



ESCOLA DE APERFEIÇOAMENTO DE OFICIAIS DA AERONÁUTICA
DIVISÃO DE ENSINO
CURSO DE APERFEIÇOAMENTO DE OFICIAIS 1º/2025

JOSÉ BRACONNOT VELLOSO, Cap Eng

Desenvolvimento de *software* embarcado no âmbito do DCTA

Rio de Janeiro

2025

ESCOLA DE APERFEIÇOAMENTO DE OFICIAIS DA AERONÁUTICA
DIVISÃO DE ENSINO
CURSO DE APERFEIÇOAMENTO DE OFICIAIS 1º/2025

JOSÉ BRACONNOT **VELLOSO**, Cap Eng

Desenvolvimento de *software* embarcado no âmbito do DCTA

Trabalho de conclusão de curso apresentado à Escola de Aperfeiçoamento de Oficiais da Aeronáutica como requisito parcial para aprovação no Curso de Pós-Graduação *Lato Sensu* em Liderança com Ênfase em Gestão no COMAER.

Linha de Pesquisa: Ciência, Tecnologia e Inovação

Orientador: Danilo Bichir, Maj Int

Rio de Janeiro

2025

JOSÉ BRACONNOT **VELLOSO**, Cap Eng

Desenvolvimento de *software* embarcado no âmbito do DCTA

Trabalho de conclusão de curso apresentado ao
Curso de Aperfeiçoamento de Oficiais da Escola
de Aperfeiçoamento de Oficiais da Aeronáutica.

Aprovado por:

Presidente, Jaqueline de Azevedo Bruno, Ten Cel Int - IEFA

Danilo Bichir, Maj Int - EAOAR

Rio de Janeiro

2025

RESUMO

Existem atualmente, em algumas Organizações Militares no Departamento de Ciência e Tecnologia Aeroespacial (DCTA), pequenas equipes trabalhando de forma autônoma, para aplicações e projetos bastante específicos, utilizando diferentes *hardwares* e linguagens de programação, o que dificulta a reutilização dos códigos desenvolvidos. Devido a esta estrutura de equipes autônomas, existe ainda, a dificuldade de evoluir os profissionais e gerenciar o conhecimento nesta área. Com isso, este ensaio defende a implantação de um Centro de Desenvolvimento de *Software* Embarcado no DCTA, com objetivo de proporcionar um gerenciamento do conhecimento, gerando equipes mais experientes e produtivas para desenvolver os *softwares* que embarcam as soluções científico-tecnológicas e inovadoras. A implantação de um Centro de *Software* Embarcado permite construir critérios para definição dos *hardwares* de forma a otimizar a reutilização de códigos, melhorando a produtividade com o uso de códigos já homologados. Este Centro proporciona ainda o gerenciamento do conhecimento em *software* embarcado, formando e mantendo equipes experientes para desenvolver os projetos do DCTA, em uma área de alta complexidade técnica. No futuro, o Centro deve tornar-se uma referência para soluções de *software* embarcado, integrando os desenvolvedores, a fim de atender às demandas de interesse do Comando da Aeronáutica, assim como os Centros de Computação da Aeronáutica o fazem, para *softwares* corporativos e simuladores.

Palavras-chave: *software* embarcado; gerenciamento do conhecimento; reutilização de código; padronização de *hardware*.

1 INTRODUÇÃO

Em meados de 2014, o cenário do desenvolvimento de *softwares* corporativos no Comando da Aeronáutica (COMAER) encontrava-se com uma certa desorganização e descentralização. As Organizações Militares (OM) desenvolviam, localmente em seus setores de Tecnologia da Informação (TI), vários *softwares* para automatizar seus procedimentos administrativos. Normalmente, eram *softwares* para internet para serem executados na rede interna da OM, como os amplamente utilizados formulários, que são preenchidos em navegadores e executam alguma tarefa específica, como por exemplo: solicitação de reserva de hotel de trânsito. O melhor exemplo desta desorganização foram as aplicações desenvolvidas para geração automática de publicação de apresentação em boletim (missão, férias, viagens, etc), em que foram identificadas inúmeras OM desenvolvendo sistemas com a mesma finalidade, qual seja: apresentação de militares. De 2014 até os dias atuais, houve muitas melhorias. Existe atualmente, um portal do militar, no qual são concentradas várias tarefas administrativas, com acesso a todas as OM do COMAER e sendo desenvolvido e suportado por apenas um órgão responsável, evitando o retrabalho de outrora.

