

(PÁGINA EM BRANCO)

CONTEÚDO

PREFÁCIO.....	4
INTRODUÇÃO.....	6
1 Escopo.....	8
1.1 * Propósito.....	8
1.2 * Área de Aplicação.....	8
1.3 Relacionamento com outras normas.....	8
1.4 Conformidade.....	8
2 * Referências normativas.....	9
3 * Termos e definições.....	9
4 * Requisitos gerais.....	13
4.1 * Sistema de gestão da qualidade.....	13
4.2 * GERENCIAMENTO DE RISCO.....	13
4.3 * Classificação de segurança de software.....	13
5 PROCESSO de desenvolvimento de software.....	14
5.1 * Planejamento de desenvolvimento de software.....	14
5.2 * Análise de requisitos de software.....	17
5.3 * Projeto da arquitetura de software.....	20
5.4 * Projeto detalhado do software.....	20
5.5 * Implementação e verificação da UNIDADE DE SOFTWARE.....	21
5.6 * Integração de software e ensaio de integração.....	22
5.7 * Ensaio do SISTEMA DE SOFTWARE.....	23
5.8 * Liberação do software.....	24
6 PROCESSO de manutenção do software.....	25
6.1 * Estabelecimento do plano de manutenção de software.....	25
6.2 * Análise de problemas e modificações.....	25
6.3 * Implementação de modificações.....	26
7 * PROCESSO DE GERENCIAMENTO DE RISCO de software.....	26
7.1 * Análise de softwares que contribuem para situações perigosas.....	26
7.2 Medidas de CONTROLE DE RISCO.....	26
7.3 Medidas de VERIFICAÇÃO de CONTROLE DE RISCO.....	27
7.4 GERENCIAMENTO DE RISCO de mudanças de software.....	27
8 *PROCESSO de gerenciamento de configuração de software.....	28
8.1 * Identificação de configuração.....	28
8.2 * Controle de mudanças.....	28
8.3 * Contabilidade do status de configuração.....	29
9 * PROCESSO de resolução de problema de software.....	29
9.1 Elaborar RELATÓRIOS DE PROBLEMA.....	29
9.2 Investigar o problema.....	29
9.3 Informar as partes interessadas.....	29
9.4 Processo de controle de alteração de uso.....	29
9.5 Manter registros.....	29
9.6 Problemas para análise de tendências.....	30
9.7 Verificar a resolução de problemas de software.....	30
9.8 Conteúdo da documentação de testes.....	30
Anexo A (informativo) Justificativa para os requisitos desta norma.....	31
Anexo B (informativo) Orientações sobre as disposições da presente norma.....	33
Anexo C (informativo) Relacionamento com outras normas.....	46
Anexo D (informativo) Implementação.....	65
Bibliografia.....	67
Índice dos termos definidos.....	68

Figura 1 – Visão global dos PROCESSOS e ATIVIDADES do desenvolvimento de software.....	6
Figura 2 - Visão global dos PROCESSOS e ATIVIDADES da manutenção de software	7
Figura B.1 – Exemplo de particionamento de ITENS DE SOFTWARE.....	37
Figura C.1 – Relacionamento das principais normas de PRODUTOS PARA A SAÚDE com a IEC 62304....	46
Figura C.2 – Software como parte do Modelo-V.....	49
Figura C.3 – Aplicação da IEC 62304 com IEC 61010-1.....	57

Tabela A.1 – Resumo dos requisitos por classe de segurança de software.....	32
Tabela B.1 – Estratégias de desenvolvimento (modelo) conforme definido na ISO/IEC 12207.....	33
Tabela C.1 – Relacionamento com ISO 13485:2003.....	47
Tabela C.2 – Relacionamento com ISO 14971:2000.....	48
Tabela C.3 – Relacionamento com IEC 60601-1.....	50
Tabela C.4 – Relacionamento com IEC 60601-1-4.....	55
Tabela C.5 – Relacionamento com ISO/IEC 12207.....	58
Tabela D.1 – Check-list para pequenas empresas sem certificação QMS.....	66

COMISSÃO ELETROTÉCNICA INTERNACIONAL

SOFTWARE DE PRODUTO PARA A SAÚDE – PROCESSO DO CICLO DE VIDA DO SOFTWARE

PREFÁCIO

- 1) A Comissão Eletrotécnica Internacional (IEC) é uma organização de âmbito mundial para a normatização que engloba todos os comitês eletrotécnicos nacionais (Comitês Nacionais IEC). O objetivo da IEC é promover a cooperação internacional em todas as questões relativas à normatização nas áreas elétrica e eletrônica. Com este objetivo, além de outras atividades, A IEC publica Normas Internacionais, Especificações Técnicas, Relatórios Técnicos, Especificações Disponíveis Publicamente (EDP (PAS)) e Guias (daqui por diante denominados "A(s) Publicação(ões) da IEC"). Sua preparação é confiada aos comitês técnicos; qualquer Comitê Nacional da IEC interessado no assunto abordado pode participar deste trabalho preparatório. Organizações internacionais, governamentais e não-governamentais parceiras do IEC também participam desta preparação. A IEC colabora bastante com a Organização Internacional para Normatização (International Organization for Standardization (ISO)), seguindo as condições determinadas pelo acordo entre as duas organizações.
- 2) As decisões formais ou acordos da IEC relacionados a assuntos técnicos expressam, tanto quanto possível, o consenso internacional sobre os assuntos relevantes, já que cada comitê técnico possui representação de todos os Comitês Nacionais da IEC interessados.
- 3) As Publicações da IEC têm o formato de recomendações para utilização internacional e são aceitas pelos Comitês Nacionais da IEC neste sentido. Embora todos os esforços razoáveis sejam empregados para garantir que o conteúdo técnico das Publicações IEC seja exato, a IEC não é responsável pela forma com que estas são empregadas ou por qualquer falha de interpretação do usuário final.
- 4) Visando promover a uniformidade internacional, os Comitês Nacionais da IEC se esforçam para aplicar as Publicações IEC da forma mais transparente possível em suas publicações nacionais e regionais. Qualquer divergência entre uma Publicação da IEC e sua publicação correspondente, nacional ou regional, deve estar claramente indicada na última.
- 5) A própria IEC não fornece qualquer atestado de conformidade. Entidades de certificação independentes fornecem serviços de avaliação de conformidade e, em algumas áreas, acesso a rótulos de conformidade com a IEC. A IEC não é responsável por quaisquer serviços executados por entidades de certificação independentes.
- 6) Todos os usuários devem se certificar de estarem com a versão mais atualizada desta publicação.
- 7) Nenhuma responsabilidade legal pode ser atribuída à IEC ou a seus diretores, funcionários, servidores ou agentes, incluindo especialistas e membros dos comitês técnicos e dos Comitês Nacionais da IEC em relação a lesões corporais, danos à propriedade ou outros danos de qualquer natureza, diretos ou indiretos, ou em relação a custos e a despesas (incluindo-se as despesas legais) que possam ser decorrentes da publicação, do uso ou da dependência desta Publicação da IEC ou de quaisquer outras Publicações da IEC.
- 8) Deve-se prestar atenção às Referências Normativas citadas nesta publicação. A utilização das publicações referenciadas é indispensável à aplicação correta desta publicação.
- 9) Deve-se prestar atenção à possibilidade de que alguns elementos desta Publicação da IEC estejam sujeitos a direitos de patentes. A IEC não se responsabiliza pela identificação de quaisquer destes direitos de patentes.

A norma internacional IEC 62304 foi preparada por um grupo de trabalho conjunto do subcomitê 62A: Aspectos comuns aos equipamentos elétricos utilizados na prática médica do comitê técnico 62: Equipamentos elétricos na prática médica e Comitê Técnico 210 da ISO, Gestão da qualidade e aspectos gerais e aspectos gerais correspondentes de PRODUTOS PARA A SAÚDE. A Tabela C.5 foi preparada pela ISO/IEC JTC 1/SC 7, Software e engenharia de sistemas.

Ela foi publicada como uma norma de logotipo duplo.

O texto desta norma colateral foi baseado nos seguintes documentos da IEC:

FDIS	Relatório sobre a votação
62A/523/FDIS	62A/528/RVD

Informações completas sobre a votação para a aprovação desta norma colateral podem ser encontradas no relatório sobre a votação indicado na tabela acima. Na ISO, a norma foi aprovada por 23 Membros-P, de um total de 23 que votaram.

Esta publicação foi redigida, de acordo com as Diretrizes da ISO/IEC, Parte 2.

Nesta norma colateral, são utilizados os seguintes tipos:

- prescrições e definições: em tipo romano;
- material informativo aparecendo fora das tabelas, tais como notas, exemplos e referências: em tipo menor. Texto normativo de tabelas também em tipo menor;
- termos usados ao longo desta norma que foram definidos na Cláusula 3 e também determinados neste índice: em versalete.

Um asterisco (*) como primeiro caractere de um título ou no início de um parágrafo indica que há um guia relacionado a aquele item no Anexo B.

O comitê decidiu que o conteúdo desta publicação permanecerá inalterado até a data de estabilidade indicada na página da IEC na internet no endereço "<http://webstore.iec.ch>" nos dados relacionados à publicação em específico. Nesta data a publicação será

- reconfirmada;
- recolhida;
- substituída por uma edição revisada, ou
- alterada.

INTRODUÇÃO

O software é muitas vezes parte integrante da tecnologia de um PRODUTO PARA A SAÚDE. Determinar a SEGURANÇA e eficácia de um PRODUTO PARA A SAÚDE contendo um software requer o conhecimento do uso pretendido do software e demonstração de que o uso do software preenche estas intenções, sem causar quaisquer RISCOS inaceitáveis.

Esta norma estabelece uma plataforma de PROCESSOS de ciclo de vida com ATIVIDADES e TAREFAS necessárias para o projeto e manutenção seguros de um PRODUTO PARA A SAÚDE. Esta norma fornece os requisitos para cada PROCESSO do ciclo de vida. Cada PROCESSO do ciclo de vida é ainda dividido em um conjunto de ATIVIDADES, com a maioria das ATIVIDADES sendo por sua vez divididas em um conjunto de TAREFAS.

Se o software é um fator que contribui para um perigo, é determinado durante a ATIVIDADE de identificação de PERIGO do PROCESSO DE GERENCIAMENTO DE RISCO. PERIGOS que poderiam ser indiretamente causados por software (por exemplo, fornecer informações errôneas que poderia causar tratamento inapropriado a ser administrado) precisam ser considerados ao determinar se o software é um fator contribuinte. A decisão de usar o software para controlar o RISCO é feita durante a ATIVIDADE DE CONTROLE DE RISCO do PROCESSO DE GERENCIAMENTO DE RISCO. O PROCESSO DE GERENCIAMENTO DE RISCO de software requerido nesta norma tem que estar incorporado no PROCESSO DE GERENCIAMENTO DE RISCO do produto de acordo com a ISO 14971.

O PROCESSO de desenvolvimento do software consiste de uma série de ATIVIDADES. Essas ATIVIDADES são mostradas na Figura 1 e descritas na Cláusula 5. Como muitos incidentes no campo estão relacionados a serviços ou à manutenção do SISTEMA de PRODUTO PARA A SAÚDE, incluindo atualizações de software e melhorias inapropriadas, o PROCESSO de manutenção do software é considerado tão importante quanto o PROCESSO de desenvolvimento do software. O PROCESSO de manutenção do software é muito similar ao PROCESSO de desenvolvimento do software. Isto é mostrado na Figura 2 e descrito na Cláusula 6.

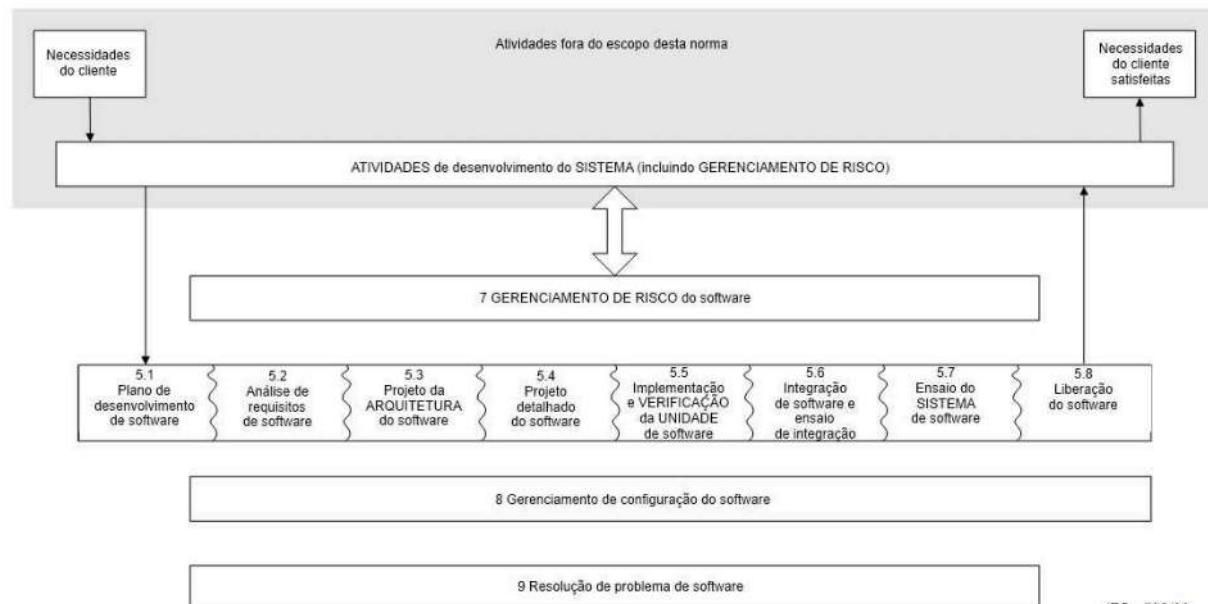


Figura 1 – Visão global das ATIVIDADES e PROCESSOS de desenvolvimento do software

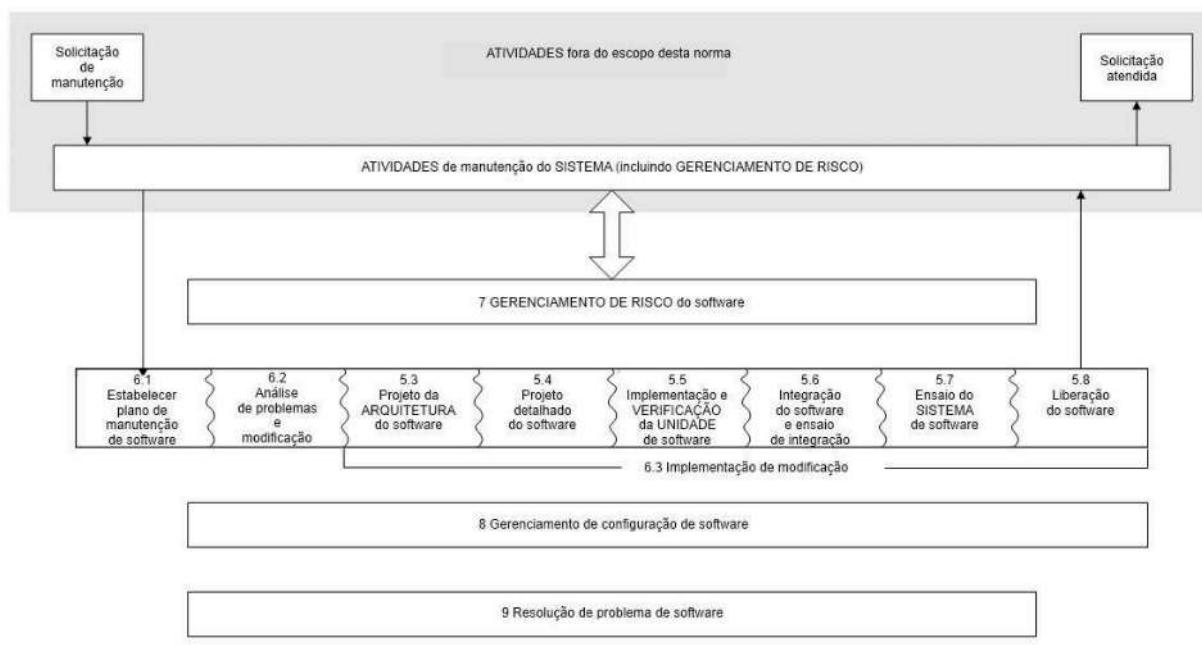


Figura 2 – Visão global das ATIVIDADES e PROCESSOS de manutenção do software

Esta norma identifica dois PROCESSOS adicionais considerados essenciais para o desenvolvimento seguro do SOFTWARE DE PRODUTO PARA A SAÚDE. Eles são o PROCESSO de gerenciamento da configuração do software (Cláusula 8) e o PROCESSO de resolução de problema do software (Cláusula 9).

Esta norma não especifica uma estrutura organizacional para o FABRICANTE ou qual a parte da organização executa qual PROCESSO, ATIVIDADE ou TAREFA. Esta norma requer somente que o PROCESSO, ATIVIDADE ou TAREFA seja completado de forma a estabelecer conformidade com esta norma.

Esta norma não prescreve o nome, o formato, ou explicita o conteúdo da documentação a ser produzida. Esta norma requer a documentação das TAREFAS, mas a decisão de como formatar essa documentação é deixada a cargo do usuário da norma.

Esta norma não prescreve um modelo de ciclo de vida específico. Os usuários desta norma são responsáveis por selecionar um modelo de ciclo de vida para o projeto do software e por mapear os PROCESSOS, as ATIVIDADES, e as TAREFAS desta norma no modelo.

O Anexo A apresenta a justificativa para as cláusulas desta norma. O Anexo B apresenta uma orientação sobre as disposições desta norma.

Para os propósitos desta norma:

- “deve” significa que a conformidade com um requisito é mandatória para conformidade com esta norma;
 - “deveria” significa que a conformidade com um requisito é recomendada mas não é mandatória para conformidade com esta norma;
 - “pode” é usado para descrever uma possível forma de atingir conformidade com um requisito;
 - “estabelecer” significa definir, documentar, e implementar; e
 - quando esta norma usa o termo “como apropriado” em conjunto com um PROCESSO, uma ATIVIDADE, uma TAREFA ou saída requeridos, a intenção é que o FABRICANTE deve usar o PROCESSO, a ATIVIDADE, a TAREFA ou saída, a menos que o FABRICANTE possa documentar uma justificativa para não proceder desta forma.

SOFTWARE DE PRODUTOS PARA A SAÚDE – PROCESSOS DE CICLO DE VIDA DE SOFTWARE

1 Escopo

1.1 * Propósito

Esta norma define os requisitos de ciclo de vida para os software DE produto PARA A SAÚDE. O conjunto de PROCESSOS, ATIVIDADES e TAREFAS descritas nesta norma estabelece uma plataforma comum para os PROCESSOS de ciclo de vida de software DE produto PARA A SAÚDE.

1.2 * Área de Aplicação

Esta norma é voltada para o desenvolvimento e manutenção de software DE produto PARA A SAÚDE.

Esta norma é voltada para o desenvolvimento e manutenção de software DE produto PARA A SAÚDE quando o software é em si um PRODUTO PARA A SAÚDE ou quando o software é uma parte incorporada ou integral de um produto PARA A SAÚDE final.

Esta norma não abrange a validação e versão final de um produto PARA A SAÚDE, mesmo quando o produto PARA A SAÚDE consiste inteiramente de software.

1.3 Relacionamento com outras normas

Esta norma de ciclo de vida de software DE produto PARA A SAÚDE é para ser usado junto com outras normas apropriadas quando do desenvolvimento de um produto PARA A SAÚDE. O Anexo C mostra a relação entre esta norma e outras normas relevantes.

1.4 Conformidade

A conformidade com esta norma é definida como a implementação de todos os PROCESSOS, ATIVIDADES e TAREFAS identificadas nesta norma de acordo com a classe de segurança de software.

NOTA As classes de segurança do software relacionadas com cada requisito são identificadas no texto normativo após o requisito.

A conformidade é determinada pela inspeção de toda documentação requerida por esta norma incluindo o ARQUIVO DE GERENCIAMENTO DE RISCO, e avaliando os PROCESSOS, ATIVIDADES e TAREFAS requeridas pela classe de segurança de software. Veja Anexo D.

NOTA 1 Essa avaliação poder ser executada por auditoria interna ou externa.

NOTA 2 Embora os PROCESSOS, ATIVIDADES e TAREFAS especificadas sejam executados, existe flexibilidade nos métodos de implementação desses PROCESSOS e na execução dessas ATIVIDADES e TAREFAS.

NOTA 3 Sempre que quaisquer requisitos contiverem “como apropriado” e não forem executados, a documentação para a justificativa é necessária para essa avaliação.

NOTA 4 O termo “conformidade” (conformance) é usado na ISO/IEC 12207 onde o termo “cumprimento” (compliance) é usado nesta norma.

2 * Referências normativas

Os documentos de referência relacionados a seguir são indispensáveis para a aplicação deste documento. Para referências datadas, somente a versão citada se aplica. Para referências sem data, a última edição do documento de referência (incluindo qualquer adendo) se aplica.

ISO 14971, *PRODUTOS PARA A SAÚDE – Aplicação de gerenciamento de risco em PRODUTOS PARA A SAÚDE*.

3 * Termos e definições

Para efeitos deste documento, a seguinte terminologia e definições são aplicáveis.

3.1

ATIVIDADE

o conjunto de uma ou mais TAREFAS interrelacionadas ou interativas

3.2

ANOMALIA

qualquer condição que desvie do esperado baseado em requisitos especificados, documentos de projeto, padrões, etc. ou a partir da percepção ou experiência de alguém. ANOMALIAS podem ser encontradas durante, mas não limitada a, revisão, teste, análise, compilação ou uso de PRODUTOS DE SOFTWARE ou documentação aplicável

[IEEE 1044:1993, definição 3.1]

3.3

ARQUITETURA

estrutura organizacional de um SISTEMA ou componente

[IEEE 610.12:1990]

3.4

SOLICITAÇÃO DE MUDANÇA

especificação documentada de uma mudança a ser feita em um PRODUTO DE SOFTWARE

3.5

ITEM DE CONFIGURAÇÃO

entidade que pode ser identificada de forma única em um dado ponto de referência

NOTA Baseado na ISO/IEC 12207:1995, definição 3.6.

3.6

ITEM DE ENTREGA

resultado ou produto (inclui documentação) de uma ATIVIDADE ou TAREFA

3.7

AVALIAÇÃO

determinação sistemática da extensão com que uma entidade atende suas especificações

[ISO/IEC 12207:1995, definição 3.9]

3.8

DANO

ferimento ou prejuízo físico, ou ambos, à saúde de uma pessoa ou prejuízo a um bem material ou ao meio ambiente

[ISO/IEC Guide 51:1999, definição 3.3]

3.9

PERIGO

potencial fonte de DANO

[ISO/IEC Guide 51:1999, definition 3.5]

3.10

FABRICANTE

pessoa física ou jurídica responsável por projetar, fabricar, embalar ou rotular um PRODUTO PARA A SAÚDE; montar um SISTEMA; ou adaptar um PRODUTO PARA A SAÚDE antes do mesmo ser colocado no mercado e/ou colocado em serviço, independente se sua operação será realizada por aquela pessoa ou por um terceiro autorizado pela mesma.

[ISO 14971:2000, definição 2.6]

3.11

PRODUTO PARA A SAÚDE

qualquer instrumento, aparato, utensílio, máquina, equipamento, implante, reagente ou calibrador in vitro, software, material ou outro artigo similar ou relacionado, definido pelo FABRICANTE para ser utilizado, sozinho ou em combinação, por seres humanos para um ou mais dos propósitos específicos de

- diagnóstico, prevenção, monitoração, tratamento ou alívio de doença,
- diagnóstico, monitoração, tratamento, alívio de ou compensação de uma lesão,
- investigação, substituição, modificação ou manutenção da anatomia ou de um PROCESSO fisiológico,
- apoio ou sustentação da vida,
- controle da concepção,
- desinfecção de PRODUTOS PARA A SAÚDE,
- disponibilização de informação para propósitos médicos através de exame in-vitro de amostras provenientes do corpo humano

e aquele que não atinge sua ação pretendida primária no ou sobre o corpo humano por meios farmacológicos, imunológicos ou metabólicos, mas que pode ser auxiliado em suas funções por esses meios

NOTA 1 Esta definição foi elaborada pela Global Harmonization Task Force (GHTF). Veja referência bibliográfica [15] (em ISO 13485:2003).

[ISO 13485:2003, definição 3.7]

NOTA 2 Algumas diferenças podem ocorrer nas definições utilizadas na normativa de cada país.

3.12

SOFTWARE DE PRODUTO PARA A SAÚDE

SISTEMA DE SOFTWARE desenvolvido com a finalidade de ser incorporado em um PRODUTO PARA A SAÚDE em desenvolvimento ou de ser utilizado como PRODUTO PARA A SAÚDE em sua própria forma.

3.13

RELATÓRIO DE PROBLEMA

um registro do comportamento real ou potencial de um PRODUTO DE SOFTWARE que um usuário ou outra pessoa interessada acredita ser inseguro, inapropriado para o uso pretendido ou contrário às especificações

NOTA 1 Esta norma não requer que um RELATÓRIO DE PROBLEMA resulte em mudança do PRODUTO DE SOFTWARE. O FABRICANTE pode rejeitar o RELATÓRIO DE PROBLEMA por considerar que o mesmo apresenta mal entendido, erro ou evento insignificante.

NOTA 2 Um RELATÓRIO DE PROBLEMA pode estar relacionado a um PRODUTO DE SOFTWARE lançado ou a um PRODUTO DE SOFTWARE que ainda se encontra em desenvolvimento.

NOTA 3 Esta norma requer que o FABRICANTE execute um procedimento passo a passo de decisão adicional (ver Cláusula 6) para um RELATÓRIO DE PROBLEMAS relacionado a um produto lançado de modo a garantir que ações regulatórias sejam identificadas e implementadas.

