Mitigação de Riscos de Certificação Civil em um Processo de Certificação de Aeronave Militar: Uma Abordagem para Software Embarcado

Johnny Cardoso Marques [1,2], Luiz Alberto Vieira Dias [3]
[1] Empresa Brasileira de Aeronáutica, Av. Brigadeiro Faria Lima, 2170 - 12227-901, São José dos Campos, SP
[2] Faculdade Anhanguera de Taubaté, Av. Charles Schneider, 585 -12040-001, Taubaté, SP
[3] Instituto Tecnológico de Aeronáutica / Divisão de Computação, Praça Mal. Eduardo Gomes, 50 - 12228-900, São José dos Campos, SP

***Resumo* — Este trabalho apresenta um modelo de mitigação de riscos do requerente de um projeto de aeronave militar em um processo de certificação civil de software embarcado. O trabalho apresenta justificativas para aumentar a atuação do fabricante abordando as atividades adicionais que são esperadas através da classificação por tipos de envolvimento definido pelo autor.**

***Palavras-Chave* — Software, Certificação, Aeronave.**

I. INTRODUÇÃO

A aviação comercial tem como guia principal a segurança de vôo, assim, todo o desenvolvimento de software não visa só um aumento de funcionalidades. A segurança de vôo é a prioridade no software embarcado de uso civil. Podendo todo um projeto de sistema computacional embarcado, cujo software faz parte, ser descartado se não cumprir normas de segurança previstas no processo de certificação [1].

Na aviação civil, principalmente nos grandes jatos, o uso de tecnologias avançadas incorporadas aos sistemas, traduz bem a necessidade de softwares que atendam a um alto grau de tecnologia, combinado com a preocupação em segurança [13].

A aviação de defesa possui um outro foco sobre o software embarcado concentrando primariamente nas inovações tecnológicas (novas ferramentas, processos e tecnologias) e mantendo segurança, mas não sendo este o foco principal. No mercado de defesa, o desenvolvimento de software pode ser definido por duas grandes características:

- Novas tecnologias; e
- Conteúdo preservado, já que envolve informações sensíveis de forças militares de diversos países, na maior parte das vezes.

As aeronaves de defesa possuem um conjunto de características bem definidas, diferente do transporte normal de passageiros. Uma aeronave de defesa, quando projetada, normalmente atende a uma força militar de algum país. Assim, normalmente é definido um conjunto de normas entre fabricante e força militar, para definir o sigilo de informações de determinados sistemas embarcados.

J.C. Marques, johnny.marques@gmail.com, Tel +55-12-39276847,
L.A.V. Dias, vdias@ita.br, Fax +55-12-39475986.

Em um projeto híbrido, onde um avião militar requer certificação civil, é importante o balanceamento entre as duas realidades: certificação civil e projeto militar. É neste escopo que se encaixa este trabalho.

A. *Tipos de Software Presentes em um Projeto Híbrido*

Em um projeto híbrido é necessário agrupar os tipos de software em quatro categorias:

I. Software Embarcado – Plataforma Básica: software destinado à operação e manutenção básica da aeronave, sem considerar suas funcionalidades de missão.

II. Software Embarcado – Especialista: software destinado à operação e manutenção das funcionalidades específicas de missão.

O escopo deste trabalho, embora pudesse ser aplicável em qualquer uma das duas classificações acima, é restrito apenas ao Tipo I, Software Embarcado – Plataforma Básica. Tipicamente, o Tipo I é o software que dentro da aeronave militar é objeto de certificação civil.

II. A DO-178B E SUA APLICAÇÃO

Criada na década de oitenta pela *Radio Technical Commission for Aeronautics, Inc.* (RTCA), o padrão DO-178B estabelece os procedimentos necessários para que o fabricante de aeronaves com uso de software embarcado demonstre que seu produto foi construído atendendo aos requisitos de aeronavegabilidade [14].