Em contexto semelhante, até por uma proximidade das atividades, encontra-se o desenvolvimento de *softwares* embarcados. Para melhor situar o contexto, vale citar que desenvolvedores de *software* corporativos, normalmente, não se aventuram no desenvolvimento de *softwares* embarcados e vice-versa. Ainda que, por acaso, usem a mesma linguagem de programação (o que é totalmente possível), não costumam migrar de área. Em resumo, são abordagens tão distintas que tornaram-se praticamente áreas separadas, com profissionais e experiências próprios.

Observando-se esta evolução no desenvolvimento dos *softwares* corporativos no COMAER, torna-se ainda mais marcante a falta desta, no desenvolvimento de *softwares* embarcados no Departamento de Ciência e Tecnologia Aeroespacial (DCTA). Nos dias atuais, existem pequenas equipes trabalhando de forma autônoma, utilizando diferentes *hardwares* e linguagens de programação, o que traz dificuldades em: reutilizar os códigos desenvolvidos; evoluir os profissionais; e gerenciar o conhecimento nesta área. Particularmente, esta dificuldade de evoluir os profissionais foi notada em uma equipe de um projeto estratégico do COMAER, quando houve adversidades na absorção de mão de obra especializada de bolsistas nesta área para suprir a falta de profissionais, deixando a equipe com força de trabalho em torno de 50% deficitária, o que obviamente gerou percalços no progresso do projeto. Outro impacto que pode ser causado é a busca por empresas especializadas, quando é iniciado um

projeto de maior complexidade, diminuindo o controle sobre o processo de desenvolvimento do projeto pelo DCTA.

Com isso, este ensaio defende a implantação de um Centro de Desenvolvimento de *Software* Embarcado no DCTA, com objetivo de proporcionar um gerenciamento do conhecimento, gerando equipes mais experientes e produtivas para desenvolver os *softwares* que embarcam as soluções científico-tecnológicas e inovadoras.

Ainda que os *softwares* embarcados possuam finalidades diferentes em cada projeto, a implantação de um Centro de *Software* Embarcado permite construir critérios para definição dos *hardwares* de forma a otimizar a reutilização de códigos, melhorando a produtividade com o uso de códigos já homologados. Este Centro proporciona ainda o gerenciamento do conhecimento em *Software* Embarcado, formando e mantendo equipes experientes para desenvolver os projetos do DCTA, em uma área de alta complexidade técnica.

2 DESENVOLVIMENTO

Apesar deste ensaio ter ênfase na parte gerencial das equipes de desenvolvimento dos produtos aeroespaciais e de defesa, tornam-se oportunas algumas definições, para melhor entendimento das intervenções propostas nos argumentos e seus resultados esperados. Existem muitas definições sobre o que são sistemas embarcados, que diferem em alguns pontos e este ensaio utilizará aquela proposta por Oshana e Kraeling (2019), que é suficiente para o entendimento do trabalho: um sistema embarcado é um conjunto de *hardware* (placas eletrônicas) e *software* (programa) que, juntos, formam um equipamento para executar uma tarefa específica. Diferentemente de um computador, que é desenvolvido para executar múltiplas tarefas genéricas, os sistemas embarcados são restritos à sua aplicação.

O *software* embarcado nada mais é então que: um programa desenvolvido para ser executado em um sistema embarcado e que controla o *hardware*, para desempenhar a aplicação para o qual foi projetado. Um exemplo simples seria um televisor, que possui um *hardware* que permite que a imagem seja projetada na tela e um *software* que executa as tarefas da interface com o usuário, quando este pressiona os botões. Sistemas mais complexos também utilizam *software* embarcado, como os computadores de bordo de um Veículo Aéreo não Tripulado (VANT) ou de um Míssil. Nestes últimos casos, os *softwares* executam inúmeras tarefas, como por exemplo, receber informações de sensores e gerar deflexão nas superfícies de controle da plataforma para voo.

2.1 CRITÉRIOS PARA ESCOLHA DE *HARDWARES* E REUTILIZAÇÃO DE CÓDIGOS

Como pode ser percebido na definição, os *softwares* embarcados possuem uma intrínseca relação com o *hardware*. Em sistemas complexos, como são muitas vezes os dos projetos do DCTA, pode-se criar uma dependência do *software* com o *hardware* utilizado.