3.14

PROCESSO

um conjunto de ATIVIDADES interrelacionadas ou interativas que transformam entradas em saídas

[ISO 9000:2000, definição 3.4.1]

NOTA O termo "ATIVIDADES" cobre o uso de recursos.

3.15

ENSAIO DE REGRESSÃO

testes necessários para se determinar se a mudança em um componente do SISTEMA não afetou negativamente funcionalidade, confiabilidade ou desempenho, e não introduziu problemas adicionais

[ISO/IEC 90003:2004, definição 3.11]

3.16

RISCO

combinação da probabilidade de ocorrência de um DANO com a severidade deste DANO

[ISO/IEC Guia 51:1999 definição 3.2]

3.17

ANÁLISE DE RISCO

uso sistemático das informações disponíveis para identificar PERIGOS e estimar o RISCO

[ISO/IEC Guia 51:1999 definição 3.10]

3.18

CONTROLE DE RISCO

PROCESSO no qual decisões são tomadas e RISCOS são reduzidos a, ou mantidos em, níveis especificados

[ISO 14971:2000 definição 2.16, modificado]

3.19

GERENCIAMENTO DE RISCO

aplicação sistemática de políticas, procedimentos e práticas de gerenciamento a TAREFAS para análise, avaliação e controle do RISCO

[ISO 14971:2000 definição 2.18]

3.20

ARQUIVO DE GERENCIAMENTO DE RISCO

conjunto de registros e outros documentos, não necessariamente contínuos, produzidos pelo PROCESSO DE GERENCIAMENTO DE RISCO

[ISO 14971:2000 definição 2.19]

3.21

SEGURO

livre de RISCOS inaceitáveis

[ISO/IEC Guia 51:1999 definição 3.1]

3.22

SEGURANÇA

proteção de dados e informações para que pessoas ou SISTEMAS não autorizados não possam lê-los ou modificá-los e para que pessoas ou SISTEMAS autorizados não tenham acesso negado a eles

[ISO/IEC 12207:1995 definição 3.25]

3.23

FERIMENTO SÉRIO

ferimento ou doença que direta ou indiretamente:

- a) representa risco à vida,
- b) resulta em prejuízo permanente de uma função do corpo ou dano permanente a uma estrutura do corpo, ou
- c) necessitar intervenção médica ou cirúrgica para prevenir prejuízo permanente de uma função do corpo ou dano permanente de uma estrutura do corpo

NOTA Prejuízo permanente significa um dano ou prejuízo irreversível para uma estrutura ou função do corpo excluindo danos ou prejuízos triviais.

3.24

MODELO DE CICLO DE VIDA DO DESENVOLVIMENTO DE SOFTWARE

estrutura conceitual abrangendo a voda de um software da definição de seus requisitos até sua liberação para fabricação, que:

- identifica PROCESSOS, ATIVIDADES e TAREFAS envolvidas no desenvolvimento de um PRODUTO DE SOFTWARE,
- descreve a sequência de e dependência entre as ATIVIDADES e TAREFAS, e
- identifica as etapas em que a totalidade dos resultados especificados é verificada.

NOTA Baseado na ISO/IEC 12207:1995, definição 3.11

3.25

ITEM DE SOFTWARE

qualquer parte identificável de um programa de computador

[ISO/IEC 90003:2004 definição 3.14 modificada]

NOTA Três termos identificam a decomposição de um software. O nível superior é o SISTEMA DE SOFTWARE. O nível inferior que não é mais decomposto é a UNIDADE DE SOFTWARE. Todos os níveis de composição, incluindo os níveis superior e inferior, podem ser chamados ITENS DE SOFTWARE. Um SISTEMA DE SOFTWARE, então, é composto de um ou mais ITENS DE SOFTWARE, e cada ITEM DE SOFTWARE é composto de uma ou mais UNIDADES DE SOFTWARE ou ITENS DE SOFTWARE decomponíveis. É dada ao FABRICANTE a responsabilidade de fornecer a definição e detalhamento dos ITENS DE SOFTWARE e UNIDADES DE SOFTWARE.

3.26

PRODUTO DE SOFTWARE

conjunto de programas de computador, procedimentos e documentação e dados possivelmente associados

[ISO/IEC 12207:1995 definição 3.26]

3.27

SISTEMA DE SOFTWARE

coleção integrada de ITENS DE SOFTWARE organizada para desempenhar uma função específica ou um conjunto de funções

3.28

UNIDADE DE SOFTWARE

ITEM DE SOFTWARE que não pode ser subdividido em outros itens

3.29

SDPD

software de procedência desconhecida (acrônimo)

ITEM DE SOFTWARE que já está desenvolvido e geralmente disponível e que não foi desenvolvido para o propósito de ser incorporado a um PRODUTO PARA A SAÚDE (também conhecido como software “de prateleira”) ou software previamente desenvolvido para o qual os registros adequados do seu desenvolvimento não estão disponíveis

3.30

SISTEMA

composto integrado consistindo de um ou mais dos PROCESSOS, hardware, software, instalações e pessoas, que oferece uma capacidade para satisfazer uma necessidade declarada ou objetivo

3.31

TAREFA

uma parte simples de trabalho que precisa ser realizada

3.32

RASTREABILIDADE

grau em que uma relação pode ser estabelecida entre dois ou mais produtos do PROCESSO de desenvolvimento

[IEEE 610.12:1990]

3.33

VERIFICAÇÃO

confirmação através do fornecimento de evidência objetiva de que os requisitos especificados foram cumpridos

NOTA 1 “Verificado” é usado para designar uma situação correspondente.

[ISO 9000:2000, definição 3.8.4]

NOTA 2 No projeto e desenvolvimento, VERIFICAÇÃO diz respeito ao PROCESSO de examinar o resultado de uma dada ATIVIDADE para determinar a conformidade com os requisitos expressos para a ATIVIDADE.

3.34

VERSÃO

instância identificada de um ITEM DE CONFIGURAÇÃO

NOTE1 Modificação para uma VERSÃO de um PRODUTO DE SOFTWARE, resultando em uma nova versão, requer a ação de gerenciamento de configuração de software.

NOTA 2 Baseado na ISO/IEC 12207:1995, definição 3.37.

4 * Requisitos gerais

4.1 * Sistema de gestão da Qualidade

O FABRICANTE do SOFTWARE DE PRODUTO PARA A SAÚDE deve demonstrar a capacidade de prover um SOFTWARE DE PRODUTO PARA A SAÚDE que satisfaça consistentemente os requisitos dos consumidores e os requisitos regulatórios aplicáveis.

NOTA 1 A demonstração desta capacidade pode ser através do uso de um sistema de gerenciamento de qualidade que cumpre com:

- ISO 13485[7]; ou
- uma norma nacional de sistema de gerenciamento de qualidade; ou
- uma regulamentação nacional de um sistema de gerenciamento da qualidade.

NOTA 2 Um Guia para aplicação dos requisitos de um sistema de gerenciamento da qualidade para software pode ser encontrado na ISO/IEC 90003 [11].

4.2 * GERENCIAMENTO DE RISCO.

O FABRICANTE deve aplicar o PROCESSO DE GERENCIAMENTO DE RISCO conforme a ISO 14971.

4.3 * Classificação de segurança do software

a) O FABRICANTE deve especificar para cada SISTEMA DE SOFTWARE uma classe de segurança (A, B ou C) de acordo com os efeitos possíveis no paciente, operador, ou outra pessoa, resultado de um PERIGO para qual o SISTEMA DE SOFTWARE possa ter contribuído.

A classe de segurança do software deve inicializar a especificação baseada nas severidades seguintes:

Classe A: Sem possíveis ferimentos ou danos para a saúde

Classe B: Nenhum ferimento sério possível

Classe C: Morte ou ferimentos sérios possível

Se um PERIGO pode surgir de uma falha de um SISTEMA DE SOFTWARE comportando-se como especificado, a probabilidade desta falha deve ser assumida com 100 por cento.

Se o RISCO de morte ou FERIMENTO SÉRIO surgido de uma falha de software for posteriormente reduzido a um nível aceitável (como definido na ISO 14971) por uma medida de CONTROLE DE RISCO física, tanto por redução das consequências da falha quanto por redução na probabilidade de morte ou FERIMENTO SÉRIO surgido da falha, a classificação de segurança do software poderá ser reduzida de C para B; e se o RISCO de um FERIMENTO não SÉRIO, surgido de uma falha de software for similarmente reduzido para uma nível aceitável por uma medida de CONTROLE DE RISCO física, a classificação de segurança de software pode ser reduzida de B para A.

b) O FABRICANTE deve especificar para cada SISTEMA DE SOFTWARE que contribui para aplicação de uma medida de CONTROLE DE RISCO, uma classe de segurança de software baseada nos efeitos possíveis dos PERIGOS que a medida de CONTROLE DE RISCO está controlando.

c) O FABRICANTE deve documentar a classe de segurança do software especificado para cada SISTEMA DE SOFTWARE no ARQUIVO DE GERENCIAMENTO DE RISCO.

d) Quando um SISTEMA DE SOFTWARE é decomposto em ITENS DE SOFTWARE, e quando um ITEM DE SOFTWARE é decomposto em outros ITENS DE SOFTWARE, estes ITENS DE SOFTWARE devem herdar a classificação de segurança de software do ITEM DE SOFTWARE original (ou SISTEMA DE SOFTWARE) a menos que os documentos do FABRICANTE fundamentem uma classificação de segurança diferente. Esta fundamentação deve explicar como os novos ITENS DE SOFTWARE são separados do todo e que por isso devem ser classificados separadamente.

e) O FABRICANTE deve documentar a classe de segurança do software de cada ITEM DE SOFTWARE se essa classificação for diferente da classe do ITEM DE SOFTWARE do qual este foi criado por decomposição.

f) Para cumprimento desta norma, onde quer que um PROCESSO for requisitado por um ITEM DE SOFTWARE de uma classificação específica e o PROCESO é necessariamente aplicado a um grupo de ITEMS DE SOFTWARE, o FABRICANTE deve usar PROCESSOS e TAREFAS que são requisitos da classificação mais alta do ITEM DE SOFTWARE no grupo, a menos que a documentação do FABRICANTE no ARQUIVO DE GERENCIAMENTO DE RISCO fundamente o uso de uma classificação menor.

g) Para cada SISTEMA DE SOFTWARE deve ser aplicada a classe C, a menos que outra classe de segurança de software seja especificada.,.

NOTA Nos requisitos seguintes, a identificação das classes de segurança de software que os requisitos devem ser executados seguem a forma [Classe ...].

5 PROCESSO de desenvolvimento de Software

5.1 * Planejamento de desenvolvimento de software

5.1.1 Plano de desenvolvimento de Software

O FABRICANTE deve estabelecer um plano de desenvolvimento de software (ou planos) para conduzir as ATIVIDADES do PROCESSO de desenvolvimento de software apropriado ao escopo, magnitude e classificações de segurança do SISTEMA DE SOFTWARE a ser desenvolvido. O MODELO DE CICLO DE VIDA DE DESENVOLVIMENTO DO SOFTWARE deve tanto ser completamente definido ou ser referenciado no plano (ou planos). O plano deve tratar o seguinte:

- a) Os PROCESSOS que serão usados no desenvolvimento do SISTEMA DE SOFTWARE (veja Nota 4);
- b) Os ITENS DE ENTREGA (incluindo documentação) das ATIVIDADES e TAREFAS;
- c) A RASTREABILIDADE entre requisitos do SISTEMA, requisitos de software, teste de SISTEMA DE SOFTWARE, e implementação de medidas de CONTROLE DE RISCO no software;
- d) Configuração do software e gerenciamento de mudanças, incluindo ITENS DE CONFIGUAÇÃO de SDPD e software usados para dar suporte ao desenvolvimento; e
- e) Resolução de problemas no software para problemas detectados na execução dos PRODUTOS DE SOFTWARE, ITENS DE ENTREGA e ATIVIDADES de cada estágio do ciclo de vida.

[Classe A, B, C]

NOTA 1 O MODELO DE CICLO DE VIDA DE DESENVOLVIMENTO DO SOFTWARE pode identificar elementos diferentes (PROCESSO, ATIVIDADES, TAREFAS e ITENS DE ENTREGA) para diferentes ITENS DE SOFTWARE de acordo com a classificação de segurança do software de cada ITEM SOFTWARE do SISTEMA DE SOFTWARE.

NOTA 2 Estas ATIVIDADES e TAREFAS podem se sobrepor ou interagir e podem ser realizadas iterativamente ou recursivamente. Não é o objetivo sugerir que um modelo de ciclo de vida específico deve ser usado.

NOTA 3 Outros PROCESSOS são descritos, nesta norma, separadamente do PROCESSO de desenvolvimento. O que não implica que eles devam ser implementados em separado das ATIVIDADES e TAREFAS. As ATIVIDADES e TAREFAS de outros PROCESSOS podem ser integradas ao PROCESSO de desenvolvimento.

NOTA 4 O plano de desenvolvimento do software pode fazer referência a PROCESSOS existentes ou definir novos.

NOTA 5 O plano de desenvolvimento do software pode estar integrado a um plano de desenvolvimento do SISTEMA geral.

5.1.2 Manter o plano de desenvolvimento atualizado

O FABRICANTE deve atualizar o plano conforme o desenvolvimento prossegue, de maneira apropriada. [Classe A, B, C]

5.1.3 Referência do plano de desenvolvimento de Software ao projeto e desenvolvimento do SISTEMA

- a) Como entradas para o desenvolvimento do software, os requisitos do SISTEMA devem ser referenciados no plano de desenvolvimento de software pelo FABRICANTE.
- b) O FABRICANTE deve incluir ou referenciar no plano de desenvolvimento de software procedimentos para coordenar o desenvolvimento do software, o projeto e a validação do desenvolvimento necessários para satisfazer a 4.1.

[Classe A, B, C]

NOTA Não deve haver uma diferença entre requisitos do SISTEMA DE SOFTWARE e requisitos do SISTEMA se o SISTEMA DE SOFTWARE for um SISTEMA autônomo (dispositivo somente de software).

5.1.4 Normas, métodos e ferramentas de planejamento de desenvolvimento de software.

O FABRICANTE deve incluir ou referenciar no plano de desenvolvimento de software:

- a) normas,

b) métodos, e

c) ferramentas

associados ao desenvolvimento de ITENS DE SOFTWARE da classe C. [Classe C]

5.1.5 Integração do software e plano de teste de integração.

O FABRICANTE deve incluir ou referenciar no plano de desenvolvimento de software, um plano para integrar os ITENS DE SOFTWARE (incluindo SDPD) e executar testes durante a integração. [Classe B, C]

NOTA É aceitável combinar testes de integração e testes do SISTEMA DE SOFTWARE em um único plano e conjunto de ATIVIDADES.

5.1.6 Planejamento da VERIFICAÇÃO de software

O FABRICANTE deve incluir ou referenciar no plano de desenvolvimento de software as seguintes informações de VERIFICAÇÃO:

a) ITENS DE ENTREGAS que requerem VERIFICAÇÃO;

b) As TAREFAS de VERIFICAÇÃO requeridas em cada ATIVIDADE do ciclo de vida;

c) marcos em que os ITENS DE ENTREGA são VERIFICADOS; e

d) o critério de aceitação para a VERIFICAÇÃO dos ITENS DE ENTREGA.

[Classe A, B, C]

5.1.7 Planejamento do gerenciamento de risco do software

O FABRICANTE deve incluir ou referenciar no plano de desenvolvimento de software, um plano para conduzir as ATIVIDADES e TAREFAS do PROCESSO DE GERENCIAMENTO DE RISCO do software, incluindo o gerenciamento de RISCO relativo ao SDPD. [Classe A, B,C]

NOTA Veja Cláusula 7.

5.1.8 Planejamento da documentação

O FABRICANTE deve incluir ou referenciar no plano de desenvolvimento de software informação sobre os documentos a serem produzidos durante o ciclo de vida de desenvolvimento do software. Para cada documento identificado ou tipo de documento, as informações seguintes devem ser incluídas ou referenciadas:

a) título, nome ou nomeação convencionada;

b) propósito;

c) público-alvo do documento; e

d) procedimentos e responsabilidades para desenvolvimento, revisão, aprovação e modificação.

[Classe A, B, C]

5.1.9 Planejamento do gerenciamento da configuração do software

O FABRICANTE deve incluir ou referenciar informações do gerenciamento de configuração de software no plano de desenvolvimento de software. As informações de gerenciamento de configuração de software devem incluir ou referenciar:

- a) as classes, tipos, categorias ou listas de itens a serem controlados;
- b) as TAREFAS e ATIVIDADES do gerenciamento de configuração de software;
- c) a(s) organização(ões) responsável(is) por realizar o gerenciamento da configuração de software e das ATIVIDADES.
- d) o relacionamento com outras organizações, como desenvolvimento de software ou manutenção.
- e) quando os itens devem ser colocados sob controle de configuração; e
- f) quando o PROCESSO de resolução de problemas deve ser usado.

[Classe A, B, C]

5.1.10 Itens de suporte a serem controlados

Os itens que serão controlados devem incluir ferramentas, itens ou configurações, usados no desenvolvimento do SOFTWARE DE PRODUTO PARA A SAÚDE, que possa impactar no SOFTWARE DE PRODUTO PARA A SAÚDE. [Classe B,C]

NOTA Exemplos de cada item incluindo versão de compilador/montador, arquivos de montagem, arquivos de bloco e configurações específicas de ambiente.

5.1.11 Controle de ITEM DE CONFIGURAÇÃO DE SOFTWARE antes da VERIFICAÇÃO

O FABRICANTE deve planejar para colocar os ITENS DE CONFIGURAÇÃO sob controle documentado do gerenciamento de configuração antes destes serem VERIFICADOS. [Classe B, C]

5.2 *Análise dos requisitos do software

5.2.1 Defina e documente os requisitos de software dos requisitos do SISTEMA

Para cada SISTEMA DE SOFTWARE do PRODUTO PARA A SAÚDE, o FABRICANTE deve definir e documentar os requisitos do SISTEMA DE SOFTWARE do nível de requisitos do SISTEMA. [Classe A, B, C]

NOTE Não deve haver uma diferença entre requisitos do SISTEMA DE SOFTWARE e requisitos do SISTEMA se o SISTEMA DE SOFTWARE for um SISTEMA autônomo (dispositivo somente de software).

5.2.2 Conteúdo dos requisitos de software

Como apropriado ao SOFTWARE DE PRODUTO PARA A SAÚDE, o FABRICANTE deve incluir nos requisitos de software:

- a) requisitos funcionais e de capacidade;

NOTA 1 Exemplos Incluem:

- desempenho (exemplo, propósito do software, requisitos de tempo),

- características físicas (exemplo, linguagem do código, plataforma, sistema operacional),
- ambiente computacional (exemplo, hardware, tamanho de memória, unidade de processamento,fuso horário, infra-estrutura de rede) sob o qual o software é executado, e
- necessidade de compatibilidade com atualizações ou múltiplos SDPD ou outras versões do produto.

b) Entradas e saídas do SISTEMA DE SOFTWARE;

NOTA 2 Exemplos Incluem:

- características dos dados (exemplo, numéricos, alfanuméricos, formato),
- escalas,
- limites, e
- valores padrão.

c) Interfaces entre o SISTEMA DE SOFTWARE e outros SISTEMAS;

d) Alarmes, avisos e mensagens ao operador por software;

e) requisitos de SEGURANÇA;

NOTA 3 Exemplos Incluem:

- aqueles relacionados ao comprometimento de informações confidenciais,
- autenticação,
- autorização,
- rastreamento de operações,
- integridade de comunicação.

f) requisitos de engenharia de usabilidade que são sensíveis à erros humanos e de treinamento.

NOTA 4 Exemplos Incluem:

- suporte para operações manuais,
- interações homem-máquina,
- limitações de pessoal, e
- áreas que necessitam de atenção humana concentrada.

NOTA 5 Informações relacionadas à requisitos de engenharia de usabilidade podem ser encontradas na IEC 60601-1-6. (citar também a norma específica?)

g) requisitos de definição de dados e de base de dados;

NOTA 6 Exemplos incluem:

- formulário;
- disposição;
- função.

- h) requisitos de instalação e aceitação do SOFTWARE DE PRODUTO PARA A SAÚDE entregue no local ou locais de operação e manutenção;
- i) requisitos relacionados à métodos de operação e manutenção;
- j) documentação de usuário a ser desenvolvida;
- k) requisitos de manutenção do usuário; e
- l) requisitos regulatórios.

[Classe A, B, C]

NOTA 7 Todos estes requisitos podem não estar disponíveis no início do desenvolvimento de software

NOTA 8 A ISO/IEC 9126-1 [8] fornece informação sobre as características de qualidade que podem ser úteis para definir os requisitos de software.

5.2.3 Inclusão de medidas de CONTROLE DE RISCO nos requisitos de software

O FABRICANTE deve incluir medidas de CONTROLE DE RISCO implementadas no software para falhas no hardware ou potenciais defeitos no software nos requisitos, quando aplicável ao SOFTWARE DE PRODUTO PARA A SAÚDE. [Classe A, B, C]

NOTA Estes requisitos podem não estar disponíveis no início do desenvolvimento de software e podem mudar enquanto o software é projetado e as medidas de CONTROLE DE RISCO são mais definidas.

5.2.4 ReAVALIAÇÃO da ANÁLISE DE RISCO DO PRODUTO PARA A SAÚDE

O FABRICANTE deve reAVALIAR a ANÁLISE DE RISCO DO PRODUTO PARA A SAÚDE quando os requisitos de software são estabelecidos e atualizá-lo como apropriado. [Classe A, B, C]

5.2.5 Atualização de requisitos do SISTEMA

O FABRICANTE deve assegurar que os requisitos existentes, incluindo os requisitos do SISTEMA, são re-AVALIADOS e atualizados quando necessário como resultado da ATIVIDADE de análise de requisitos de software. [Classe A, B, C]

5.2.6 Verificação dos requisitos de software

O FABRICANTE deve verificar e documentar que os requisitos de software:

- a) implementam os requisitos do SISTEMA, incluindo aqueles relacionados ao CONTROLE DE RISCO;
- b) não contradizem um ao outro;
- c) são expressos em termos que evitam ambiguidade.
- d) são apresentados em termos que permitam o estabelecimento de um critério de teste e desempenho de testes para determinar quando o critério de teste foi alcançado;
- e) podem ser identificados de forma única; e
- f) são rastreáveis aos requisitos do SISTEMA ou outra fonte.

[Classe A, B, C]

NOTA Esta norma não requer o uso de um formato específico de linguagem.

5.3 *Projeto da ARQUITETURA do software

5.3.1 Transformar os requisitos de software em uma ARQUITETURA

O FABRICANTE deve transformar os requisitos do SOFTWARE DE PRODUTO PARA A SAÚDE em uma ARQUITETURA documentada que descreva a estrutura do software e identifique os ITENS DE SOFTWARE. [Classe B, C]

5.3.2 Desenvolver uma ARQUITETURA para as interfaces dos ITENS DE SOFTWARE

O FABRICANTE deve desenvolver e documentar uma ARQUITETURA para as interfaces entre os ITENS DE SOFTWARE e os componentes externos dos ITENS DE SOFTWARE (ambos software e hardware), e entre os ITENS DE SOFTWARE. [Classe B, C]

5.3.3 Especificação dos requisitos funcionais e de desempenho dos itens SDPD

Se um ITEM DE SOFTWARE é identificado como SDPD, o FABRICANTE deve especificar requisitos funcionais e de desempenho para o item SDPD que são necessários para seu uso pretendido. [Classe B, C]

5.3.4 Especificação do SISTEMA de software e hardware necessários para o item SDPD

Se um ITEM DE SOFTWARE é identificado como SDPD, o FABRICANTE deve especificar o SISTEMA de hardware e software necessários para dar suporte à operação do item SDPD. [Classe B,C]

NOTA Exemplos incluem tipo e velocidade do processador, tipo e tamanho de memória, tipo do SISTEMA de software, comunicação e requisitos de exibição do software.

5.3.5 Identificação da separação necessária para o CONTROLE DE RISCO.

O FABRICANTE deve identificar a separação entre os ITENS DE SOFTWARE que é essencial para o CONTROLE DE RISCO, e de que forma assegurar que a separação é efetiva. [Classe C]

NOTA Um exemplo de separação é ter a execução de ITENS DE SOFTWARE em diferentes processadores. A efetividade da separação pode ser assegurada pelo não compartilhamento de recursos entre os processadores.

5.3.6 Verificação da ARQUITETURA de software

O FABRICANTE deve verificar e documentar que:

- a) a ARQUITETURA do software implementa os requisitos de SISTEMA e do software incluindo os relacionados ao CONTROLE DE RISCO;
- b) a ARQUITETURA de software é capaz de suportar as interfaces entre ITENS DE SOFTWARE e entre ITENS DE SOFTWARE e hardware; e
- c) a ARQUITETURA DO PRODUTO PARA A SAÚDE suporta o funcionamento adequado de quaisquer itens SDPD. [Classe B, C]

5.4 Projeto detalhado do software

5.4.1 Refinar a ARQUITETURA do software em UNIDADES DE SOFTWARE

O FABRICANTE deve refinar a ARQUITETURA do software até ela ser representada por UNIDADES DE SOFTWARE. [Classe B, C]

5.4.2 Desenvolver projeto detalhado de cada UNIDADE DE SOFTWARE

O FABRICANTE deve desenvolver e documentar um projeto detalhado de cada UNIDADE DE SOFTWARE do ITEM DE SOFTWARE. [Classe C]

5.4.3 Desenvolver projeto detalhado das interfaces

O FABRICANTE deve desenvolver e documentar um projeto detalhado para qualquer interface entre a UNIDADE DE SOFTWARE e componentes externos (hardware ou software), assim como quaisquer interfaces entre UNIDADES DE SOFTWARE. [Classe C]

5.4.4 Verificar projeto detalhado

O FABRICANTE deve verificar e documentar que o projeto detalhado do software:

- a) implementa a ARQUITETURA do software; e
 - b) é livre de contradições com a ARQUITETURA do software.
- [Classe C]

5.5 * Implementação e verificação da UNIDADE DE SOFTWARE

5.5.1 Implementar cada UNIDADE DE SOFTWARE

O FABRICANTE deve implementar cada UNIDADE DE SOFTWARE. [Classe A, B, C]

5.5.2 Estabelecer PROCESSO DE VERIFICAÇÃO DA UNIDADE DE SOFTWARE

O FABRICANTE deve estabelecer estratégias, métodos e procedimentos para verificar cada UNIDADE DE SOFTWARE. Onde a VERIFICAÇÃO for realizada por ensaio, os procedimentos de ensaio devem ser AVALIADOS por exatidão. [Classe B, C]

62304 © IEC:2006 21 – COPYRIGHT © IEC. NOT FOR COMMERCIAL USE OR REPRODUCTION
62304 IEC:2006 – 43 –

NOTA É aceitável combinar ensaio de integração e ensaio de SISTEMA DE SOFTWARE em um único plano e conjunto de ATIVIDADES.

