

# Proposta de um Framework Ágil para Software Aeronáutico

Andréia Aparecida Costa<sup>1,2</sup>, Johnny Marques<sup>1</sup>

<sup>1</sup>Programa de Pós-graduação em Ciências e Tecnologias Espaciais  
Instituto Tecnológico de Aeronáutica (ITA) São José dos Campos – SP – Brasil

<sup>2</sup>EMBRAER S.A.  
São José dos Campos – SP – Brasil

**Resumo.** Este artigo apresenta uma proposta de framework para desenvolvimento de software aeronáutico, baseado em requisitos obtidos a partir de um Mapeamento Sistemático da Literatura e seus resultados. Esse trabalho apresenta uma pesquisa de mestrado em andamento, onde esse framework apresenta compliance com a RTCA DO-178C, incorporando os valores e princípios do Manifesto Ágil e uma extensão do Scrum.

## 1. Introdução

Tipicamente, os *Software Safety Critical* são desenvolvidos em ambientes regulados por normas e padrões. Exemplos deste tipo de Software são encontrados em domínios, tais como: Aeronáutico, Biomédico, Ferroviário e Nuclear. Nesta pesquisa, os produtos de software destes domínios são tratados como Software em Ambientes Regulados (SAR) [Marques and Cunha 2019]. Segundo Marques et al. (2019), na aviação, o uso de tecnologias avançadas incorporadas aos sistemas, traduz bem a necessidade de Software Aeronáutico que atenda um alto desempenho, combinado com a preocupação em segurança e qualidade [Yelisetty et al. 2021].

A RTCA DO-178C [RTCA 2012] fornece recomendações para o desenvolvimento e certificação de software de aviação. Em sua estrutura, os diversos processos do ciclo de vida do software são descritos por meio de atividades e objetivos que devem ser atendidos [Yelisetty et al. 2015]. A RTCA DO-178C contém 71 objetivos que devem ser cumpridos [Marques and Cunha 2017].

Os métodos ágeis de desenvolvimento de software surgiram no final da década de 90. Segundo Boehm & Turner (2003) “(...) a disciplina é a base para qualquer esforço bem-sucedido. A disciplina cria memórias bem organizadas, histórias e experiências(...)” e ainda completam que “a agilidade aplica memória e história para ajustar-se a novos ambientes”. Percebe-se então algumas características comuns entre agilidade e disciplina.

O Scrum é um *framework* ágil que utiliza ciclos de trabalho curtos chamados *sprints* para promover a colaboração e a entrega incremental de produtos, incluindo papéis, artefatos e cerimônias [Larman and Basili 2003].

O objetivo geral desta pesquisa é, partindo da norma RTCA DO-178C, identificar pontos que possam comprometer a adoção de métodos ágeis no desenvolvimento de software embarcado de uso aeronáutico e propor adaptações no *framework* Scrum.

## 2. Metodologia de Pesquisa

A metodologia de pesquisa envolveu 4 etapas: (i) Execução de um Mapeamento Sistemático da Literatura (MSL); (ii) Organização dos Requisitos para o *Framework*; (iii)

Concepção do *Framework*; e (iv) Avaliação do *Framework* por Grupo Focal de Especialistas e/ou por Experimento. A Figura 1 apresenta um fluxo com as etapas de pesquisa.

Na Etapa 1, o MSL foi realizado compreendendo trabalhos de 2001 até 2022, nas bases: IEEEExplore, SBC Online e Google Scholar. A extração automática revelou 181 trabalhos recuperados nas três bases. Na sequência foi realizada a aplicação dos critérios de inclusão e exclusão, onde restaram 76 artigos. Finalmente, os artigos então foram avaliados quanto aos seus resumos e conclusões, restando após essa etapa 43 trabalhos considerados inclusos para o Mapeamento.