A DO-178B [2] foi desenvolvida para estabelecer considerações para desenvolvedores, instaladores, e usuários quando o projeto de um equipamento aeronáuticos é implementado usando software. A RTCA é a entidade responsável por montar os comitês de discussão e revisão de diversas normas de uso aeronáutico. Os comitês da RTCA são formados por membros da comunidade aeronáutica, que inclui autoridades de certificação, fabricantes de aeronaves, fornecedores e fabricantes de sistemas e software, entre outros. A DO-178B é datada de 1992, com sua concepção original em 1982 (DO-178) e revisão posterior. em 1985 (DO-178A). A DO-178B estabelece o que deve ser feito sem estabelecer o formato [10].

Por meio da AC 20-115B [6], o Federal Aviation Agency, agência de certificação americana, reconhece a DO-178B

como um método aceitável para aprovação de sistemas e/ou equipamentos com uso de software, não complementando a DO-178B. Embora a AC 20-115B permita outros métodos alternativos a DO-178B, os projetos de aeronaves com certificação civil, os projetos recentes enxergam apenas a DO-178B como método aceitável de aprovação. A razão para essa abordagem é justificada por dois argumentos:

1) Não existe outra norma com critérios de desenvolvimento de software voltados para a certificação civil.

2) A aceitação de outro método alternativo à DO-178B, embora seja previsto pela autoridade de certificação, é um processo não usual e não existem casos documentados que comprovem essa aceitação.

A DO-178B especifica cinco níveis de software. Cada nível de software possui um rigor diferenciado de processo de desenvolvimento e um conjunto de objetivos que devem ser cumpridos para a aprovação do software perante a autoridade de certificação. Quanto mais crítico é o software, mais rigoroso é o seu processo de desenvolvimento. Dentre os cinco níveis de software (A, B, C, D e E), o nível A é o mais rigoroso, e cada nível inferior (B-D) é feita uma degradação no processo de desenvolvimento, ou seja, alguns objetivos passam a não serem requeridos. O nível E é o mais degradado, sendo o nível de software que não é requerido um processo de desenvolvimento de software.

De acordo com [7], cada falha de sistema deve ser classificada com uma criticalidade associada e essa classificação é também feita em cinco categorias, sendo elas:

- Catastrófica (Catastrophic)
- Perigosa (Hazardous)
- Maior (Major)
- Menor (Minor)
- Sem Impacto em Segurança (No Safety Impact)

Para os sistemas que possuem uso de software, é necessário que se obtenha qual o efeito do mau funcionamento de software [2][9]. Por exemplo, se um mau funcionamento de um determinado software levará à uma condição de falha catastrófica [11] se espera que resulte em mortes de vários ocupantes, ou incapacidade ou ferimento fatal de um tripulante de vôo normalmente com a perda do avião.

Assim, associa-se a classificação da condição de falha com níveis definidos na DO-178B, conforme a TABELA I. É importante salientar que quanto maior a criticalidade, torna-se necessário o cumprimento de um maior número de objetivos, tornando o processo mais rigoroso.

TABELA I CLASSIFICAÇÃO EM NÍVEIS DE SOFTWARE PELA DO-178B

Condição de Falha	Nível de Software Associado	Número de Objetivos a cumprir pela DO-178B
Catastrófica	A	66
Perigosa	B	65
Maior	C	57
Menor	D	28
Sem Impacto	E	Não aplicável

Os 66 objetivos da DO-178B são organizados em 10 tabelas, publicadas no anexo A da norma. As tabelas identificam objetivos de processo com as seguintes características, sendo estas:

- Planejamento (Tabela A-1)
- Desenvolvimento (Tabela A-2)
- Verificação dos Requisitos de Alto, Baixo Nível e Arquitetura de Software (Tabelas A-3, A-4 e A-5)
- Verificação do Código Fonte e Executável (Tabelas A-5 e A-6)
- Testes e Análise (Tabela A-7)
- Controle de Configuração (Tabela A-8)
- Garantia da Qualidade (Tabela A-9)
- Certificação (Tabela A-10)

III. O ENVOLVIMENTO DO REQUERENTE

A. *Requerente*

Um requerente, em inglês *applicant*, é o responsável por obtenção de uma certificação de um produto aeronáutico [12]. O requerente pode ser qualquer pessoa, física ou jurídica, que esteja desenvolvendo ou deseja obter certificação em um produto aeronáutico: aeronave, motor ou hélice.