5.5.3 Critério de aceitação de UNIDADE DE SOFTWARE

O FABRICANTE deve estabelecer critérios de aceitação para UNIDADES DE SOFTWARE antes da integração em ITENS DE SOFTWARE como apropriado, e garantir que as UNIDADES DE SOFTWARE atendam os critérios. [Classe B, C]

NOTA Exemplos de critérios de aceitação são:

- o código do software implementa requisitos incluindo medidas de CONTROLE DE RISCO?
- o código do software é livre de contradições com a interface documentada no projeto detalhado da UNIDADE DE SOFTWARE?
- o código do software está em conformidade com os procedimentos de programação ou padrões de codificação?

5.5.4 Critérios de aceitação de UNIDADE DE SOFTWARE adicional

Quando presente no projeto, o FABRICANTE deve incluir critérios de aceitação adicionais como apropriado para:

- a) sequência de eventos apropriada;
 - b) dados e controle de fluxo;
 - c) alocação de recursos planejada;
 - d) tratamento de falhas (definição de erro, isolação, e recuperação);
 - e) inicialização de variáveis;
 - f) auto-diagnóstico;
 - g) gerenciamento de memória e estouro de memória; e
 - h) condições limite.
- [Classe C]

5.5.5 VERIFICAÇÃO DA UNIDADE DE SOFTWARE

O FABRICANTE deve realizar a VERIFICAÇÃO DA UNIDADE DE SOFTWARE e documentar os resultados. [Classe B, C]

5.6 * Integração de software e ensaio da integração

5.6.1 Integrar as UNIDADES DE SOFTWARE

O FABRICANTE deve integrar as UNIDADES DE SOFTWARE em conformidade com o plano de integração (ver 5.1.5). [Classe B, C]

5.6.2 Verificar a integração de software

O FABRICANTE deve verificar e registrar os seguintes aspectos da integração de software em conformidade com o plano de integração (ver 5.1.5):

- a) as UNIDADES DE SOFTWARE foram integradas nos ITENS DE SOFTWARE e SISTEMA DE SOFTWARE; e
- b) os itens de hardware, ITENS DE SOFTWARE, e suporte para operações manuais (exemplo, interface homem-máquina, menus de ajuda on-line, reconhecimento de fala, controle por voz) do SISTEMA foram integrados no SISTEMA. [Classe B, C]

NOTA Esta VERIFICAÇÃO apenas coloca que os itens foram integrados em conformidade com o plano, não que foram realizados conforme pretendido. Esta VERIFICAÇÃO é muito provavelmente implementada por alguma forma de inspeção.

22 – 6

COPYRIGHT © IEC. NOT FOR COMMERCIAL USE OR REPRODUCTION 62304 IEC:2006 – 45 –

5.6.3 Teste do software integrado

O FABRICANTE deve testar os ITENS DE SOFTWARE integrados em conformidade com o plano de integração (ver 5.1.5) e documentar os resultados. [Classe B, C]

5.6.4 Conteúdo do ensaio de integração

Para os ensaios de integração do software, o FABRICANTE deve abordar se o ITEM DE SOFTWARE atua conforme previsto. [Classe B, C]

NOTA 1 Exemplos a serem considerados são:

- a funcionalidade requerida do software;
- implementação de medidas de CONTROLE DE RISCO;
- temporização especificada e outros comportamentos;
- funcionamento especificado das interfaces internas e externas ; e
- ensaios sob condições anormais incluindo má utilização previsível.

NOTA 2 É aceitável combinar ensaio de integração e ensaio do SISTEMA DE SOFTWARE em um único plano e conjunto de ATIVIDADES.

5.6.5 Verificar procedimentos de teste de integração

O FABRICANTE deve AVALIAR a exatidão dos procedimentos de teste de integração. [Classe B, C]

5.6.6 Conduzir ensaio de regressão

Quando itens de software são integrados, o FABRICANTE deve conduzir ENSAIOS DE REGRESSÃO apropriados para demonstrar que falhas não foram introduzidas nos softwares integrados previamente. [Classe B, C]

5.6.7 Conteúdo dos registros dos teste de integração

O FABRICANTE deve:

- a) documentar os resultados do teste (aprovação/reprovação e uma lista de ANOMALIAS);
- b) reter registros suficientes para permitir a repetição dos teste; e
- c) identificar o testador.

[Classe B, C]

NOTA o requisito b) pode ser implementado mantendo, por exemplo:

- especificações de casos de teste que demonstrem as ações requeridas e resultados esperados;
- registros do equipamento; e
- registros do ambiente de teste (incluindo ferramentas de software) usados para o teste.

5.6.8 Usar PROCESSO de resolução de problema de software

O FABRICANTE deve incluir as ANOMALIAS encontradas durante a integração do software e os ensaios de integração no PROCESSO de resolução de problema de software. [Classe B, C]

NOTA Ver Cláusula 9.

62304 © IEC:2006 23 –

COPYRIGHT © IEC. NOT FOR COMMERCIAL USE OR REPRODUCTION 62304 IEC:2006 – 47 –

5.7 * Ensaio do SISTEMA DE SOFTWARE

5.7.1 Estabelecer testes para os requisitos de software

O FABRICANTE deve estabelecer e realizar um conjunto de testes, expressos como estímulos de entrada, resultados esperados, critérios de aprovação/reprovação e procedimentos, para condução do ensaio do SISTEMA DE SOFTWARE, de tal forma que todos os requisitos de software sejam cobertos. [Classe B, C]

NOTA 1 É aceitável combinar ensaio de integração e ensaio do SISTEMA DE SOFTWARE em um único plano e conjunto de ATIVIDADES. É também aceitável testar requisitos de software em fases anteriores.

NOTA 2 Não testar apenas cada requisito em separado, mas testes das combinações de requisitos também devem ser realizados, especialmente se houver dependências entre eles.

5.7.2 Usar PROCESSO de resolução de problema de software

O FABRICANTE deve incluir as ANOMALIAS encontradas durante o ensaio de integração do software no PROCESSO de resolução de problema de software. [Classe B, C]

5.7.3 Retestar depois das mudanças

Quando mudanças forem feitas durante o ensaio do SISTEMA DE SOFTWARE, o FABRICANTE deve:

- a) repetir os testes, realizar testes modificados ou realizar testes adicionais, como apropriado, para verificar a efetividade da mudança na correção do problema;
- b) conduzir testes apropriados para demonstrar que efeitos colaterais não foram introduzidos; e
- c) realizar ATIVIDADES DE GERENCIAMENTO DE RISCO relevantes conforme definido em 7.4. [Classe B, C]

5.7.4 Verificar o ensaio do SISTEMA DE SOFTWARE

O FABRICANTE deve verificar que:

- a) a estratégia de VERIFICAÇÃO e os procedimentos de teste utilizados são apropriados;
- b) o procedimento de teste do SISTEMA DE SOFTWARE é rastreável aos requisitos de software;
- c) todos os requisitos de software foram testados ou, de outra forma, VERIFICADOS; e
- d) o resultado dos testes atende os critérios de aprovação/reprovação.

[Classe B, C]

5.7.5 Conteúdo dos registros do teste do SISTEMA DE SOFTWARE

O FABRICANTE deve:

- a) documentar o resultado dos teste (aprovão/reprovação e uma lista de ANOMALIAS);
- b) manter registros suficientes para permitir a repetição dos testes; e
- c) identificar o testador.

[Classe B, C]

NOTA O requisito b) pode ser implementado mantendo, por exemplo:

- especificações de casos de teste que demonstrem as ações requeridas e resultados esperados;
- registros do equipamento; e
- registros do ambiente de teste (incluindo ferramentas de software) usados para o teste.

24 – 6

COPYRIGHT © IEC. NOT FOR COMMERCIAL USE OR REPRODUCTION 62304 IEC:2006 – 49 –

5.8 * Liberação do software

5.8.1 Garantir que a VERIFICAÇÃO do software está completa

O FABRICANTE deve garantir que a VERIFICAÇÃO do software foi completada e os resultados AVALIADOS antes do software ser liberado. [Classe B, C]

5.8.2 Documentar ANOMALIAS residuais conhecidas

O FABRICANTE deve documentar todas as ANOMALIAS residuais conhecidas. [Classe B, C]

5.8.3 AVALIAR ANOMALIAS residuais conhecidas

O FABRICANTE deve garantir que todas as ANOMALIAS residuais conhecidas foram AVALIADAS para garantir que elas não contribuem para um RISCO inaceitável. [Classe B, C]

5.8.4 Documentar liberação de VERSÕES

O FABRICANTE deve documentar as VERSÕES do PRODUTO DE SOFTWARE que está sendo liberado. [Classe A, B, C]

5.8.5 Documentar como o software liberado foi criado

O FABRICANTE deve documentar o procedimento e ambiente usado para criar o software liberado. [Classe B, C]

5.8.6 Garantir que ATIVIDADES e TAREFAS estão completas

O FABRICANTE deve garantir que todas ATIVIDADES e TAREFAS estão completas, juntamente com todos os documentos associados. [Classe B, C]

5.8.7 Arquivar o software

O FABRICANTE deve arquivar:

- a) o PRODUTO DE SOFTWARE e ITENS DE CONFIGURAÇÃO; e
- b) a documentação

por, pelo menos, um período de tempo determinado como o mais longo: a vida útil do dispositivo conforme definida pelo FABRICANTE ou o tempo especificado pelos requisitos regulatórios relevantes. [Classe B, C]

5.8.8 Assegurar a repetibilidade do software liberado

O FABRICANTE deve estabelecer procedimentos para garantir que o PRODUTO DE SOFTWARE liberado pode ser entregue de forma confiável ao ponto de uso sem corrupção ou mudança não autorizada. Estes procedimentos devem abordar a produção e manuseio da mídia contendo o PRODUTO DE SOFTWARE incluindo como apropriado:

- replicação,
- rotulagem da mídia,
- embalagem,
- proteção,
- armazenamento, e
- entrega.

[Classe B, C]

6 PROCESSO de manutenção do software

6.1 * Estabelecer plano de manutenção do software

O FABRICANTE deve estabelecer um plano de manutenção do software (ou planos) para conduzir as ATIVIDADES e TAREFAS do PROCESSO de manutenção. O plano deve abordar o seguinte:

a) procedimentos para:

- receber,
- documentar,
- avaliar,
- resolver e
- rastrear

feedback recebidos após a liberação do SOFTWARE DE PRODUTO PARA A SAÚDE;

b) critérios para determinar se o feedback é considerado um problema;

c) uso do PROCESSO DE GERENCIAMENTO DE RISCO do software;

d) uso do PROCESSO de resolução de problema de software para análise e resolução de problemas recebidos após a liberação do SOFTWARE DE PRODUTO PARA A SAÚDE;

e) uso do PROCESSO de gerenciamento de configuração de software (Cláusula 8) para gerenciamento de modificações no SISTEMA existente; e

f) procedimentos para AVALIAR e implementar:

- atualizações,
- correção de falhas,
- pacotes de atualização e
- obsolescência do SDPD.

[Classe A, B, C]

6.2 * Análise do problema e modificação

6.2.1 Documentar e AVALIAR o feedback

6.2.1.1 Monitorar o feedback

O FABRICANTE deve monitorar o feedback sobre o PRODUTO DE SOFTWARE liberado tanto de dentro da organização quanto dos usuários. [Classe A, B, C]

6.2.1.2 Documentar e AVALIAR o feedback

O feedback deve ser documentado e AVALIADO para determinar se o problema existe no PRODUTO DE SOFTWARE liberado. Qualquer problema deste tipo deve ser registrado como um RELATÓRIO DE PROBLEMA (ver Cláusula 9). RELATÓRIOS DE PROBLEMA devem incluir os eventos adversos reais ou potenciais, assim como os desvios de especificações. [Classe A, B, C]

6.2.1.3 Avaliar os efeitos do RELATÓRIO DE PROBLEMA na SEGURANÇA

Cada RELATÓRIO DE PROBLEMA deve ser AVALIADO para determinar como ele afeta a SEGURANÇA do PRODUTO DE SOFTWARE liberado e se uma mudança ao PRODUTO DE SOFTWARE liberado é necessária para atender o problema. [Classe A, B, C]

26 – 6

COPYRIGHT © IEC. NOT FOR COMMERCIAL USE OR REPRODUCTION 62304 IEC:2006 – 53 –

6.2.2 Uso do PROCESSO de resolução de problema de software

O FABRICANTE deve usar o PROCESSO de resolução de problema de software (ver Cláusula 9) para abordar os RELATÓRIOS DE PROBLEMA. [Classe A, B, C]

NOTA Quando esta ATIVIDADE for realizada, qualquer mudança na classificação de segurança do SISTEMA DE SOFTWARE ou de seus ITENS DE SOFTWARE deve ser conhecida.

6.2.3 Analisar SOLICITAÇÕES DE MUDANÇA

Adicionalmente à análise requerida pela Cláusula 9, o FABRICANTE deve analisar cada SOLICITAÇÃO DE MUDANÇA pelo seu efeito na organização, PRODUTO DE SOFTWARE liberado, e SISTEMAS com o qual ele faz interface. [Classe B, C]

6.2.4 Aprovação de SOLICITAÇÃO DE MUDANÇA

O FABRICANTE deve AVALIAR e aprovar SOLICITAÇÕES DE MUDANÇA que modifiquem o PRODUTO DE SOFTWARE liberado. [Classe A, B, C]

6.2.5 Comunicar usuários e órgãos regulatórios

O FABRICANTE deve identificar as SOLICITAÇÕES DE MUDANÇA que afetem os PRODUTOS DE SOFTWARE liberado.

Conforme exigido pela regulamentação local, o FABRICANTE deve informar os usuários e os órgãos regulatórios sobre:

- a) qualquer problema no PRODUTO DE SOFTWARE liberado e das consequências de seu uso contínuo inalterado; e
- b) a natureza de qualquer mudança disponível para o PRODUTO DE SOFTWARE liberado e como obter e instalar estas mudanças.

[Classe A, B, C]

6.3 * Implementação de modificação

6.3.1 Usar PROCESSOS estabelecidos para implementar modificações

O FABRICANTE deve usar o PROCESSO de desenvolvimento de software (ver Cláusula 5) ou um PROCESSO de manutenção estabelecido para implementar as modificações. [Classe A, B, C]

NOTA Para requisitos relativos a mudanças no GERENCIAMENTO DE RISCO do software ver 7.4.

6.3.2 Relançamento de SISTEMA DE SOFTWARE modificado

O FABRICANTE deve liberar SISTEMAS DE SOFTWARE conforme 5.8. Modificações podem ser liberadas como parte de um relançamento completo de um SISTEMA DE SOFTWARE ou como um kit de modificação compreendendo ITENS DE SOFTWARE modificados e as ferramentas necessárias para instalar as mudanças como modificações em um SISTEMA DE SOFTWARE existente. [Classe A, B, C]

62304 © IEC:2006 27 –
COPYRIGHT © IEC. NOT FOR COMMERCIAL USE OR REPRODUCTION 62304 IEC:2006 – 55 –

7 * PROCESSO DE GERENCIAMENTO DE RISCO de software

7.1 * Análise do software contribuindo para situações de perigo

7.1.1 Identificar ITENS DE SOFTWARE que possam contribuir para situações de perigo

O FABRICANTE deve identificar ITENS DE SOFTWARE que possam contribuir para uma situação de perigo identificada na ATIVIDADE de ANÁLISE DE RISCO DO PRODUTO PARA A SAÚDE da ISO 14971 (ver 4.2). [Classe B, C]

NOTA A situação de perigo pode ser resultado direto de uma falha de software ou o resultado de uma falha da medida de CONTROLE DE RISCO implementada no software.

28 – 6
COPYRIGHT © IEC. NOT FOR COMMERCIAL USE OR REPRODUCTION 62304 IEC:2006 – 57 –

7.2 Medidas de CONTROLE DE RISCO

7.2.1 Definir as medidas de CONTROLE DE RISCO

Para cada potencial causa do item de software que contribua para uma situação de perigo documentada no arquivo de gerenciamento de risco, o FABRICANTE deve definir e documentar medidas de CONTROLE DE RISCO. [Classe B, C]

NOTA As medidas de CONTROLE DE RISCO podem ser implementadas no hardware, software, no ambiente de trabalho ou instrução de usuário.

7.2.2 Medidas de CONTROLE DE RISCO implementadas no software

Se uma medida de CONTROLE DE RISCO é implementada como parte das funções do ITEM DE SOFTWARE, o FABRICANTE deve:

- a) incluir a medida de CONTROLE DE RISCO nos requisitos de software;
- b) atribuir uma classe de segurança de software ao ITEM DE SOFTWARE com base nos possíveis efeitos do PERIGO que a medida de CONTROLE DE RISCO está controlando; e
- c) desenvolver o ITEM DE SOFTWARE em conformidade com a Cláusula 5.

[Classe B, C]

NOTA Este requisito fornece detalhes adicionais para os requisitos de CONTROLE DE RISCO da ISO 14971

7.3 VERIFICAÇÃO das medidas de CONTROLE DE RISCO

7.3.1 Verificar medidas de CONTROLE DE RISCO

A implementação de cada medida de CONTROLE DE RISCO documentada em 7.2 deve ser VERIFICADA, e esta VERIFICAÇÃO deve ser documentada. [Classe B, C]

7.3.2 Documentar qualquer nova sequência de eventos

Se uma medida de CONTROLE DE RISCO é implementada como um ITEM DE SOFTWARE, o FABRICANTE deve AVALIAR a medida de CONTROLE DE RISCO para identificar e documentar no ARQUIVO DE GERENCIAMENTO DE RISCO quaisquer novas sequências de eventos que possam resultar em situação de perigo. [Classe B, C]

7.3.3 Documentar a RASTREABILIDADE

O FABRICANTE de documentar a RASTREABILIDADE dos PERIGOS de software como apropriado:

- a) da situação de perigo para o ITEM DE SOFTWARE;
- b) do ITEM DE SOFTWARE para a específica causa no software;
- c) da causa no software para a medida de CONTROLE DE RISCO; e
- d) da medida de CONTROLE DE RISCO para a VERIFICAÇÃO da medida de CONTROLE DE RISCO. [Classe B, C]

NOTA Ver ISO 14971 – relatório de GERENCIAMENTO DE RISCO

62304 © IEC:2006 29 –

COPYRIGHT © IEC. NOT FOR COMMERCIAL USE OR REPRODUCTION 62304 IEC:2006 – 59 –

7.4 GERENCIAMENTO DE RISCO de mudanças no software

7.4.1 Analisar as mudanças no SOFTWARE DE PRODUTO PARA A SAÚDE relacionadas a SEGURANÇA

O FABRICANTE de analisar as mudanças ao SOFTWARE DE PRODUTO PARA A SAÚDE (incluindo o SDPD) para determinar se:

- a) causas potenciais adicionais são introduzidas contribuindo para uma situação de perigo; e
- b) medidas de CONTROLE DE RISCO de software adicionais são necessárias.

[Classe A, B, C]

7.4.2 Análise do impacto de mudanças no software em medidas de CONTROLE DE RISCO existentes

O FABRICANTE deve analisar as mudanças no software, incluindo mudanças no SDPD, para determinar se a modificação no software poderia interferir com medidas de CONTROLE DE RISCO existentes. [Classe B, C]

7.4.3 Realizar ATIVIDADES DE GERENCIAMENTO DE RISCO com base em análises

O FABRICANTE deve realizar ATIVIDADES DE GERENCIAMENTO DE RISCO relevantes definidas em 7.1, 7.2 e 7.3 com base nestas análises [Classe B, C]

8 PROCESSO de gerenciamento de configuração de software

8.1 Identificação da configuração

8.1.1 Estabelecer os meios para identificar os ITENS DE CONFIGURAÇÃO

O FABRICANTE deve estabelecer um esquema para a identificação única dos ITENS DE CONFIGURAÇÃO e de suas VERSÕES a serem controladas para o projeto. Este esquema deve incluir outros PRODUTOS DE SOFTWARE ou entidades tais como SDPD e documentação. [Classe A, B, C]

8.1.2 Identificar SDPD

Para cada ITEM DE CONFIGURAÇÃO SDPD sendo utilizado, incluindo bibliotecas padrão, o FABRICANTE deve documentar:

- a) o título;
 - b) o FABRICANTE, e
 - c) o designador único para o SDPD
- de cada ITEM DE CONFIGURAÇÃO SDPD sendo usado. [Classe A, B, C]

NOTA O designador único para o SDPD poderia ser, por exemplo, uma VERSÃO, uma data de liberação, um número de pacote de atualização ou uma designação de atualização.

8.1.3 Identificar documentação de configuração do SISTEMA

O FABRICANTE deve documentar o conjunto de ITENS DE CONFIGURAÇÃO e suas VERSÕES que compreendem a configuração do SISTEMA DE SOFTWARE. [Classe A, B, C]

8.2 * Controle de mudanças

8.2.1 Aprovar SOLICITAÇÃO DE MUDANÇA

O FABRICANTE deve alterar ITENS DE CONFIGURAÇÃO somente em resposta a uma SOLICITAÇÃO DE MUDANÇA aprovada. [Classe A, B, C]

NOTA 1 A decisão para aprovar uma SOLICITAÇÃO DE MUDANÇA pode ser integral para o PROCESSO de controle de mudança ou parte de um outro PROCESSO. Esta sub-cláusula somente requer que a aprovação de uma mudança preceda sua implementação.

NOTA 2 PROCESSOS diferentes de aceitação podem ser usados para SOLICITAÇÕES DE MUDANÇA em diferentes estágios do ciclo de vida, conforme indicado nos planos, ver 5.1.1 e) e 6.1 e).

8.2.2 Implementar mudanças

O FABRICANTE deve implementar a mudança conforme especificado na SOLICITAÇÃO DE MUDANÇA. O FABRICANTE deve identificar e realizar qualquer ATIVIDADE que necessite ser repetida como resultado de uma mudança, incluindo mudanças na classificação de segurança do software do SISTEMA DE SOFTWARE e dos ITENS DE SOFTWARE. [Classe A, B, C]

NOTA Esta sub-cláusula estabelece como a mudança deve ser implementada para atingir um adequado controle de mudança. Isto não implica que a implementação é uma parte integrante do PROCESSO de controle de mudança. A implementação deve usar PROCESSOS planejados, ver 5.1.1 e) e 6.1 e).

8.2.3 Verificar alterações

O FABRICANTE deve verificar a mudança, inclusive repetindo qualquer VERIFICAÇÃO que tenha sido invalidada por uma mudança e tendo em conta 5.7.3 e 9.7. [Classe A, B, C]

NOTA Esta sub-cláusula somente requer que mudanças sejam VERIFICADAS. Isto não implica que a VERIFICAÇÃO é uma parte integral do PROCESSO de controle de mudança. A VERIFICAÇÃO deve usar PROCESSOS planejados, ver 5.1.1 e) e 6.1 e).

8.2.4 Providenciar meios para RASTREABILIDADE de mudanças

O FABRICANTE deve criar uma auditoria em que cada:

- a) SOLICITAÇÃO DE MUDANÇA;
- b) RELATÓRIO DE PROBLEMA relevante; e
- c) aprovação de SOLICITAÇÃO DE MUDANÇA

possa ser rastreada. [Classe A, B, C]

8.3 * Contabilidade do estado da configuração

O FABRICANTE deve reter registros recuperáveis do histórico dos ITENS DE CONFIGURAÇÃO controlados incluindo a configuração do SISTEMA. [Classe A, B, C]

9 * PROCESSO de resolução de problemas de software

9.1 Preparar RELATÓRIO DE PROBLEMAS

O FABRICANTE deve preparar um RELATÓRIO DE PROBLEMAS para cada problema detectado em um PRODUTO DE SOFTWARE. RELATÓRIOS DE PROBLEMAS devem ser classificados como segue:

- a) tipo;
EXEMPLO 1 corretivo, preventivo ou adaptável a um novo ambiente
- b) escopo; e

EXEMPLO 2 tamanho da mudança, número de dispositivos modelo afetados, acessórios de suporte afetados, recursos envolvidos, tempo para mudança

- c) criticidade.

EXEMPLO 3 efeito sobre a performance, SEGURANÇA

[Classe A, B, C]

NOTA Problemas podem ser descobertos antes ou depois da liberação, dentro da organização do FABRICANTE ou fora dela.

9.2 Investigar o problema

O FABRICANTE deve:

- a) investigar o problema e se possível identificar as causas;
- b) AVALIAR a relevância do problema quanto à SEGURANÇA usando o PROCESSO DE GERENCIAMENTO DE RISCO de software (Cláusula 7);
- c) documentar o resultado da investigação e avaliação; e
- d) criar SOLICITAÇÃO(ÕES) DE MUDANÇA para ações necessárias à correção do problema, ou documentar a justificativa para a não tomada de ação.

