

XI CONGRESSO DE INICIAÇÃO CIENTÍFICA DO IFSP ITAPETININGA

Itapetininga, 27, 28 e 29 de maio de 2025

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Campus Itapetininga

ANÁLISE DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE BASEADO NO MODELO ESPIRAL EM UM PROJETO DE INOVAÇÃO

Mariana Ferraz Marcelino – ITI/IFSP¹
Prof. Dr. Rodrigo Palucci Pantoni - IFSP²
Prof. Dr. Eduardo André Mossin - IFSP³

Introdução

Um modelo de processo de desenvolvimento de software define as etapas e abordagens utilizadas para planejar, projetar, implementar, testar e manter um sistema. Ele orienta o desenvolvimento, ajudando a gerenciar requisitos, riscos e qualidade do software. Entre os modelos clássicos, que são amplamente estudados na Engenharia de Software, destacam-se os Modelos Cascata, Prototipação e o Espiral.

Segundo Morais (2017), o Modelo Cascata segue uma abordagem sequencial, exigindo que as fases sejam concluídas antes do avanço para a próxima. É eficaz para projetos bem definidos, mas apresenta dificuldades quando há mudanças nos requisitos. Já o Modelo de Prototipação, conforme exposto por Pfleeger (2004), permite o desenvolvimento de um protótipo para esclarecer requisitos, refinando-o com base no retorno dos usuários. No entanto, o protótipo é descartado ao final do processo. O Modelo Espiral, proposto por Boehm (1988), combina a estrutura do Modelo Cascata com a flexibilidade da Prototipação, organizando o desenvolvimento em ciclos iterativos que incluem planejamento, prototipagem e análise de riscos. É ideal para projetos complexos ou sujeitos a mudanças.

Cada modelo apresenta vantagens e limitações, e a escolha deve considerar os requisitos do projeto, a complexidade, a experiência da equipe e os riscos envolvidos. Independentemente do modelo adotado, é essencial garantir a qualidade do software e atender às expectativas dos usuários dentro dos prazos e orçamentos estabelecidos.

Objetivos

O objetivo geral deste trabalho é relatar e analisar o processo de desenvolvimento de software baseado no modelo espiral em um projeto inovador, desenvolvido colaborativamente entre o IFSP e a empresa Nova Smar S/A. O objetivo geral é composto pelos seguintes objetivos específicos: 1 - Descrever brevemente o projeto, os requisitos, o processo de desenvolvimento aplicado e como estão estruturadas a equipe e as instituições participantes; e 2 - Relacionar e analisar os requisitos com as etapas do processo de

¹Estudante do Curso de Engenharia Elétrica, IFSP - Sertãozinho/SP. E-mail do autor: marianaferrazm@outlook.com. Bolsa Captada Externamente (BCE) - APPDI - PD&I Nº 07/2024 - IFSP e Nova Smar S/A. ORCID: <https://orcid.org/0009-0004-4882-3994>

²Doutor. IFSP – Sertãozinho/SP. E-mail do autor: rpantoni@ifsp.edu.br. ORCID: <https://orcid.org/0000-0001-8644-7118>

³Doutor. IFSP – Sertãozinho/SP. E-mail do autor: emossin@ifsp.edu.br. ORCID: <https://orcid.org/0000-0001-5144-518X>

XI CONGRESSO DE INICIAÇÃO CIENTÍFICA DO IFSP ITAPETININGA

Itapetininga, 27, 28 e 29 de maio de 2025

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Campus Itapetininga

desenvolvimento de software baseado no modelo espiral com abordagem iterativo-incremental.

Metodologia

A metodologia deste trabalho está organizada com a descrição breve do projeto, dos requisitos e de como estão estruturadas a equipe e as instituições participantes (objetivo específico 1).

Os desenvolvedores são quatro alunos bolsistas de iniciação científica, dois docentes pesquisadores e a equipe deste projeto de desenvolvimento na empresa, que conta com alguns engenheiros para análise e discussão das decisões de projeto. A duração do projeto foi de um ano (doze meses).

O contexto deste projeto trata da padronização e adoção de tecnologias nos sistemas de automação industrial. Desde a década de 1980, diversos padrões digitais foram introduzidos, permitindo a comunicação entre dispositivos. No entanto, a ausência de um padrão unificado para a interconexão desses protocolos gerou discussões sobre qual tecnologia deveria prevalecer.

