

Integration of the Waterfall Model with ISO/IEC/IEEE 29148:2018 for the Development of Military Defense System

R. R. J. Jardim, M. Santos, E. Neto, E. da Silva and F. de Barros

Abstract—Software development for military organizations follows standards and a set of guidelines established in the requirements engineering domain. However, these procedures are not always documented and it is difficult to reach understanding among stakeholders with varied backgrounds and interests. This article aims to obtain lessons learned through a case study of the development of a defense system for the Armed Forces of Brazil. From the validation of requirements specialist, a software development lifecycle was obtained that integrates the waterfall model and ISO/IEC/IEEE 29148 recommendations. The use of the model obtained facilitated the understanding of the processes that involved military teams, detailed specification of system requirements and success in the geoinformation based prototype.

Index Terms—Defense System, Geoinformation, ISO/IEC/IEEE 29148:2018, Military, Software Development Life Cycle.

I. INTRODUÇÃO

Inicialmente o desenvolvimento de software consistia de um programador escrevendo um código para resolver um problema ou automatizar um processo. Porém, atualmente, os sistemas são tão complexos que demandam equipes de analistas, arquitetos, desenvolvedores e testadores para atenderem às necessidades do cliente, constituindo mais uma equipe de *stakeholder*.

Para atender essas necessidades, a área de engenharia de requisitos apoia-se em modelos de Ciclo de Vida de Desenvolvimento de Software (*Software Development Life Cycle* - SDLC) para realizar a produção de um sistema tecnológico dentro do escopo, custo e prazo esperados [1].

Um modelo de SDLC é uma caracterização de como o software deve ser desenvolvido. De acordo com [2], todos modelos SDLC possuem um mesmo arquétipo, compreendendo as fases e a ordem em que são executados, constituindo: requisitos, design, desenvolvimento e teste, em

que cada fase do ciclo produz resultados que servirão de insumos para a próxima. Em requisitos, é realizada uma análise das necessidades do cliente; em design, é elaborada a especificação técnica da solução; no desenvolvimento, a proposta é constituída em um software; e na etapa de teste, é verificado se o entregável está em conformidade com os requisitos especificados.

Alguns modelos estão bastante disseminados, como o modelo-v, espiral, Agile e *waterfall* [3] [4], inclusive a análise de SDLC em organizações militares é tema de estudo em muitas pesquisas [5] [6]. Outros trabalhos destacam a importância de se aprofundar a análise do processo de desenvolvimento de software visando sua otimização [7]. O modelo *waterfall* é mais conhecido, no qual uma sequência de estágio gera uma saída para o próximo.

Frequentemente as organizações militares, a saber marinha, exército e aeronáutica, necessitam de softwares específicos para atenderem suas necessidades únicas, tornando difícil encontrar no mercado sistemas prontos, por isso, o desenvolvimento de softwares para as Forças Armadas (FFAA) do Brasil é um desafio. Trata-se de um setor peculiar que necessita de sistemas de defesa próprios [8].

De forma geral, as FFAA, operam de forma independentes, atentas às ordens do Ministério da Defesa (MD). Foi instituído o projeto Sistema de Geoinformação de Defesa (SisGEODEF), responsável pela integração da geoinformação de cada FFAA em um sistema de defesa [9]. A natureza desse projeto requereu a formação de equipes militares com experiência em operações com geoinformação de cada uma das FFAA.

Consultando a experiência dos militares das FFAA em projetos anteriores, confirmou-se a tradicional utilização do modelo *waterfall* para o desenvolvimento de software e apoio na ISO/IEC/IEEE 29148 [10]. Portanto, o objetivo desta pesquisa é obter lições aprendidas junto as organizações militares por meio do estudo de caso do desenvolvimento de um sistema de defesa baseado em geoinformação voltado ao MD e as FFAA.

O escopo deste artigo abrange os domínios das informações geográficas aplicadas aos interesses do MD, requeridas no planejamento, execução e controle de operações militares, sejam elas conjuntas, singulares, missões de Garantia da Lei e Ordem ou de ações subsidiárias [11].

Este trabalho teve o apoio do Ministério da Defesa do Brasil e do Centro de Análises de Sistemas Navais (CASNAV).

Rafael R. J. Jardim, Universidade Federal do Rio de Janeiro e Marinha do Brasil (rafaelrisala@ufrj.br).

Marcos Santos, Universidade Federal Fluminense e Marinha do Brasil (marcosdossantos_doutorado_uff@yahoo.com.br).

Edgard Neto, Instituto Militar de Engenharia e Marinha do Brasil (edgard@marinha.mil.br).

Eliseu D. da Silva, Ministério da Defesa (eliseu.silva@defesa.gov.br).

Frederico C. M. de Barros, Universidade Estadual do Rio de Janeiro e Ministério da Defesa (muthz.barros@defesa.gov.br).