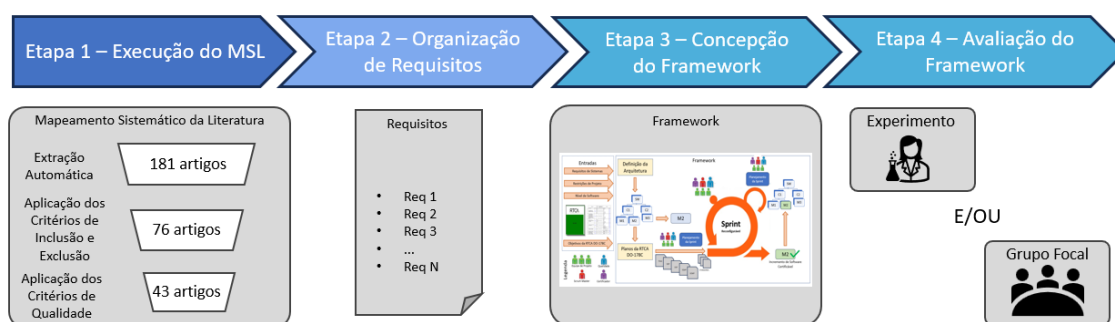


Figura 1. Etapas de pesquisa

Na Etapa 2, a partir da leitura dos 43 trabalhos resultantes, foram identificados 9 requisitos que deveriam ser incorporados ao *framework*. A Tabela 1 apresenta o levantamento dos requisitos para a concepção do *framework*. Na Etapa 3, foi realizada a concepção do *framework*. Na Etapa 4, será feita a avaliação do *framework* a se decidir entre um experimento e/ou a organização de um grupo focal com especialistas.

### 3. Framework

A Figura 2 apresenta, de forma visual, a primeira de duas camadas de abstração do *framework*. A operação do *framework* inicia-se com entradas que vão: a) viabilizar a construção de uma arquitetura inicial para a aplicação que precisará ser desenvolvida; b) fornecer informações para a escrita dos planos; e c) delinear o modo de execução do *framework*. O *framework* necessita de 4 entradas:

- Requisitos de Sistemas: São as especificações funcionais e não-funcionais que descrevem as necessidades de sistemas;
- Restrições de Projeto: São as limitações e condições que afetam a maneira como a aplicação será concebida, desde definições de hardware onde a aplicação será carregada/integrada, padrões de barramento, necessidades de particionamento, taxas de falhas aceitáveis, entre outras;
- Nível de Software: Cada nível de software corresponde então a uma categoria de condição de falha (relacionada ao efeito no nível da aeronave), onde: A corresponde a *Catastrófico*, B para *Perigoso*, C para *Maior*, D para *Menor* e E para aplicações sem impacto em segurança; e
- Objetivos da RTCA DO-178C: Para cada nível de software, a norma apresenta um conjunto de objetivos aplicáveis que devem ser satisfeitos dentro do processo de certificação aeronáutica.

**Tabela 1. Organização dos Requisitos**

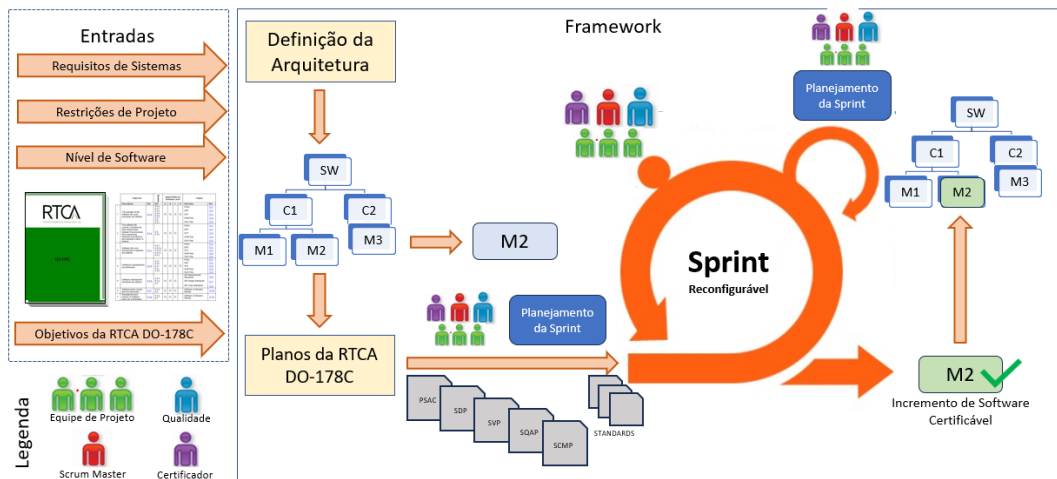
ID	Descrição do Requisito
Req1	O <i>framework</i> deverá, para cada projeto de desenvolvimento de uma aplicação de software, receber como entrada os requisitos de sistemas, o nível do software e as restrições do projeto.
Req2	O <i>framework</i> deverá, para os níveis de A a D, endereçar todos os processos de desenvolvimento previstos na RTCA DO-178C.
Req3	O <i>framework</i> deverá prever a criação de uma arquitetura de software de alto nível, compondo a organização espacial das partes de software e suas camadas de abstração (componentes, módulos ou funcionalidades).
Req4	O <i>framework</i> deverá especializar, com adaptações implementadas no <i>framework</i> Scrum.
Req5	O <i>framework</i> deverá, para cada projeto de desenvolvimento de uma aplicação de software, ter um modelo de sprint de planejamento, onde os documentos de planejamento dos processos previstos pela RTCA DO-178C serão escritos, revisados e submetidos para aprovação.
Req6	O <i>framework</i> deverá ter sprints de tamanho, duração, tipo e quantidade de atividades reconfiguráveis, onde um conjunto de incrementos será selecionado para desenvolvimento e integração, incremental e continuamente.
Req7	O <i>framework</i> deverá prever a extensão papéis previstos do Scrum.
Req8	O <i>framework</i> deverá prever a extensão de cerimônias previstas no Scrum.
Req9	O <i>framework</i> deverá conter um conjunto de métricas para medição e melhoria contínua.