O requerente é o responsável por demonstrar cumprimento com o requisitos aplicáveis ao projeto a ser certificado, incluindo requisitos que se aplicam ao software embarcado de Tipo 1. Um requerente pode contratar fornecedores, sendo estes outras pessoas físicas ou jurídicas para desenvolver partes do projeto, incluindo a parte de software. Porém a responsabilidade e a condução do processo de certificação perante a autoridade de certificação é de responsabilidade do requerente.

B. *O Critério para Envolvimento do Requerente*

Em [8] são identificados os dez pontos chaves para falha de projetos de software:

i. Os objetivos do projeto não são bem definidos e/ou os envolvidos não são identificados;
ii. Os objetivos do projeto são definidos de forma apropriada, mas as mudanças não são controladas de forma apropriada;
iii. O projeto não é planejado de forma apropriada;
iv. O projeto não é gerenciado de forma apropriada;
v. O projeto é planejado de forma apropriada, porém seus recursos não são gerenciados;
vi. Não são criados planos de contingência;
vii. As expectativas dos envolvidos com relação ao projeto não são gerenciadas;
viii. O projeto é planejado de forma apropriada, mas seu progresso não é monitorado e controlado;
ix. Relatórios de progresso são inadequados ou inexistentes;
x. Quando ocorre problema no projeto, as pessoas acreditam que os mesmo podem ser resolvidos de forma simples.

Nos últimos anos, os projetos de desenvolvimento de aeronaves com certificação civil tem sido alvo de sucessivos atrasos. Muitos desses atrasos são oriundos da dificuldade em demonstrar requisitos de certificação para a autoridade. Cabe ao requerente, responsável pelo processo de certificação, mitigar esses dez pontos, principalmente o itens vii, viii e ix . Existem casos conhecidos de projetos de aeronaves que tiveram suas certificações postergadas por problemas em garantir que o software embarcado é adequado para sua utilização.

Eventualmente esses fornecedores possuem experiência nos produtos que fornece. Mas nem sempre essa premissa é verdadeira já que muitas vezes por conta da inovação tecnológica que está sendo agregada ao projeto da aeronave, toda experiência anterior de um determinado fornecedor pode não ser um ponto de sucesso garantido.

Além disso, os fabricantes aeronáuticos têm interesse em desenvolver novos fornecedores. Por exemplo, uma empresa que faz projeto de interior para automóveis, pode ser desenvolvida a desenvolver projetos de interiores para aeronaves.

No campo de software, a mesma premissa é verdadeira. Muitos fornecedores de sistemas que no passado eram puramente mecânicos, hoje se capacitam e buscam um re-posicionamento no mercado, oferecendo sistemas com software embarcado [5].

Porém o desenvolvimento de software embarcado exige o estabelecimento de um rigor de processo com aderência à DO-178B. Na maioria dos casos, estabelecer um novo modelo de desenvolvimento que considere o uso de software e processos de desenvolvimento não acontece de maneira automática. A falta de maturidade na área de software e na DO-178B é objeto de risco que deve ser monitorado proximamente pelo fabricante da aeronave que será certificada.

Uma aeronave possui em média cerca de 60 itens de software embarcados. Cada item de software é desenvolvido por um fornecedor distinto, logo o acompanhamento de 60 desenvolvimentos de software distintos requer um esforço de pessoal que muitas vezes não está disponível.

Do lado da autoridade de certificação, também é inviável ter um envolvimento grande em todos esses projetos de software que farão parte da aeronave a ser certificada. Nesse aspecto é necessário que o requerente estabeleça um critério para definir qual o seu envolvimento na mitigação de riscos de certificação.