Assim, alcançar a portabilidade do *software* desenvolvido para um outro *hardware*, torna-se um desafio, principalmente quando se trata das partes do código que interagem diretamente com o *hardware*. Tradicionalmente, a adaptação do código necessária para essa portabilidade é feita manualmente, o que é oneroso e acarreta erros. Atualmente, existe uma diversidade de *hardwares* e uma crescente dependência entre estes e os *softwares* embarcados (Martins Gomes; Aichernig; Baunach, 2024).

As equipes autônomas, existentes nas OM do DCTA, escolhem *hardwares* em seus projetos, sem levar em consideração os utilizados em outros projetos ou mesmo sem pensar em uma abordagem diferente de desenvolvimento de *software*, que otimize uma futura portabilidade, como por exemplo, a proposta por Martins Gomes, Aichernig e Baunach (2024). A implantação do Centro de *Software* Embarcado visa justamente auxiliar a superar este desafio, criando uma estrutura de apoio à decisão de quais *hardwares* utilizar, nos diversos projetos do DCTA. A proposta para o Centro de Desenvolvimento é padronizar *hardwares* menores, que requerem programação muito específica, mantendo a mesma família ou fabricante, até que haja algum real motivo para migração para outro padrão. E para sistemas maiores, como computadores de bordo, que utilizam *hardwares* mais completos, uma padronização é importante devido à complexidade do *software* empregado e devido à possibilidade de utilizar um *hardware* que já passou por diversos testes de estresse em outros projetos.

O objetivo principal deste direcionamento na escolha dos *hardwares* ou uso de metodologias que permitam a portabilidade é: permitir a reutilização de códigos, que formam um artefato ou um produto, que pode ser reutilizado como um todo ou em partes (módulos). O desenvolvimento de *softwares* embarcados é uma área bastante desafiadora, principalmente nos sistemas mais complexos, com restrições temporais, de segurança, etc. Assim, a reutilização de códigos reduz o tempo de desenvolvimento e de testes (Al Maruf; Azim; Alam, 2022). Com o crescimento no número de linhas de código dos *softwares* embarcados (com os sistemas tornando-se mais complexos), a reutilização é inevitável e cada vez mais são combinadas partes de códigos de dois ou mais projetos. Porém, se o desenvolvimento dos *softwares* não é bem organizado, gerenciando as dependências entre os módulos do *software*,

esta reutilização é prejudicada (Kambe *et al.*, 2017). Isto torna-se ainda mais relevante quando trata-se de sistemas críticos, nos quais falhas devem ser contornadas em vista a impedir um acidente, como por exemplo, em um VANT. A reutilização de códigos de outros projetos, não só evita o tempo dispendido na codificação, mas, mais importante que isso, a parte do código reutilizada já passou por várias fases do desenvolvimento no projeto anterior, como por exemplo: as fases de testes de *software*; correção de erros de codificação (*bugs*); e testes de homologação em ambiente real.

Como apresentado por Dullemond e Van Solingen (2013), as equipes podem trabalhar no mesmo local ou distribuídas em locais diferentes. O cenário atual no DCTA, em que cada equipe trabalha localmente e isolada das demais, não se caracteriza como uma configuração distribuída, dificultando a reutilização de códigos de dois ou mais projetos. Esta reutilização, tão importante conforme citado por Al Maruf, Azim e Alam, (2022) e apontada como inevitável por Kambe *et al.* (2017), é possibilitada com a criação de um Centro de Desenvolvimento, que agregue as equipes, para que os desenvolvedores possam interagir, organizar e gerenciar a produção dos códigos dos softwares, tornando as entregas mais eficientes e robustas, para os projetos complexos do DCTA na área aeroespacial e de defesa.

2.2 GERENCIAMENTO DO CONHECIMENTO EM SOFTWARE EMBARCADO

Conforme já mencionado, o *software* embarcado possui uma relação intrínseca com o *hardware* e nesse contexto, as equipes de *software* embarcado não trabalham isoladas, e sim, se relacionando com as equipes de *hardware* e outras. Neste ambiente multifuncional, construir e gerenciar o conhecimento dos desenvolvedores não é trivial e tornou-se uma necessidade real em pequenas, médias e grandes organizações que desenvolvem *software* (Rabelo; Conte, 2017). O gerenciamento do conhecimento (*Knowledge Management, KM*) é então o conceito de coletar dados, gerar informações e usar este conhecimento. O aprendizado em equipe é reconhecido como um importante fator de sucesso para o desenvolvimento de novos produtos (Kettunen, 2003) e a eficiência em fazer o gerenciamento deste conhecimento aprendido é crucial no compartilhamento da experiência (Hanafiah *et al.*, 2020).