II. ABORDAGEM DO PROBLEMA

A extração de requisitos dos *stakeholders* é algo complexo e muito estudada. A pesquisa de [12] sugere abordar o problema sob diferentes perspectivas: dos requisitos, do contexto e do comportamento. Essa complexidade aumenta no contexto militar em que há: curto prazo de produção para longo período de vigência; especificações voláteis de requisitos; dificuldade de testes; ambiente extremo de operação; e padrões para documentação impostos pelo cliente [13], principalmente no caso de se envolver todas as FFAA em um mesmo projeto. O analista de requisitos precisa estar familiarizado com a cultura militar e conhecer previamente suas práticas habituais a fim de facilitar a comunicação para operar juntamente com essas equipes.

Apesar de seguirem uma cultura militar, cada FFAA possui suas particularidades e diretrizes, o que acarreta em dificuldade do entendimento entre os membros para a definição de funcionalidades e extração de requisitos para um sistema sinérgico entre elas.

Até recentemente, todas as FFAA do Brasil manejavam geoinformação de forma particular e independente, o que aumenta o desafio da integração e automatização de processos executados em sistemas antigos ou manuais.

Conforme apontado por [14] cada modelo de SDLC possuem características típicas, obtendo maior aproveitamento de seu uso de acordo com o tipo de projeto e sua adoção deve levar em consideração o contexto organizacional [15]. O modelo *waterfall* é tradicionalmente adotado no âmbito militar desde sua consolidação pelo Departamento de Defesa dos Estados Unidos [16].

Algumas normas, como a ISO/IEC/IEEE 29148, orientam o trabalho do analista de requisitos a fim de guiar o entendimento entre os desenvolvedores e patrocinadores.

Sendo assim, objetiva-se identificar como as FFAA operam o desenvolvimento de software e normas relacionadas a fim de ser beneficiar outras organizações militares e futuros projetos de concepção e desenvolvimento de sistemas de defesa.

III. FUNDAMENTAÇÃO TEÓRICA

A. Organizações Militares

A melhoria da expressão de requisitos para o desenvolvimento de sistemas de defesa tem sido foco de interesse, tanto pela comunidade acadêmica quanto pelas organizações militares. Várias metodologias têm sido estudadas e aplicadas nesse cenário [17] [18] [19].

A integração de sistema em ambiente militar também é foco de estudo [20]. Um problema recorrente que engenheiros enfrentam é “unir suas metodologias e ferramentas selecionadas para fornecer um ambiente de engenharia de sistemas integrado que suporte todo o processo de desenvolvimento de sistemas” [21].

Arelado ao desenvolvimento tecnológico está a inclusão de dados geoespaciais como informação privilegiada, sendo utilizada pela primeira vez em 1983 em Granada [22]. A partir daí, geoinformação passou a ser de grande interesse para as operações militares para vários países.

B. Modelo Waterfall

Um SDLC é uma ferramenta de apoio para a produção de software, visando o rigor do escopo, custo e prazo em conformidade com o gerenciamento do projeto [23]. O SDLC contempla um plano para desenvolver, alterar, manter e substituir um sistema de software. Sua adequada utilização visa garantir a qualidade geral da produção do sistema e permite eliminar trabalhos redundantes, antecipar erros dispendiosos em fases posteriores e facilitar a comunicação entre todas os *stakeholders* [24].

Para se alcançar o sucesso do projeto, segundo [25], “*Entire project team should have a unified understanding of the entire product that they have to produce. They should have a clear view of requirements, rationale behind requirement changes, architecture, and interface specifications*”, corroborando a necessidade de se ter processos bem definidos e explícitos a todos *stakeholders*.

Um dos SDLC mais tradicionais é o modelo *waterfall*, que estipula que o desenvolvimento de software é constituído em etapas [26], apesar de não ter utilizado este termo na ocasião de sua definição. Desde que o US DoD [16] consolidou sua abordagem na norma DOD-STD-2167A, esse modelo é amplamente utilizado por organizações militares, além da indústria de tecnologia em geral, estabelecendo seis etapas do ciclo de desenvolvimento de software. A Fig. 1 apresenta o modelo *waterfall* possa apresentar variações dependendo do autor.

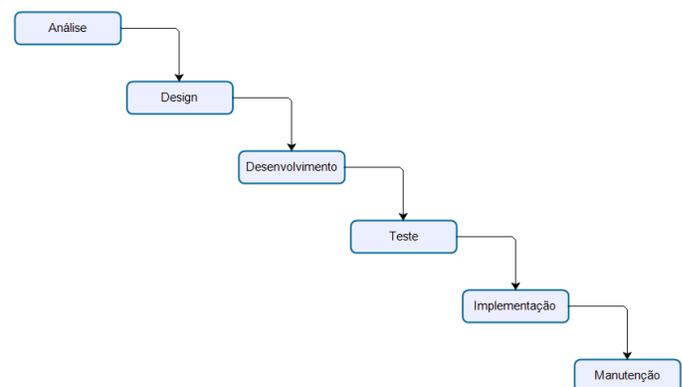


Fig. 1. Modelo *waterfall* de desenvolvimento de software, adaptado de [14].

A etapa de Análise captura a visão do negócio e a fase de Design converte-a para uma especificação técnica. Nessas etapas, constrói-se uma representação do futuro sistema, que é codificado no Desenvolvimento. Na fase de Teste, é conferido se as funcionalidades desenvolvidas estão de acordo com os requisitos especificados. Uma vez aprovado, a funcionalidade é Implementada em servidor de produção e, em seguida, entra em modo de Manutenção, a fim de atender possíveis alterações do software.