[Classe A, B, C]

NOTA Um problema não tem que ser corrigido para o FABRICANTE cumprir com o PROCESSO de resolução de problema de software, desde que o problema não seja relevante à SEGURANÇA.

9.3 Informar as partes relevantes

O FABRICANTE deve informar as partes relevantes da existência do problema, como apropriado.

[Classe A, B, C]

NOTA Problemas podem ser descobertos antes ou depois da liberação, dentro da organização do FABRICANTE ou fora dela. O FABRICANTE determina as partes relevantes dependendo da situação.

9.4 Usar PROCESSOS de controle de mudança

O FABRICANTE deve aprovar e implementar todas as SOLICITAÇÕES DE MUDANÇA, observando os requisitos do PROCESSO de controle de mudança (ver 8.2). [Classe A, B, C]

9.5 Manter registros

O FABRICANTE deve manter registros dos RELATÓRIOS DE PROBLEMAS e suas resoluções, incluindo suas VERIFICAÇÕES.

O FABRICANTE deve atualizar o ARQUIVO DE GERENCIAMENTO DE RISCO como apropriado (ver 7.4) [Classe A, B, C]

9.6 Analisar problemas para tendências

O FABRICANTE deve realizar análises para detectar tendências nos RELATÓRIOS DE PROBLEMAS. [Classe A, B, C]

9.7 Verificar resolução de problema de software

O FABRICANTE deve verificar as resoluções para determinar se:

- a) o problema foi resolvido e o RELATÓRIO DE PROBLEMA foi fechado;
- b) tendências adversas foram revertidas;
- c) SOLICITAÇÕES DE MUDANÇA foram implementadas nos PRODUTOS DE SOFTWARE e ATIVIDADES apropriadas; e
- d) problemas adicionais foram introduzidos.

[Classe A, B, C]

9.8 Conteúdo da documentação de ensaio

Quando ensaiar, reensaiar, ou quando ENSAIOS DE REGRESSÃO de ITENS DE SOFTWARE e SISTEMAS seguirem uma mudança, o FABRICANTE deve incluir na documentação de ensaio:

- a) resultados dos ensaios;
- b) ANOMALIAS encontradas;
- c) a VERSÃO do software ensaiada;
- d) configurações relevantes de hardware e software para o ensaio;
- e) ferramentas de ensaios relevantes;
- f) data do ensaio; e
- g) identificação do ensaio.

[Classe A, B, C]

Anexo A
(informativo)

Justificativa para os requisitos desta norma

A justificativa para as cláusulas desta norma é fornecida neste anexo.

A.1 Justificativa

O requisito primário desta norma é que um conjunto de PROCESSOS seja seguido no desenvolvimento e manutenção do SOFTWARE DE PRODUTOS PARA A SAÚDE, e que a escolha dos PROCESSOS seja adequada aos RISCOS para o paciente e outras pessoas. Isto resulta da crença de que o ensaio do software não é suficiente para determinar que ele é seguro em sua execução.

Os PROCESSOS requeridos por esta norma se enquadram em duas categorias:

- PROCESSOS que são requeridos para determinar os RISCOS decorrentes da operação de cada ITEM DE SOFTWARE no software;
- PROCESSOS que são requeridos para atingir uma probabilidade de falha de software apropriadamente baixa para cada ITEM DE SOFTWARE, escolhidos com base nestes RISCOS determinados.

Esta norma requer que a primeira categoria seja aplicada seja executada para todos os SOFTWARE DE PRODUTOS PARA A SAÚDE e a segunda categoria seja executada para ITENS DE SOFTWARE selecionados.

Uma reivindicação de atendimento a esta norma deve, portanto, incluir uma ANÁLISE DE RISCO documentada que identifique sequências previsíveis de eventos que incluem software e que possam resultar em uma situação perigosa (ver ISO 14971). PERIGOS que podem ser indiretamente causados pelo software (por exemplo, pelo fornecimento de informação enganosa que poderia causar tratamento inapropriado a ser administrado) devem ser incluídos na ANÁLISE DE RISCO.

Todas as ATIVIDADES que são requeridas como parte desta categoria de PROCESSOS são identificadas no texto normativo como “[Classe A, B, C]”, indicando que são exigidas independentemente da classificação do software à qual são aplicadas.

ATIVIDADES são exigidas para todas as classes A, B e C pelas seguintes razões:

- a ATIVIDADE produz um plano relevante para GERENCIAMENTO DE RISCO ou classificação de segurança de software;
- a ATIVIDADE produz uma saída que é uma entrada para o GERENCIAMENTO DE RISCO ou classificação de segurança de software;
- a ATIVIDADE é parte do GERENCIAMENTO DE RISCO ou classificação de segurança de software;
- a ATIVIDADE estabelece um sistema de administração, documentação ou manutenção de registros que suporte o GERENCIAMENTO DE RISCO ou classificação de segurança de software;
- a ATIVIDADE normalmente ocorre quando a classificação do software relacionado é desconhecida;
- a ATIVIDADE pode causar uma mudança que poderia invalidar a classificação de segurança de software atual do software associado. Isto inclui a descoberta e análise de problemas relacionados à segurança após o lançamento.

Outros PROCESSOS são exigidos somente para SISTEMAS DE SOFTWARE ou ITENS DE SOFTWARE classificados nas classes B ou C de segurança de software. As ATIVIDADES exigidas como partes destes PROCESSOS são identificadas neste texto normativo como “[Classe B, C]”, ou “[Classe C]” indicando que elas são exigidas seletivamente dependendo da classificação do software a que elas se aplicam.

ATIVIDADES são exigidas seletivamente para software classe B e C pelas seguintes razões:

- a ATIVIDADE aumenta a confiabilidade do software exigindo mais detalhe e rigor no projeto, ensaios ou outras VERIFICAÇÃO;
- a ATIVIDADE é uma ATIVIDADE administrativa que auxilia outra ATIVIDADE exigida para as classes B e C;
- a ATIVIDADE auxilia a correção de problemas relacionados à segurança;
- a ATIVIDADE produz registros de projeto, implementação, VERIFICAÇÃO e liberação de software relacionado à segurança.

ATIVIDADES são exigidas seletivamente para software classe C pelas seguintes razões:

- a ATIVIDADE melhora ainda mais a confiabilidade do software por exigir mais detalhes, ou mais rigor, ou atenção a problemas específicos no projeto, ensaio ou outra VERIFICAÇÃO.

Notar que todos os PROCESSOS e ATIVIDADES definidos nesta norma são considerados valiosos por assegurar o desenvolvimento e manutenção de software de alta qualidade. A omissão de muitos destes PROCESSOS e ATIVIDADES como requisitos de software na classe A que não pode por definição causar um PERIGO não deve implicar em que estes PROCESSOS e ATIVIDADES não seriam de valor ou que não são recomendados. Sua omissão se destina a reconhecer que o software que não pode causar um PERIGO pode ser facilmente assegurado da SEGURANÇA e eficácia principalmente através da ATIVIDADE de validação global durante o projeto do PRODUTO PARA A SAÚDE (que está fora do escopo desta norma) e através de alguns simples controles de ciclo de vida de software.

A.2 Resumo de requisitos por classe

A tabela A.1 resume quais classes de segurança de software são atribuídas a cada requisito. Esta tabela é informativa e fornecida por conveniência. A secção normativa identifica as classes de segurança de software para cada requisito.

Tabela A.1 – Resumo de requisitos por classe de segurança de software

Cláusulas e sub-cláusulas		Classe A	Classe B	Classe C
Cláusula 4	Todos os requisitos	X	X	X
5.1	5.1.1, 5.1.2, 5.1.3, 5.1.6, 5.1.7, 5.1.8, 5.1.9 5.1.5, 5.1.10, 5.1.11 5.1.4	X 	X 	X
5.2	5.2.1, 5.2.2, 5.2.4, 5.2.5, 5.2.6 5.2.3	X 	X 	X
5.3	5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.6 5.3.5		X 	X
5.4	5.4.1 5.4.2, 5.4.3, 5.4.4		X 	X
5.5	5.5.1 5.5.2, 5.5.3, 5.5.5 5.5.4	X 	X 	X
5.6	Todos os requisitos		X 	X
5.7	Todos os requisitos		X 	X
5.8	5.8.4 5.8.1, 5.8.2, 5.8.3, 5.8.5, 5.8.6, 5.8.7, 5.8.8	X 	X 	X
6.1	6.1	X	X	X
6.2	6.2.1, 6.2.2, 6.2.4, 6.2.5 6.2.3	X 	X 	X
6.3	Todos os requisitos	X	X	X
7.1	Todos os requisitos		X 	X
7.2	Todos os requisitos		X 	X
7.3	Todos os requisitos		X 	X
7.4	7.4.1 7.4.2, 7.4.3	X 	X 	X
Cláusula 8	Todos os requisitos	X	X	X
Cláusula 9	Todos os requisitos	X	X	X

Anexo B
(informativo)

Orientação sobre as provisões desta norma

B.1 Escopo

B.1.1 Propósito

O propósito desta norma é fornecer um PROCESSO de desenvolvimento que irá, de forma consistente, produzir SOFTWARE DE DISPOSITIVO PARA A SAÚDE seguro e de alta qualidade. Para realizar isto, a norma identifica as atividades e tarefas mínimas que necessitam ser atendidas para garantir que o software foi desenvolvido de uma forma apropriada a gerar PRODUTOS DE SOFTWARE altamente seguros e confiáveis.

Este anexo fornece orientação para a aplicação dos requisitos desta norma. Ele não adiciona, ou altera os requisitos da norma. Este anexo pode ser usado para um melhor entendimento dos requisitos desta norma.

Note que nesta norma, ATIVIDADES são subclasses chamadas dentro dos PROCESSOS e TAREFAS são definidas dentro das ATIVIDADES. Por exemplo, as ATIVIDADES definidas para o PROCESSO de desenvolvimento de software são planejamento de desenvolvimento de software, análise de requisitos do software, projeto da arquitetura do software, projeto do detalhe do software, implementação da UNIDADE DE SOFTWARE e VERIFICAÇÃO, integração do software e teste da integração, teste do SISTEMA DE SOFTWARE, e liberação do software. As TAREFAS dentro destas atividades são os requisitos individuais.

Esta norma não requer um MODELO DE CICLO DE VIDA PARA DESENVOLVIMENTO DE SOFTWARE particular. Contudo, a concordância com esta norma implica em dependências entre PROCESSOS, pois as entradas de um PROCESSO são geradas por outros PROCESSOS. Por exemplo, a classificação de segurança de software do SISTEMA DE SOFTWARE deveria ser completada depois do PROCESSO DE ANÁLISE DE RISCO ter estabelecido que um DANO poderia surgir a partir da falha do SISTEMA DE SOFTWARE.

Devido a cada uma das dependências lógicas entre os processos, é mais fácil descrever os processos nesta norma em uma sequência, sugerindo um modelo de ciclo de vida em “cascata” ou “passo-único”. Contudo, outros modelos de ciclos de vida podem também ser usados. Algumas estratégias de desenvolvimento (modelos) como definido na ISO/ IEC 12207 [9] incluem (ver também a tabela B.1):

- Cascata. A estratégia “passo-único”, também chamada “cascata”, consiste na realização do PROCESSO de desenvolvimento uma única vez. Simplisticamente: determinar as necessidades do cliente, definir requisitos, projeto do SISTEMA, implementação do sistema, teste, reparo e entrega.
- Incremental. A estratégia “incremental” determina as necessidades do cliente e define os requisitos do SISTEMA, então realiza o resto do desenvolvimento em uma sequência de construções. A primeira construção incorpora parte das capacidades planejadas, a próxima adiciona mais capacidades, e assim por diante, até o SISTEMA estar completo.
- Evolucionário. A estratégia “evolucionária” também desenvolve um SISTEMA em construção, mas difere da estratégia incremental ao reconhecer que a necessidade do usuário não é completamente entendida e todos os requisitos não podem ser definidos antecipadamente. Nesta estratégia, as necessidades do cliente e requisitos do SISTEMA são parcialmente definidas antecipadamente, então são refinadas em cada sucessiva construção.

Tabela B.1 – Estratégias de desenvolvimento (modelos) conforme definido na ISO/ IEC 12207

Estratégia de Desenvolvimento	Define todos os requisitos primeiro?	Múltiplos ciclos de desenvolvimento?	Distribui software provisório?
Cascata (uma vez por fase)	Sim	Não	Não
Incremental (melhoria de produto pré-planejada)	Sim	Sim	Talvez
Evolucionário	Não	Sim	Sim

Qualquer que seja o ciclo de vida escolhido é necessário que ele mantenha as dependências lógicas entre as saídas de PROCESSOS como especificações, documentos de projeto e software. O modelo de ciclo de vida cascata consegue isto retardando o início de um PROCESSO até as entradas para aquele PROCESSO estarem completas e aprovadas.

Outros ciclos de vida, particularmente os ciclos de vida evolucionários, permitem que as saídas de PROCESSOS sejam produzidas antes que todas as entradas para o PROCESSO sejam avaliadas. Por exemplo, um novo ITEM DE SOFTWARE pode ser especificado, classificado, implementado e VERIFICADO antes que toda a ARQUITETURA de software tenha sido finalizada. Tais ciclos de vida carregam o RISCO de que uma mudança ou desenvolvimento em uma saída de PROCESSO invalide outra saída. Todos os ciclos de vida, portanto usam uma configuração abrangente do sistema de gerenciamento para garantir que todas saídas de PROCESSOS sejam levadas a um estado consistente e as dependências conservadas.

Os princípios a seguir são importantes independentes do ciclo de vida para desenvolvimento de software usado:

- Todas as saídas de PROCESSO devem ser mantidas em um estado consistente; sempre que qualquer saída de PROCESSO for criada ou alterada, todas as saídas relacionadas devem ser atualizadas em seguida para manter sua consistência com as demais e para manter todas as dependências explícitas e implícitas necessárias por esta norma;
- Todas as saídas de PROCESSOS devem estar disponíveis quando necessárias como entradas para outros trabalhos no software;
- Antes de qualquer SOFTWARE DE PRODUTO PARA A SAÚDE ser liberado, todas as saídas de PROCESSOS devem estar consistentes com as demais e todas as dependências entre saídas de PROCESSO requisitadas explícita ou implicitamente por esta norma devem ser observadas.

B.1.2 Campo de aplicação

Esta norma se aplica não só ao desenvolvimento e manutenção de SOFTWARE DE PRODUTOS PARA A SAÚDE como também ao do PRODUTO PARA A SAÚDE que inclui SDPD.

O uso desta norma requer que o FABRICANTE realize GERENCIAMENTO DE RISCO DE PRODUTO PARA A SAÚDE compatível com a ISO 14971. Portanto, quando a ARQUITETURA DO SISTEMA DE PRODUTO PARA A SAÚDE incluir um componente adquirido (este pode ter sido comprado ou ser um componente de origem desconhecida), como uma impressora/ plotadora que inclui SDPD, o componente adquirido se torna responsabilidade do FABRICANTE e deve ser incluído na ANÁLISE DE RISCO do PRODUTO PARA A SAÚDE. É assumido que por meio da realização apropriada do GERENCIAMENTO DE RISCO DE PRODUTO PARA A SAÚDE, o FABRICANTE deveria tomar conhecimento do componente e reconhecer que ele inclui SDPD. O FABRICANTE usando esta norma deveria invocar o PROCESSO DE GERENCIAMENTO DE RISCO DE SOFTWARE como parte do PROCESSO DE GERENCIAMENTO DE RISCO DE PRODUTO PARA A SAÚDE.

A manutenção do SOFTWARE DE PRODUTO PARA A SAÚDE liberado se aplica à experiência de pós-produção com o SOFTWARE DE PRODUTO PARA A SAÚDE. A manutenção do software inclui a combinação de todos os recursos administrativos e técnicos, incluindo ações de supervisão, atuar sobre o informe de um problema para reter um item em, ou recuperá-lo para, um estado no qual ele possa realizar uma função requerida assim como solicitações de modificação relacionadas ao(s) PRODUTO(S) DE SOFTWARE lançados. Por exemplo, isto inclui problemas de retificação, informes regulatórios, ações preventivas ou de re-validação. Ver ISO/ IEC 14764 [10].

B.2 Referências normativas

A ISO/IEC 90003 [11] fornece um guia para a aplicação de um sistema de gerenciamento de qualidade para desenvolvimento de software. Este guia não é exigido por esta norma mas é altamente recomendado.

B.3 Termos e definições

Onde foi possível, os termos foram definidos usando como referência as normas internacionais.

Esta norma escolheu usar três termos para descrever a decomposição de um SISTEMA DE SOFTWARE (alto nível). O SISTEMA DE SOFTWARE pode ser um subsistema de um PRODUTO PARA A SAÚDE (ver IEC 60601-1-4 [2]) ou um PRODUTO PARA A SAÚDE em si. O menor nível que não é mais decomposto para os propósitos de teste ou gerenciamento de configuração de software é a UNIDADE DE SOFTWARE. Todos os níveis de composição, incluindo os níveis altos e baixos, podem ser chamados ITENS DE

SOFTWARE. Um SISTEMA DE SOFTWARE, então, é composto de um ou mais ITENS DE SOFTWARE, e cada ITEM DE SOFTWARE é composto de uma ou mais UNIDADES DE SOFTWARE ou de ITENS DE SOFTWARE passíveis de decomposição. Ao FABRICANTE é deixada a responsabilidade de providenciar a definição e granularidade dos ITENS DE SOFTWARE e das UNIDADES DE SOFTWARE. Se mantidos de forma vaga estes termos permitem o seu uso para os mais diversos métodos de desenvolvimento e tipos de software usados em PRODUTOS PARA A SAÚDE.

B.4 Requisitos gerais

Não há método conhecido que garanta 100% de segurança para qualquer tipo de software.

Há três princípios muito importantes que promovem segurança para SOFTWARE DE PRODUTO PARA A SAÚDE:

- GERENCIAMENTO DE RISCO;
- Gerenciamento de qualidade;
- Engenharia de software.

Para o desenvolvimento e manutenção de SOFTWARE DE PRODUTO PARA A SAÚDE seguro é necessário estabelecer o GERENCIAMENTO DE RISCO como parte integral de um sistema de gerenciamento de qualidade numa conjuntura geral para a aplicação apropriada das técnicas e métodos de engenharia de software. A combinação destes três conceitos permite a um FABRICANTE DE PRODUTO PARA A SAÚDE seguir um PROCESSO claramente estruturado e consistentemente reproduzível de tomada de decisão para promover a SEGURANÇA do SOFTWARE DE PRODUTO PARA A SAÚDE.

B.4.1 Sistema de gerenciamento de qualidade

Um conjunto efetivo e disciplinado de PROCESSOS de software inclui PROCESSOS organizacionais como, por exemplo, gerenciamento, infra-estrutura, melhoria e treinamento. Para evitar duplicação e focar esta norma na engenharia de software, estes processos têm sido omitidos. Estes PROCESSOS são tratados pelo sistema de gerenciamento de qualidade. A ISO 13485 [7] é uma Norma Internacional especificamente destinada a aplicação dos conceitos de gerenciamento de qualidade aos PRODUTOS PARA A SAÚDE. Estar em conformidade com os requisitos do sistema de gerenciamento de qualidade ISO 13485 não constitui conformidade automática aos requisitos regulatórios nacionais ou regionais. É responsabilidade do fabricante identificar e estabelecer conformidade com os requisitos regulatórios relevantes.

B.4.2 Gerenciamento de risco

O desenvolvimento de software participa nas ATIVIDADES DE GERENCIAMENTO DE RISCO de forma suficiente para garantir que todos os RISCOS razoavelmente previsíveis associados ao SOFTWARE PARA PRODUTO MÉDICO são considerados.

Ao invés de tentar definir um PROCESSO DE GERENCIAMENTO DE RISCO apropriado nesta norma de engenharia de software, é requerido que o FABRICANTE aplique um PROCESSO DE GERENCIAMENTO DE RISCO em conformidade com a ISO 14971, que lida explicitamente com GERENCIAMENTO DE RISCO para PRODUTO PARA A SAÚDE. ATIVIDADES DE GERENCIAMENTO DE RISCO de software específicas resultantes dos PERIGOS que tem o software como causa contribuinte são identificados nos PROCESSOS de apoio descritos na Cláusula 7.

B.4.3 Classificação de segurança de software

O RISCO associado ao software como parte de um PRODUTO PARA A SAÚDE, como acessório ou como PRODUTO PARA A SAÚDE em si, é usado como entrada para um esquema de classificação de segurança de software, que então determina os PROCESSOS a serem usados durante o desenvolvimento e manutenção do software.

Considera-se RISCO a uma combinação da severidade do dano e a probabilidade de sua ocorrência. Contudo, não há consenso em como determinar a probabilidade de ocorrência de falhas de software usando métodos estatísticos tradicionais. Nesta norma, portanto, a classificação de SISTEMA DE SOFTWARE é baseada na severidade do PERIGO resultante da falha do software, assumindo que a falha irá ocorrer. SISTEMAS DE SOFTWARE que contribuem para a implementação de medidas para CONTROLE DE RISCO são classificados com base na severidade do PERIGO que eles controlam.

Se um SISTEMA DE SOFTWARE é decomposto em ITENS DE SOFTWARE, então cada ITEM DE SOFTWARE pode ter sua própria classificação de segurança de software.

Somente é possível determinar o RISCO associado a falhas de um ITEM DE SOFTWARE:

- Se uma ARQUITETURA DE SISTEMA e uma ARQUITETURA DE SOFTWARE definem o papel do ITEM DE SOFTWARE em termos de seu propósito e de suas interfaces com outros itens de software e hardware;
- Se as mudanças no SISTEMA são controladas;
- Depois da ANÁLISE DE RISCO ter sido feita na ARQUITETURA e da especificação de medidas de CONTROLE DE RISCO.

Esta norma requer um número mínimo de ATIVIDADES que atingirão as condições acima para todas as classes de software.

O fim da ATIVIDADE DE ARQUITETURA de software é a primeira etapa no desenvolvimento em que o conjunto completo de ITENS DE SOFTWARE é definido e a ATIVIDADE DE GERENCIAMENTO DE RISCO tem identificado como os ITENS DE SOFTWARE se relacionam com SEGURANÇA. Isto é, portanto a primeira etapa na qual os ITENS DE SOFTWARE podem ser classificados definitivamente de acordo com suas regras de SEGURANÇA.

Este etapa corresponde a etapa onde o CONTROLE DE RISCO é iniciado na ISO 14971.

Antes deste etapa, o PROCESSO DE GERENCIAMENTO DE RISCO identifica medidas de ARQUITETURA DE CONTROLE DE RISCO, por exemplo, adicionando subsistemas de proteção, ou reduzindo as oportunidades de falhas do software causar DANO. Depois deste etapa, o PROCESSO DE GERENCIAMENTO DE RISCO usa PROCESSOS destinados a redução da probabilidade de falha de ITENS DE SOFTWARE. Em outras palavras, a classificação de um ITEM DE SOFTWARE específica as medidas de CONTROLE DE RISCO baseada no PROCESSO a serem aplicadas ao item.

É esperado que os FABRICANTES achem útil classificar o software antes desta etapa, por exemplo, para focar atenção em áreas a serem investigadas, mas cada classificação deveria ser considerada como preliminar e não deveria ser usada para justificar a omissão de PROCESSOS.

O esquema de classificação de segurança de software não pretende se alinhar às classificações de RISCO da ISO 14971. Enquanto o esquema da ISO 14971 classifica RISCO de acordo com sua severidade e probabilidade, o esquema de classificação de segurança de software classifica SISTEMAS DE SOFTWARE e ITENS DE SOFTWARE de acordo com o PROCESSO a ser aplicado em seu desenvolvimento e manutenção.

Conforme o projeto evolui, novos RISCOS podem se tornar evidentes. Portanto, o GERENCIAMENTO DE RISCO deveria ser aplicado como parte integrante do PROCESSO de desenvolvimento. Isto permite o desenvolvimento de um projeto ARQUITETÔNICO que identifica um conjunto completo de ITENS DE SOFTWARE, incluindo aqueles que são necessários que funcionem corretamente para garantir operações seguras e aqueles que prevêem falhas que causem DANO.

A ARQUITETURA de software deveria promover segregação de itens de software que são necessários para operação segura e deveria descrever os métodos usados para garantir a segregação efetiva daqueles ITENS DE SOFTWARE.

Como declarado em B.3, esta norma escolhe usar três termos para descrever a decomposição de um SISTEMA DE SOFTWARE (alto nível).

A Figura B.1 ilustra os possíveis particionamentos para ITENS DE SOFTWARE dentro de um SISTEMA DE SOFTWARE e como as classes de segurança de software poderiam ser aplicadas ao grupo de ITENS DE SOFTWARE na decomposição.