O conhecimento em engenharia de *software* é diverso e as proporções são imensas e crescentes. Este cenário amplo e dinâmico dificulta as organizações em rastrear o conteúdo deste conhecimento, onde está localizado dentro da instituição ou quem tem cada conhecimento específico (Rabelo; Conte, 2017). Em organizações, cuja complexidade

tecnológica dos projetos elevam o nível de conhecimento gerado (como o DCTA), torna-se ainda mais importante gerenciar o capital intelectual.

Os desenvolvedores e as lideranças foram identificadas como fatores influenciadores das atividades de KM. Os desenvolvedores devem sentir vontade de compartilhar o conhecimento dentro da organização e as lideranças devem contribuir para criar um ambiente favorável a este compartilhamento (Rabelo; Conte, 2017).

Visto isto, a criação de um Centro no DCTA possibilita que as lideranças na área de desenvolvimento de *software* embarcado, possam exercer um papel agregador, tornando o ambiente propício ao crescimento do conhecimento nesta área de alta complexidade técnica e essencial para o desenvolvimento dos produtos aeroespaciais e de defesa do DCTA. Com a implantação da proposta, em consonância com o preconizado por Kettunen (2003), os dados coletados, as informações geradas e o conhecimento, serão direcionados para a área de *software* embarcado e com acesso aos demais desenvolvedores de todos os projetos do DCTA. Atualmente, com o cenário de equipes autônomas, atuando em diferentes OM do DCTA, essas informações estão espalhadas nas documentações dos projetos específicos, seguindo padrões diversos, definidos por cada equipe ou OM e não há compartilhamento entre os projetos, não configurando assim um gerenciamento do conhecimento.

Segundo Hanafiah *et al.* (2020), gerenciar o conhecimento é ainda mais desafiador em equipes distribuídas. Assim, de forma mais concreta, o Centro de *Software* Embarcado no DCTA faria a integração das equipes que atualmente trabalham autonomamente nesta área. Pelo alto grau de interação com as equipes de *hardware*, alguns integrantes do Centro poderiam atuar deslocados numa estrutura distribuída, conforme a necessidade, mas mantendo a integração com o Centro. Ou seja, gerando produtos de acordo com o preconizado, seja na escolha dos *hardwares*, gerando códigos ou documentando o projeto, mantendo assim uma estrutura que permite o gerenciamento do conhecimento.

Os Centros de Computação da Aeronáutica (CCA) centralizam os trabalhos na área de *softwares* corporativos e simuladores no âmbito do COMAER. Com a estruturação e o futuro amadurecimento do Centro de Desenvolvimento de *Software* do DCTA, este poderá se tornar a referência para o desenvolvimento e manutenção dos *softwares* embarcados, assim como o são os CCA, em suas áreas de atuação. Com isto, o Centro completará as competências em *software* do COMAER, com centros específicos para cada área, com profissionais qualificados e experientes prontos para integrar as equipes dos projetos.

3 CONCLUSÃO

Concluindo, os *softwares* embarcados possuem uma intrínseca relação com o *hardware*, tornando um desafio a portabilidade de um *software* embarcado para outro *hardware*, acarretando uma crescente relação de dependência. A implantação do Centro de Desenvolvimento de *Software* Embarcado visa justamente criar uma estrutura de apoio a decisão de quais *hardwares* utilizar nos diversos projetos, padronizando fabricantes ou mesmo modelos específicos até que haja algum real motivo para migração para outro padrão. Com a portabilidade entre *hardwares*, torna-se possível a reutilização de códigos, reduzindo o tempo de desenvolvimento e de testes, utilizando soluções já homologadas em ambiente real. Este Centro de desenvolvimento proporcionará que os desenvolvedores possam interagir, organizar e gerenciar a produção dos códigos dos *softwares*, promovendo a reutilização.

As equipes de desenvolvimento de *software* embarcado trabalham em ambiente multifuncional e, em organizações, cuja complexidade tecnológica dos projetos elevam o nível de conhecimento gerado (como o DCTA), torna-se ainda mais importante gerenciar o capital intelectual. O conhecimento em engenharia de *software* é diverso e as proporções são imensas e crescentes. Isto dificulta as organizações em rastrear o conteúdo deste conhecimento, onde está ou quem o tem. Nesse contexto, a implantação do Centro visa criar a oportunidade para que as lideranças possam gerar um ambiente favorável, para que as experiências dos desenvolvedores possam ser armazenadas e compartilhadas, o que atualmente não é possível com o cenário de equipes autônomas atuando em diferentes OM do DCTA.

