Uma Análise das Características de Especificação de Requisitos de Software em Normas de Ambientes Regulados

Johnny Cardoso Marques[1][0000-0002-1551-435X]

Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brasil
johnny@ita.br

Abstract. Requirements Engineering is the set of activities involved in managing, surveying, documenting, and maintaining a set of requirements for a product. Engineering involves the use of systematic repeatability techniques to ensure that the Software Requirements are complete, consistent, valid and verifiable. Software Requirements Specification is an organized process oriented towards defining, documenting and maintaining Software Requirements during the development life cycle. Many authors suggest that requirements should always focus their statements on what the software product needs to address without specifying how to implement them. However, the detail of the Software Requirements is influenced by several factors such as: organizational thoughts; existing specification standards; and regulatory needs. This work fits exactly with regulatory needs, where the characteristics of Software Requirements Specification in Regulated Environments such as aeronautics, rail and medical. This article briefly presents the three standards used in these regulated environments (RTCA DO-178C, IEC 62279 and IEC 62304) and analyzes their similarities from a Requirements Specification perspective.
Keywords: software, certification, requirements, standards.

1 Introduction

Normalmente, um Software Crítico é desenvolvido em ambientes regulamentados por normas e padrões. Exemplos são encontrados em domínios como: aviação, automotivo, médico, ferroviário, espacial e nuclear. Neste trabalho, o software destes domínios é definido como Software em Ambientes Regulados (SAR).

O Software em Ambientes Regulados pode integrar, por exemplo, dispositivos médicos, equipamentos de energia nuclear, satélites, aeronaves, veículos e outros produtos críticos em segurança. Uma característica bastante presente nas normas e padrões desses domínios é a Especificação de Requisitos (ER).

A literatura tem abordado os diversos problemas na Especificação de Requisitos, que podem envolver requisitos incompletos, incorretos, ambíguos, conflitantes ou inconsistentes.

O desenvolvimento de Software em Ambientes Regulados não envolve uma área completamente heterogênea, mas consiste em muitas culturas diferentes de desenvolvimento, que possuem características comuns que permitem estabelecer uma correlação entre elas, tais como: a) Tipo de produto de Software; b) O papel do Software no sistema; c) O tamanho do sistema; e d) O nível de risco do sistema.

As normas publicadas por comitês, entidades técnicas internacionais ou agências reguladoras influenciam o desenvolvimento de SAR, por meio de diretrizes para processos e produtos de Software [1], considerando o risco citado anteriormente.

Este trabalho possui o seguinte objetivo: apresentar as normas de software (RTCA DO-178C [2], IEC 62279 [3] e IEC 62304 [4]), abordando suas similaridades no escopo de Especificação de Requisitos, incluindo um critério para seleção destas três normas.

Além desta seção 1, o trabalho inclui mais outras 3 seções. A seção 2 descreve, suscintamente as três normas que fazem parte do escopo deste trabalho: RTCA DO-178C, IEC 62279 e IEC 62304. uma das principais normas para sistemas militares aeronáuticos. A seção 3 apresenta as características de similaridade e possíveis diferenças. A seção 4 apresenta as conclusões.

2 Normas de Software para Ambientes Regulados

Os ambientes regulados são aqueles que trazem impactos à sociedade em geral e, por isto, precisam de padrões que legislem sobre os produtos e serviços entregues por empresas. Existe a expectativa da sociedade de receber serviços e produtos seguros e confiáveis. Em todos os diversos ambientes regulados, como: aeronáutico, ferroviário, automotivo, nuclear, médico, entre outros, existem padrões que abrangem diversas tecnologias, incluindo o desenvolvimento de Software. Como consequência direta, existem normas que regulamentam as necessidades exigidas para demonstrar que um produto de Software é seguro e confiável para se operar neste tipo de ambiente.

Munch et. al [1] consideram que o número de organizações que precisam verificar aderência aos padrões regulatórios vem aumentando. Muitos destes regulamentos, que se apresentam como normas, requerem a presença de processos de desenvolvimento de Software explícitos. Neste aspecto, as atividades realizadas devem apresentar repetibilidade e rastreabilidade dentro do processo de desenvolvimento de Software proposto.

Essas normas possuem objetivos ou atividades que precisam ser satisfeitas para que o produto de Software seja aprovado para operação em seu ambiente de utilização. Agências reguladoras, ou outras entidades, normalmente, exigem a aderência às normas e padrões estabelecidos.

2.1 Seleção das Normas de Software para Ambientes Regulados

Para análise e seleção das normas de SAR utilizadas neste trabalho de pesquisa, identificou-se 7 atributos normalmente presentes em Especificação de Requisitos, visando estabelecer um critério de comparação e diferenciação das normas de SAR:

– At1 - Rastreabilidade entre Requisitos de Software e Requisitos de Sistemas;

- At2 - Rastreabilidade entre Casos de Testes e Requisitos de Software;
- At3 - Descrição de Requisitos de Software indicando com o desempenho esperado;
- At4 - Compatibilidade de Requisitos de Software com o ambiente computacional requerido;
- At5 - Descrição dos Requisitos de Software em termos de interfaces com outros Softwares e/ou sistemas;
- At6 - Consistência entre Requisitos de Software; e
- At7 - Alocação de Requisitos de Software na Arquitetura de Software.