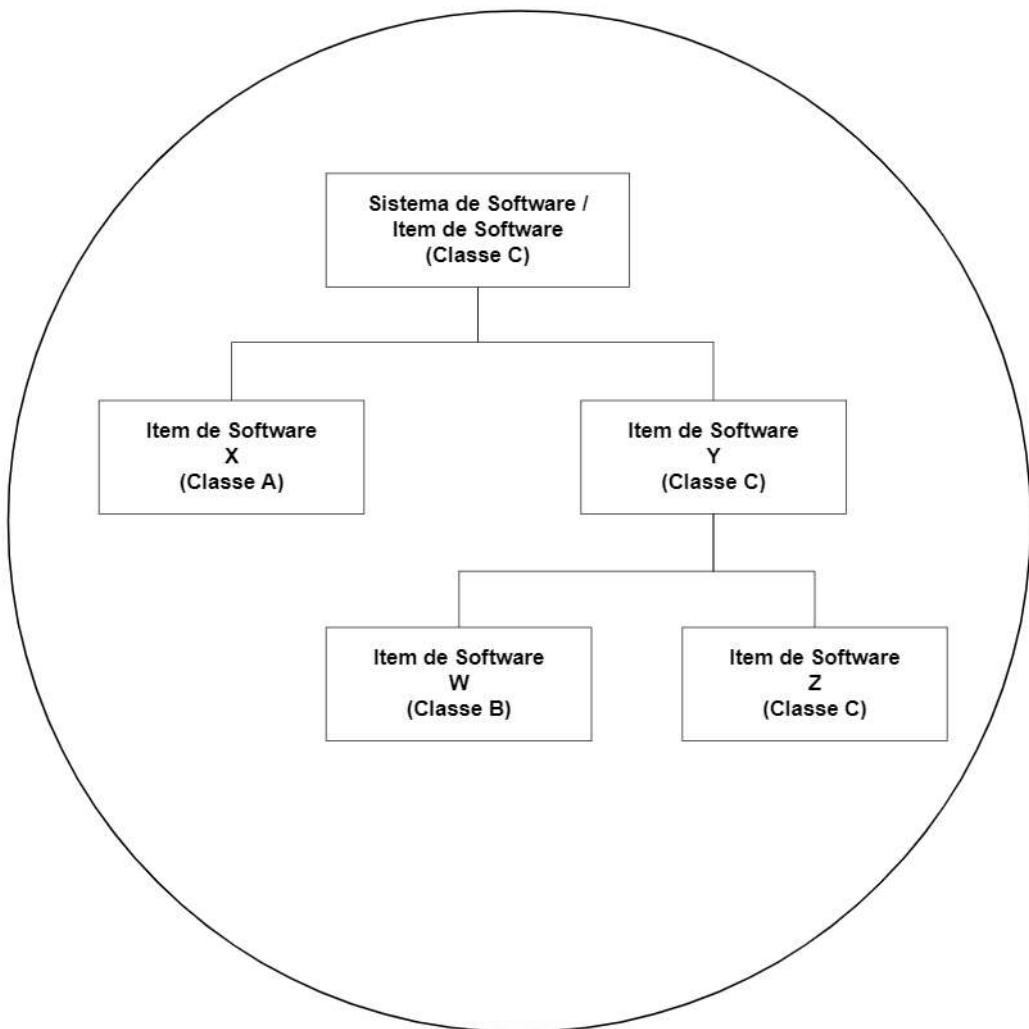


Figura B.1 – Exemplo de particionamento de ITENS DE SOFTWARE

Por este exemplo, os FABRICANTES sabem, devido ao tipo de SOFTWARE DE PRODUTO PARA A SAÚDE que está sendo desenvolvido, que a classificação preliminar de segurança de software para o SISTEMA DE SOFTWARE é software de segurança classe C. Durante o projeto da ARQUITETURA o FABRICANTE decide partitionar o SISTEMA, como mostrado, com 3 ITENS DE SOFTWARE – X, W e Z. O FABRICANTE é capaz de separar todas as contribuições do SISTEMA DE SOFTWARE para PERIGOS que poderiam resultar em morte ou FERIMENTO SÉRIO para ITEM DE SOFTWARE Z e todas contribuições do SISTEMA DE SOFTWARE remanescentes para PERIGOS que poderiam resultar em FERIMENTO NÃO SÉRIO para ITEM DE SOFTWARE W. ITEM DE SOFTWARE W é classificado como software de segurança Classe B e ITEM DE SOFTWARE Z como software de segurança Classe C. O ITEM DE SOFTWARE Y, portanto, deve ser classificado como Classe C, por 4.3 d). O SISTEMA DE SOFTWARE também é classificado como Classe C por este requisito. O ITEM DE SOFTWARE X foi classificado como software de segurança Classe A. O FABRICANTE é capaz de documentar uma justificativa para a separação entre os ITENS DE SOFTWARE X e Y, assim como para os ITENS DE SOFTWARE W e Z, para garantir a integridade da separação. Se partitionar não for possível os ITENS DE SOFTWARE X e Y devem ser classificados na classe de segurança de software C.

B.5 PROCESSO de desenvolvimento de software

B.5.1 Planejamento do desenvolvimento de software

O objetivo desta ATIVIDADE é planejar as TAREFAS de desenvolvimento de software afim de reduzir os RISCOS causados pelo software, comunicar procedimentos e metas para os membros da equipe de desenvolvimento, e assegurar que os requisitos de qualidade do SISTEMA para o SOFTWARE DE PRODUTO PARA A SAÚDE sejam cumpridos.

A ATIVIDADE de planejamento de desenvolvimento do software pode ser documentada em um plano único ou em múltiplos planos. Alguns FABRICANTES podem ter estabelecido políticas e procedimentos que se aplicam para o desenvolvimento de todos os seus SOFTWARES DE PRODUTOS PARA A SAÚDE. Neste caso, o plano pode simplesmente referenciar as políticas e os procedimentos existentes. Alguns FABRICANTES podem preparar um plano ou um conjunto de planos específicos para cada SOFTWARE DE PRODUTO PARA A SAÚDE que expõem em detalhes ATIVIDADES específicas e procedimentos de referência geral. Outra possibilidade é que um plano ou um conjunto de planos é adaptado para o desenvolvimento de cada SOFTWARE DE PRODUTO PARA A SAÚDE. O planejamento deve ser especificado no nível de detalhe necessário para a realização do PROCESSO de desenvolvimento e deve ser proporcional ao RISCO. Por exemplo, SISTEMAS ou itens com maior risco seriam sujeitos a um PROCESSO de desenvolvimento com mais rigor e TAREFAS deveriam ser definidas com maior nível de detalhes.

Planejamento é uma ATIVIDADE iterativa que deve ser reexaminada e atualizada conforme o desenvolvimento progride. O plano pode evoluir para incorporar mais e melhor informação quanto mais se souber sobre o SISTEMA e o nível de esforço necessário para desenvolver o SISTEMA. Por exemplo, uma classificação de segurança de software inicial do SISTEMA pode mudar como resultado do exercício do PROCESSO DE GERENCIAMENTO DE RISCOS e desenvolvimento da ARQUITETURA do software. Ou pode ser decidido que um SDPD deve ser incorporado ao SISTEMA. É importante que o(s) plano(s) seja(m) atualizado(s) para refletir(em) o conhecimento atual do SISTEMA e o nível de rigor necessário para o SISTEMA ou itens no SISTEMA para permitir um controle adequado sobre o PROCESSO de desenvolvimento.

B.5.2 Análise de requisitos de software

Esta ATIVIDADE exige que o FABRICANTE estabeleça e verifique os requisitos de software para o SOFTWARE DE PRODUTO PARA A SAÚDE. Estabelecer requisitos verificáveis é essencial para determinar o que está para ser construído, para determinar que o SOFTWARE DE PRODUTO PARA A SAÚDE apresente um comportamento aceitável, e para demonstrar que o SOFTWARE DE PRODUTO PARA A SAÚDE está pronto para uso. Para demonstrar que os requisitos foram implementados conforme o desejado, cada requisito deve ser indicado de modo que possam ser estabelecidos critérios objetivos para determinar se os mesmos foram implementados corretamente. Se o PROCESSO DE GERENCIAMENTO DE RISCO impôs requisitos ao software para controlar os RISCOS identificados, esses requisitos devem ser identificados nos requisitos de software de forma a tornar possível traçar medidas de CONTROLE DE RISCOS para os requisitos de software. Todos os requisitos de software devem ser identificados de forma a tornar possível demonstrar RASTREABILIDADE entre o requisito e testes do SISTEMA DE SOFTWARE. Se a aprovação dos órgãos reguladores em alguns países exigirem conformidade com regulamentos específicos ou normas internacionais, esta exigência de conformidade deve ser documentado nos requisitos de software. Como os requisitos de software estabelecem o que está para ser implantado no software, uma avaliação dos requisitos é necessária antes que a ATIVIDADE de análise de requisitos seja concluída.

Uma área de frequente conflito é a distinção entre necessidades do cliente entradas de projetos, requisitos de software, especificações funcionais do software e especificações de projetos do software. As entradas do projeto são as interpretações das necessidades dos clientes nos requisitos dos PRODUTOS PARA A SAÚDE formalmente documentados. Requisitos de software são as especificações formalmente documentadas do que o software faz para satisfazer as necessidades do cliente e das entradas de projeto. As especificações funcionais do software são frequentemente incluídas com os requisitos do software e definem em detalhe o que o software faz para satisfazer os seus requisitos, embora muitas alternativas diferentes possam também cumprir os requisitos. As especificações do projeto do software definem como o software será concebido e decomposto para implementar as seus requisitos e especificações funcionais.

Tradicionalmente, requisitos do software, especificações funcionais e especificações de projeto têm sido escritos como um conjunto de um ou mais documentos. Agora, é possível considerar essas informações como itens de dados dentro de uma base de dados comum. Cada item teria um ou mais atributos que definem o seu objetivo e vínculo com os outros itens na base de dados. Esta abordagem permite a apresentação e visualização de diferentes pontos de vista da informação mais adequada para cada conjunto de usuários previstos (por exemplo, marketing, FABRICANTES, testadores, auditores) e suporta RASTREABILIDADE para demonstrar adequada implementação e a medida em que os casos de teste testam os requisitos. As ferramentas para apoiar esta abordagem podem ser simples como um documento hipertexto usando hyperlinks HTML ou tão complexa e capaz como uma ferramenta de engenharia auxiliada por software - (Computer Aided Software Engineering - CASE) e ferramentas de análise de requisitos.

O PROCESSO de requisitos do SISTEMA está fora do escopo desta norma. No entanto, a decisão de implantar funcionalidades de PRODUTO PARA A SAÚDE com software é normalmente realizada durante o projeto do SISTEMA. Alguns ou todos os requisitos do SISTEMA poderão ser alocados para serem

implementados no software. A ATIVIDADE de análise de requisitos do software consiste em analisar os requisitos alocados ao software pelo PROCESSO de requisitos do SISTEMA e resultar um conjunto completo de requisitos de software que refletem os requisitos alocados.

Para garantir a integridade do SISTEMA, o FABRICANTE deve fornecer um mecanismo para a negociação de alterações e esclarecimentos para os requisitos do SISTEMA para corrigir dificuldades práticas, inconsistências ou ambiguidades nos requisitos do SISTEMA principal ou nos requisitos do software.

O PROCESSO de captura e análise dos requisitos de SISTEMA e de software pode ser iterativo. Esta norma não tem a intenção de exigir que os PROCESSOS sejam rigidamente separados em duas camadas. Na prática, a ARQUITETURA DO SISTEMA e ARQUITETURA do software são frequentemente descritas de maneira simultânea e os requisitos de software e do SISTEMA são posteriormente documentados em um formulário padrão.

B.5.3 Projeto da ARQUITETURA do software

Esta atividade requer que o fabricante defina os principais componentes estruturais do software, suas propriedades visíveis externamente, e a relação entre eles. Se o comportamento de um componente pode afetar outros componentes, este comportamento deve ser descrito na ARQUITETURA do software. Esta descrição é especialmente importante para comportamentos que podem afetar componentes do PRODUTO PARA A SAÚDE que estão fora do software. Decisões ARQUITETURAIS são extremamente importantes para implantação de medidas de CONTROLE DE RISCOS. Sem a compreensão (e documentação) do comportamento de um componente que pode afetar outros componentes, será quase impossível demonstrar que o SISTEMA é seguro. A ARQUITETURA de software é necessária para garantir a correta implantação dos requisitos de software. A ARQUITETURA do software não está completa a menos que todos os requisitos do software possam ser implementados pelos ITENS DE SOFTWARE identificados. Devido ao projeto e a implementação do software serem dependentes da ARQUITETURA, a ARQUITETURA é VERIFICADA para completar esta ATIVIDADE. A VERIFICAÇÃO da ARQUITETURA é geralmente feita através de AVALIAÇÃO técnica.

A classificação dos ITENS DE SOFTWARE durante a ATIVIDADE DE ARQUITETURA do software cria uma base para futuras escolhas dos PROCESSOS do software. Os registros de classificação são colocados sob controle de alteração, como parte do ARQUIVO DE GERENCIAMENTO DE RISCO.

Diversos acontecimentos posteriores podem invalidar uma classificação. Estas incluem, por exemplo:

- alterações da especificação do SISTEMA, especificação do software ou da ARQUITETURA;
- descoberta de erros na ANÁLISE DE RISCO, especialmente para PERIGOS imprevistos; e
- descoberta de inviabilidade de um requisito, especialmente uma medida de CONTROLE DE RISCOS;

Portanto, durante todas as ATIVIDADES seguindo o projeto da ARQUITETURA de software, a classificação do SISTEMA DE SOFTWARE e ITENS DE SOFTWARE devem ser reAVALIADAS e podem precisar ser revistas. Isto poderia desencadear retrabalho para aplicar PROCESSOS adicionais para um ITEM DE SOFTWARE como resultado de uma atualização para uma classe superior. O PROCESSO de gerenciamento de configuração do software (Cláusula 8) é utilizado para garantir que todo retrabalho necessário é identificado e concluído.

B.5.4 Projeto detalhado de software

Esta ATIVIDADE requer que o FABRICANTE refine os ITENS DE SOFTWARE e interfaces definidas na ARQUITETURA para criar UNIDADES DE SOFTWARE e suas interfaces. Embora UNIDADES SOFTWARE sejam muitas vezes consideradas como sendo uma única função ou módulo, este ponto de vista nem sempre é o mais adequado. Nós definimos a UNIDADE DE SOFTWARE como sendo um ITEM DE SOFTWARE que não é subdividido em itens menores. UNIDADES DE SOFTWARE podem ser testadas separadamente. O FABRICANTE deve definir o nível de detalhe de uma UNIDADE DE SOFTWARE. O projeto detalhado especifica algoritmos, representações de dados, interfaces entre as diferentes UNIDADES DE SOFTWARE e interfaces entre UNIDADES DE SOFTWARE e estruturas de dados. O projeto detalhado deve também considerar o empacotamento do PRODUTO DE SOFTWARE. É necessário documentar o projeto de cada UNIDADE DE SOFTWARE e sua interface para que a UNIDADE DE SOFTWARE possa ser implementada corretamente. O projeto detalhado completa em detalhes o necessário para construir o software. Deve ser completo o suficiente para que o programador não seja obrigado a tomar decisões próprias (ad hoc).

Um ITEM DE SOFTWARE pode ser decomposto de modo que somente alguns dos novos ITENS DE

SOFTWARE implementem o requisito relativo a SEGURANÇA do ITEM DE SOFTWARE original. Os ITENS DE SOFTWARE restantes, não implementam funções relacionadas com à SEGURANÇA e podem ser reclassificados em uma classe de segurança de software mais baixa. No entanto, a decisão de fazer isto em si é parte do PROCESSO DE GERENCIAMENTO DE RISCO, e é documentado no ARQUIVO DE GERENCIAMENTO DE RISCO.

Como a implantação depende do projeto detalhado, é necessário verificar o projeto detalhado antes da atividade ser concluída. A VERIFICAÇÃO do projeto detalhado é geralmente realizada por uma AVALIAÇÃO técnica. A sub-cláusula 5.4.4 requer que o FABRICANTE verifique as saídas das ATIVIDADES do projeto detalhado. O projeto especifica como os requisitos são implementados. Se o projeto contém problemas, o código não irá implementar os requisitos corretamente.

Quando presente no projeto, o FABRICANTE deve verificar as características do projeto que ele considera importantes para a SEGURANÇA. Exemplos dessas características incluem:

- Execução dos eventos destinados, entradas, saídas, interfaces, fluxo lógico, alocação da CPU, alocação dos recursos de memória, definição de erro e exceção, isolação de erros e exceções, e recuperação de erros;
- Definição do estado padrão, ao qual todos as falhas que podem resultar em uma situação perigosa são destinadas, com eventos e transições;
- Inicialização de variáveis, gerenciamento de memória, e
- Inicialização a quente e a frio, modo de espera, e outras mudanças de estado que podem afetar as medidas de controlo de risco.

B.5.5 – Verificação e Implantação de Unidade Software

Esta ATIVIDADE requer que o FABRICANTE escreva e verifique o código para as UNIDADES DE SOFTWARE. O projeto detalhado deve ser traduzido em código fonte. Codificar representa o ponto onde termina a decomposição das especificações e se inicia a composição do software executável. Para atingir de forma consistente as características do código desejadas, padrões de codificação devem ser adotados para especificar um estilo preferencial de codificação. Exemplos de padrões de codificação incluem requisitos de inteligibilidade, regras ou restrições de uso da linguagem, e gerenciamento da complexidade. O código de cada unidade é VERIFICADO para garantir que o mesmo funciona como especificado pelo projeto detalhado e que esteja em conformidade com os padrões especificados para codificação.

A sub-cláusula 5.5.5 requer que o FABRICANTE verifique o código. Se o código não implementar o projeto corretamente, o SOFTWARE DE PRODUTO PARA A SAÚDE não terá um desempenho como pretendido.

B.5.6 Integração de software e teste de integração

Esta ATIVIDADE requer que o FABRICANTE planeje e execute a integração das UNIDADES DE SOFTWARE em ITENS DE SOFTWARE agregados, bem como a integração dos ITENS DE SOFTWARE em ITENS DE SOFTWARE agregados maiores e verificar que os ITENS DE SOFTWARE resultantes se comportem conforme esperado.

A abordagem de integração pode variar de integração não incremental a qualquer outra forma de integração incremental. As propriedades do ITEM DE SOFTWARE sendo montado ditam a escolha do método de integração.

Software de teste de integração concentra a transferência de dados e controle através das interfaces internas e externas dos ITENS DE SOFTWARE. As interfaces externas são aquelas com outros softwares, incluindo software de sistema operacional, e hardware de outros PRODUTOS PARA A SAÚDE.

O rigor dos testes de integração e o nível de detalhe da documentação associada com os testes de integração devem ser proporcionais ao RISCO associado com o dispositivo, a dependência do dispositivo de software para funções potencialmente perigosas, e ao papel de determinados ITENS DE SOFTWARE em funções do dispositivo de elevado RISCO. Por exemplo, embora todos os ITENS DE SOFTWARE devam ser testados, itens que tenham um efeito sobre a SEGURANÇA devem ser submetidos a testes mais diretos, minuciosos e detalhados.

Quando aplicável, os testes de integração demonstram o comportamento do programa nos limites de seus domínios de entrada e saída, e confirmam as respostas do programa para entradas inválidas, inesperadas, e especiais. As ações do programa são reveladas quando determinadas combinações de entradas ou sequências inesperadas de entradas, ou mesmo quando requisitos de tempo definidos são violados. Os

requisitos de teste do plano devem incluir, quando apropriado, os tipos de testes de caixa branca a serem realizados como parte dos testes de integração.

O teste de caixa branca, também conhecido como caixa de vidro, estrutural, caixa transparente e teste de caixa aberta, é uma técnica de teste em que o conhecimento explícito do funcionamento interno do ITEM DE SOFTWARE sendo testado é utilizado para selecionar os dados de teste. O teste de caixa branca utiliza o conhecimento específico do ITEM DE SOFTWARE para examinar as saídas. A precisão do teste dependerá somente do conhecimento do testador sobre o que o ITEM DE SOFTWARE é proposto a fazer. O testador pode, então, ver se o ITEM DE SOFTWARE se afasta da sua meta. O teste de caixa branca não pode garantir que a especificação completa foi implementada uma vez que este é focado na teste da implementação do ITEM DE SOFTWARE. O teste de caixa preta, também conhecida como teste comportamental, funcional, caixa opaca, e teste de caixa fechada, está focado no teste da especificação funcional e não pode garantir que todas as partes da implementação foram testadas. Assim, o teste de caixa preta é o teste contra a especificação e irá descobrir falhas de omissão, indicando que parte da especificação não foi cumprida. O teste de caixa branca é o teste contra a implementação, e irá descobrir falhas de comissão, indicando que parte da implementação está falha. Para testar completamente um PRODUTO DE SOFTWARE tanto os testes de caixa preta e branca devem ser exigidos.

Os planos e documentação de teste identificados em 5.6 e 5.7 podem ser documentos individuais vinculados a fases específicas de desenvolvimento ou protótipos evolutivos. Eles também podem ser combinados em um único documento ou conjunto de documentos que abrange os requisitos de múltiplas subseções. Tudo ou parte dos documentos poderiam ser incorporados em um documento do projeto de nível superior, tais como um software ou um projeto de garantia da qualidade, ou um plano de teste mais abrangente que aborde todos os aspectos de teste para hardware e software. Nestes casos, uma referência cruzada deve ser criada, que identifica como os diversos documentos de projeto se relacionam com cada uma das TAREFAS de integração de software.

Os testes de integração de software podem ser realizados em um ambiente simulado, no hardware real, ou no PRODUTO PARA A SAÚDE completo.

A sub-cláusula 5.6.2 requer que o FABRICANTE verifique a saída da ATIVIDADE de integração de software. A saída da ATIVIDADE de integração são os ITENS DE SOFTWARE integrados. Estes ITENS de SOFTWARE integrados devem funcionar adequadamente para que todo o SOFTWARE PARA PRODUTO PARA A SAÚDE funcione corretamente e com segurança.

B.5.7 Testes de SISTEMA DE SOFTWARE

Esta ATIVIDADE requer que o FABRICANTE verifique a funcionalidade do software verificando se que os requisitos para o software foram implementados com sucesso.

O teste do SISTEMA DE SOFTWARE demonstra que determinada funcionalidade existe. Este teste verifica a funcionalidade e o desempenho do programa como foram concebidos em relação aos requisitos para o software.

O teste do SISTEMA DE SOFTWARE foca no ensaio funcional (caixa preta), embora possa ser desejável utilizar o método da caixa branca (ver secção anterior) para realizar determinados ensaios mais eficientemente, iniciar condições de estresse ou falhas, ou aumentar a cobertura do código dos testes de qualificação. A organização dos testes por tipos e estágio do teste é flexível, mas a cobertura dos requisitos, do CONTROLE DE RISCO, usabilidade, e tipos de teste (por exemplo, falha, instalação, estresse), deve ser demonstrada e documentada.

O teste do SISTEMA DE SOFTWARE testa o software integrado e pode ser realizado em um ambiente simulado, no hardware real, ou no PRODUTO PARA A SAÚDE completo.

Quando uma alteração é realizada em um SISTEMA DE SOFTWARE (mesmo uma pequena mudança), o grau de TESTES DE REGRESSÃO (não apenas os testes de uma alteração individual) deve ser determinado para garantir que nenhum efeito colateral foi introduzido. Este TESTE DE REGRESSÃO (e as razões para não repetir completamente o teste do SISTEMA DE SOFTWARE) deve ser planejado e documentado.

As responsabilidades de teste do SISTEMA DE SOFTWARE podem ser dispersadas, ocorrendo em diferentes locais e sendo conduzido por diferentes organizações. No entanto, independentemente da distribuição das TAREFAS, das relações contratuais, fonte dos componentes, ou do ambiente de desenvolvimento, o FABRICANTE do produto mantém a responsabilidade final por garantir que o software funcione corretamente para o uso pretendido.

Se ANOMALIAS reveladas durante os testes se repetirem, mas a decisão tomada for por não corrigi-las, estas ANOMALIAS devem ser AVALIADAS em relação à análise dos PERIGOS para verificar se elas não afetam a SEGURANÇA do produto. A causa-raiz e os sintomas das ANOMALIAS devem ser entendidos, e as razões para a não corrigi-las devem ser documentadas.

A sub-cláusula 5.7.4 requer que os resultados dos testes do SISTEMA DE SOFTWARE sejam AVALIADOS para garantir que os resultados esperados foram obtidos.

B.5.8 Liberação do software

Esta ATIVIDADE requer que o FABRICANTE documente a VERSÃO do SOFTWARE DE PRODUTO PARA A SAÚDE que está sendo lançada, especifique como ela foi criada, e siga procedimentos adequados para a liberação do software.

O FABRICANTE deve ser capaz de demonstrar que o software que foi desenvolvido utilizando o PROCESSO de desenvolvimento é o software que está sendo lançado. O FABRICANTE também deve ser capaz de recuperar o software e as ferramentas utilizadas para a sua geração em caso de necessidade futura e deve armazenar, empacotar e entregar o software de forma a minimizar seu dano ou má utilização. Procedimentos definidos devem ser estabelecidos para garantir que essas TAREFAS sejam executadas adequadamente e com resultados consistentes.

B.6 PROCESSO de manutenção do software

B.6.1 Estabelecer plano de manutenção do software

O PROCESSO de manutenção do software difere do PROCESSO de desenvolvimento de software de duas formas:

- O FABRICANTE pode utilizar um PROCESSO menor que o PROCESSO de desenvolvimento de software completo para implementar mudanças rápidas em resposta a problemas urgentes.
- Em resposta aos RELATÓRIOS DE PROBLEMAS de software referentes ao produto lançado, o FABRICANTE não só resolve o problema, mas também atende às regulamentações locais (normalmente através de um esquema de vigilância pró-ativa para recolher dados dos problemas do campo e da comunicação com clientes e órgãos reguladores sobre o problema).

A sub-cláusula 6.1 requer que estes PROCESSOS sejam estabelecidos em um plano de manutenção.

Esta ATIVIDADE requer que o FABRICANTE crie ou identifique procedimentos para implementação das ATIVIDADES e TAREFAS de manutenção. Para implantar ações corretivas, controle de mudanças durante a manutenção, e gerenciar a liberação de software revisado, o FABRICANTE deve documentar e resolver os problemas reportados e as solicitações dos usuários, bem como gerenciar as alterações ao PRODUTO PARA A SAÚDE. Este PROCESSO é ativado quando o SOFTWARE DO PRODUTO PARA A SAÚDE sofre alterações no código e documentação associada, por causa de um problema ou de uma necessidade de melhoria ou adaptação. O objetivo é o de modificar o SOFTWARE DO PRODUTO PARA A SAÚDE lançado preservando sua integridade. Este PROCESSO inclui a migração do SOFTWARE DO PRODUTO PARA A SAÚDE para ambientes ou plataformas para os quais não foi originalmente lançado. As ATIVIDADES previstas nesta cláusula são específicas para o PROCESSO de manutenção, contudo, o PROCESSO de manutenção pode utilizar outros PROCESSOS desta norma.