Dentro desse critério alguns aspectos relevantes aumentam o risco de certificação, sendo estes:

- Nível de Software: quanto maior o nível, maior é a preocupação das autoridades de certificação;
- Uso de Metodologias e Tecnologias ainda sem abordagem na DO-178B, como por exemplo, desenvolvimento baseado em modelos; e
- Experiência do desenvolvedor de software em projetos de sistemas similares e com a DO-178B.

A proposta desse trabalho é a definição de uma metodologia que deve seguir algumas etapas, sendo elas:

1) Levantar informações sobre o desenvolvimento de software que pode trazer riscos à certificação.
2) Quantificar o resultado das respostas recebidas e obter a pontuação.
3) Classificar o desenvolvimento de software em faixas de risco apropriadas.
4) Realizar as atividades de mitigação de riscos previstas.

Na primeira etapa é necessário obter informações mais detalhadas sobre o nível de software a ser desenvolvido. O modelo mais adequado é o de aplicar um questionário no desenvolvedor com perguntas sugeridas conforme a TABELA II. Cada tipo de resposta provável recebe uma pontuação de 0 (zero) à 2 (dois) pontos.

TABELA II QUESTIONÁRIO A SER APLICADO NO DESENVOLVEDOR DE SOFTWARE

Questão	Opções de Resposta
Em qual dos grupos de sistemas, o software será utilizado?	(2 pontos) Grupo 1: Aviônica, Comandos de Vôo, Piloto-Automático, Sensores. (1 ponto) Grupo 2: Sistemas Ambientais (Controlador de pressão da cabine, aquecedor de pára-brisa), Trem de Pouso (Controle de Freio, Sensor de Proximidade), Combustível, Motor, Sistema Elétrico (Primário e Secundário de Distribuição de Energia). (0 ponto) Grupo 3: Outros sistemas / equipamentos não mencionados nos Grupos 1 ou 2.
O software já foi previamente desenvolvido e certificado pela DO-178B?	(2 pontos) Não. O software é completamente novo. (1 ponto) Sim. Existem grandes modificações a partir de um software já certificado em outra aeronave. (0 ponto) Sim. Existem pequenas modificações a partir de um software já certificado em outra aeronave.
O fornecedor tem experiência prévia no sistema e/ou software a ser desenvolvido para a aeronave?	(2 pontos) Não. O fornecedor não possui experiência no sistema e/ou software a ser desenvolvido. (1 ponto) Sim. O fornecedor tem experiência no sistema a ser fornecido, mas nunca implementou o sistema usando software. (0 ponto) Sim. O fornecedor tem experiência no sistema a ser fornecido e em implementar usando software.
O desenvolvedor tem experiência em desenvolver software no nível requerido?	(2 pontos) Não. O desenvolvedor não tem experiência em desenvolver software pela DO-178B no nível requerido ou superior ao requerido. (1 ponto) Sim. O fornecedor possui experiência de até 7 anos no desenvolvimento de software pela DO-178B no nível requerido ou superior. (0 ponto) Sim. O fornecedor possui experiência de mais de 7 anos em desenvolver software pela DO-178B no nível requerido ou superior.
O desenvolvedor de software vai subcontratar alguma atividade de desenvolvimento para uma outra empresa?	(2 pontos) Sim. Mais de uma atividade de desenvolvimento será subcontratada. (1 ponto) Sim. Uma atividade de desenvolvimento será subcontratada. (0 ponto) Não haverá nenhuma subcontratação de outra empresa para o desenvolvimento de software.
O software será desenvolvido usando técnicas de orientação a objeto durante a implementação do design ou código?	(2 pontos) Sim. No design e código. (1 ponto) Sim. Apenas no código. (0 ponto) Não.
O software será desenvolvido usando desenvolvimento baseado em modelos?	(2 pontos) Sim, em Requisitos de alto nível, design e requisitos de baixo nível. (1 ponto) Sim, em design e requisitos de baixo nível. (0 ponto) Não.