A escolha dos sete atributos listados anteriormente encontra-se justificada em [5]. Dentre os diversos domínios Safety-Critical, escolheram-se oito normas, devido às suas representatividades em seus domínios de aplicação, para análise de satisfação dos 7 atributos: RTCA DO-178C [2], IEC 62279 [3], IEC/ISO 62304[4], RTCA DO-278A [6], ISSO 26262-6 [7], ECSS-E-ST-40C [8] e IAEA SSG-39 [9].

A **Tabela 1** apresenta os resultados desta avaliação de análise de satisfação dos 7 atributos descritos nesta seção. Portanto, as 8 normas selecionadas foram avaliadas considerando-se os 7 atributos definidos pelo autor desta pesquisa. Cada atributo foi avaliado com um grau 0, 1 ou 2. O grau 0 indica que a norma não possui ou menciona o referido atributo. O grau 1 indica que ela menciona o atributo, porém não o considera mandatório necessitando ser avaliado para cumprimento com algum objetivo ou atividade da norma. No grau 1, o atributo pode ser mencionado como um exemplo, dentro do texto da norma, mas não representa obrigatoriedade. Por fim, o grau 2 indica que a norma requer uma atividade ou objetivo explícito em relação ao atributo. Assim, o atributo passa a ser considerado uma obrigatoriedade dentro da norma avaliada.

Tabela 1. Avaliação das Normas de Software em Ambientes Regulados

Norma	Domínio	At 1	At 2	At 3	At 4	At 5	At 6	At 7	Total
RTCA DO-178C [2]	Aeronáutico	2	2	1	2	1	2	2	12
IEC 62279 [3]	Ferroviário	2	2	2	2	1	2	2	13
IEC 62304 [4]	Médico	2	2	2	1	2	1	2	12
RTCA DO-278A [6]	Tráfego Aéreo	2	2	1	2	1	2	2	12
ISSO 26262-6 [7]	Automotivo	1	2	2	1	1	2	1	10
ECSS-E-ST-40C [8]	Satélites	1	2	1	1	1	0	1	7
IAEA SSG-39 [9]	Nuclear	0	1	1	1	1	1	1	6

Pela similaridade entre a RTCA DO-178C e a DO-278A, as mesmas apresentaram avaliações idênticas quanto aos atributos. Ambas foram definidas pelo mesmo comitê SC-205, onde a RTCA DO-178C tem enfoque no desenvolvimento de sistemas aeronáuticos embarcados e a RTCA DO-278A, enfoca os sistemas de solo de apoio aeronáutico, como os de comunicação, navegação e vigilância para o controle de trafego aéreo.

Baseado nos resultados consolidados na última coluna da **Tabela 1**, que reflete o somatório da pontuação obtida para cada atributo definido, a RTCA DO-178C, a IEC 62279 e a IEC 62304 foram as normas que apresentaram mais aderência aos atributos presentes em Especificação de Requisitos.

2.2 RTCA DO-178C

O início da década de 80 se caracterizou pelo rápido aumento no uso do Software em sistemas e equipamentos de aeronaves e motores. Esta tendência resultou na necessidade de uma orientação para o desenvolvimento de Software aceita pela indústria para satisfazer os requisitos de aeronavegabilidade. A RTCA DO-178C existe para satisfazer esta necessidade. Este documento oferece, à comunidade aeronáutica, diretrizes sobre os processos de desenvolvimento de Software que os sistemas e equipamentos a bordo precisarão apresentar.

A RTCA DO-178C [2] é uma evolução da DO-178 (1982), DO-178A (1985) e DO-178B (1992). Ao longo dos anos, a Agência Nacional de Aviação Civil (ANAC), a *Federal Aviation Administration* (FAA) e a *European Aviation Safety Agency* (EASA) reconhecem as revisões da DO-178 como um meio aceitável para desenvolvimento de Software aeronáutico. Atualmente, a FAA AC 20-115D [10] reconhece a RTCA DO-178C como um método aceitável para aprovação de sistemas e/ou equipamentos com uso de Software.

Cada um dos seus 5 níveis de Software desdobra-se em objetivos que devem ser satisfeitos para viabilizar sua aprovação, como parte do processo de certificação de uma aeronave. Dentre os cinco níveis de Software existentes (A, B, C, D e E), o nível A possui maior rigor e exige o cumprimento de todos os objetivos da norma. Já o nível E refere-se aos produtos de Softwares cujo mal funcionamento não acarreta em perda das margens de segurança.

A ARP 4754A classifica cada falha de sistema com uma criticalidade associada em cinco categorias [11]. Assim, associa-se a classificação da condição de falha com níveis definidos na RTCA DO-178C, conforme a **Tabela 2** e se torna necessária a satisfação de um conjunto de objetivos associados.