O FABRICANTE deve planejar como as ATIVIDADES e TAREFAS do PROCESSO de manutenção serão realizadas.

B.6.2 Análise de problema e modificação

Esta ATIVIDADE requer que o FABRICANTE analise o retorno de seus efeitos; verifique problemas relatados; e considere, selecione e obtenha aprovação para implementar uma opção de modificação. Problemas e outras solicitações de alterações podem afetar o desempenho, SEGURANÇA, ou autorização regulatória de um PRODUTO PARA A SAÚDE. Uma análise é necessária para determinar se quaisquer efeitos existem devido ao RELATÓRIO DE PROBLEMA ou se quaisquer efeitos irão resultar de uma modificação para corrigir um problema ou implementar uma solicitação. É especialmente importante verificar através de rastreamento ou análise de regressão que as medidas de CONTROLE DE RISCOS integradas ao produto não sejam prejudicialmente modificadas ou alteradas pela alteração do software que está sendo implementada como parte de uma ATIVIDADE de manutenção de software. É também

importante verificar que o software modificado não cause um PERIGO ou que atenua um RISCO no software que anteriormente não causava ou atenuava.

É importante distinguir entre manutenção de software (Cláusula 6) e resolução de problema de software (Cláusula 9).

O foco do PROCESSO de manutenção de software é numa resposta adequada aos comentários recebidos após a liberação do PRODUTO DE SOFTWARE. Como parte de um PRODUTO PARA A SAÚDE, o PROCESSO de manutenção de software precisa garantir que:

- RELATÓRIOS DE PROBLEMAS relacionados a SEGURANÇA são dirigidos e reportados às autoridades regulatórias apropriadas e usuários afetados;
- PRODUTOS DE SOFTWARE são revalidados e relançados após a modificação com controles oficiais que garantem a retificação do problema e prevenção de novos problemas;
- o FABRICANTE considera que outros PRODUTOS DE SOFTWARE podem ser afetados e toma medidas apropriadas.

O foco da resolução do problema de software é a operação de um sistema de controle abrangente que:

- analise RELATÓRIOS DE PROBLEMA e identifique todas as implicações do problema;
- decida sobre uma série de mudanças e identifique todos os seus efeitos colaterais;
- implemente a mudança enquanto mantém a consistência dos ITENS DE CONFIGURAÇÃO de software incluindo o ARQUIVO DE GERENCIAMENTO DE RISCO;
- VERIFIQUE a implementação das mudanças.

O PROCESSO de manutenção de software usa o PROCESSO de resolução de software. O PROCESSO de manutenção de software trata as decisões de alto-nível relacionada só RELATÓRIO DE PROBLEMA (se o problema existe, se ele tem efeito significante na SEGURANÇA, que mudanças são necessárias e quando implementá-las), e usa o PROCESSO de resolução de software na análise do RELATÓRIO DE PROBLEMA para descobrir todas as implicações e gerar possíveis SOLICITAÇÕES DE MUDANÇA que identifiquem todos os ITENS DE CONFIGURAÇÃO que precisam ser alterados e todos os passos de VERIFICAÇÃO necessários.

B.6.3 Implementação de modificação

Esta ATIVIDADE requer que o FABRICANTE use um PROCESSO estabelecido para realizar a modificação. Se um PROCESSO de manutenção não foi definido, as TAREFAS DE PROCESSO de desenvolvimento adequadas podem ser usadas para realizar a modificação. O FABRICANTE deve também garantir que a modificação não cause efeito negativo em outras partes do SOFTWARE DE PRODUTO PARA A SAÚDE. A menos que o SOFTWARE DE PRODUTO PARA A SAÚDE seja tratado como um novo desenvolvimento, a análise do efeito de uma modificação em todo o SOFTWARE DE PRODUTO PARA A SAÚDE é necessária. Uma análise lógica deve ser feita para justificar a quantidade de ENSAIOS DE REGRESSÃO que serão realizados para garantir que as partes do SOFTWARE DE PRODUTO PARA A SAÚDE que não estão sendo modificadas ainda desempenham como antes da modificação ser realizada.

B.7 PROCESSO DE GERENCIAMENTO DE RISCO DE SOFTWARE

GERENCIAMENTO DE RISCO DE Software é uma parte de todo o GERENCIAMENTO DE RISCO DE PRODUTO PARA A SAÚDE e não podem ser tratados adequadamente de forma isolada. Esta norma requer o uso de um PROCESSO DE GERENCIAMENTO DE RISCO em conformidade com a ISO 14971. GERENCIAMENTO DE RISCO, como definido na ISO 14971, trata especificamente com uma estrutura de gerenciamento eficaz dos RISCOS associados ao uso do PRODUTO PARA A SAÚDE. Uma parte da ISO 14971 diz respeito ao controle dos RISCOS identificados associados com cada PERIGO identificado durante a ANÁLISE DE RISCOS. O PROCESSO DE GERENCIAMENTO DE RISCO de software nesta norma destina-se a fornecer requisitos adicionais para CONTROLE DE RISCOS do software, incluindo o software que foi identificado durante a ANÁLISE DE RISCO como sendo potencialmente contribuinte para uma situação de perigo, ou software que é usado para controlar RISCOS DE PRODUTO PARA A SAÚDE. O PROCESSO DE GERENCIAMENTO DE RISCO de software está incluído nesta norma por duas razões:

- a) o público-alvo desta norma necessita entender os requisitos mínimos para medidas de CONTROLE DE RISCOS em sua área de responsabilidade – software;
- b) a norma geral de GERENCIAMENTO DE RISCO, ISO 14971, fornecida como referência

nesta norma, não trata especificamente o CONTROLE DE RISCO de software e o posicionamento do CONTROLE DE RISCO no ciclo de vida do desenvolvimento do software.

O GERENCIAMENTO DE RISCO de software é uma parte de todo o GERENCIAMENTO DE RISCO DO PRODUTO PARA A SAÚDE. Planos, procedimentos e documentação necessárias para as ATIVIDADES DE GERENCIAMENTO DE RISCO podem ser uma série de documentos separados ou um único documento, ou podem estar integrados à documentação e às ATIVIDADES DE GERENCIAMENTO DE RISCO DO PRODUTO PARA A SAÚDE desde que todos os requisitos desta norma sejam cumpridos.

B.7.1 Análise de software contribuindo para situações de perigo

É esperado que a análise do dispositivo PERIGOSO identifique situações perigosas e medidas de CONTROLE DE RISCO correspondentes para reduzir a probabilidade e/ou severidade daquelas situações de perigo a um nível aceitável. É também esperado que as medidas de CONTROLE DE RISCO sejam atribuídas a funções do software que se espera implementem aquelas medidas de CONTROLE DE RISCOS.

No entanto, não é esperado que todas as situações perigosas do dispositivo possam ser identificadas até a ARQUITETURA do software ter sido produzida. Nesse momento é conhecido como as funções do software serão implementadas em componentes de software e a praticidade das medidas de CONTROLE DE RISCO atribuídas às funções do software podem ser AVALIADAS. Nesse momento, a análise do dispositivo PERIGOSO deve ser revisado para incluir:

- situações de perigo revisadas;
- medidas de CONTROLE DE RISCO e requisitos de software revisados;
- novas situações perigosas devido ao software, por exemplo, situações perigosas relacionadas a fatores humanos.

A ARQUITETURA do software deve incluir estratégias confiáveis para segregar componentes de software de tal forma que não interajam de forma insegura.

B.8 PROCESSO de gerenciamento de configuração do software

O PROCESSO de gerenciamento de configuração do software é um PROCESSO de aplicação de procedimentos técnicos e administrativos ao longo do ciclo de vida do software para identificar e definir ITENS DE SOFTWARE, incluindo documentação, num SISTEMA; controlar modificações e liberação de itens; e documentar e relatar o status dos itens e SOLICITAÇÕES DE MUDANÇA. O gerenciamento de configuração de software é necessário para recriar o ITEM DE SOFTWARE, para identificar suas partes constituintes, e para fornecer um histórico das mudanças que tenham sido feitas a ele.

B.8.1 Identificação de configuração

Esta ATIVIDADE requer que o FABRICANTE unicamente identifique os ITENS DE CONFIGURAÇÃO do software e suas VERSÕES. Esta identificação é necessária para identificar os ITENS DE CONFIGURAÇÃO de software e as VERSÕES que são incluídas no SOFTWARE DE PRODUTO PARA A SAÚDE.

B.8.2 Controle de mudança

Esta ATIVIDADE requer que o FABRICANTE controle as alterações dos ITENS DE CONFIGURAÇÃO do software e documente informações identificando as SOLICITAÇÕES DE MUDANÇA e forneça documentação sobre sua disposição. Esta ATIVIDADE é necessária para garantir que mudanças não autorizadas ou involuntárias não sejam feitas aos ITENS DE CONFIGURAÇÃO do software e garantir que SOLICITAÇÕES DE MUDANÇA aprovadas são implementadas completamente e verificadas.

SOLICITAÇÕES DE MUDANÇA podem ser aprovadas por um conselho de controle de alterações ou por um gerente ou por um líder técnico de acordo com o plano de gerenciamento de configuração de software. SOLICITAÇÕES DE MUDANÇA aprovadas são rastreáveis à modificação e VERIFICAÇÃO real do software. O requisito é que cada mudança real seja ligada a uma SOLICITAÇÃO DE MUDANÇA e que exista documentação para mostrar que a SOLICITAÇÃO DE MUDANÇA foi aprovada. A documentação pode ser a ata do conselho de controle de mudanças, uma assinatura de aprovação ou um registro em uma base de dados.

B.8.3 Relato de situação da configuração

Esta ATIVIDADE requer que o FABRICANTE mantenha registros do histórico dos ITENS DE CONFIGURAÇÃO do software. Esta ATIVIDADE é necessária para determinar quando e porque as alterações foram feitas. O acesso a esta informação é necessário para garantir que os ITENS DE CONFIGURAÇÃO do software contem somente modificações autorizadas.

B.9 PROCESSO de resolução de problema do software

O PROCESSO de resolução de problema do software é um PROCESSO para analisar e resolver os problemas (incluindo não-conformidades), qualquer que seja a fonte ou natureza, incluindo aqueles descobertos durante a execução do desenvolvimento, manutenção ou outros PROCESSOS. O objetivo é fornecer um meio oportuno, responsável e documentado para garantir que os problemas descobertos são analisados e resolvidos e que tendências são reconhecidas. Este PROCESSO é às vezes chamado “rastreamento de falha” na literatura de engenharia de software. É chamado “resolução de problema” na ISO/IEC 12207 [9] e IEC 60601-1-4 [2], Emenda 1. Nesta norma escolheu-se chamar “resolução de problema de software”.

Esta ATIVIDADE requer que o FABRICANTE use o PROCESSO de resolução de problema de software quando um problema ou não-conformidade é identificado. Esta ATIVIDADE é necessária para garantir que os problemas descobertos são analisados e AVALIADOS para possível relevância à SEGURANÇA (conforme especificado na ISO 14971).

Plano(s) de desenvolvimento de software ou procedimentos, conforme requerido em 5.1, são para abordar como problemas ou não-conformidades serão tratados. Isto inclui especificar em cada estágio do ciclo de vida os aspectos do PROCESSO de resolução do problema de software que serão formais e documentados bem como quando problemas e não-conformidades estão para ser inseridos no PROCESSO de resolução do problema de software.

Anexo C (informativo)

Relacionamento com outras normas

C.1 Geral

Esta norma se aplica ao desenvolvimento e manutenção do SOFTWARE DE PRODUTO PARA A SAÚDE. O software é considerado um subsistema do PRODUTO PARA A SAÚDE ou é ele mesmo um PRODUTO PARA A SAÚDE. Esta norma é para ser usada em conjunto com outras normas apropriadas no desenvolvimento de um PRODUTO PARA A SAÚDE.

Normas de gerenciamento de PRODUTOS PARA A SAÚDE como a ISO 13485 [7] (ver C.2 e Anexo D) e ISO 14971 (Ver Anexo 0) fornecem um ambiente de gerenciamento que estabelece uma fundação para uma organização desenvolver produtos. Normas de segurança como a IEC 60601-1 [1] (ver Anexo C.4) e IEC 61010-1 [4] (ver Anexo C.5) dão direção específica para a criação de PRODUTOS PARA A SAÚDE seguros. Quando o software é uma parte destes PRODUTOS PARA A SAÚDE, a IEC 62304 fornece um direcionamento mais detalhado no que é necessário para desenvolver e manter seguro um PRODUTO PARA A SAÚDE. Muitas outras normas, como ISO/IEC 12207 [9] (ver Anexo C.6), IEC 61508-3 [3] (ver Anexo C.7) e ISO/IEC 90003 [11] podem ser vistas como fonte de métodos, ferramentas e técnicas que podem ser usadas para implementar os requisitos da IEC 62304. A Figura C.1 mostra o relacionamento entre estas normas.

Onde cláusulas ou requisitos de outras normas são citados, termos definidos nos itens citados são termos que são definidos em outra norma, e não termos definidos nesta norma.

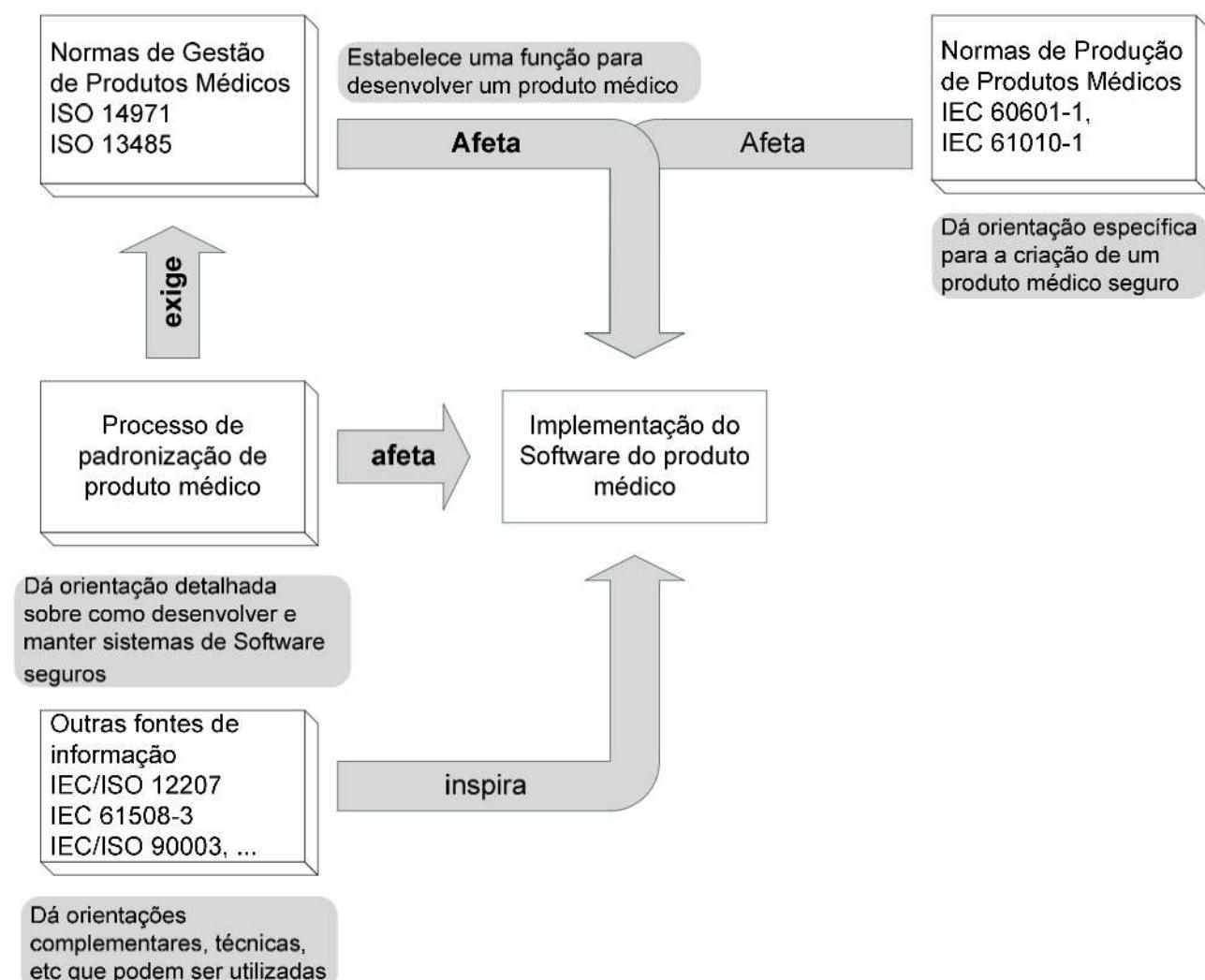


Figura C.1 – Relacionamento de normas fundamentais de PRODUTOS PARA A SAÚDE com a IEC 62304

C.2 Relacionamento com ISO 13485

Esta norma requer que o FABRICANTE empregue um sistema de gerenciamento de qualidade. Quando um FABRICANTE utiliza a ISO 13485 [7], os requisitos da ISO 62304 se relacionam diretamente a alguns dos requisitos da ISO 13485 conforme mostrado na Tabela C.1.

Tabela C.1 - Relacionamento com ISO 13485:2003

Cláusula IEC62304	Cláusula da ISO 13485:2003 relacionada
5.1 Planejamento de desenvolvimento de software	7.3.1 Planejamento de desenvolvimento e projeto
5.2 Análise de requisitos de software	7.3.2 Entradas de projeto e desenvolvimento
5.3 Projeto da ARQUITETURA de software	
5.4 Projeto detalhado do software	
5.5 Verificação e implementação da UNIDADE DE SOFTWARE	
5.6 Integração de software e ensaio de integração	
5.7 Ensaio do SISTEMA DE SOFTWARE	7.3.3 Saídas de projeto e desenvolvimento 7.3.4 Revisão de projeto e desenvolvimento
5.8 Liberação de software	7.3.5 Verificação de projeto e desenvolvimento 7.3.6 Validação de projeto e desenvolvimento
6.1 Estabelecer plano de manutenção de software	7.3.7 Controle de projeto e de mudanças de desenvolvimento
6.2 Análise de problema e modificação	
6.3 Implementação de modificação	7.3.5 Verificação de projeto e desenvolvimento 7.3.6 Validação de projeto e desenvolvimento
7.1 Análise de softwares que contribuem para situações perigosas	
7.2 Medidas de CONTROLE DE RISCO	
7.3 Medidas de CONTROLE DE RISCO e verificação	
7.4 GERENCIAMENTO DE RISCO de mudança de software	
8.1 Identificação de configuração	7.5.3 Identificação e RASTREABILIDADE
8.2 Controle de mudança	7.5.3 Identificação e RASTREABILIDADE
8.3 Contabilidade do status de configuração	
9 PROCESSO de resolução de problema de software	

C.3 Relacionamento com ISO 14971

A Tabela C.2 exibe as áreas onde a IEC 62304 amplia os requisitos para o PROCESSO DE GERENCIAMENTO DE RISCO necessários à ISO 14971.

Tabela C.2 - Relacionamento com ISO 14971:2000

Cláusula ISO 14971:2000	Cláusula da IEC 62304 relacionada
4.1 Procedimento de ANÁLISE DE RISCO	
4.2 Uso pretendido/ propósito pretendido e identificação das características relacionadas à SEGURANÇA do PRODUTO PARA A SAÚDE	
4.3 Identificação de PERIGOS conhecidos ou previsíveis	7.1 Análise de software que contribui para situações perigosas
4.4 Estimativa do(s) RISCO(O) de cada PERIGO	4.3 Classificação da segurança de software
5 Avaliação do risco	
6.1 Redução do risco	
6.2 Análise de opções	7.2.1 Definir medidas de CONTROLE DE RISCO
6.3 Implementação de medidas de CONTROLE DE RISCO	7.2.2 Medidas de CONTROLE DE RISCO implementadas no software 7.3.1 Verificar medidas de CONTROLE DE RISCO
6.4 Avaliação do RISCO residual	
6.5 Análise do risco/ benefício	
6.6 Outros PERIGOS gerados	7.3.2 Documentar qualquer nova sequência de eventos
6.7 Perfeição da avaliação do RISCO	
7 Avaliação do RISCO residual global	
8 Relatório de GERENCIAMENTO DE RISCO	7.3.3 RASTREABILIDADE de documentos
9 Informação de pós-produção	7.4 GERENCIAMENTO DE RISCO de mudanças de software

C.4 Relacionamento com requisitos de um SEMP de IEC 60601-1:2005

C.4.1 Geral

Os requisitos para um software são um subconjunto dos requisitos para um sistema eletro médico programável (SEMP). Esta norma identifica os requisitos de software que estão além, mas não são incompatíveis com, os requisitos da IEC 60601-1 [1] para SEMP. Como os SEMP incluem elementos que não são software, nem todos os requisitos da IEC 60601-1 para SEMP são abordados nesta norma.

C.4.2 Relacionamento do software ao desenvolvimento de SEMP

Com o uso do modelo-V ilustrado na Figura C.2 para descrever o que ocorre durante o desenvolvimento de SEMP, pode ser visto que os requisitos desta norma de software se aplicam ao SEMP em nível de componente, da especificação dos requisitos de software à integração dos ITENS DE SOFTWARE em um SISTEMA DE SOFTWARE. Este SISTEMA DE SOFTWARE é uma parte do subsistema elétrico programável (SSEP), que é parte do SEMP.

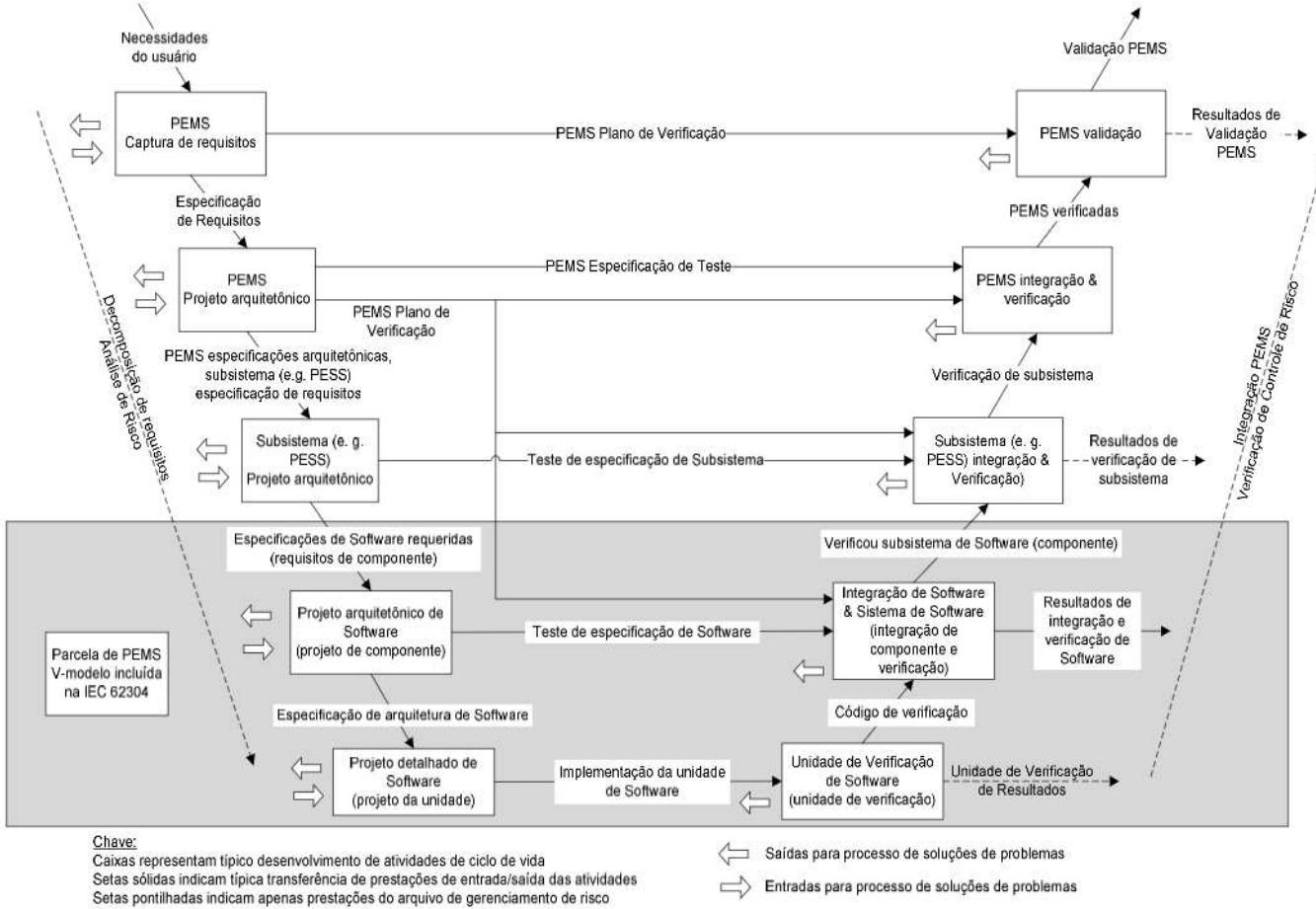


Figura C.2 – Software como parte do modelo-V

C.4.3 PROCESSO de desenvolvimento

Conformidade com PROCESSO de desenvolvimento de software desta norma (Cláusula 5) requer que o plano de desenvolvimento do software seja especificado e então seguido; isto não requer que qualquer modelo de ciclo de vida em particular seja usado, mas requer que o plano inclua certas ATIVIDADES e que tenha certos atributos. Estes requisitos dizem respeito aos requisitos do SEMP na IEC 60601-1 para ciclo de vida de desenvolvimento, especificação de requisito, ARQUITETURA, projeto e implementação e VERIFICAÇÃO. Os requisitos nesta norma fornecem maiores detalhes sobre desenvolvimento de software que na IEC 60601-1.