C. Norma ISO/IEC/IEEE 29148:2018

Conforme apontado na pesquisa [27] “os processos estabelecidos na ISO/IEC/IEEE 29148 ajudam a garantir exata classificação dos requisitos, validação dos requisitos, identificação dos *stakeholders* e controle de documentos de

evidências”, a fim de atender o desenvolvimento de um sistema de gerenciamento de requisitos. Essa pesquisa corrobora as vantagens na utilização desta norma como parte da análise de requisitos no desenvolvimento de sistemas.

Em Engenharia de Requisitos, a norma “*ISO/IEC/IEEE 29148:2018 - Systems and Software Engineering - Life Cycle Processes - Requirements Engineering*” [10] aborda processos e produtos que auxiliam o desenvolvimento de sistemas. Essa norma orienta a prática da análise de requisitos, como em atividades de levantamento, análise e especificação da solução, prévios e necessários ao desenvolvimento do software. Assim, ela visa atender às necessidades do cliente, evitando-se erros que impactariam na qualidade de seu desenvolvimento.

Assim, para o desenvolvimento adequado de software, a ISO/IEC/IEEE 29148 recomenda a elaboração da documentação técnica apresentada na Tabela I.

TABELA I

DESCRIBÇÃO DOS DOCUMENTOS SUGERIDOS PELA ISO/IEC/IEEE 29148.

Documento	Conceito
ConOps (<i>Concept of Operations</i>)	Apresenta a concepção da organização por seus líderes, ou seja, descreve o nível organizacional como a gerência imagina sua operação considerando os sistemas existentes e aquele que será desenvolvido.
OpsCon (<i>Operational Concept</i>)	É um documento que descreve as características que o sistema deve atender, de acordo com a perspectiva do usuário.
StRS (<i>Stakeholder Requirements Specification</i>)	Contém a motivação da organização para alterar ou desenvolver o sistema.
SyRS (<i>System Requirements Specification</i>)	Apresenta características que o sistema deve possuir a fim de satisfazer as necessidades dos <i>stakeholders</i> .
SRS (<i>Software Requirements Specification</i>)	É o documento mais específico e abrange informações técnicas referentes ao software e não ao sistema como um todo.

A ISO/IEC/IEEE 29148 menciona a possibilidade de agregação a um processo de desenvolvimento de software, mas não deixa claro como se dá sua integração a um modelo específico. Sendo esta determinação o propósito deste artigo, conforme mencionado anteriormente.

D. Trabalhos Relacionados

Foi pesquisado nas bases de publicações científicas SCOPUS, IEEE e ACM, trabalhos relacionados ao desta publicação. Alguns trabalhos esbarram em temática similar. Suas semelhanças e diferenças são destacados a seguir.

Os seguintes trabalhos abordam o desenvolvimento de sistemas de defesa: [13] reconhece os problemas no desenvolvimento de sistemas militares e propõe medidas técnicas e administrativas; [28] foca na avaliação quantitativa de softwares militares; e [29] sugerem que o teste de software pode ser aplicado em todas as etapas do SDLC. Entretanto, esses trabalhos não exploraram seus processos de

desenvolvimento, nem demonstram relação com a ISO/IEC/IEEE 29148.

[30] propõem que os requisitos de segurança do software sejam obtidos a partir da ideia completa do sistema e se basearam no modelo-v. Esta proposta entende que é da natureza das organizações militares revisitarem as funcionalidades idealizadas para um sistema de defesa, preocupa-se com o processo como um todo e baseia-se no modelo *waterfall*.

[31] reconhecem a efetiva vantagem da integração de sistemas, principalmente no setor militar. Essa ideia é totalmente aderente à proposta do presente artigo, visto que o desenvolvimento do protótipo integra a geoinformação das FFAA brasileiras. Apesar de não proporem um SDLC, este artigo o faz.

[32] abordam um projeto baseado no modelo *waterfall* e atentos à ISO/IEC/IEEE 29148. Entretanto não atende às especificidades das forças militares nem detalharam seus processos.

[33] reconhecem que requisitos mal especificados podem ter consequências custosas ao longo do SDLC e propõem aplicar técnicas para detectá-los. O presente trabalho propõe efetivamente a integração das referências e considera o contexto militar.

Não há publicações interligando organizações militares com a ISO/IEC/IEEE 29148 e os processos de desenvolvimento de sistemas de defesa, o que atesta a contribuição deste trabalho.

IV. METODOLOGIA

A construção desta metodologia segue as orientações de [34] para a realização do estudo de caso de engenharia de software, que é exposta a seguir.

A. Design do Estudo de Caso

O objetivo deste estudo de caso em engenharia de software é determinar como funciona na realidade o processo de desenvolvimento de sistemas de defesa nas FFAA brasileiras.

A fim de atender as necessidades de governança de dados geoespaciais no interesse da segurança geográfica do Brasil, o MD estabeleceu o projeto SisGEODEF no qual promoveu o desenvolvimento de um sistema tecnológico capaz de integrar as funcionalidades e a geoinformação de cada FFAA, estabelecido como o Sistema para Governança da Infraestrutura de Dados Espaciais de Defesa (SisGIDE).