Tabela 2. Condições de falha de sistemas pela RTCA DO-178C e total de objetivos associados

Condição de Falha	Nível de Software Requerido	Objetivos Associados
Catastrófica	A	71
Perigosa	B	69
Maior	C	62
Menor	D	24
Sem Impacto	E	0

Os 71 objetivos da DO-178C encontram-se organizados em 10 tabelas específicas de objetivos dentro da norma. As tabelas agrupam objetivos de planejamento, desenvolvimento, qualidade, entre outros. A **Fig. 1** apresenta uma visão geral da organização destas tabelas.

A RTCA DO-178C possui um número expressivo de objetivos associados ao desenvolvimento de Requisitos de Software, utilizando como entrada, os Requisitos de Sistemas que serão implementados por Software.

Existem dois níveis de Requisitos de Software na RTCA DO-178C. Os Requisitos de Software de Alto Nível, em inglês, *Software High-Level Requirements* (SW-HLR), geralmente, representam “o quê” deve ser concebido. Os SW-HLR incluem características funcionais, de performance, interface e requisitos relacionados com segurança.

Os Requisitos de Software de Baixo Nível, em inglês, *Software Low-Level Requirements* (SW-LLR), geralmente, representam “o como”', fornecendo detalhes sobre a implementação de Software em código [12]. Os SW-LLR incluem as características necessárias para o desenvolvimento de código fonte, como características de acoplamento de dados e de controle.

A justificativa para a existência de dois níveis de Requisitos de Software consiste na necessidade em fornecer rastreabilidade e refinamento, a partir dos requisitos do sistema, até o nível de implementação em código-fonte.

Os objetivos do *Software Requirements Process* da RTCA DO-178C preveem que os Requisitos de Software de Alto Nível (SW-HLR) sejam definidos e que os SW-HLR, considerados como derivados, não possuam rastreabilidade para os Requisitos de Sistemas e sejam realimentados para o processo de desenvolvimento de sistemas, para análise.

A RTCA DO-178C exige a definição de um *Software Requirements Standards* (SRSt) que deve definir os métodos, notações, regras e ferramentas a serem utilizados para desenvolver os SW-HLR, que devem ser aderentes ao SRSt. Dentre as atividades associadas ao desenvolvimento dos Requisitos de Software de Alto Nível, destacam-se:

- Cada requisito de sistema alocado para Software deve ser especificado em Requisitos de Software de Alto Nível (RTCA DO-178C Seção 5.1.2(c));
- Cada Requisito de Software de Alto Nível deve ser aderente ao *Software Requirements Standards* (SRSt), além de ser verificável e consistente (RTCA DO-178C Seção 5.1.2(e));
- Cada Requisito de Software de Alto Nível deve ser estabelecido em termos quantitativos com tolerâncias, quando aplicável (RTCA DO-178C Seção 5.1.2(f)); e
- Cada Requisito de Software de Alto Nível derivado deve ter uma razão justificável para sua existência (RTCA DO-178C Seção 5.1.2(h)).

A revisão de Requisitos de Software de Alto Nível deve garantir que estes são:

- Rastreáveis e cumpram com os Requisitos de Sistemas (RTCA DO-178C Seção 6.3.1(a) e (f));
- Precisos e consistentes (RTCA DO-178C Seção 6.3.1(b));
- Compatíveis com o hardware (RTCA DO-178C Seção 6.3.1(c));
- Verificáveis, ou seja, possível de fornecer evidências de satisfação (RTCA DO-178C Seção 6.3.1(d)); e
- Aderentes ao *Software Requirements Standards* (SRSt) (RTCA DO-178C Seção 6.3.1(e)).

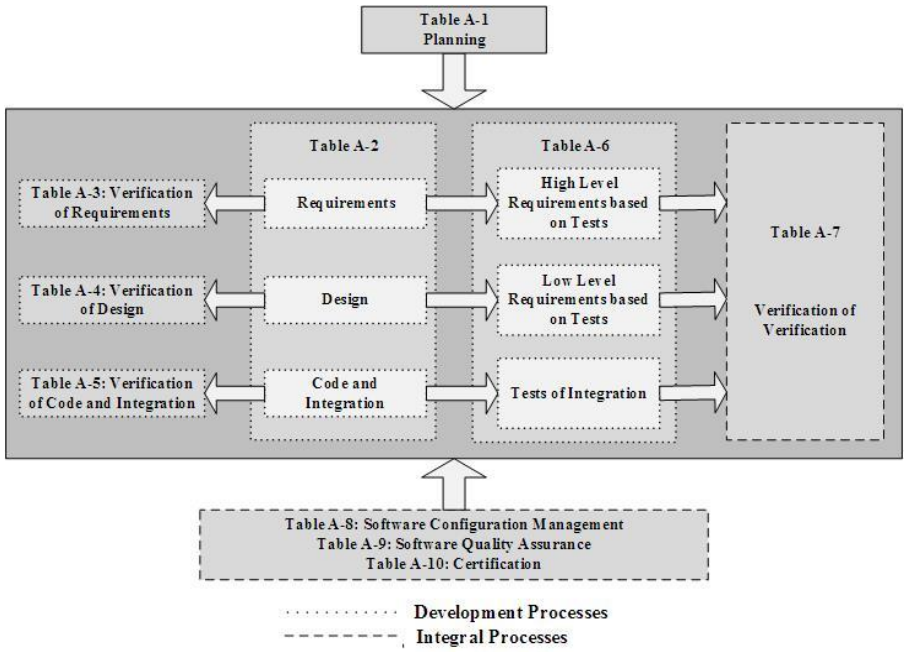

Fig. 1. Organização de tabelas da RTCA DO-178C [13]