C.4.4 PROCESSO de manutenção

Conformidade com PROCESSO de manutenção de software desta norma (Cláusula 6) requer que procedimentos sejam estabelecidos e seguidos quando mudanças forem feitas ao software. Estes requisitos correspondem aos requisitos na IEC 60601-1 para modificação de um SEMP. Os requisitos nesta norma de manutenção de software fornecem maiores detalhes sobre o que deve ser feito para manutenção do software que os requisitos para modificação de SEMP na IEC 60601-1.

C.4.5 Outros PROCESSOS

Os outros PROCESSOS nesta norma especificam requisitos adicionais para o software além dos similares para SEMP na IEC 60601-1. Na maioria dos casos, há um requisito geral para SEMP na IEC 60601-1, que os PROCESSOS nesta norma espande.

O PROCESSO DE GERENCIAMENTO DE RISCO do software nesta norma corresponde aos requisitos de GERENCIAMENTO DE RISCO adicionais identificados para SEMP na IEC 60601-1.

O PROCESSO de resolução de problema de software nesta norma corresponde aos requisito de resolução de problema para SEMP na IEC 60601-1.

O PROCESSO de gerenciamento de configuração de software nesta norma especifica requisitos adicionais que não estão presentes para SEMP na IEC 60601-1, exceto para documentação.

C.4.6 Cobertura de requisitos de SEMP na IEC 60601-1

A Tabela C.3 mostra os requisitos de SEMP da IEC 60601-1 e os requisitos correspondentes nesta norma.

Tabela C.3 – Relacionamento com IEC 60601-1

Requisitos para SEMP na IEC 60601-1:2005	Requisitos da IEC 62304 relacionados a subsistemas de software de um SEMP
14.1 Geral Os requisitos desta cláusula devem se aplicar a SEMP a menos que: <ul style="list-style-type: none"> • o SSEP não forneça SEGURANÇA BÁSICA ou DESEMPENHO ESSENCEIAL; ou • a aplicação da ISO 14971 demonstra que a falha do SSEP não leva a um RISCO inaceitável. 	4.3 Classificação de segurança de software Os requisitos do SEMP na IEC 60601 somente se aplicariam para software com classificação de segurança B e C. Esta norma inclui alguns requisitos para software com classificação de segurança A.
14.2 Documentação Além dos registros e documentos exigidos pela ISO 14971, os documentos produzidos pela aplicação da Cláusula 14 devem fazer parte do ARQUIVO DE GERENCIAMENTO DE RISCO.	4.2 GERENCIAMENTO DE RISCO
Os documentos exigidos pela Cláusula 14 devem ser revisados, aprovados, emitido e alterado de acordo com um procedimento de controle formal do documento.	5.1 Planejamento do desenvolvimento de software Além dos requisitos específicos na ATIVIDADE de planejamento do desenvolvimento de software, documentos que são parte do ARQUIVO DE GERENCIAMENTO DE RISCO são necessários serem mantido pela ISO 14971. Além disso, para documentos que são necessários pelo sistema de qualidade, a ISO 13485 [7] requer controle dos documentos.
14.3 Plano de GERENCIAMENTO DE RISCO O plano de GERENCIAMENTO DE RISCO exigido por 3.5 da ISO 14971 deve também incluir uma referência para o plano de VALIDAÇÃO do SEMP (ver 14.11).	Não especificamente exigido.
Um CICLO DE VIDA DE DESENVOLVIMENTO de SEMP deve ser documentado.	Não há plano de validação de software específico. O plano de VALIDAÇÃO do SEMP está a nível do sistema e portanto fora do escopo desta norma de software. Esta norma requer RASTREABILIDADE do PERIGO de uma causa de software específica para a medida de CONTROLE DE RISCO, para VERIFICAÇÃO da medida de CONTROLE DE RISCO (VER 7.3).
14.4 CICLO DE VIDA DE DESENVOLVIMENTO de SEMP O CICLO DE VIDA DE DESENVOLVIMENTO de SEMP deve conter um conjunto de pontos de controle	5.1 Planejamento do desenvolvimento de software 5.1.1 Plano de desenvolvimento de software Os itens abordados pelo plano de desenvolvimento de software constituem um ciclo de vida de desenvolvimento de software.

Requisitos para SEMP na IEC 60601-1:2005	Requisitos da IEC 62304 relacionados a subsistemas de software de um SEMP
definidos.	
A cada ponto de controle, as ATIVIDADES a serem completadas e os métodos de VERIFICAÇÃO a serem aplicados àquelas ATIVIDADES devem ser definidos.	5.1.6 Planejamento de VERIFICAÇÃO do software TAREFAS de VERIFICAÇÃO, pontos de controle e critérios de aceitação devem ser planejados.
Cada ATIVIDADE deve ser definida incluindo suas entradas e saídas.	5.1.1 Plano de desenvolvimento de software ATIVIDADES são definidas nesta norma. A documentação a ser produzida é definida em cada ATIVIDADE.
Cada ponto de controle deve identificar as ATIVIDADES de GERENCIAMENTO DE RISCO que devem ser completadas antes de atingir este ponto.	
O CICLO DE VIDA DE DESENVOLVIMENTO do SEMP deve ser adaptado através de planos que detalham ATIVIDADES, pontos de controle e cronogramas.	5.1.1 Plano de desenvolvimento de software Esta norma permite que o CICLO DE VIDA DE DESENVOLVIMENTO seja documentado no plano de desenvolvimento. Isto significa que o plano de desenvolvimento contém um CICLO DE VIDA DE DESENVOLVIMENTO adaptado.
O CICLO DE VIDA DE DESENVOLVIMENTO deve incluir os requisitos de documentação.	5.1.1 Plano de desenvolvimento de software 5.5.8 Planejamento de documentação
14.5 Resolução de problema Onde for apropriado, um sistema documentado para resolução de problema dentro e entre todas as fases e ATIVIDADES do CICLO DE VIDA DE DESENVOLVIMENTO de SEMP deve ser desenvolvido e mantido.	9 PROCESSO de resolução de problema de software
Dependendo do tipo de produto, o SISTEMA de resolução de problema pode: <ul style="list-style-type: none"> • ser documentado como parte do CICLO DE VIDA DE DESENVOLVIMENTO do SEMP; • permitir o relato de problemas potenciais ou existentes afetando a SEGURANÇA BÁSICA ou DESEMPENHOS ESSENCIAL; • incluir uma avaliação de cada problema para RISCOS associados; • identificar o critério que deve ser cumprido para que o assunto seja fechado; • identificar a ação a ser tomada para resolver cada problema. 	5.1.1 Plano de desenvolvimento de software 9.1 Preparar RELATÓRIOS DE PROBLEMA
14.6 PROCESSO DE GERENCIAMENTO DE RISCO	7 PROCESSO DE GERENCIAMENTO DE RISCO de software
14.6.1 Identificação de PERIGOS conhecidos e previsíveis Ao compilar a lista de PERIGOS conhecidos e previsíveis, o FABRICANTE deve considerar aqueles PERIGOS associados aos aspectos de software e hardware do SEMP, incluindo aqueles associados ao ACOPLAMENTO REDE/DADOS, componentes de terceiros e subsistemas legados.	7.1 Análise de software contribuindo para situações de PERIGO Esta norma não menciona especificamente o ACOPLAMENTO REDE/DADOS.
14.6.2 CONTROLE DE RISCO	5.1.4 Normas de desenvolvimento de

Requisitos para SEMP na IEC 60601-1:2005	Requisitos da IEC 62304 relacionados a subsistemas de software de um SEMP
<p>Ferramentas devidamente validadas e PROCEDIMENTOS devem ser selecionados e identificados para implementar cada medida de CONTROLE DE RISCO. Estas ferramentas e PROCEDIMENTOS devem ser apropriados para assegurar que cada medida de CONTROLE DE RISCO reduz satisfatoriamente o(s) RISCO(S) identificado(s).</p>	<p>software, métodos e ferramentas de planejamento</p> <p>Esta norma requer a identificação de ferramentas e métodos específicos a serem usados para o desenvolvimento em geral, não para cada medida de CONTROLE DE RISCO.</p>
14.7 Especificação de requisitos	5.2 Análise de requisito de software
<p>Para o SEMP e cada um dos seus subsistemas (exemplo, para um SSEP) deveria haver uma especificação de requisito documentada.</p>	<p>Esta norma trata apenas de subsistemas de software de um SEMP.</p>
<p>Esta especificação de requisito de um sistema ou subsistema deve incluir e distinguir qualquer DESEMPENHO ESSENCEIAL e qualquer medida de CONTROLE DE RISCO implementada pelo sistema ou subsistema.</p>	<p>5.2.1 Definir e documentar requisitos de software dos requisitos de SISTEMA.</p> <p>5.2.2 Conteúdo dos requisitos de software.</p> <p>5.2.3 Incluir medidas de CONTROLE DE RISCO nos requisitos de software.</p> <p>Esta norma não requer que os requisitos relacionados ao desempenho essencial e medidas de CONTROLE DE RISCO sejam distinguidos de outros requisitos, mas ela requer que todos os requisitos sejam identificados unicamente.</p>
14.8 Arquitetura	5.3 Projeto da ARQUITETURA do software
<p>Para o SEMP e cada um de seus subsistemas, uma arquitetura deve ser especificada de forma que satisfaça à especificação dos requisitos.</p>	
<p>Onde for apropriado, para reduzir o RISCO a um nível aceitável, a especificação da ARQUITETURA deve fazer uso de:</p> <ul style="list-style-type: none"> a) COMPONENTES COM CARACTERÍSTICAS DE ALTA INTEGRAÇÃO; b) funções seguras quanto a falha; c) redundância; d) diversidade; e) particionamento de funcionalidade; f) projeto defensivo, por exemplo limitar efeitos potencialmente perigosos pela restrição da potência de saída disponível ou a introdução de meios para limitar o curso de atuadores; 	<p>5.3.5 Identificar a separação necessária para o CONTROLE DE RISCO</p> <p>O particionamento é a única técnica identificada, e somente é identificada porque há um requisito para indicar como a integridade da partição é garantida.</p>
<p>A especificação da ARQUITETURA deve levar em consideração:</p> <ul style="list-style-type: none"> g) alocação de medidas de CONTROLE DE RISCO para subsistemas e componentes do SEMP; h) modos de falha de componentes e seus efeitos; i) causas comuns de falha; j) falhas sistêmicas; k) duração do intervalo de teste e cobertura do diagnóstico; l) manutenibilidade; 	<p>Isto não está incluído nesta norma.</p>

Requisitos para SEMP na IEC 60601-1:2005	Requisitos da IEC 62304 relacionados a subsistemas de software de um SEMP
<p>m) proteção contra mau uso razoavelmente previsível;</p> <p>n) a especificação do ACOPLAMENTO REDE/DADOS, se aplicável.</p>	
14.9 Projeto e implementação	5.4 Projeto detalhado do software
<p>Onde for apropriado, o projeto deve ser decomposto em subsistemas, cada um tendo a sua especificação de projeto e de teste.</p>	5.4.2 Projeto detalhado do desenvolvimento para cada UNIDADE DE SOFTWARE Esta norma não requer uma especificação de teste para projeto detalhado.
<p>Dados descritivos sobre o ambiente de projeto devem ser incluídos no ARQUIVO DE GERENCIAMENTO DE RISCO.</p>	5.4.2 Projeto detalhado do desenvolvimento para cada UNIDADE DE SOFTWARE
14.10 VERIFICAÇÃO	5.1.6 Plano de VERIFICAÇÃO de software
<p>A VERIFICAÇÃO é necessária para todas as funções que implementem SEGURANÇA BÁSICA, DESEMPENHO ESSENCIAL ou medidas de CONTROLE DE RISCO.</p>	A VERIFICAÇÃO é exigida para cada ATIVIDADE.
<p>Um plano de VERIFICAÇÃO deve ser produzido para mostrar como estas funções devem ser verificadas. O plano deve incluir:</p> <ul style="list-style-type: none"> • em que ponto(s) de controle a VERIFICAÇÃO deve ser realizada para cada função; • a seleção e documentação de estratégias de VERIFICAÇÃO, atividades, técnicas, e o nível apropriado de independência do pessoal realizando a VERIFICAÇÃO; • a seleção e utilização de ferramentas de VERIFICAÇÃO; • critérios de cobertura para a VERIFICAÇÃO. 	5.1.6 Plano de VERIFICAÇÃO de software A independência do pessoal não está incluída nesta norma. Ela é considerada coberta na ISO 13485.
<p>A VERIFICAÇÃO deve ser realizada de acordo com um plano de VERIFICAÇÃO. Os resultados das atividades de VERIFICAÇÃO devem ser documentados.</p>	Requisitos de VERIFICAÇÃO estão na maioria das ATIVIDADES.
14.11 VALIDAÇÃO de SEMP	Esta norma não cobre validação de software. A validação de SEMP é uma ATIVIDADES em nível de SISTEMA e está fora do escopo desta norma.
<p>Um plano de VALIDAÇÃO de SEMP deve incluir a validação de SEGURANÇA BÁSICA e DESEMPENHO ESSENCIAL e deve exigir a checagem para funcionamento indesejado do SEMP.</p>	Esta norma não cobre validação de software. A validação de SEMP é uma ATIVIDADES em nível de SISTEMA e está fora do escopo desta norma.
<p>A VALIDAÇÃO de SEMP deve ser realizada de acordo com um plano de VALIDAÇÃO de SEMP. Os resultados das atividades de VALIDAÇÃO do SEMP devem ser documentados.</p>	Esta norma não cobre validação de software. A validação de SEMP é uma ATIVIDADES em nível de SISTEMA e está fora do escopo desta norma.
<p>A pessoal que tenha total responsabilidade pela VALIDAÇÃO do SEMP deve ser independente do time de projeto. O FABRICANTE deve documentar a justificativa para o nível de independência.</p>	Esta norma não cobre validação de software. A validação de SEMP é uma ATIVIDADES em nível de SISTEMA e está fora do escopo desta norma.
<p>Nenhum membro do time de projeto deve ser responsável pela VALIDAÇÃO do SEMP de seu próprio projeto.</p>	Esta norma não cobre validação de software. A validação de SEMP é uma ATIVIDADES em nível de SISTEMA e está fora do escopo desta norma.
<p>Todo relacionamento profissional dos membros do time de VALIDAÇÃO de SEMP com os membros do time de projeto deve ser documentado no ARQUIVO DE GERENCIAMENTO DE RISCO.</p>	Esta norma não cobre validação de software. A validação de SEMP é uma ATIVIDADES em nível de SISTEMA e está fora do escopo desta norma.

Requisitos para SEMP na IEC 60601-1:2005	Requisitos da IEC 62304 relacionados a subsistemas de software de um SEMP
Uma referência para os métodos e resultados da VALIDAÇÃO do SEMP devem ser incluídos no ARQUIVO DE GERENCIAMENTO DE RISCO.	Esta norma não cobre validação de software. A validação de SEMP é uma ATIVIDADE em nível de SISTEMA e está fora do escopo desta norma.
14.12 Modificação Se algo ou tudo de um projeto resulta de uma modificação de um projeto antigo então toda esta cláusula se aplica como se fosse um novo projeto ou a prorrogação da validade de qualquer documentação de projeto prévia deve ser assegurada mediante um PROCEDIMENTO de modificação/ alteração documentado.	6 PROCESSO de manutenção do software Esta norma adota a abordagem de que a manutenção do software deve ser planejada e que a implementação das modificações deve usar o PROCESSO de desenvolvimento de software ou um PROCESSO de manutenção de software estabelecido.
14.13 Conexão de SEMP via ACOPLAMENTO REDE/ DADOS a outro equipamento Se o SEMP se destina a ser conectado via ACOPLAMENTO REDE/ DADOS a outro equipamento que está fora do controle do FABRICANTE do SEMP, a descrição técnica deve: a) especificar as características do ACOPLAMENTO REDE/ DADOS necessárias para que o SEMP atinja o seu USO PRETENDIDO; b) listar as SITUAÇÕES DE PERIGO resultantes de uma falha do ACOPLAMENTO REDE/ DADOS em fornecer as características especificadas; c) instruir a ORGANIZAÇÃO RESPONSÁVEL que: <ul style="list-style-type: none">• a conexão de SEMP em um ACOPLAMENTO REDE/ DADOS que inclui outro equipamento pode resultar em RISCOS ao PACIENTE, OPERADORES ou terceiros, não previamente identificados;• a ORGANIZAÇÃO RESPONSÁVEL deve identificar, analisar, avaliar e controlar estes RISCOS;• subsequentes mudanças ao ACOPLAMENTO REDE/ DADOS podem introduzir novos riscos e requerer análises adicionais; e• mudanças ao ACOPLAMENTO REDE/ DADOS incluem:<ul style="list-style-type: none">◦ mudanças na configuração do ACOPLAMENTO REDE/ DADOS;◦ conexão de itens adicionais ao ACOPLAMENTO REDE/ DADOS;◦ desconexão de itens do ACOPLAMENTO REDE/ DADOS;◦ atualização de equipamento conectado ao ACOPLAMENTO REDE/ DADOS;◦ melhoria no equipamento conectado ao ACOPLAMENTO REDE/ DADOS.	Requisitos para acoplamento rede/ dados não estão incluídos nesta norma.

C.4.7 Relacionamento com requisitos na IEC 60601-1-4

A IEC 60601-1-4 continuará a ser utilizada até o período de transição para IEC 60601-1:2005 estar completo.

A Tabela C.4 mostra o relacionamento entre os requisitos da IEC 60601-1-4 e os requisitos desta norma. Isto não indica que os requisitos relacionados nesta norma cobrem completamente os requisitos na IEC 60601-1-4. Muitas partes dos requisitos da IEC 60601-1-4 são cobertos pela conformidade com a ISO 14971. Alguns requisitos da IEC 60601-1-4 não são tratados pela IEC 62304.

Tabela C.4 – Relacionamento com IEC 60601-1-4

Requisitos para SEMP da ISO 60601-1-4:1996 mais Emenda 1:1999	Requisitos relacionados da IEC 62304
6.8 Acompanhamento de documentos	
6.8.201	4.2 e 4.3 c)
52.201 Documentação	
52.201.1	4.1
52.201.2	4.1 e 4.2
52.201.3	4.2
52.202 PLANO DE GERENCIAMENTO DE RISCO	
52.202.1	4.2
52.202.2	5.1.1, 5.1.5
52.202.3	4.1, 5.1.2
52.203 Ciclo de vida do desenvolvimento	
52.203.1	5.1.1
52.203.2	5.1.1
52.203.3	
52.203.4	5.1.7
52.203.5	7
52.204 Processo de gerenciamento de risco	
52.204.1	4.2
52.204.2	4.2, 7
52.204.3	
52.204.3.1	
52.204.3.1.1	4.2, 7.1
52.204.3.1.2	4.2, 7.1.2
52.204.3.1.3	4.2
52.204.3.1.4	4.2, 7.1.2 e)
52.204.3.1.5	4.2, 7.1.2
52.204.3.1.6	4.2, 7.1
52.204.3.1.7	4.2
52.204.3.1.8	4.2
52.204.3.1.9	4.2
52.204.3.1.10	4.2
52.204.3.2	
52.204.3.2.1	4.2

Requisitos para SEMP da ISO 60601-1-4:1996 mais Emenda 1:1999	Requisitos relacionados da IEC 62304
52.204.3.2.2	4.2, 4.3
52.204.3.2.3	
52.204.3.2.4	
52.204.3.2.5	4.2
52.204.4	
52.204.4.1	4.2
52.204.4.2	4.2
52.204.4.3	4.2
52.204.4.4	4.2
52.204.4.5	
52.204.4.6	4.2
52.205 Qualificação de pessoal	4.1
52.206 Especificação de requisitos	
52.206.1	5.2
52.206.2	7.2.2
52.206.3	
52.207 Arquitetura	
52.207.1	5.3.1
52.207.2	5.3
52.207.3	
52.207.4	
52.207.5	
52.208 Projeto e implementação	
52.208.1	5
52.208.2	
52.209 Verificação	
52.209.1	5.7.1
52.209.2	5.1.5, 5.1.6
52.209.3	5.2.6, 5.3.6, 5.4.4, 5.5.5, 5.6, 5.7
52.209.4	
52.210 Validação	
52.210.1	4.1
52.210.2	4.1
52.210.3	4.1
52.210.4	
52.210.5	
52.210.6	
52.210.7	
52.211 Modificação	
52.211.1	6
52.211.2	4.1, 6
52.212 Avaliação	

Requisitos para SEMP da ISO 60601-1-4:1996 mais Emenda 1:1999	Requisitos relacionados da IEC 62304
52.212.1	4.1

C.5 Relacionamento com IEC 61010-1

O escopo da IEC 61010-1 [4] abrange equipamentos elétrico de teste e medição, equipamento elétrico de controle e equipamento elétrico de laboratório. Somente parte do equipamento de laboratório é usada em um ambiente médico ou como em um equipamento de diagnóstico in vitro (DIV).

Devido à regulamentação legal ou referências normativas, um equipamento de DIV é alocado como PRODUTO PARA A SAÚDE sem, contudo, ser abrangido pelo escopo da IEC 60601-1 [1]. Isto é atribuível não somente pelo fato de, estritamente falando, instrumentos de DIV não são PRODUTOS PARA A SAÚDE que entram em contato direto com pacientes, mas também pelo fato de que cada produto é fabricado para muitas aplicações diferentes em vários laboratórios. O uso como instrumento de DIV ou como um acessório para um instrumento de DIV é então raro.

Se um equipamento de laboratório é usado como equipamento de DIV, os resultados medidos obtidos devem ser AVALIADOS de acordo com critérios médicos. A aplicação da ISO 14971 é necessária para o GERENCIAMENTO DE RISCO. Se estes produtos também contiverem software que pode levar a um PERIGO, por exemplo uma falha causada por um software que resulta em uma mudança indesejada em um dado médico (resultados de medidas), IEC 62304 deve ser levada em conta.

O fluxograma na Figura C.3 fornece uma útil ajuda para explicar a forma básica do PROCESSO DE GERENCIAMENTO DE RISCO e a aplicação da IEC 62304:

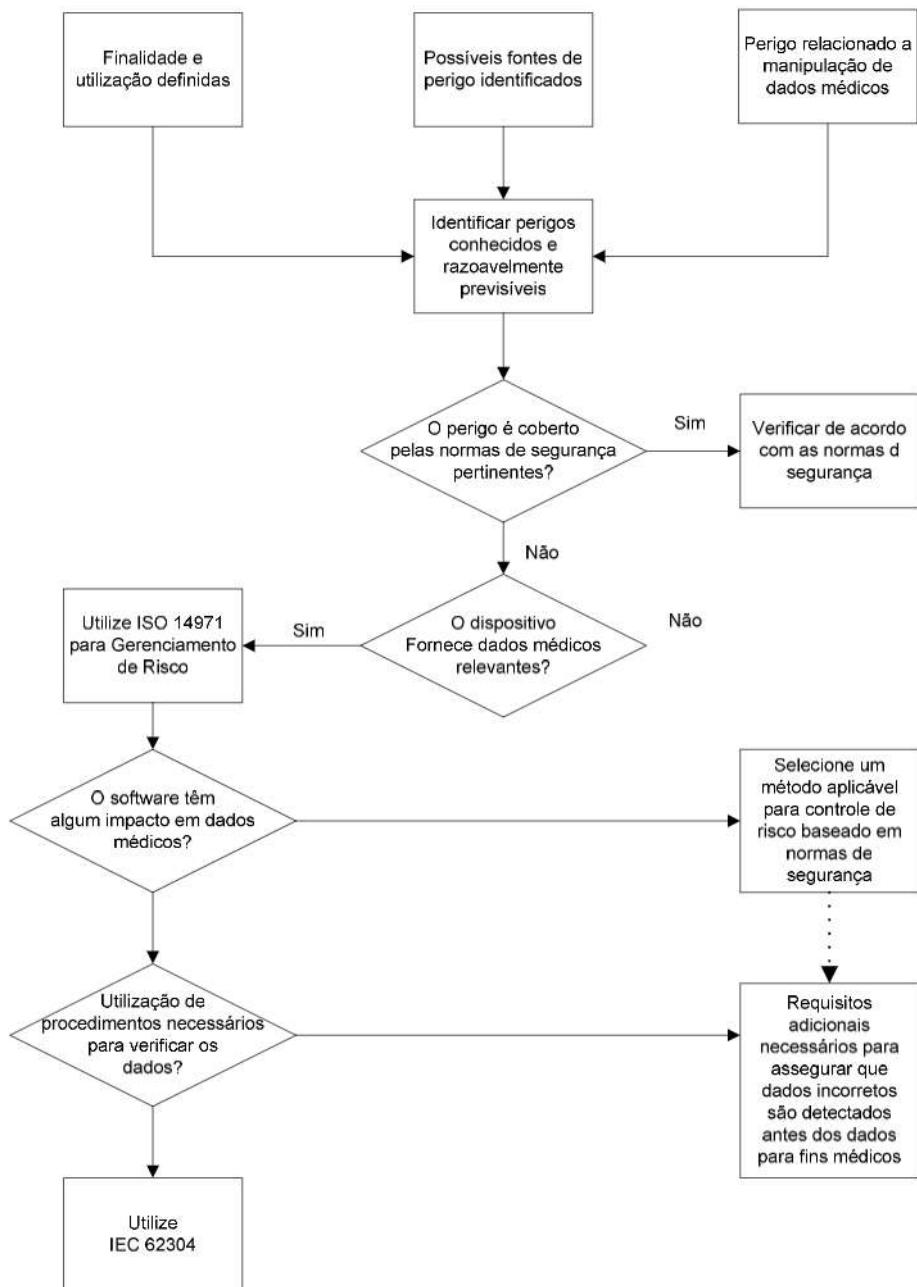


Figura C.3 – Aplicação da IEC 62304 com IEC 61010-1

C.6 Relacionamento com ISO/IEC 12207

Esta norma foi derivada da abordagem e conceitos da ISO/IEC 12207 [9], que define requisitos para PROCESSOS de ciclo de vida de software em geral, isto é, não restrito a PRODUTOS PARA A SAÚDE.