Durante a pesquisa exploratória, foi documentado os processos que envolveram a concepção e desenvolvimento do software nas FFAA, identificado os documentos de referência, e realizado uma comparação com os tradicionais regulamentos da engenharia de software. Ao final, as informações qualitativas foram úteis para a composição e descrição de um SDLC que aderente às FFAA do Brasil.

B. Preparação para Coleta de Dados

Baseado nas experiências em geoinformação da gerência militar do projeto, foi definido sete domínios do conhecimento para dar suporte ao desenvolvimento do SisGIDE, sendo elas: Doutrina, Cartografia, Tecnologia da Informação e

Comunicações, Inteligência, Operações, Meteorologia e Oceanografia e Engenharia.

Essas áreas de conhecimento foram efetivadas em grupos de trabalho formados por especialistas em operações com geoinformação de cada FFAA. Assim, apesar de todos os membros serem militares operadores de geoinformação, cada grupo foi formado por profissionais de formações diversas, com culturas militares, interesses e necessidades distintas.

Foi programado um workshop para se coordenar os esforços com os grupos militares. Também foi agendado reuniões com esses grupos para ocorrerem quinzenalmente por videoconferência e bimestralmente presencial, as quais eram seriam guiadas por questionários e documentadas.

C. Coleta de Evidências

Os analistas de requisitos entrevistaram membros do MD e dos grupos de trabalho documentando o procedimento. Os militares discutiram com os analistas suas necessidades para se trabalhar com informações georreferenciadas em consonância com os interesses do MD.

Apoiando-se principalmente na ISO/IEC/IEEE 29148, o analista de requisitos elaborou o ConOps do SisGEODEF, documento que permitiu estabelecer as intenções da gerência em relação a operações globais do SisGIDE. Outros documentos foram consultados nesse momento, como a ISO/IEC/IEEE 25010:2011, Plano de projeto, Estudo de viabilidade e Declaração de escopo.

Por fim, este documento foi encaminhado à gerência do SisGEODEF para que fosse revisado e as sugestões validadas, passando algumas vezes por esse ciclo antes de ser aprovado.

A finalização do ConOps propiciou a elaboração do OpsCon do SisGIDE. O analista de requisitos começou a escrever o OpsCon, tendo principalmente como referência a ISO/IEC/IEEE 29148 e o ConOps. Os militares, que futuramente serão os usuários do sistema, prestaram entrevistas acerca das intenções e expectativas desejadas para o sistema.

Nessa oportunidade, foi detalhada a situação dos processos correntes nos diversos cenários de operação de geoinformação em cada FFAA, identificando carências nos processos a serem preenchidas e estruturando funcionalidades a serem cobertas pelo sistema proposto, além de descrever cenários de operações conjuntas.

Uma vez assinado o ConOps e o OpsCon do SisGIDE, iniciou-se a etapa de design da solução, uma das mais elaboradas. O analista pôde informar-se e estudar acerca do projeto SisGEODEF, visto que os documentos facilitaram o entendimento e coleta de requisitos iniciais para a elaboração do SRS.

À medida que a especificação técnica foi elaborada, o analista contactou as equipes militares para elucidarem detalhes e conferirem aspectos idealizados para o sistema.

Os militares foram as principais referências para o detalhamento e elucidação das necessidades desejadas para o SisGIDE, que ocorreram por meio de reuniões presenciais e videoconferências, sendo essencial para a validação dos requisitos.

A partir dos documentos analisados e entrevistas prestadas, os analistas de requisitos puderam realizar a especificação dos Requisitos Funcionais e Requisitos Não-Funcionais desejados para o SisGIDE, em ferramenta apropriada para a análise de requisitos.

Essa etapa da documentação persiste na ferramenta de requisitos e culmina com a elaboração do SRS, que contém os “Requisitos de Alto Nível”, “Requisitos Funcionais”, “Não-Funcionais”, “Regras de Negócio”, “Casos de Uso” e “Modelos de Interface”, exibidos na Fig. 2.



Fig. 2. Componentes do documento de especificação de requisitos de software (SRS) adotado neste estudo de caso.

Em seguida, o MD validou o SRS e ajustes puderam ser feitos juntamente com o analista.

Então, o analista de requisitos dotado do ConOps, OpsCon e do SRS iniciou uma nova fase de coleta de informações, submetendo os militares a uma nova fase de entrevistas para identificação, principalmente, de Regras de Negócio para as funcionalidades identificadas do sistema proposto.

O analista documenta todo o Caso de Uso na ferramenta de requisitos, detalhando as Regras de Negócio, modelando o Caso de Uso e ilustrando o Modelo de Interface, compondo o SRS.

A etapa de desenvolvimento foi guiada pelos documentos consolidados. Principalmente, o SRS proporcionou aos desenvolvedores a autonomia para o desenvolvimento de forma sequencial, transformando as especificações técnicas em um conjunto de instruções codificadas para o sistema. A cada finalização de uma funcionalidade, os desenvolvedores realizaram teste unitário certificando sua efetividade, passando ao desenvolvimento das próximas funcionalidades. Ao finalizá-las, o sistema passou para a fase de teste.