Diante desse desafio, empresas como Nova Smar S/A, Intel e IBM fazem parte do *The Open Group*, responsável pelo desenvolvimento do padrão O-PAS (*Open Process Automation™ Specification*). Esse novo padrão, baseado em avanços da Tecnologia da Informação (TI), define uma arquitetura aberta, segura e interoperável, abrangendo áreas como inteligência artificial e computação em nuvem (Qamsane et al., 2022). Com isso, o O-PAS™ busca se tornar o “padrão dos padrões” dos sistemas de automação.

Neste sentido, o projeto está contextualizado como parte do desenvolvimento padrão O-PAS. Para mais detalhes do projeto desenvolvido, o trabalho de Pantoni et al. (2024) pode ser consultado. O objetivo do projeto é implementar o Bloco Funcional Definido pelo Usuário (UDFB - *User Defined Function Block*). Esse tipo de Bloco difere-se dos atuais Blocos Padrões, que são previamente especificados pela norma. Em suma, um Bloco Funcional é um programa de software, que recebe entradas, realiza um processamento e gera saídas para o sistema de automação industrial. Os requisitos principais do projeto são divididos em requisitos funcionais (RF) e de qualidade (RQ):

RF1 - Conformidade à especificação O-PAS, que ainda está em desenvolvimento;

RF2 - Tradução de programas em ST (Texto Estruturado) para linguagem C. Compilação do código em C, assim como sua execução;

RF3 - Troca de informações do Servidor OPC-UA para inicialização e execução do Bloco, pois este último faz a mediação do software do Bloco com o sistema de automação industrial em conformidade com o O-PAS;

RQ1 - Executado em containers (docker) com sistema operacional Linux.

O modelo espiral segue quatro etapas principais em cada ciclo (Figura 1). Primeiramente, são determinados os objetivos, alternativas e restrições, onde se definem as metas do ciclo, levantam-se os requisitos e analisam-se possíveis soluções. Em seguida, ocorre a análise e resolução de riscos, etapa em que são identificados riscos técnicos, financeiros e operacionais, podendo incluir a criação de protótipos para validação. Após essa análise, inicia-se o desenvolvimento e validação, implementação e realização de testes, resultando em uma versão funcional. Por fim, há o planejamento do próximo ciclo, no qual se avalia o progresso, fazem-se ajustes necessários e decide-se a continuidade ou reformulação do projeto (Summerville, 2018). Foi acordado entre as instituições que cada

XI CONGRESSO DE INICIAÇÃO CIENTÍFICA DO IFSP ITAPETININGA

Itapetininga, 27, 28 e 29 de maio de 2025

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Câmpus Itapetininga

	haver mudanças futuras.	foram os sinais digitais e analógicos.		11 (ST). Refatoração e testes de regressão.
RF2-Tradução de programas em ST (Texto Estruturado) para linguagem C. Compilação do código em C, assim como sua execução.	Tecnicamente muito complexa. Exige o desenvolvimento de um tradutor de linguagem de programação.	Dois frentes: procura por biblioteca ou rotinas <i>open-source</i> . Estudos de desenvolvimento de compiladores.	Implementação de protótipos usando biblioteca ou rotinas <i>open-source</i> . Estudos para implementação própria.	Validação de uma das duas frentes para seguir. Refatoração e testes de regressão.
RF3-Troca de informações do Servidor OPC-UA para inicialização e execução do Bloco.	Tecnicamente complexa, mas factível. Exige modificações na máquina de estado de threads e no Servidor OPC-UA.	Estudar o protocolo de trocas de mensagens entre um componente existente (FB Engine - desenvolvido e em funcionamento pela empresa) e o Servidor OPC-UA.	Implementar o processo de inicialização de um bloco padrão (existente) e aos poucos modificar o UFDB e o Servidor OPC-UA.	Validação e testes usando um Cliente OPC-UA. Refatoração e testes de regressão.
RQ1 - Executado em containers (docker) com sistema operacional Linux.	Portabilidade do Linux com Windows. Cada container pode executar um ou mais UDFBs.	Desenvolvimento em C no Sistema Linux Desktop, com testes de portabilidade.	Desenvolvimento de um bloco por container primeiramente, para depois analisar mais de um bloco.	Validação, integração e testes no sistema com comunicação entre os containers. Refatoração e testes de regressão.

O risco do RF1 está na especificação do O-PAS, ainda em desenvolvimento. Para mitigar isso, cada tipo de dado do padrão é trabalhado separadamente, começando pelo mais simples. Quando inconsistências são identificadas, a equipe da empresa participa da validação para viabilizar o desenvolvimento.