A partir das resposta as perguntas típicas apresentadas na TABELA II, é necessário criar um mecanismo de correlação dessas informações e identificar o risco de certificação que o projeto de software oferece. Nesse sentido, a TABELA III

define o critério para classificar cada desenvolvimento nos riscos alto, médio e baixo.

TABELA III CRITÉRIO PARA CLASSIFICAÇÃO DO RISCO DE CERTIFICAÇÃO

Risco	Critério
Alto	Total de pontos obtidos é maior ou igual a 9.
Médio	Total de pontos obtidos é maior ou igual a 4 e menor que 9.
Baixo	Total de pontos é menor que 4.

Com a classificação de risco obtida, é necessário então fazer uma segunda correlação, que considera o nível de software conforme estabelecido na DO-178B. Essa segunda correlação define o tipo de envolvimento do requerente no projeto de desenvolvimento de software. Essa correlação é apresentada na TABELA IV.

TABELA IV CRITÉRIO PARA CLASSIFICAÇÃO DO RISCO DE CERTIFICAÇÃO

Nível de Software	Risco Alto	Risco Médio	Risco Baixo
A	Tipo 1	Tipo 2	Tipo 3
B	Tipo 2	Tipo 3	Não Aplicável
C	Tipo 3	Não Aplicável	Não Aplicável
D / E	Não Aplicável	Não Aplicável	Não Aplicável

Com a definição do tipo de envolvimento do requerente no projeto de desenvolvimento de software é necessário que para cada um dos tipos seja definido o conjunto de atividades que o requerente planeja realizar para mitigar riscos na certificação. Pela correlação apresentada na TABELA IV, o Tipo 1 deve ser o que possui mais ações de mitigação de riscos, seguido pelos Tipos 2 e 3, nessa ordem.

Para cada um dos tipos é necessário identificar o conjunto de atividades para mitigação de riscos de certificação, a TABELA V define esse conjunto.

TABELA V ATIVIDADES DE MITIGAÇÃO DE RISCO DE CERTIFICAÇÃO POR TIPO DE ENVOLVIMENTO DO REQUERENTE

Tipo	Atividades de Mitigação de Risco de Certificação
1	• Auditoria do Requerente 1 (AR1) • Auditoria do Requerente 2 (AR2) • Auditoria do Requerente 3 (AR3) • Auditoria do Requerente 4 (AR4)
2	• Auditoria do Requerente 1 (AR1) • Auditoria do Requerente 3 (AR3)
3	• Auditoria do Requerente 1 (AR1)

De acordo com [4], durante as auditorias da autoridade os itens identificados pela autoridade podem ser classificados como:

- *Finding*: Falta de cumprimento com um ou mais itens da DO-178B.
- *Action:* Ação de esclarecimento que demanda resposta de uma pessoa ou organização com uma data específica. Caso não seja esclarecido pode ser reclassificado um *finding*.
- *Observation:* Identificação de melhoria no processo de desenvolvimento de software. Um *observation* não é um item de falha de cumprimento com a DO-178B e não é obrigatória sua implementação.

C. *Auditoria do Requerente 1 (AR1)*

A AR1 deve ser conduzida quando os planos e padrões para desenvolvimento de software previstos pela DO-178B estiverem emitidos.

Critério de Entrada:

- Planos de Desenvolvimento de Software e Padrões de Desenvolvimento de Software foram desenvolvidos, controlados, revisados e aprovados pelo grupo de desenvolvimento de software ou fornecedor.

A TABELA VI define quais as entradas necessárias para a realização da AR1 para cada tipo definido na TABELA V.

TABELA VI ENTRADAS NECESSÁRIA PARA A REALIZAÇÃO DA AR1

Artefatos (conforme definidos na DO-178B)
Plano dos Aspectos de Software para a Certificação (PSAC)
Plano de Desenvolvimento de Software (SDP)
Plano de Verificação de Software (SVP)
Plano de Controle de Configuração de Software (SCMP)
Plano de Qualidade de Software (SQAP)
Padrões de Requisitos, Projeto e Codificação de Software
Registros da Qualidade de Software

Objetivos da DO-178 a serem verificados na AR1:

- Tabela A-1 (Objetivos 1 a 7)
- Tabela A-9 (Objetivo 1)
- Tabela A-10 (Objetivo 1 e 2)

Critério de Saída:

Planos de Desenvolvimento de Software e Padrões de Desenvolvimento de Software foram avaliados e aprovados pelo requerente.