A arquitetura de Software é desenvolvida, a partir dos Requisitos de Software de Alto Nível (DO-178C Seção 5.2.1 a). Adicionalmente, o fabricante deve desenvolver e documentar a arquitetura, incluindo as interfaces entre os itens e os componentes externos (DO-178C Seção 5.2.2d).

Apesar da aderência à RTCA DO-178C ser tipicamente avaliada pelos objetivos existentes nas 10 tabelas, cada um destes rastreia para seções no corpo da norma que apresentam o detalhamento completo de cada objetivo. Como esta característica é exclusiva desta norma, para fins de harmonização, o autor deste trabalho preferiu fazer a análise de aderência às seções da norma.

2.3 IEC 62279

O software é amplamente utilizado no sistema ferroviário, como sistema de propulsão de trem, sistema de freio, sistema de controle de trem, sistema de detecção de trens e unidade de exibição do motorista (*display*) [14].

Ao desenvolver softwares no setor ferroviário, a norma IEC 62279 é a mais comum a ser seguidas em termos de RAMS (Confiabilidade, Disponibilidade, Manutenção e Segurança) [14].

A IEC 62279 é uma norma que regula o desenvolvimento, implantação e manutenção de sistemas de segurança software destinado a aplicações ferroviárias. Contém requisitos da organização em desenvolvimento (funções e competências), ciclo de vida (fases, documentação e métodos) e garantia de software (teste, verificação, validação e garantia de qualidade e avaliação) [15].

A IEC 62279 requer que fabricantes atribuam um *Safety Integration Level* (SIL) para os sistemas com Software. Esta classificação é baseada no potencial perigo que pode resultar em um prejuízo para o usuário, em caso de um comportamento anormal do sistema. O conceito de SIL envolve uma classe de requisitos de segurança para funções, sistemas, subsistemas ou componentes. Um SIL consiste em dois fatores:

- Um intervalo de valores para uma Taxa de Risco Tolerável (TRT);
- Medidas a serem implementadas no projeto durante o processo de design.

Um SIL pode ser atribuído a qualquer função ou sistema relevante de segurança ou subsistema ou componente que pode ser categorizado em cinco níveis distintos de 0 até 4. O SIL 4 possui o maior rigor e exige o cumprimento de todas as atividades obrigatórias. O SIL 2 é equivalente ao SIL 1, assim como o SIL 4 é equivalente ao SIL 3, no entanto, o SIL 2 e 4 possuem atividades que necessitam de independência em sua execução. O total de atividades de cada SIL é apresentado na **Tabela 3**.

A IEC 62279 possui 11 fases: Planejamento, Desenvolvimento do Sistemas, Requisitos, Projeto de Arquitetura, Projeto de Componentes, Implementação, Testes, Integração, Validação, Manutenção e Avaliação. A **Fig. 2** apresenta uma visão geral do ciclo de vida da IEC 62279.

A IEC 62279 possui um grande conjunto de atividades associadas ao desenvolvimento de Requisitos de Software descritas ao longo da norma. A fase Especificação de Requisitos de Software deve expressar as propriedades do software a ser desenvolvido, incluindo a sua funcionalidade, robustez, manutenibilidade, eficiência, usabilidade e portabilidade (IEC 62279 Seção 7.2.4.2). Adicionalmente, a Especificação de Requisitos de Software deve ser estruturada de forma a garantir que esta é completa, clara, precisa, verificável, testável e rastreável para as entradas utilizadas na sua definição (IEC 62279 Seção 7.2.4.4).

Tabela 3. *Safety Integration Levels* pela IEC 62279 e total de atividades associadas

SIL	Taxa de Risco Tolerável (TRT)	Total de Atividades Obrigatórias	Total de Atividades Altamente Recomendadas	Total de Atividades Recomendadas
0	Não aplicável	4	16	53
1	10^{-5}=<TRT<10^{-6}	5	54	53
2	10^{-6}=<TRT<10^{-7}	5	54	53
3	10^{-7}=<TRT<10^{-8}	19	84	69
4	10^{-8}=<TRT<10^{-9}	19	84	69