Esta norma difere da ISO/IEC 12207 principalmente com respeito ao seguinte. Ela:

- exclui aspectos de SISTEMA, tais como requisitos de SISTEMA, ARQUITETURA DE SISTEMA e validação;
- omite alguns PROCESSOS vistos como duplicação de ATIVIDADES documentadas em outro local para PRODUTOS PARA A SAÚDE;
- adiciona o PROCESSO DE GERENCIAMENTO DE RISCO (SEGURO) e PROCESSO de liberação de software;
- incorpora a documentação e os PROCESSOS de VERIFICAÇÃO de apoio nos PROCESSOS de manutenção e desenvolvimento;
- combina a implementação do PROCESSO e ATIVIDADES de planejamento de cada PROCESSO em uma ATIVIDADE única nos PROCESSOS de manutenção e desenvolvimento;
- classifica os requisitos conforme as necessidades de SEGURANÇA; e
- não classifica explicitamente PROCESSOS como primários ou de apoio, nem PROCESSOS de grupo, como faz a ISO/IEC 12207.

A maioria destas mudanças foram motivadas pelo desejo de adequar a norma às necessidades do setor de PRODUTOS PARA A SAÚDE por:

- focar os aspectos de SEGURANÇA e a norma ISO 14971 de GERENCIAMENTO DE RISCO DE PRODUTOS PARA A SAÚDE;
- selecionar PROCESSOS apropriados úteis em um ambiente regulado;
- levar em conta que o desenvolvimento de software está incorporado ao sistema da qualidade (que cobre alguns dos PROCESSOS e requisitos da ISO/IEC 12207); e
- reduzir o nível de abstração para facilitar o uso.

Esta norma não é contraditória à ISO/IEC 12207. A ISO/IEC 12207 pode ser útil como um auxílio na criação de um MODELO DE CICLO DE VIDA DE DESENVOLVIMENTO DE SOFTWARE bem estruturado que inclui os requisitos desta norma.

A Tabela C.5, preparada pela ISO/IEC JTC1/SC7, mostra o relacionamento entre IEC 62304 e ISO/IEC 12207.

Tabela C.5 – Relacionamento com ISO/IEC 12207

Processos da ISO/ IEC 62304		Processos da ISO/ IEC 12207	
Atividade	Tarefa	Atividade	Tarefa
5 Processo de desenvolvimento de software		5.3 Processo de desenvolvimento 6.1 Processo de documentação 6.2 Processo de gerenciamento de configuração 6.4 Processo de verificação 6.5 Processo de validação 6.8 Processo de resolução de problema 7.1 Processo de gerenciamento	
5.1 Plano de desenvolvimento de software		5.3.1 Processo de implementação 5.3.3 Projeto arquitetural do sistema 5.3.7 Codificação e ensaio do software 5.3.8 Integração do software 5.3.9 Ensaio de qualificação do software 5.3.10 Integração do sistema 6.1.1 Implementação de processo 6.2.1 Implementação de processo 6.2.2 Identificação de configuração 6.4.1 Implementação de processo 6.5.1 Implementação de processo 6.8.1 Implementação de processo 7.1.2 Planejamento 7.1.3 Execução e controle 7.2.2 Estabelecimento da infraestrutura 7.2.3 Manutenção da infraestrutura	
5.1.1 Plano de desenvolvimento do software		5.3.1 Implementação do processo 7.1.2 Planejamento	5.3.1.1 5.3.1.3 5.3.1.4 7.1.2.1
5.1.2 Manter plano de desenvolvimento do software atualizado		7.1.3 Execução e controle	7.1.3.3
5.1.3 Referência do plano de desenvolvimento do software para projeto e desenvolvimento do SISTEMA		5.3.3 Projeto arquitetural do sistema 5.3.10 Integração do sistema 6.5.1 Implementação de processo	5.3.3.1 5.3.10.1 6.5.1.4
5.1.4 Normas, métodos e		5.3.1 Implementação de processo	5.3.1.3

Processos da ISO/ IEC 62304		Processos da ISO/ IEC 12207	
Atividade	Tarefa	Atividade	Tarefa
5.1 Análise de requisitos de software	ferramentas de planejamento de desenvolvimento de software		5.3.1.4
	5.1.5 Integração do software e planejamento do ensaio de integração	5.3.8 Integração do software	5.3.8.1
	5.1.6 Planejamento de VERIFICAÇÃO do software	6.4.1 Implementação de processo 5.3.7 Codificação e ensaio do software 5.3.8 Integração do software 5.3.9 Ensaio de qualificação do software	6.4.1.4 6.4.1.5 5.3.7.5 5.3.8.5 5.3.9.3
	5.1.7 Planejamento do GERENCIAMENTO DE RISCO do software	Emenda 1:2002 – F 3.1.5 Processo de gerenciamento de risco	
	5.1.8 Planejamento de documentação	6.1.1 Implementação de processo	6.1.1.1
	5.1.9 Planejamento de gerenciamento de configuração do software	6.2.1 Implementação de processo 6.8.1 Implementação de processo	6.2.1.1 6.8.1.1
	5.1.10 Itens de apoio a serem controlados	7.2.2 Estabelecimento de infraestrutura 7.2.3 Manutenção da infraestrutura	7.2.2.1 7.2.3.1
	5.1.11 Controle de ITEM DE CONFIGURAÇÃO do software antes da VERIFICAÇÃO	6.2.2 Identificação da configuração	6.2.2.1
		5.3.3 Projeto arquitetural do sistema 5.3.4 análise de requisitos de software 6.4.2 Verificação	
	5.2.1 Define e documenta os requisitos de software dos requisitos do SISTEMA	5.3.3 Projeto arquitetural do software	5.3.3.1
5.2 Projeto ARQUITETURAL do software	5.2.2 Conteúdo dos requisitos de software	5.3.4 Análise dos requisitos de software	5.3.4.1
	5.2.3 Incluir medidas de CONTROLE DE RISCO nos requisitos de software		
	5.2.4 Re-AVALIAR ANÁLISE DE RISCO DE PRODUTO PARA A SAÚDE		Nenhuma
	5.2.5 Atualizar requisitos do SISTEMA	5.3.4 Análise dos requisitos de software	a) b)
	5.2.6 Verificar requisitos de software	5.3.4 Análise dos requisitos de software 6.4.2 Verificação	5.3.4.2 6.4.2.3
		5.3.5 Projeto arquitetural do software	
5.3 Projeto ARQUITETURAL do software	5.3.1 Transformar requisitos de software em uma ARQUITETURA	5.3.5 Projeto arquitetural do software	5.3.5.1
	5.3.2 Desenvolver uma		5.3.5.2

Processos da ISO/ IEC 62304		Processos da ISO/ IEC 12207	
Atividade	Tarefa	Atividade	Tarefa
	ARQUITETURA para as interfaces dos ITENS DE SOFTWARE 5.3.3 Especificar requisitos funcionais e de desempenho de item SDPD 5.3.4 Especificar hardware e software do SISTEMA necessário para o item SDPD 5.3.5 Identificar separação necessária para o CONTROLE DE RISCO 5.3.6 Verificar a ARQUITETURA do software		Nenhuma Nenhuma Nenhuma Nenhuma
5.4 Projeto detalhado do software		5.3.5 Projeto arquitetural do software	5.3.5.6
		5.3.6 Projeto detalhado do software 6.4.2 Verificação	
	5.4.1 Refinar a ARQUITETURA nos UNIDADES DE SOFTWARE	5.3.6 Projeto detalhado do software	5.3.6.1
	5.4.2 Desenvolver projeto detalhado para cada UNIDADE DE SOFTWARE		
	5.4.3 Desenvolver projeto detalhado para interfaces		5.3.6.2
	5.4.4 Verificar projeto detalhado	6.4.2 Verificação	5.3.6.7
5.5 Implementação e Verificação da UNIDADE DE SOFTWARE		5.3.6 Projeto detalhado do software 5.3.7 Codificação e ensaio do software 6.4.2 Verificação	
	5.5.1 Implementar cada UNIDADE DE SOFTWARE	5.3.7 Codificação e ensaio do software	5.3.7.1
	5.5.2 Estabelecer PROCESSO DE VERIFICAÇÃO DE UNIDADE DE SOFTWARE	5.3.6 Projeto detalhado do software 5.3.7 Codificação e ensaio do software	5.3.6.5 5.3.7.5
	5.5.3 Critérios de aceitação da UNIDADE DE SOFTWARE	5.3.7 Codificação e ensaio do software	5.3.7.5
	5.5.4 Critérios adicionais de aceitação da UNIDADE DE SOFTWARE	5.3.7 Codificação e ensaio do software 6.4.2 Verificação	5.3.7.5 6.4.2.5
	5.5.5 VERIFICAÇÃO DA UNIDADE DE SOFTWARE	5.3.7 Codificação e ensaio do software	5.3.7.2
5.6 Integração do software e ensaio de integração		5.3.8 Integração do software 5.3.9 Ensaio de qualificação do software 5.3.10 Integração do software 6.4.1 Integração de processo 6.4.2 Verificação	
	5.6.1 Integrar UNIDADES DE SOFTWARE	5.3.8 Integração do software	5.3.8.2
	5.6.2 Verificar integração do	5.3.8 Integração do software	5.3.8.2

Processos da ISO/ IEC 62304		Processos da ISO/ IEC 12207	
Atividade	Tarefa	Atividade	Tarefa
	software		5.3.10.1
	5.6.3 Teste de software integrado	5.3.9 Ensaio de qualificação do software	5.3.9.1
	5.6.4 Conteúdo do ensaio de integração		5.3.9.3
	5.6.5 Verificar procedimentos dos testes de integração	6.4.2 Verificação	6.4.2.2
	5.6.6 Conduzir testes de integração	5.3.8 Integração do software	5.3.8.2
	5.6.7 Integração do conteúdo do registro de teste	5.3.8 Integração do software	5.3.8.2
	5.6.8 Usar PROCESSO de resolução de problema de software	6.4.1 Implementação de processo	6.4.1.6
5.7 Ensaio do SISTEMA DE SOFTWARE		5.3.8 Integração do software 5.3.9 Ensaio de qualificação de software 6.4.1 Implementação de processo 6.4.2 Verificação 6.8.1 Implementação de processo	
	5.7.1 Estabelecer teste para cada requisito de software	5.3.8 Integração do software 5.3.9 Ensaio de qualificação de software	5.3.8.4 5.3.9.1
	5.7.2 Usar PROCESSO de resolução de problema de software	6.4.1 Implementação de processo	6.4.1.6
	5.7.3 Testar novamente antes de alterar	6.8.1 Implementação de processo	6.8.1.1
	5.7.4 Verificar ensaio do SISTEMA DE SOFTWARE	6.4.2 Verificação 5.3.9 Ensaio de qualificação de software	6.4.2.2 5.3.9.3
	5.7.5 Dados do documento para cada teste do conteúdo do registro de teste de SISTEMA DE SOFTWARE	5.3.9 Ensaio de qualificação de software	5.3.9.1
5.8 Liberação de software		5.3.9 Ensaio de qualificação de software 5.4.2 Ensaio operacional 6.2.5 Avaliação de configuração 6.2.6 Gerenciamento de liberação e entrega	
	5.8.1 Garantir que VERIFICAÇÃO do software está completa	5.4.2 Ensaio operacional 6.2.6 Gerenciamento de liberação e entrega	5.4.2.1 5.4.2.2 6.2.6.1
	5.8.2 Documentar ANOMALIAS residuais conhecidas	6.2.5 Avaliação de configuração 5.3.9 Ensaio de qualificação de software	6.2.5.1 5.3.9.3
	5.8.3 Avaliar ANOMALIAS residuais conhecidas		
	5.8.4 Documentar VERSÕES liberadas	6.2.6 Gerenciamento de liberação e entrega	6.2.6.1
	5.8.5 Documentar como o software liberado foi criado		

Processos da ISO/ IEC 62304		Processos da ISO/ IEC 12207	
Atividade	Tarefa	Atividade	Tarefa
	5.8.6 Assegurar que atividade e tarefas estão completas		
	5.8.7 Arquivar o software		
	5.8.8 Assegurar repetibilidade da liberação de software		
6 PROCESSO de manutenção do software		5.5 Processo de manutenção 6.2 Processo de gerenciamento da configuração	
6.1 Estabelecer plano de manutenção do software		5.5.1 Implementação de processo	5.5.1.1
6.2 Análises de problema e modificação		5.5.1 Implementação de processo 5.5.2 Análise de problema e modificação 5.5.3 Implementação de modificação 5.5.5 Migração	
	6.2.1 Registro e avaliação de retorno		
	6.2.1.1 Monitorar o retorno	5.5.1 Implementação de processo	5.5.1.1
	6.2.1.2 Documentar e AVALIAR o retorno		5.5.1.2
	6.2.1.3 Avaliar efeitos do RELATÓRIO DE PROBLEMAS na SEGURANÇA	5.5.2 Análise de problema e modificação	5.5.2.1 5.5.2.2 5.5.2.3 5.5.2.4
	6.2.2 Usar PROCESSO de resolução do problema de software	5.5.1 Implementação de processo	5.5.1.2
	6.2.3 Analisar SOLICITAÇÃO DE MUDANÇA	5.5.2 Análise de problema e modificação	5.5.2.1
	6.2.4 Aprovar SOLICITAÇÃO DE MUDANÇA	5.5.2 Análise de problema e modificação	5.5.2.5
	6.2.5 Comunicar usuários e reguladores	5.5.3 Implementação de modificação 5.5.5 Migração	5.5.3.1 5.5.5.3
6.3 Implementação de modificação		5.5.3 Implementação de modificação 6.2.6 Gerenciamento de liberação e entrega	
	6.3.1 Usar PROCESSO estabelecido para implementar modificação	5.5.3 Implementar modificação	5.5.3.2
	6.3.2 Repetir liberação de SISTEMA DE SOFTWARE	6.2.6 Gerenciamento de liberação e entrega	6.2.6.1
7 PROCESSO de GERENCIAMENTO DE RISCO de software		Emenda 1:2002 – F 3.15 Processo de gerenciamento de risco Processo na 62304 trata de assuntos de risco/perigo que não são tratados na Emenda 1. Há alguns pontos comuns (medidas de risco, etc) mas o foco da análise é bem diferente.	
8 PROCESSO de gerenciamento de configuração de software		5.5 Processo de manutenção 6.2 Processo de gerenciamento de configuração	

Processos da ISO/ IEC 62304		Processos da ISO/ IEC 12207	
Atividade	Tarefa	Atividade	Tarefa
8.1 Identificação da configuração		6.2.2 Identificação da configuração	
	8.1.1 Estabelecer meios para identificar ITENS DE CONFIGURAÇÃO	6.2.2 Identificação da configuração	6.2.2.1
	8.1.2 Identificar SDPD		Nenhuma
	8.1.3 Identificar documentação de configuração do SISTEMA	6.2.2 Identificação da configuração	6.2.2.1
8.2 Controle de mudança		5.5.3 Implementação de modificação 6.2.3 Controle de configuração	
	8.2.1 Aprovar SOLICITAÇÕES DE MUDANÇA	6.2.3 Controle de configuração	6.2.3.1
	8.2.2 Implementar mudanças	5.5.3 Implementação de modificação 6.2.3 Controle de configuração	5.5.3.2 6.2.3.1
	8.2.3 Verificar mudanças	6.2.3 Controle de configuração	6.2.3.1
	8.2.4 Fornecer meios para RASTREABILIDADE da mudança		
8.3 Contabilidade do status da configuração		6.2.4 Contabilidade do status da configuração	6.2.4.1
9 PROCESSO de resolução de problema de software		5.5 Processo de manutenção 6.2 Gerenciamento de configuração 6.8 Processo de resolução de problema	
9.1 Prepara RELATÓRIOS DE PROBLEMA		6.8.1 Implementação de processo 6.8.2 Resolução de problema	6.8.1.1 b) 6.8.2.1
9.2 Investigar o problema		6.8.2 Resolução de problema 6.8.1 Implementação de processo	6.8.2.1 6.8.1.1 b)
9.3 Informar as partes relevantes		6.8.1 Implementação de processo	6.8.1.1 a)
9.4 Usar processo de controle de mudança		6.2.3 Controle de configuração 5.5.3 Implementação de modificação	
9.5 Manter registros		6.8.1 Implementação de processo	6.8.1.1 a)
9.6 Analisar problemas para tendências		6.8.1 Implementação de processo 6.8.2 Resolução de problema	6.8.1.1 b) 6.8.2.1
9.7 Verificar resolução de problema de software		6.8.1 Implementação de processo	6.8.1.1 d)
9.8 Conteúdo da documentação de teste			Todas as tarefas de ensaio na 12207 exigem documentação

C.7 Relacionamento com ISO/IEC 61508

A questão foi levantada se esta norma, sendo preocupada com o projeto de software de SEGURANÇA-crítica, deveria seguir os princípios da IEC 61508. A seguir é explicada a postura desta norma.

A IEC 61508 trata de 3 principais assuntos:

1. Ciclo de vida do GERENCIAMENTO DE RISCO e PROCESSOS do ciclo de vida;
2. Definição de Níveis de Integridade de Segurança;
3. Recomendação de técnicas, ferramentas e métodos de desenvolvimento de software e níveis de independência do pessoal responsável pela realização de diferentes TAREFAS.

O primeiro assunto é coberto nesta norma por uma referência normativa à ISO 14971 (o setor da norma de PRODUTOS PARA A SAÚDE para GERENCIAMENTO DE RISCO). O efeito desta referência é adotar a abordagem da ISO 14971 para GERENCIAMENTO DE RISCO como parte integral do PROCESSO de software para SOFTWARE DE PRODUTO PARA A SAÚDE.

Para o segundo assunto, esta norma tem uma abordagem mais simples que a IEC 61508. Esta classifica software em 4 "Níveis de Integridade de Segurança" definidos em termos de objetivos de confiabilidade. Os objetivos de confiabilidade são identificados depois da ANÁLISE DE RISCO, que quantifica tanto a severidade quanto a probabilidade do DANO causado por uma falha do software.

Esta norma simplifica o segundo assunto não permitindo a consideração de probabilidade de falha do software antes da classificação. A classificação em 3 classes de segurança de software é baseada somente na severidade do DANO causado por uma falha. Depois da classificação, diferentes PROCESSOS são necessários para diferentes classes de segurança de software: a intenção é reduzir ainda mais a probabilidade de falha do software.

O terceiro assunto não é tratado por esta norma. Leitores da norma são encorajados a usar a IEC 61508 como uma fonte para bons métodos, técnicas e ferramentas de software, enquanto reconhecem que outras abordagens, presentes e futuras, podem fornecer resultados igualmente bons. Esta norma não faz qualquer recomendação relativa a independência das pessoas responsáveis por uma ATIVIDADE de software (por exemplo, VERIFICAÇÃO) daquelas responsáveis por outra (por exemplo, projeto). Em particular, esta norma não requer um assessor de segurança independente, uma vez que isto é assunto da ISO 14971.

Anexo D (informativo)

Implementação

D.1 Introdução

Este anexo dá uma visão global de como esta norma pode ser implementada no PROCESSO dos FABRICANTES. Ele também considera que outras normas como a ISO 13485 [7] requerem PROCESSOS adequados e comparáveis.

D.2 Sistema de gestão da qualidade

Para FABRICANTES de PRODUTOS PARA A SAÚDE, incluindo SOFTWARE PARA PRODUTOS PARA A SAÚDE no contexto desta norma, o estabelecimento de um sistema de gestão da qualidade (SGQ) é necessário em 4.1. Esta norma não requer que o SGQ necessariamente tenha que ser certificado.

D.3 AVALIAR os PROCESSOS de gestão da qualidade

É recomendado AVALIAR quão bem os PROCESSOS estabelecidos e documentados do SGQ já cobrem os PROCESSOS do ciclo de vida do software, por meio de auditorias, inspeções, ou análises sob a responsabilidade do FABRICANTE. Eventuais lacunas podem ser acomodadas pela extensão dos PROCESSOS do SGQ, ou podem ser descritas separadamente. Se o FABRICANTE já tem descrições de PROCESSOS disponíveis que regulam o desenvolvimento, VERIFICAÇÃO e validação do software, então estes devem ser também AVALIADOS para determinar quão bem eles estão de acordo com esta norma.

D.4 Integrando requisito desta norma no PROCESSO de gerenciamento de qualidade do FABRICANTE

Esta norma pode ser implementada pela adaptação ou extensão dos PROCESSOS já instalados no SGQ, ou integrando novos PROCESSOS. Esta norma não especifica como isto deve ser feito; o FABRICANTE é livre para fazer da forma mais adequada.

O FABRICANTE é responsável por garantir que os PROCESSOS descritos nesta norma estão devidamente colocados em ação quando o SOFTWARE DE PRODUTO PARA A SAÚDE é desenvolvido por OEM (Original Equipment Manufacturers) ou sub-contratados que não tem seu próprio SGQ documentado.

D.5 Check-list para pequenos FABRICANTES sem SGQ certificado

O FABRICANTE deve determinar a classificação de segurança de software mais alta (A, B ou C) para o software. A Tabela D.1 lista todas as ATIVIDADES descritas nesta norma. A referência a ISO 13485 deve ajudar a definir o lugar no SGQ. Com base na classificação de segurança de software requerida, o FABRICANTE deverá avaliar cada ATIVIDADE requerida contra o PROCESSO existente. Se o requisito já é coberto, uma referência para a descrição do PROCESSO relevante deve ser dada.

Se há discrepância, uma ação é necessária para melhorar o PROCESSO.

A lista pode também ser usada para uma AVALIAÇÃO dos PROCESSOS depois da ação ter sido realizada.

Tabela D.1 – Check-list para pequenos FABRICANTES sem SGQ certificado

ATIVIDADE	Cláusula relacionada da ISO 13485:2003	Coberta por procedimento existente?	Se Sim: Referência	Ações a serem tomadas
5.1 Plano de desenvolvimento de software	7.3.1 Planejamento de projeto e desenvolvimento	Sim/ Não		
5.2 Análise de requisitos de software	7.3.2 Entradas de projeto e desenvolvimento	Sim/ Não		
5.3 Projeto ARQUITETURAL do software		Sim/ Não		
5.4 Projeto detalhado do software		Sim/ Não		
5.5 Implementação e verificação da UNIDADE DE SOFTWARE		Sim/ Não		
5.6 Integração do software e ensaio de integração		Sim/ Não		
5.7 Ensaio do SISTEMA DE	7.3.3 Saídas de projeto e	Sim/ Não		

ATIVIDADE	Cláusula relacionada da ISO 13485:2003	Coberta por procedimento existente?	Se Sim: Referência	Ações a serem tomadas
SOFTWARE	desenvolvimento			
5.8 Liberação do software	7.3.5 Verificação de projeto e desenvolvimento 7.3.6 Validação de projeto e desenvolvimento	Sim/ Não		
6.1 Estabelecer plano de manutenção de software	7.3.7 Controle de mudança de projeto e desenvolvimento	Sim/ Não		
6.2 Análise de problema e modificação		Sim/ Não		
6.3 Implementação de modificação	7.3.5 Verificação de projeto e desenvolvimento 7.3.6 Validação de projeto e desenvolvimento	Sim/ Não		
7.1 Análise de software contribuindo para situações de perigo		Sim/ Não		
7.2 Medidas de CONTROLE DE RISCO		Sim/ Não		
7.3 VERIFICAÇÃO de medidas DE CONTROLE DE RISCO		Sim/ Não		
7.4 GERENCIAMENTO DE RISCO de mudanças de software		Sim/ Não		
8.1 Identificação de configuração	7.5.3 Identificação e rastreabilidade	Sim/ Não		
8.2 Controle de mudança	7.5.3 Identificação e rastreabilidade	Sim/ Não		
8.3 Contabilidade do status de configuração		Sim/ Não		
9 PROCESSO de resolução de problema de software		Sim/ Não		

Bibliografia

- [1] IEC 60601-1:2005, Medical electrical equipment – Part 1: General requirements for basic safety and essential performance
- [2] IEC 60601-1-4:1996, Medical electrical equipment – Part 1: General requirements for safety – 4.Collateral standard: Programmable electrical medical systems
Amendment 1 (1999)
- [3] IEC 61508-3, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements
- [4] IEC 61010-1:2001, Safety requirements for electrical equipment for measurement, control, and laboratory use – Part 1: General requirements
- [5] ISO 9000:2005, Quality management systems – Fundamentals and vocabulary
- [6] ISO 9001:2000, Quality management systems – Requirements

[7] ISO 13485:2003, Medical devices – Quality management systems – Requirements for regulatory purposes

[8] ISO/IEC 9126-1:2001, Software engineering — Product quality — Part 1: Quality model

[9] ISO/IEC 12207:1995, Information technology – Software life cycle processes
Amendment 1 (2002)
Amendment 2 (2004)

[10] ISO/IEC 14764:1999, Information technology – Software maintenance

[11] ISO/IEC 90003:2004, Software engineering – Guidelines for the application of ISO 9001:2000 to computer software

[12] ISO/IEC Guide 51:1999, Safety aspects – Guidelines for their inclusion in standards

[13] IEEE 610.12:1990, IEEE standard glossary of software engineering terminology

[14] IEEE 1044:1993, IEEE standard classification for software anomalies

[15] IEC 60601-1-6, Medical electrical equipment - Part 1-6: General requirements for safety - Collateral standard: Usability