D. *Auditoria do Requerente 2 (AR2)*

A AR2 deve ser conduzido quando uma porção representativa (cerca de 40 a 60%) do produto de software estiver desenvolvida e revisada incluindo requisitos, projeto e código.

Critério de Entrada:

- Amostras de requisitos de software de alto nível estão documentadas, revisadas, e rastreáveis para requisitos de sistemas.
- Amostras de requisitos de software de baixo nível estão documentadas, revisadas, e rastreáveis para requisitos de software de alto nível.
- Arquitetura de Software está definida e revisões e análises foram conduzidas.
- Amostras de código fonte que implementam requisitos de baixo nível estão geradas, revisadas e rastreiam para requisitos de baixo-nível.
- Itens de ação definidos na AR1 encontram-se fechados.

A TABELA VII define quais as entradas necessárias para a realização da AR2 para cada Tipo definido na TABELA V.

TABELA VII ENTRADAS NECESSÁRIA PARA A REALIZAÇÃO DA AR2

Artefatos (conforme definidos na DO-178B)
Planos de Qualificação de Ferramentas de Desenvolvimento de Software
Padrões de Requisitos, Projeto e Codificação de Software
Documento de Requisitos de Software (SRD)
Documento de Projeto de Software (SDD)
Código Fonte
Registros de Problemas
Registros de Controle de Configuração
Registros da Qualidade de Software

Objetivos da DO-178 a serem verificados na AR2:

- Tabela A-2 (Objetivos 1a 6)

- Tabela A-3 (Objetivos 1 a 7)
- Tabela A-4 (Objetivos 1 a 13)
- Tabela A-5 (Objetivos 1 a 6)
- Tabela A-8 (Objetivos 1 a 4 e 6)
- Tabela A-9 (Objetivos 1 e 2)
- Tabela A-10 (Objetivo 3)

Critério de Saída:

Artefatos e amostras de dados de desenvolvimento foram avaliados pelo requerente.

E. Auditoria do Requerente 3 (AR3)

A AR3 deve ser conduzido quando uma porção representativa (cerca de 40 a 60%) da verificação do produto de software estiver desenvolvida e revisada.

Critério de Entrada:

- Amostras de casos de teste e procedimentos estão documentadas, controladas e revisadas.
- Amostras de resultados de testes estão documentadas, controladas e revisadas.
- Ambiente de testes está documentado e controlado.
- Itens de ação da auditoria AR2 foram fechados.

A TABELA VIII define quais as entradas necessárias para a realização da AR3 para cada tipo definido na TABELA V.

TABELA VIII ENTRADAS NECESSÁRIA PARA A REALIZAÇÃO DA AR3

Artefatos (conforme definidos na DO-178B)
Documento de Requisitos de Software (SRD)
Documento de Projeto de Software (SDD)
Código Fonte
Código Executável
Casos de Teste e Procedimentos
Resultados de Verificação de Software
Documento de Configuração do Ambiente de Desenvolvimento de Software (SECI)
Documento de Configuração de Software (SCI)
Registros de Problemas
Registros de Controle de Configuração
Registros da Qualidade de Software
Dados de Qualificação de Ferramentas
Estratégia de Rastreabilidade entre Código Fonte e Código Executável

Objetivos da DO-178 a serem verificados na AR3:

- Tabela A-2 (Objetivo 7)
- Tabela A-5 (Objetivo 7)
- Tabela A-6 (Objetivos 1 a 5)
- Tabela A-7 (Objetivos 1 a 8)
- Tabela A-8 (Objetivos 1 a 6)
- Tabela A-9 (Objetivos 1 e 2)
- Tabela A-10 (Objetivo 3)

Critério de Saída:

Artefatos e amostras de dados de verificação foram avaliados pelo requerente.