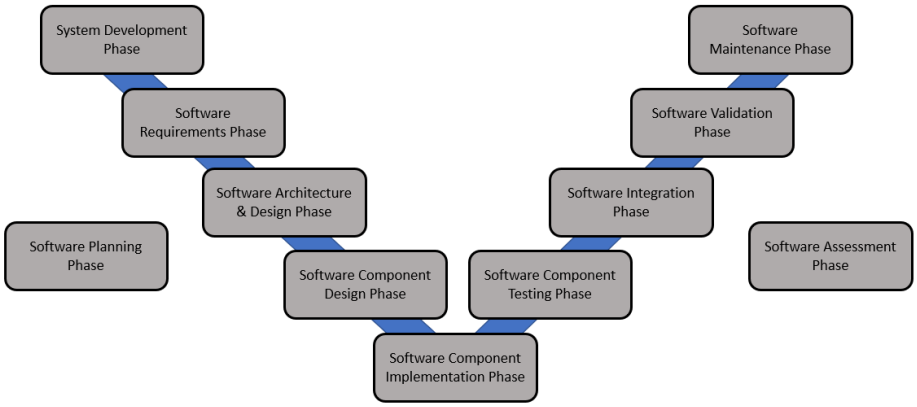

Fig. 2. Visão geral das fases da IEC62279 - adaptado de [3]

2.4 IEC 62304

De acordo com Magnusson [16], todos os dispositivos médicos precisam satisfazer a regulamentação para garantir a segurança do usuário e do paciente. Com o aumento do uso de Software em dispositivos médicos, entidades como o *Food and Drug Administration* (FDA) dos Estados Unidos identificou a necessidade de uma regulamentação específica para Software.

Em 2006, um novo padrão internacional foi lançado para Software biomédico desenvolvido por um grupo de trabalho conjunto da *International Electrotechnical Commission* (IEC). com isto, o uso da IEC 62304 [4] se tornou totalmente harmonizado nos Estados Unidos e Europa.

A IEC 62304 descreve 5 processes: *Software Development Process, Software Maintenance Process, Software Risk Management Process, Software Configuration Management Process, Software Problem Resolution Process*, conforme apresentado na **Fig. 3**.

A IEC 62304 requer que fabricantes atribuam uma classe de segurança para os sistemas com Software. Esta classificação é baseada no potencial perigo que pode resultar em um prejuízo para o usuário ou o paciente, em caso de um comportamento anormal do sistema. O Software pode ser categorizado em três classes. A classe C possui o maior rigor e exige o cumprimento de todas as atividades associadas. As classes B e A possuem um menor número de atividades requeridas, conforme apresentado na **Tabela 4**.

A IEC 62304 possui um grande conjunto de atividades associadas, descritas ao longo da norma, referentes ao desenvolvimento de Requisitos de Software. Sobre desenvolvimento de Requisitos de Software, não existe diferença entre as classes A, B e C apresentadas na **Tabela 4**.

Dentre as atividades associadas ao desenvolvimento de requisitos, destacam-se:

– Para cada Software de um dispositivo médico, o fabricante deve definir e documentar os Requisitos de Software, a partir dos Requisitos de Sistemas (IEC 62304, Seção 5.2.1); e

– Conforme apropriado ao Software de um dispositivo médico, o fabricante deve incluir nos Requisitos de Software:

a. Requisitos funcionais, incluindo performance, características físicas e ambiente computacional (IEC 62304, Seção 5.2.2(a));
b. Entradas e saídas, incluindo características dos dados, faixa de valores, limites e valores típicos (IEC 62304, Seção 5.2.2(b)); e
c. Interfaces entre Software e sistema, incluindo considerações sobre compatibilidade (IEC 62304, Seção 5.2.2(c)).

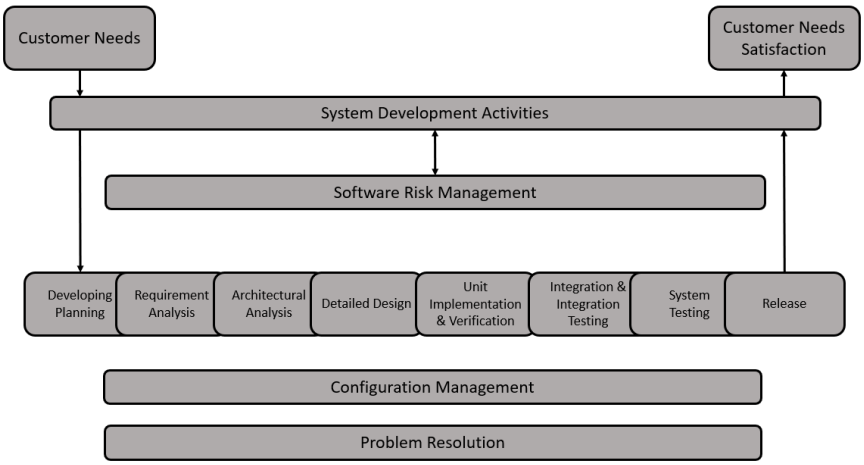

Fig. 3. Visão geral dos processos da IEC 62304 - adaptado de [4]

Tabela 4. Classes de software pela IEC 62304 e total de atividades associadas

Classe	Impacto em Usuário e Paciente	Atividades Associadas
A	Não é possível ter prejuízo ou danos à saúde	44
B	É possível ter lesão não grave	87
C	É possível ter morte ou lesão grave	92

Durante a atividade de revisão, para cada requisito de Software definido, o fabricante deve analisar, garantir e documentar que os Requisitos de Software:

– Implementam os Requisitos de Sistemas definidos (IEC 62304, Seção 5.2.6(a));
– Não possuam conflitos entre si (IEC 62304, Seção 5.2.6(b));
– Apresentam-se corretamente expressos, evitando ambiguidade (IEC 62304, Seção 5.2.6(c));
– Possuam escrita, de forma que permitem o estabelecimento de um critério de teste e determinar se este foi atingido (IEC 62304, Seção 5.2.6(d));
– Encontram-se unicamente identificados (IEC 62304, Seção 5.2.6(e)); e

– Apresentam rastreabilidade para os requisitos de sistema (IEC 62304, Seção 5.2.6(f)).