Durante os testes, o analista validou o sistema em conformidade com o documento de Caso de Teste. Na percepção de funcionalidades deficientes, o sistema era retornado para correção pelos desenvolvedores.

Posteriormente, na fase de implementação, o sistema desenvolvido foi configurado em um servidor apropriado em uma rede segregada de uso privativo para as FFAA e o MD. A Fig. 3 apresenta o sistema desenvolvido.

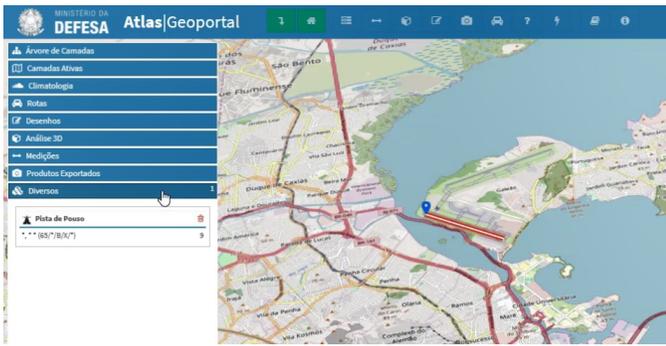


Fig. 3. Protótipo do sistema de defesa obtido no processo de desenvolvimento *waterfall* e da ISO/IEC/IEEE 29148.

Após sua implementação, iniciou-se um período de manutenção das funcionalidades previstas e de ajustes do software. Como foi homologado, o sistema foi finalizado e recebido com sucesso.

D. Análise dos Dados Coletados

Por meio da análise dos processos executados no desenvolvimento do SisGIDE ficou evidente sua alta afinidade com o modelo *waterfall*.

Observou-se nas diretrizes do projeto e no empenho prático dos analistas de requisitos o acatamento à ISO/IEC/IEEE 29148.

Durante todas as fases de desenvolvimento do sistema de defesa, principalmente na análise e design da solução, as especificidades das FFAA brasileiras guiaram a definição das necessidades do projeto e sua conduta.

Coube à gerência do SisGEODEF avaliar a conformidade do SisGIDE às expectativas definidas no ConOps, exploradas no OpsCon e detalhadas no SRS.

E. Relatórios

Os autores projetaram um SDLC capaz de integrar todas as peculiaridades apresentadas no desenvolvimento do protótipo e o submeteram para revisão e validação de analistas de requisitos militares em outro workshop de geoinformação de defesa. Pôde-se constatar na prática a utilização do modelo *waterfall* com adaptações de orientações da ISO/IEC/IEEE 29148. O modelo obtido foi aprovado pelo conjunto de analista de requisitos militares e é apresentado na Fig. 4.

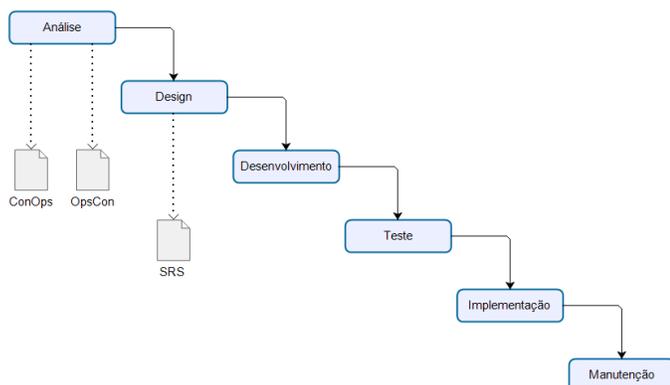


Fig. 4. Ciclo de vida do desenvolvimento de sistemas de defesa em organizações militares baseado no modelo *waterfall* e na ISO/IEC/IEEE 29148.

O SDLC obtido implica na condução de processos de engenharia de requisitos. A definição desses processos fornece as orientações para o planejamento e implementação de sistemas militares.

Em atenção a esse SDLC, a elaboração do ConOps, OpsCon e SRS requisitaram forte interação e cooperação entre os patrocinadores e as equipes militares envolvidas, particularmente em relação aos processos de negócios, práticas organizacionais e avaliação de opções técnicas.

O propósito do ConOps desenvolvido foi expressar o consenso de ideias, alinhamento de desafios e apontamento de possíveis soluções de sistemas militares, entre os patrocinadores, desenvolvedores e apoiadores. Descreveu também as necessidades operacionais, demandas, visões e expectativas dos gerentes, além de como o sistema as atenderia, de forma abrangente e sem abordar o detalhamento técnico para seu desenvolvimento.

À medida que o sistema em desenvolvimento evoluiu em complexidade, os analistas e desenvolvedores tiveram o ConOps como ferramenta de referência para as funcionalidades e para manter uma visão comum a todos os *stakeholders*, ao longo do desenvolvimento do sistema.

Nesse sentido, o ConOps foi tratado como o primeiro passo para se alcançar o desenvolvimento necessário das funcionalidades do sistema como um todo. Foi considerado o elo entre o Plano do Projeto e o SRS do sistema. Seu conteúdo proveu um meio de descrever as necessidades gerenciais, viabilizando o detalhamento necessário para os documentos decorrentes.