F. Auditoria do Requerente 4 (AR4)

A AR4 deve ser conduzida após a versão final de software estar completa, a verificação de software estiver completa, a conformidade de software foi conduzida e o software estiver pronto para fazer parte da aeronave a ser certificada.

Critério de Entrada:

- Revisão da conformidade de software foi realizada e qualquer deficiência foi resolvida.
- SAS e SCI foram finalizados e revisados.
- Todos os artefatos requeridos pela DO-178B estão completos, aprovados e colocados em controle de configuração.
- Itens de ação da auditoria AR3 foram fechados.

A TABELA IX define quais as entradas necessárias para a realização da AR3 para cada tipo definido na TABELA V.

TABELA IX ENTRADAS NECESSÁRIA PARA A REALIZAÇÃO DA AR4

Artefatos (conforme definidos na DO-178B)
Documento de Configuração de Software (SCI)
Documento de Configuração do Ambiente de Desenvolvimento de Software (SECI)
Resumo do Ciclo de Desenvolvimento e Certificação de Software (SAS)
Registros da Qualidade de Software

Objetivos da DO-178 a serem verificados na AR4:

- Tabela A-9 (Objetivos 1 a 2)
- Tabela A-10 (Objetivo 3)

Critério de Saída:

SAS e SCI foram avaliados pelo requerente.

IV. ATIVIDADES DE ENVOLVIMENTO DA AUTORIDADE CERTIFICADORA

De acordo com [3] e [4], o FAA (Federal Aviation Administration), autoridade de certificação americana pode se envolver nos desenvolvimentos de software seguindo o seguinte critério especificado na TABELA X.

TABELA X TIPO DE ENVOLVIMENTO DO FAA

Nível de Software	**Tipo de Envolvimento do FAA**
D	Baixo
C	Baixo ou Médio
B	Médio ou Alto
A	Alto

V. VALIDAÇÃO DA METODOLOGIA

Este modelo proposto pelos autores ainda não foi exercitado em um projeto militar com certificação civil.

Um protótipo anterior à este mesmo modelo foi implementado por um requerente na certificação civil de um projeto na aviação executiva. Este modelo em que foi aplicado o projeto, era o segundo de uma mesma família, composta por duas aeronaves, o modelo 1, com capacidade de até quatro passageiros e o modelo 2 com capacidade de até sete passageiros.

Na certificação do modelo 1, em que o protótipo da metodologia não foi aplicada, a certificação sofreu um atraso de seis meses em relação ao cronograma original. A falta de maturidade no desenvolvimento utilizando a DO-178B por parte da empresa contratada pelo requerente e a ausência de uma metodologia de mitigação de riscos organizado, fez com

que a reação do requerente ocorre-se tardiamente, atrasando o cronograma de certificação.

Na certificação do modelo 2, em que o protótipo da metodologia foi aplicada, a certificação não sofreu atraso em relação ao cronograma original. Embora tenha sido identificado durante essa certificação o mesmo problema do modelo 1, ou seja, falta de maturidade no desenvolvimento utilizando a DO-178B por parte da empresa contratada pelo requerente, a existência de um modelo de mitigação de riscos, fez com que as ações tomadas pelo requerente no momento adequado contornasse as dificuldades e mantendo os prazos originais de certificação.

A abordagem proposta neste trabalho baseia-se no protótipo aplicado no modelo 2, embutindo melhorias identificadas e estendendo sua aplicabilidade para projetos militares com certificação civil.

VI. CONSIDERAÇÕES FINAIS

Mitigar riscos em desenvolvimentos de software para aeronaves militares com certificação civil é de responsabilidade do requerente.

Recomenda-se o envolvimento antecipado do requerente em projetos de desenvolvimento de software, principalmente quando fornecedores externos foram contratados.