No RF2, há um desafio técnico no desenvolvimento de um tradutor de ST para C. Com o aval da empresa, foram abertas duas frentes: estudar a criação do tradutor e buscar alternativas *open-source*. A solução Matiec (2003) mostrou-se mais adequada, atendendo todas as necessidades sem exigir modificações no código, tornando desnecessário desenvolver um tradutor próprio.

O RF3 também apresenta um desafio técnico, pois envolve alterações no Servidor OPC-UA, cujo código fonte não está acessível ao IFSP. Assim, testes foram conduzidos em parceria com a empresa para validar a comunicação entre o UFDB e o Servidor OPC-UA, ajustando o cronograma conforme a empresa adaptava seu código.

Em relação ao RQ1, há um risco técnico, pois o software deve rodar em containers Linux, ambiente pouco familiar à equipe. Para mitigar isso, o desenvolvimento será feito diretamente no Linux, com testes de portabilidade para o container. Além disso, cada container precisa suportar múltiplos UDFBs, mas inicialmente foi limitado a um, conforme acordo entre as instituições devido ao risco.

Todos os requisitos passam por avaliação, refinamento, refatoração do protótipo e testes de regressão. A refatoração melhora a estrutura e qualidade do código sem alterar sua funcionalidade (Fowler, 2018), enquanto os testes de regressão garantem que mudanças no código não afetem partes já existentes (Pressman, 2016).

XI CONGRESSO DE INICIAÇÃO CIENTÍFICA DO IFSP ITAPETININGA

Itapetininga, 27, 28 e 29 de maio de 2025

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Câmpus Itapetininga

Conclusão

Portanto, com base nos resultados obtidos, conclui-se que o modelo Espiral é adequado para projetos que envolvem requisitos nebulosos, características comuns em iniciativas inovadoras. Além disso, sua eficácia está diretamente relacionada ao ajuste adequado da frequência das iterações, já que intervalos muito longos podem causar ociosidade, enquanto reuniões excessivas podem prejudicar o fluxo de trabalho.

Entretanto, neste contexto, o modelo Espiral também apresenta desafios. Um deles é a dificuldade em conciliar os horários da equipe com a disponibilidade da empresa, o que pode atrasar decisões e impactar o andamento do projeto.

Agradecimentos

Os autores agradecem à empresa Nova Smar pela colaboração e apoio no desenvolvimento deste projeto (APPDI - PD&I Nº 07/2024 - IFSP e Nova Smar S/A).

Referências

BOEHM, Barry. **A Spiral Model of Software Development and Enhancement**, *IEEE Computer*, Vol. 21, No. 5, 1988, pp. 61-72. Disponível em: <http://dx.doi.org/10.1109/2.59>. Acesso em: 29 mar. 2025.

FOWLER, Martin. **Refactoring: Improving the Design of Existing Code**. 2. ed. Boston: Addison-Wesley Professional, 2018. ISBN 9780134757681.

MATIEC. IEC 61131-3 Compiler. Disponível em: <https://github.com/nucleron/matiec>. Acesso em: 25 mar. 2025.

MORAIS, Izabelly Soares de. **Engenharia de Software**. São Paulo: Pearson Education do Brasil, 2017.

PANTONI, R. P.; MOSSIN, E. A.; HALFELD, V. T. C.; DONAIRES, O. S.; DA SILVA AFONSO, E.; DA SILVEIRA, G. S.; MARCELINO, M. F.; SAMPAIO, P. C.; DE OLIVEIRA SPAGIARI, V. **Design and implementation of O-PAS user-defined function blocks**. *Journal of Electrical Systems and Information Technology*, v. 11, p. 55-1, 2024.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software: uma abordagem profissional**. Tradução: João Eduardo Nóbrega Tortello. 8. ed. Porto Alegre: AMGH, 2016.

PFLEEGER, Shari Lawrence. **Engenharia de software: teoria e prática**. Revisão técnica Ana Regina Cavalcanti da Rocha. Tradução: Dino Franklin. 2. ed. São Paulo: Prentice Hall, 2004.

QAMSANE et al. **Open process automation and digital twin-based performance monitoring of a process manufacturing system**. *IEEE Access*, v. 10, p. 60823-60835, 2022.

SOMMERVILLE, Ian. **Engenharia de software**. Tradução: Luiz Cláudio Queiroz. 10 ed. São Paulo: Pearson Education do Brasil, 2018.