Seguindo essa proposição, os conceitos operacionais introduzidos no ConOps foram explorados na perspectiva do usuário por meio do OpsCon. E, por fim, o SRS foi elaborado com criterioso rigor e especificou os interesses dos *stakeholders* em maior grau de detalhamento técnico.

V. DISCUSSÃO

Foram observadas outras opções de SDLC, entretanto, o intuito deste estudo não foi comparar ciclos de vida, mas obter conhecimentos da execução praticada pelas organizações militares no desenvolvimento de software. Confirmou-se que o modelo mais próximo seguido pelas organizações militares foi o *waterfall* com adaptações de recomendações da ISO/IEC/IEEE 29148, culminando no modelo apresentado na Fig. 4.

Em observância ao cenário militar, o SDLC apresentado atende às necessidades do MD, comprovado pela conclusão e aprovação do protótipo.

O ConOps e o OpsCon decorrente são passíveis de atualização durante todo o ciclo de vida de desenvolvimento do sistema e são meios adequados para compartilhar as intenções dos *stakeholders*. Sendo assim, usuários e adquirentes podem debater possíveis soluções a serem avaliadas, negociadas e planejadas antecipadamente. Também permitem que a eventual entrada de novos participantes seja facilitada por meio das instruções detalhadas, organizando os acordos e facilitando a gestão do conhecimento do projeto.

Apesar da ISO/IEC/IEEE 29148 sugerir três documentos de

especificação (StRS, SyRS e SRS), na prática, no âmbito do desenvolvimento de sistema de defesa das FFAA do Brasil, o ConOps e OpsCon elaborados demonstraram suprimir a necessidade dos dois primeiros. Por sua vez, eles foram essenciais para a especificação técnica das funcionalidades no SRS.

Embora não se possa apontar um padrão para o SRS elaborado, uma estrutura bastante usada foi o IEEE/ANSI 830-1993 [35], cujos componentes foram exibidos na Fig. 2.

VI. LIÇÕES APRENDIDAS

A partir do estudo relatado foram identificadas importantes recomendações, traduzidas em lições aprendidas:

- O modelo *waterfall* requer alguma adaptação para atender o desenvolvimento de tecnologias modernas;
- A ISO/IEC/IEEE 29148 apresenta afinidade para sua integração com o modelo *waterfall*;
- A elaboração do ConOps e do OpsCon mostraram-se essenciais. Eles possibilitaram a identificação de particularidades militares e suas necessidades se refletiram no sistema desenvolvido;
- O ConOps refletiu as intenções da organização, de forma ampla. Expressou as características gerais do negócio e do sistema, do ponto de vista gerencial;
- O OpsCon apresentou o desdobramento do conceito operacional. Descreveu as funcionalidades desejadas do protótipo e suas razões, visto que apresentou como as organizações operavam com geoinformação antes do estabelecimento do SisGIDE e comunicou as características quantitativas e qualitativas do sistema;
- A especificação dos requisitos do sistema de defesa no SRS foi facilitada pelos documentos precedentes, ConOps e do OpsCon; e
- A ausência dos documentos StRS e SyRS não pareceu prejudicar o desenvolvimento do sistema, visto que seus objetivos foram contemplados pelo ConOps e OpsCon.

VII. CONCLUSÃO

Embora o modelo *waterfall* e a ISO/IEC/IEEE 29148 estejam bem consolidados, a inovação deste trabalho consiste em:

- Integração de ambas diretrizes em um SDLC;
- Desenvolvimento de um sistema de defesa por meio do SDLC descrito; e
- Contemplação das especificidades das FFAA brasileiras por meio de um sistema de defesa.

O SDLC resultante viabilizou o gerenciamento da documentação apropriada para todo o desenvolvimento do sistema.

Uma vez que o ciclo de desenvolvimento foi explicitado, os diversos *stakeholders* visualizaram sua participação no projeto e perceberam o impacto de seu trabalho.

O ConOps e o OpsCon facilitam que os analistas de requisitos e desenvolvedores entendessem o que deviam construir e sua motivação, fazendo-os mais participativos de

todo o processo de desenvolvimento do software. Estes documentos permitiram aos *stakeholders* contribuírem com os objetivos e as estratégias para alcançá-los juntos.

As diretrizes militares impactam a adoção de novas tecnologias. Espera-se que a explanação bem sucedida do desenvolvimento do software possa corroborar a adoção do SDLC obtido em várias outras organizações militares.

AGRADECIMENTOS

Os autores desta publicação agradecem ao Ministério da Defesa e ao Centro de Análises de Sistemas Navais (CASNAV) pelo apoio e desenvolvimento da tecnologia apresentada.