Diante do exposto, este ensaio defende a implantação de um Centro de Desenvolvimento de *Software* Embarcado no DCTA, com objetivo de proporcionar um gerenciamento do conhecimento, gerando equipes mais experientes e produtivas para desenvolver os *softwares* que embarcam as soluções científico-tecnológicas e inovadoras. No futuro, este Centro poderá se tornar uma referência para soluções de *software* embarcado, integrando os desenvolvedores, a fim de atender às demandas de interesse do COMAER, assim como os CCA o fazem, para *softwares* corporativos e simuladores.

REFERÊNCIAS

AL MARUF, Md; AZIM, Akramul; ALAM, Omar. Facilitating reuse of functions in embedded software. **IEEE Access**, [s. l.], v. 10, p. 88595-88605, 8 Aug. 2022. Disponível em: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9852432>. Acesso em: 24 mar. 2025.

DULLEMOND, Kevin; VAN SOLINGEN, Rini. Increasing awareness in distributed software teams: A first evaluation. *In: INTERNATIONAL CONFERENCE ON COLLABORATIVE COMPUTING: NETWORKING, APPLICATIONS AND WORKSHARING*, 9., 2013, Austin. **Proceedings** [...]. Austin: Institute of Electrical and Electronics Engineers, p. 325-334, 2013. Disponível em: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=fe5af608bdf5e9411c8a2103652d624f74929932>. Acesso em: 9 abr. 2025.

HANAFIAH, Mastura; ABDULLAH, Rusli; MURAD, Masrah Azrifah Azmi; DIN, Jamilah. The development and evaluation of experience-based factory model for software development process. **International Journal on Advanced Science, Engineering and Information Technology**, [s. l.], v. 10, n. 3, p. 1016-1024, June 2020. Disponível em: <https://core.ac.uk/download/pdf/325990424.pdf>. Acesso em: 20 mar. 2025.

KAMBE, Hidetoshi; KITAGAMI, Shinji; SAWAMOTO, Jun; MITSUI, Hiroyasu; KOIZUMI, Hisao. A Method for Analyzing and Visualizing Intermodule Relations to Support the Reuse-Based Embedded Software Development. **Electronics and Communications in Japan**, [s. l.], v. 100, n. 7, p. 18-31, 19 June 2017. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ecj.11962>. Acesso em: 24 mar. 2025.

KETTUNEN, Petri. Managing embedded software project team knowledge. **IEE Proceedings-Software**, [s. l.], v. 150, n. 6, p. 359-366, 17 Dec. 2003. Disponível em: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=07a2c058fdbbc8fde5393c5254e751015b79c64e>. Acesso em: 17 mar. 2025.

MARTINS GOMES, Renata; AICHERNIG, Bernhard; BAUNACH, Marcel. A framework for embedded software portability and verification: from formal models to low-level code. **Software and Systems Modeling**, Berlin, v. 23, n. 2, p. 289-315, 1 Feb. 2024. Disponível em: <https://link.springer.com/content/pdf/10.1007/s10270-023-01144-y.pdf>. Acesso em: 24 mar. 2025.

OSHANA, Robert; KRAELING, Mark. **Software engineering for embedded systems: Methods, practical techniques, and applications**. 2nd ed. Cambridge: Newnes, 2019. cap. 1, p. 7-8. ISBN: 978-0-12-809448-8. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=g6aeDwAAQBAJ&oi=fnd&pg=PP1&dq=%22Software+Engineering+for+Embedded+Systems%22&ots=8HRx1kYuRo&sig=x-CzNz8NuOkDuHSGJ0veSz4dLMw#v=onepage&q=%22Software%20Engineering%20for%20Embedded%20Systems%22&f=false>. Acesso em: 28 mar. 2025.

RABELO, Jacilane; CONTE, Tayana. IFactor-KM: A Process for Supporting Knowledge Management Initiatives in Software Organizations Considering Influencing Factors. *In: INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS*, 19., 2017, Porto. **Proceedings** [...]. Porto: Science and Technology Publications, v. 2, p. 166-177, 2017. Disponível em: <https://www.scitepress.org/papers/2017/63607/63607.pdf>. Acesso em: 24 mar. 2025.