Table A.2 – Software design and development: software architecture design (see 7.4.3)

	Technique/Measure*	Ref	SIL1	SIL2	SIL3	SIL4
1	Fault detection and diagnosis	C.3.1	---	R	HR	HR
2	Error detecting and correcting codes	C.3.2	R	R	R	HR
3a	Failure assertion programming	C.3.3	R	R	R	HR
3b	Safety bag techniques	C.3.4	---	R	R	R
3c	Diverse programming	C.3.5	R	R	R	HR
3d	Recovery block	C.3.6	R	R	R	R
3e	Backward recovery	C.3.7	R	R	R	R
3f	Forward recovery	C.3.8	R	R	R	R
3g	Re-try fault recovery mechanisms	C.3.9	R	R	R	HR
3h	Memorising executed cases	C.3.10	---	R	R	HR
4	Graceful degradation	C.3.11	R	R	HR	HR
5	Artificial intelligence - fault correction	C.3.12	---	NR	NR	NR
6	Dynamic reconfiguration	C.3.13	---	NR	NR	NR
7a	Structured methods including for example, JSD, MASCOT, SADT and Yourdon.	C.2.1	HR	HR	HR	HR
7b	Semi-formal methods	Table B.7	R	R	HR	HR
7c	Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	C.2.4	---	R	R	HR
8	Computer-aided specification tools	B.2.4	R	R	HR	HR

Table A.3 – Software design and development: support tools and programming language (see 7.4.4)

	Technique/Measure*	Ref	SIL1	SIL2	SIL3	SIL4
1	Suitable programming language	C.4.6	HR	HR	HR	HR
2	Strongly typed programming language	C.4.1	HR	HR	HR	HR
3	Language subset	C.4.2	---	---	HR	HR
4a	Certificated tools	C.4.3	R	HR	HR	HR
4b	Tools: increased confidence from use	C.4.4	HR	HR	HR	HR
5a	Certificated translator	C.4.3	R	HR	HR	HR
5b	Translator: increased confidence from use	C.4.4	HR	HR	HR	HR
6	Library of trusted/verified software modules and components	C.4.5	R	HR	HR	HR

Table A.4 – Software design and development: detailed design (see 7.4.5 and 7.4.6)
(This includes software system design, software module design and coding)

	Technique/Measure*	Ref	SIL1	SIL2	SIL3	SIL4
1a	Structured methods including for example, JSD, MASCOT, SADT and Yourdon	C.2.1	HR	HR	HR	HR
1b	Semi-formal methods	Table B.7	R	HR	HR	HR
1c	Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	C.2.4	---	R	R	HR
2	Computer-aided design tools	B.3.5	R	R	HR	HR
3	Defensive programming	C.2.5	---	R	HR	HR
4	Modular approach	Table B.9	HR	HR	HR	HR
5	Design and coding standards	Table B.1	R	HR	HR	HR
6	Structured programming	C.2.7	HR	HR	HR	HR
7	Use of trusted/verified software modules and components (if available)	C.2.10 C.4.5	R	HR	HR	HR

Table A.5 – Software design and development: software module testing and integration (see 7.4.7 and 7.4.8)

	Technique/Measure*	Ref	SIL1	SIL2	SIL3	SIL4
1	Probabilistic testing	C.5.1	---	R	R	HR
2	Dynamic analysis and testing	B.6.5 Table B.2	R	HR	HR	HR
3	Data recording and analysis	C.5.2	HR	HR	HR	HR
4	Functional and black box testing	B.5.1 B.5.2 Table B.3	HR	HR	HR	HR
5	Performance testing	C.5.20 Table B.6	R	R	HR	HR
6	Interface testing	C.5.3	R	R	HR	HR

Table A.6 – Programmable electronics integration (hardware and software) (see 7.5)

	Technique/Measure*	Ref	SIL1	SIL2	SIL3	SIL4
1	Functional and black box testing	B.5.1 B.5.2 Table B.3	HR	HR	HR	HR
2	Performance testing	C.5.20 Table B.6	R	R	HR	HR

Table A.7 – Software safety validation (see 7.7)

	Technique/Measure*	Ref	SIL1	SIL2	SIL3	SIL4
1	Probabilistic testing	C.5.1	---	R	R	HR
2	Simulation/modelling	Table B.5	R	R	HR	HR
3	Functional and black-box testing	B.5.1 B.5.2 Table B.3	HR	HR	HR	HR
* A numbered technique/measure shall be selected according to the safety integrity level. NOTE – Appropriate techniques/measures shall be selected according to the safety integrity level.						

Table A.8 – Modification (see 7.8)

	Technique/Measure*	Ref	SIL1	SIL2	SIL3	SIL4
1	Impact analysis	C.5.23	HR	HR	HR	HR
2	Reverify changed software module	C.5.23	HR	HR	HR	HR
3	Reverify affected software modules	C.5.23	R	HR	HR	HR
4	Revalidate complete system	C.5.23	---	R	HR	HR
5	Software configuration management	C.5.24	HR	HR	HR	HR
6	Data recording and analysis	C.5.2	HR	HR	HR	HR

Table A.9 – Software verification (see 7.9)

	Technique/Measure*	Ref	SIL1	SIL2	SIL3	SIL4
1	Formal proof	C.5.13	---	R	R	HR
2	Probabilistic testing	C.5.1	---	R	R	HR
3	Static analysis	B.6.4 Table B.8	R	HR	HR	HR
4	Dynamic analysis and testing	B.6.5 Table B.2	R	HR	HR	HR
5	Software complexity metrics	C.5.14	R	R	R	R
Software module testing and integration		See table A.5				
Programmable electronics integration testing		See table A.6				
Software system testing (validation)		See table A.7				

NOTE 1 – For convenience all verification activities have been drawn together under this table. However, this does not place additional requirements for the dynamic testing element of verification in table A.5 and table A.6 which are verification activities in themselves. Nor does this table require verification testing in addition to software validation (see table A.7), which in this standard is the demonstration of conformance to the safety requirements specification (end-end verification).

NOTE 2 – Verification crosses the boundaries of IEC 61508-1, IEC 61508-2 and IEC 61508-3. Therefore the first verification of the safety-related system is against the earlier system level specifications.

NOTE 3 – In the early phases of the software safety lifecycle verification is static, for example inspection, review, formal proof. When code is produced dynamic testing becomes possible. It is the combination of both types of information that is required for verification. For example code verification of a software module by static means includes such techniques as software inspections, walk-throughs, static analysis, formal proof. Code verification by dynamic means includes functional testing, white-box testing, statistical testing. It is the combination of both types of evidence that provides assurance that each software module satisfies its associated specification.

* A numbered technique/measure shall be selected according to the safety integrity level.

Appropriate techniques/measures shall be selected according to the safety integrity level.

Table A.10 – Functional safety assessment (see clause 8)

	Assessment/Technique*	Ref	SIL1	SIL2	SIL3	SIL4
1	Checklists	B.2.5	R	R	R	R
2	Decision/truth tables	C.6.1	R	R	R	R
3	Software complexity metrics	C.5.14	R	R	R	R
4	Failure analysis	Table B.4	R	R	HR	HR
5	Common cause failure analysis of diverse software (if diverse software is actually used)	C.6.3	---	R	HR	HR
6	Reliability block diagram	C.6.5	R	R	R	R