REFERÊNCIAS

- [1] N. Enger. "Classical and Structured Systems Life Cycle Phases and Documentation" in Cotterman, W., Conger, J, Enger, N. Harold, F. (Ed) System Analysis and Design: A foundation to the 1980s, Elsevier North Holland, NY, 1981.
- [2] P. K. Ragunath, S. Velmourougan, P. Davachelvan, S. Kayalvizhi, and R. Ravimohan. "Evolving a New Model (SDLC Model-2010) for Software Development Life Cycle (SDLC)," International Journal of Computer Science Network Security, Volume 10, no 1, pp. 112-120, 2010.
- [3] A. Alshamrani, A. Bahattab. A comparison between three SDLC models waterfall model, spiral model, and incremental/iterative model. Int. J. Comput. Sci. Issues, 12 (1), p. 106, 2015.
- [4] V. Chandra. Comparison between Various Software Development Methodologies. In International Journal of Computer Applications (Vol. 131, Issue 9), 2015. <https://doi.org/10.5120/ijca2015907294>.
- [5] S. Miller, D. Ward, M. A. Lapham, R. Williams, C. Hammons, D. Burton & A. Schenker. Update 2016: Considerations for Using Agile in DoD Acquisition (CMU/SEI-2016-TN-001). Software Engineering Institute, Carnegie Mellon University, 2016. Disponível em: https://resources.sei.cmu.edu/asset_files/TechnicalNote/2016_004_001_484651.pdf. Acesso em: 13 maio 2020.
- [6] R. Carlson, R. Turner. Review of agile case studies for applicability to aircraft systems integration. Procedia Comput. Sci. 16, 469–474, 2013.
- [7] S. A. Ribeiro, E. A. Schmitz, A. J. S. M. De Alencar, & M. F. Da Silva. Literature Review on the Theory of Constraints Applied in the Software Development Process. IEEE Latin America Transactions, 2018. <https://doi.org/10.1109/TLA.2018.8795116>.
- [8] Ministério da Defesa. Estratégia Nacional de Defesa. Brasília, DF, 2008.
- [9] Ministério da Defesa. Portaria Normativa no 40, de 11 de jul. de 2018. Brasília, DF, 2018.
- [10] ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering, in ISO/IEC/IEEE 29148:2018(E) pp.1-104, 2018.
- [11] Ministério da Defesa. Política Nacional de Defesa. Brasília, DF, 2008.
- [12] L. Lemazurier, V. Chapurlat, A. Grossetête. An MBSE Approach to Pass from Requirements to Functional Architecture, IFAC-PapersOnLine, Volume 50, Issue 1, 2017.
- [13] Kershaw, J. The special problems of military systems. Microprocessors and Microsystems, 17(1), 25–30, 1993. [https://doi.org/10.1016/0141-9331\(93\)90090-T](https://doi.org/10.1016/0141-9331(93)90090-T).
- [14] S. Balaji, M. Murugaiyan. Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. International Journal of Information Technology and Business Management 2(1), 26–30, 2012.
- [15] S. Raza, C. Standing. Towards a systemic model on information systems' adoption using critical systems thinking. Journal of Systems and Information Technology, 12 (3), pp. 196-209, 2010.
- [16] US DoD, Military Standard: Defense System Software Development, DoD-STD-2167A, US Department of Defense, February 29, 1988.
- [17] S. P. Cook, & G. Haverkamp. Challenges and Opportunities for Software Development and Verification on Military Aircraft Systems, 2020. <https://doi.org/10.2514/6.2020-0238>.
- [18] T. Farooqui, T. Rana, and F. Jafari. "Impact of Human-Centered Design Process (HCDP) on Software Development Process," 2nd International Conference on Communication, Computing and Digital systems (C-