O fabricante deve transformar os requisitos do Software de um dispositivo médico em uma arquitetura documentada que descreve a estrutura do Software, identificando os módulos e componentes de Software presentes (IEC 62304, Seção 5.3.2). Adicionalmente, o fabricante deve desenvolver e documentar a arquitetura, incluindo as interfaces entre os módulos e os componentes externos (IEC 62304, Seção 5.3.3).

3 Análise das Características das Normas de Software em Engenharia de Requisitos

Tipicamente um bom conjunto de requisitos precisa de características mínimas que permitam que ele seja consistente, não ambíguo, consistente, verificável e rastreável [17][18].

3.1 Consistência Interna e Externa

Um conjunto de Requisitos de Software está consistente se e somente se, todas as exigências expressas em algum dos requisitos não conflita com os demais. Existem dois tipos de consistência previstos na literatura [19][20][21][22]:

– A Consistência Externa refere-se a garantia de que os Requisitos de Software são consistentes com as entradas necessárias para sua formulação; e
– A Consistência Interna refere-se a garantia de que os Requisitos de Software são consistentes entre si, garantindo que não são contraditórios uns com os outros.

A RTCA DO-178C fala explicitamente em consistência interna na seção 6.3.1b (Objetivo A3-2) durante o processo de revisão de Requisitos de Software, onde os requisitos devem ser avaliados para garantir que não estão em conflito entre si. A IEC 62304 trata da consistência na seção 5.2.6b, onde é citado que os Requisitos de Software devem ser verificados para garantir que eles não se contradizem uns com ou outros. A IEC 62279 reforça que a consistência interna deve ser mantida na revisão do conjunto de Requisitos de Software na ocasião de sua revisão formal, conforme citado explicitamente na seção 7.2.4.22e.

Sobre a consistência externa, a RTCA DO-178C fala em satisfazer os Requisitos de Sistemas com os Requisitos de Software na seção 6.3.1a (Objetivo A3-1). A IEC 62304 já afirma que os Requisitos de Software devem implementam os Requisitos de Sistemas na seção IEC 62304, Seção 5.2.6a. Por fim a IEC 62279 também afirma, na seção 7.2.4.22a, que os Requisitos de Software devem cumprir com o conjunto de Requisitos de Sistemas. De certa maneira, a consistência externa está entendida nestas três normas, embora não explicitamente definida, já que ao desdobrar os Requisitos de Sistema, os possíveis conflitos precisam ser avaliados neste desdobramento.

3.2 Não Ambiguidade

Um conjunto de Requisitos de Software é não ambíguo se todas as exigências expressas nele têm apenas uma única interpretação. Como mínimo, isto requer que cada característica do produto de software seja descrita usando termos simples e únicos. Nas situações em que um termo ao ser utilizado em determinado contexto possa adquirir múltiplos significados, esse termo deve ser incluído num glossário onde o seu significado é tornado mais específico [19][20].

A RTCA DO-178C trata a questão ambiguidade como parte da precisão e consistência, explicitando na seção 6.3.1b que o objetivo A3-1 deve garantir que um Requisito de Software não é ambíguo. Já a IEC 62304 prevê na seção 5.2.6c que os Requisitos de Software devem ser expressos em termos que evitem ambiguidade. Por fim, a IEC 62279 prevê na seção 7.2.4.4 que os Requisitos de Software devem ser estabelecidos de maneira inequívoca.

3.3 Verificabilidade

Um conjunto de Requisitos de Software é considerado verificável e testável se for possível checar que os requisitos funcionais e atributos de qualidade foram adequadamente implementados no design e código [17].

De acordo com *o International Standardization Organization* [18], um requisito é verificável, se e só se, existir um processo finito e de custo aceitável através do qual uma pessoa ou uma máquina pode verificar que o produto de software cumpre essa exigência. Em geral um requisito ambíguo não é verificável.

Na RTCA DO-178C, existe a necessidade de comprovar que, tanto os Requisitos de Software de Alto Nível (SW-HLR), quanto Requisitos de Software de Baixo Nível (SW-LLR), são verificáveis. Isso é definido na RTCA DO-178C Seção 6.3.1(d), para os SW-HLR, e na RTCA DO-178C Seção 6.3.2 (d), para os SW-LLR.