A partir das informações de entrada para o framework, a definição da Arquitetura Inicial é o primeiro passo. Esta deverá apresentar uma visão completa do contexto da aplicação que será desenvolvida, delimitando interfaces externas, modularizando partes do software, chamados aqui de incrementos, e apresentando as suas interações entre si. Estas partes deverão nortear a prioridade e o modelo de *Sprint* mais adequado (escopo de atividades, duração e entregáveis).

Posteriormente, a escrita dos planos, conforme o Nível do Software estabelecido como entrada para o *framework*, deverá orientar os processos previstos pela RTCA DO-178C (Desenvolvimento, Verificação, Gestão da Configuração, Garantia da Qualidade e Certificação), associando-os ao *framework* e respectivos modelos de Sprint (Sprint reconfigurável).

Com a priorização do incremento, um pedaço da Arquitetura (componente, módulo, ou qualquer outra porção) será completamente especificado, desenvolvido e verificado dentro da *Sprint*. Espera-se que a *Sprint* seja reconfigurável, pois a cada nova entrega e incremento, existe uma necessidade de integração contínua para que o produto seja desenvolvido constantemente e aos poucos chegando à sua completude.

O uso de *User Stories* (US) está sendo considerado como sendo mapeadas a partir de Requisitos de Sistemas, possibilitando estimar esforços para implementá-los, de forma simples e direta e com elas usar essas informações para a definição de requisitos de software de alto-nível previstos na RTCA DO-178C. As *User Stories*, seguem um formato



**Figura 2. Primeira camada de abstração do framework**

específico, adaptado de Leffingwell (2011). As *User Stories* deverão conter descrições comportamentais ou propriedades compostas por: (i) Aspectos de desempenho; (ii) Funcionalidades; (iii) Interfaces externas; e (iv) Limites, intervalos e dados.

## Referências

- Boehm, B. and Turner, R. (2003). *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley, Estados Unidos da América.
- Larman, C. and Basili, V. (2003). Iterative and incremental development: A brief history. *Computer*, 36:47–56.
- Leffingwell, D. (2011). *Agile Software Requirements, Lean Requirements Practices for Teams Programs and the Enterprise*. Addison-Wesley.
- Marques, J. (2019). Uma análise das características de especificação de requisitos de software em normas de ambientes regulados. In *Anais do WER19 - Workshop em Engenharia de Requisitos*.
- Marques, J. and Cunha, A. (2017). Verification scenarios of onboard databases under the rtca do-178c and the rtca do-200b. In *36th IEEE/AIAA Digital Avionics Systems Conference*.
- Marques, J. and Cunha, A. (2019). Ares: An agile requirements specification process for regulated environments. *International Journal of Software Engineering and Knowledge Engineering*.
- RTCA (2012). *DO-178C - Software Considerations in Airborne Systems and Equipment Certification*.
- Yelisetty, S., Barros, L., Slavov, T., and Marques, J. (2021). Um levantamento de fatores humanos e técnicos em garantia de qualidade de projeto de software embarcado aeronáutico no brasil. In *Anais do VI Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software*, pages 1–10, Porto Alegre, RS, Brasil. SBC.
- Yelisetty, S., Marques, J., and Tasinaffo, P. (2015). A set of metrics to assess and monitos compliance with rtca do-178c. In *34th AIAA/IEEE Digital Avionics System Conference*.