- CODE), Islamabad, Pakistan, pp. 110-114, 2019. doi: 10.1109/C-CODE.2019.8680978.
- [19] J. Marques and A. M. da Cunha. "A Set of Requirements for Certification of Airborne Military Software," IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, CA, USA, pp. 1-7, 2019. doi: 10.1109/DASC43569.2019.9081775.
- [20] D. Guan-dong. (The 28th Research Institute of CETC, Nanjing 210007, China); Research on Integration of Military Information System[J]; Command Control & Simulation, 2016.
- [21] N. Hoang, M. Jenkins and N. Karangelen. "Data integration for military systems engineering," Proceedings IEEE Symposium and Workshop on Engineering of Computer-Based Systems, Friedrichshafen, Germany, pp. 11-15, 1996.
- [22] R. H. Cole. Grenada, Panama and Haiti: Joint operational reform. Joint Forces Quarterly Autumn/Winter Edition 57-64, 1998.
- [23] Project Management Institute. A Guide To The Project Management Body Of Knowledge (PMBOK-Guide). Sixth version, Pennsylvania, USA:Project Management Institute, Inc, 2017.
- [24] S. M. Salve, S. N. Samreen, & N. K. Valmik. A Comparative Study of Software Development Life Cycle Models. International Journal of Application or Innovation in Engineering & Management (IJAEM), 5(02), 696-700, 2018.
- [25] R. Jain, U. Suman. A systematic literature review on global software development life cycle. SIGSOFT Softw. Eng. Notes 40(2), 1-14, 2015.
- [26] W. Royce. "Managing the Development of Large Software Systems", Proceedings of IEEE WESCON, 26 (August): 1-9, 1970.
- [27] H. R. Martins and J. Fahy. "Development of an Automotive Regulatory Requirements Management System Based on ISO/IEC/IEEE 29148," 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, pp. 0939-0944, 2018.
- [28] J. J. McGarry & C. L. Jones. Application of a quantitative software metrics assessment process to military software development programs. Electro International, Conference Proceedings, 239-250, 1994. <https://doi.org/10.1109/electr.1994.472701>.
- [29] C. Bayrak, M. Sahinoglu, & T. Cummings. High assurance software testing in business and DoD. Proceedings of IEEE International Symposium on High Assurance Systems Engineering, 207-211, 2000. <https://doi.org/10.1109/HASE.2000.895463>.
- [30] J. Li & B. Chen. Military software safety engineering. Journal of Interdisciplinary Mathematics, 21(5), 1133-1137, 2018. <https://doi.org/10.1080/09720502.2018.1493042>.
- [31] B. L. Summers. Effective methods for software and systems integration. In Effective Methods for Software and Systems Integration, 2016. <https://doi.org/10.1201/b12149>.
- [32] Á. Gómez, L. M. Aristizábal, C. A. Zuluaga, J. C. Correa, & R. E. Vásquez. Development and Implementation of a High-Level Control System for the Underwater Remotely Operated Vehicle VISOR3. IFAC-PapersOnLine, 50(1), 1151-1156, 2017. <https://doi.org/10.1016/j.ifacol.2017.08.400>.
- [33] H. Femmer, D. M. Fernández, E. Juergens, M. Klose, I. Zimmer, & J. Zimmer. Rapid requirements checks with requirements smells: Two case studies. 1st International Workshop on Rapid Continuous Software Engineering, RCoSE 2014 - Proceedings, 10-19, 2014. <https://doi.org/10.1145/2593812.2593817>.
- [34] P. Runeson, M. Höst. Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering, 14, pp. 131-164, 2009.
- [35] IEEE/ANSI. Recommended Practice for Software Requirements Specifications, International Standard 830-1993, IEEE, 1993.



Rafael Ris-Ala José Jardim, mestrando em Ciência da Computação na Universidade Federal do Rio de Janeiro (UFRJ).

Atuou como Gerente de Projetos de Infraestrutura da PUC-Rio, responsável pela criação do Data Center, gestão de infraestrutura e criação de Política de Segurança. Como professor, ministrou aulas de Gerenciamento de Projetos na UFRJ e de Data Science na UNISUAM. Na Marinha do Brasil, possui 7 anos

de experiência, como analista de sistemas. Atualmente atua como analista de requisitos do SisGEODEF.



Marcos Santos, professor do Programa de Pós-graduação em Sistemas e Computação (PPgSC) do Instituto Militar de Engenharia (IME). Pesquisador de Pós-doutorado em Ciências e Tecnologias Espaciais do Instituto Tecnológico da Aeronáutica (ITA). Pós-doutor e Doutor em Engenharia de Produção pela

Universidade Federal Fluminense (UFF) em Sistemas, Logística e Apoio à Decisão. Mestre em Engenharia de Produção pela COPPE/UFRJ em Pesquisa Operacional. Oficial de carreira com 27 anos de serviço na Marinha do Brasil onde desempenha a função de Gerente de Projetos e Pesquisador no Centro de Análises de Sistemas Navais (CASNAV). É professor da graduação do IME e do SENAI CETIQT. Membro da Diretoria da Sociedade Brasileira de Pesquisa Operacional (SOBRAPO). Possui uma produção acadêmica composta por livros, artigos científicos publicados em periódicos e trabalhos publicados em anais de eventos nacionais/internacionais. Faz parte do Comitê Avaliador da Revista Gestão em Análise (ReGA), Revista Pesquisa Operacional para o Desenvolvimento (PODes) e da Revista Gestão da Produção Operações e Sistemas (GEPROS). Referee dos congressos SBPO, ENEGEP, CNEG, EMEMPRO, SIMEPRO, SIMEP, SIMPEP, SEGeT, ENCEPRO, ENFEPRO e CONTEXMOD.



Edgard Candido de Oliveira Neto, possui mestrado em Engenharia de Sistemas - Informática pelo Instituto Militar de Engenharia (1988).

Vasta experiência profissional em banco de dados. Comandante de Mar e Guerra e Gerente de Projetos do Centro de Análises de Sistemas Navais e Gerente de Sistemas do Ministério da Defesa - Comando da Marinha.



Eliseu Dias da Silva, possui graduação em Matemática pela Universidade Federal do Rio de Janeiro (1997) e especialização em Especialização em Geoprocessamento pela Universidade de Brasília (2006). Atualmente é Tenente Coronel da Força Aérea Brasileira e Tenente Coronel do Ministério da Defesa. Tem experiência na área de Geociências, com ênfase em Geodésia.



Frederico C. Muthz M. Barros possui graduação em Ciências Náuticas pela Escola Naval (1993) e mestrado em Geomática pela Universidade do Estado do Rio de Janeiro (2006). Atualmente é oficial de marinha - posto Capitão-de-Mar-e-Guerra - Marinha do Brasil. Tem experiência na área de Geociências, com ênfase em Cartografia Hidrográfica, Geoinformação e Aerolevanteamento.