Na IEC 62304, também existe a necessidade que assegurar a verificabilidade dos requisitos de software. A IEC 62304 Seção B.5.2 descreve que estabelecer requisitos verificáveis é essencial para determinar o que deve ser construído, e para determinar se o software do dispositivo médico apresenta um comportamento aceitável e que está pronto para uso. Para demonstrar que os requisitos foram implementados conforme desejado, cada requisito deve ser declarado de tal forma que critérios objetivos possam ser estabelecidos para determinar se ele foi implementado corretamente.

Na IEC 62279, a verificabilidade dos requisitos também é mandatória. No entanto, ela simplesmente informa na seção 7.2.4.4 (a) que a especificação de software deve ser expressa de maneira verificável, sem fornecer maiores detalhes.

3.4 Rastreabilidade

De acordo com Lauesen [21], o rastreio de requisitos é necessário para comparar os requisitos com outras informações correlatas. A rastreabilidade de requisitos, em particular, é definida por Gotel & Finkelstein [22] como “a capacidade de descrever desde

suas origens, desenvolvimento e especificação, até sua implantação e uso subsequentes e por períodos de refinamento contínuo e iteração em qualquer fase".

Na RTCA DO-178C, a rastreabilidade de requisitos acontece em vários sentidos. Os Requisitos de Software de Alto Nível (SW-HLR) devem apresentar rastreabilidade bidirecional para os Requisitos de Sistemas, Requisitos de Software de Baixo Nível (SW-LLR) e casos de testes. Já os Requisitos de Software de Baixo Nível (SW-LLR), que fazem parte do Design, precisam apresentar rastreabilidade bidirecional para os de Alto Nível (SW-HLR), Código Fonte e Testes.

Na IEC 62304, a rastreabilidade de requisitos é menos extensa que na RTCA DO-178C. Os Requisitos de Software precisam apresentar rastreabilidade bidirecional para os Requisitos de Sistemas e Testes. Embora não seja obrigatória, a IEC 62304 prevê que a rastreabilidade entre a Arquitetura, que faz parte do Design, e os Requisitos de Software, pode ser útil para a verificação.

Na IEC 62279, a rastreabilidade bidirecional de requisitos acontece dos Requisitos de Software, para os Requisitos de Sistema, o Design e os Testes. A **Fig. 4** apresenta uma síntese envolvendo a rastreabilidade entre Requisitos de Software e os demais artefatos para as três normas de software em análise.

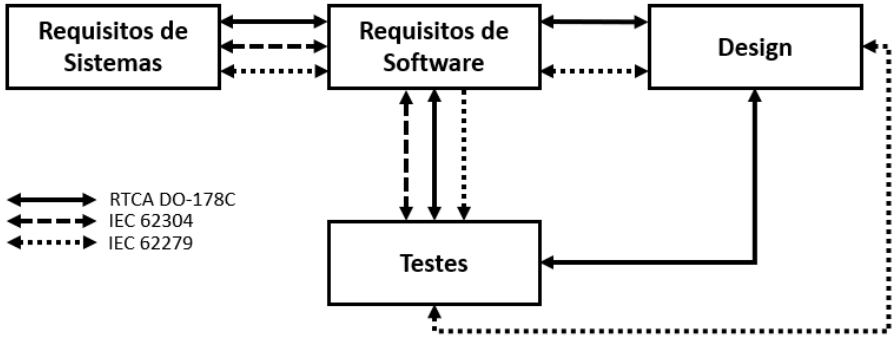

Fig. 4. Síntese das rastreabilidades com Requisitos de Software

4 Conclusão

Conforme apresentado na seção 1, o objetivo deste trabalho foi o de apresentar as normas de software (RTCA DO-178C [2], IEC 62279 [3] e IEC 62304 [4]), abordando suas similaridades no escopo de Especificação de Requisitos. Este trabalho representa um esforço inicial de pesquisa, conduzido no Instituto Tecnológico de Aeronáutica, visando a possibilidade de propor um *framework* universal que possa simultaneamente produzir aderências às três normas que fazem parte do escopo deste trabalho DO-178C [2], IEC 62279 [3] e IEC 62304 [4].

O *framework* a ser concebido deve não só capturar as necessidades de Especificação de Requisitos, mas também as demais característica tipicamente observadas nos processos de desenvolvimento de Software em Ambientes Regulados (SAR), como a parte de Arquitetura, Design, Código, Testes, dentre outros aspectos.

Observa-se, pela análise concebida na seção 3, que no cenário de Especificação de Requisitos, as normas utilizadas apresentam fortes características de similaridade. Foram analisadas as seguintes características: consistência, não ambiguidade, verificabilidade e rastreabilidade. A **Tabela 5** apresenta uma síntese com as similaridades e diferenças entre as três normas avaliadas, sob o aspecto de Especificação de Requisitos de Software.