Ainda são poucas as experiências de projetos militares que foram certificados pela autoridade civil. O requerente responsável por um projeto militar deve se antecipar aplicando uma metodologia de mitigação de riscos, como a proposta neste trabalho. Do outro lado do processo, a autoridade de certificação também precisa ter uma preocupação adicional, já que um requerente de um projeto militar, notadamente, pode ter pouca experiência em certificação civil.

Existe a tendência de projetos de aeronaves militares requererem a certificação civil. Os tipos de projetos militares mais aplicáveis são os destinados para transporte de tropas, autoridades diplomáticas e resgate de pessoas.

Um outro aspecto é que um projeto militar pode ser adaptado, após sua certificação, e tornar-se um produto aeronáutico para aviação civil. Um projeto de uma aeronave para transporte de tropas, pode ser adaptado pelo requerente e posteriormente se transformar numa aeronave cargueira para uso comercial. Se na certificação da aeronave para o transporte de tropas, for realizada a certificação civil, ao adaptar essa aeronave para o transporte de cargas comerciais, o projeto adaptado terá um processo de certificação civil mais rápido e grande parte do crédito da certificação anterior poderá ser requerida.

A autoridade de certificação civil não possui diferenças de abordagem entre projetos de aeronaves comerciais e militares. Os requisitos publicados não diferenciam o escopo de aplicação, e sendo assim, é esperado que um projeto militar tenha mesmo rigor de aplicação dado a projetos de aeronaves comerciais como nos jatos e turboélices.

REFERÊNCIAS

[1] J. C. Marques, "Processo de Reuso de Software Aeronáutico: Uma Proposta para a EMBRAER", Instituto Tecnológico de Aeronáutica, São José dos Campos, Brasil, 2005.

[2] Radio Technical Commission for Aeronautics, "DO-178B - Software Considerations in Airborne Systems and Equipment Certification", Washington, Estados Unidos, 1992.

[3] Federal Aviation Administration, "Software Approval Guidelines", Washington, Estados Unidos, 2003.

[4] Federal Aviation Administration, "Conducting Software Reviews Prior to Certification", Washington, Estados Unidos, 2004.

[5] M. J. R. Lemes, F. O., Altoé, A. J. Domiciano, A. J. Carbonari "Software certification in airborne systems: process and challenges" no Primeiro Simpósio Latino-Americano em Computação Dependente, São Paulo, Brasil, 2003.

[6] Federal Aviation Administration, "AC 20-115B – RTCA Inc, Document RTCA/DO-178B", Washington, Estados Unidos, 1993.

[7] SAE, "ARP 4761 - Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment", Warrendale, Estados Unidos, 1996.

[8] F. Amaral, "Por que projetos de software falham?", Disponível em: < http://www.fernandoamaral.com.br/Default.aspx?Artigo=52>. Acesso em: 26/04/2010.

[9] T. Baghai, V. Hildeman, "Avionics Certification: A Complete Guide to DO-178 (Software), DO-254 (Hardware)", Estados Unidos, 2007.

[10] J. C. Marques, Notas de Aula do Curso "Critérios de Certificação de Software Embarcado", EMBRAER, São José dos Campos, Brasil, 2009.

[11] Federal Aviation Administration, "AC 23-1309-1D – System Safety Analysis and Assessment for Part 23 Airplanes", Washington, Estados Unidos, 2009.

[12] Federal Aviation Administration, "14 CFR PART 21 – Certification Procedures for Products and Parts". Disponível em: <http://ecfr.gpoaccess.gov/cgi/t/text/text-idx?c=ecfr&sid=8018efd3816a57c4a17b2a72032aa058&tpl=/ecfrbrowse/Title14/14cfr21_main_02.tpl>. Acesso em: 18/05/2010.

[13] V. I. Quandt, "Processo Para Análise De Compatibilidade De Software E Hardware Em Modificações De Sistemas", ITA, São José dos Campos, Brasil, 2009.

[14] R.C. Martins, S. R. M . Pellegrino, J. Santellano, "Tratamento de dead codes em Software de Uso Aeronáutico", no DINCON 2010, 9th Conference on Dynamics, Controls and their Applications, Serra Negra, Brasil, 2010.