Tabela 5. Similaridades e Diferenças entre as Três Normas

Similaridades	Diferenças
1. Todas as três normas tratam sobre a característica da não ambiguidade de maneira equivalente; 2. A Rastreabilidade entre Requisitos de Sistemas e Requisitos de Software aparece de maneira equivalente nas três normas analisadas; 3. A Rastreabilidade entre Requisitos de Software e Testes aparece de maneira equivalente nas três normas analisadas; 4. A Rastreabilidade entre Requisitos de Software e o Design aparece, de maneira equivalente, apenas na RTCA DO-178C e IEC 62279; e 5. A Rastreabilidade entre o Design e Testes aparece de maneira equivalente apenas na RTCA DO-178C e IEC 62279.	1. A RTCA DO-178C e a IEC 62034 não utilizam a nomenclatura consistência interna e externa. Já a IEC 62279 fala explicitamente em consistência interna. 2. A RTCA DO-178C exige a comprovação que, tanto os Requisitos de Software de Alto Nível (SW-HLR), quanto Requisitos de Software de Baixo Nível (SW-LLR), são verificáveis. Na IEC 62304, também existe a necessidade que assegurar a verificabilidade dos requisitos de software, mas nesta não são especificados dois níveis de requisitos de software, apenas um. Por fim na IEC 62279, a verificabilidade dos requisitos também é mandatória. No entanto, esta é suscinta sem fornecer maiores detalhes. 3. A IEC 62304 não prevê a rastreabilidade entre Requisitos de Software e Design e também entre Design e Testes, diferentemente da RTCA DO-178C e IEC 62279.

A análise das três primeiras características (Consistência, Não ambiguidade e Verificabilidade) aponta que as normas exigem que tais características sejam verificadas, embora não representem algum diferencial exigido pelas normas que fujam ao que se espera também em especificação de requisitos para ambientes não regulados. No entanto, o fato destas três características serem previstas nestas normas, torna estas como obrigatórias, sob o ponto de vista de regulamentação, impedindo as empresas de cada a domínio realizem suas especificações de requisitos sem considerá-las. Já na característica Rastreabilidade há uma indicação um pouco mais concreta de como cada norma se diferencia das demais, conforme apresentado na seção 4.4.

Referências

1. Munch, J., Armbrunt, O., Kowalczyk, M., Soto, M.: Software Process Definition and Management. 1st edn. Springer-Verlag, Germany 2012.
2. Radio Technical Commission for Aeronautics: DO-178C - Software Considerations in Airborne Systems and Equipment Certification. Estados Unidos (2011).

3. International Electrotechnical Commission: ISO 62279:2015 Railway applications – Communications, signaling and processing systems – Software for railway control and protection systems. Estados Unidos (2015).
4. International Electrotechnical Commission: ISO 62304:2015 Medical device software - Software life-cycle processes. Estados Unidos (2015).
5. Marques, J., Cunha, A.M.: Especificação de Requisitos de Software: Um Modelo Ágil para Ambientes Regulados. Brasil (2017).
6. Radio Technical Commission for Aeronautics: DO-278A - Software Integrity Assurance Considerations for Communication, Navigation, Surveillance and Air Traffic Management (CNS/ATM) Systems. Estados Unidos (2011).
7. International Standardization Organization: ISO26262-6 Road vehicles – Functional safety – Part 6: Product development at the software level. (2011).
8. European Cooperation for Safety Standardization, ECSS-E-ST-40C Space Engineering – Software. Norway (2009).
9. International Atomic Energy Agency: Specific Safety Guidance SSG-39: Design of Instrumentation and Control Systems for Nuclear Power Plants. Austria (2016).
10. Federal Aviation Administration: AC20-115D Airborne Software Development Assurance Using EUROCAE ED-12( ) and RTCA DO-178( ). Estados Unidos (2018).
11. Society of Automotive Engineers: ARP 4761 - Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment. Estados Unidos (1996).
12. European Aviation Safety Agency: "Certification Memo SW-CEH 002". Alemanha (2011).
13. Rierson, L.: Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance. CRC Press, Estados Unidos (2013).
14. Joung, E. J., Lee C. M., Lee H. M., Kim G. D.: Software Safety Criteria and Application Procedure for the Safety Critical Railway System. In: 2009 Transmission & Distribution Conference & Exposition: Asia and Pacific, pp: 1-4. Coréia do Sul (2009).
15. Chen, T.: Non-safety-related software in the context of railway RAMS standards. In: The Second International Conference on Reliability Systems Engineering (ICRSE), pp: 1-5. China (2017).
16. Magnuson, A.: IEC/ISO 62304 Regulations for the Development of Medical Software Devices. Tese da Chalmers University of Technology. Suécia (2012).
17. Institute of Electrical and Electronics Engineers: IEEE 830-1998 - IEEE Recommended Practice for Software Requirements Specifications. Estados Unidos (1998).
18. International Standardization Organization: ISO/IEC/IEEE 29148:2011 ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes --Requirements engineering. Estados Unidos (2011).
19. Dick, J., Hull, E., Jackson, K.: Requirements Engineering. 4th edn. Springer. Estados Unidos (2017).
20. Laplante, P.: Requirements Engineering for Software and Systems, 4th edn. CRC Press. Estados Unidos (2013).
21. Lauesen, S.: Software Requirements – Styles and Techniques. 1st edn. Pearson. Grã-Bretanha (2002).
22. Gotel, O. C. Z., Finkelstein, C. W.: An analysis of requirements traceability problem. In: Proceedings of IEEE International Conference on Requirements Engineering. Estados Unidos (1994).