

COMPARATIVO ENTRE MODELOS DE PROCESSO DE SOFTWARE: Cascata, RUP e Ágil

Yasmin P. PASQUA¹; Paulo C. SANTOS²

RESUMO

A escolha adequada do modelo de processo de software impacta diretamente na qualidade, produtividade e adaptabilidade dos projetos. Este artigo realiza uma revisão bibliográfica comparativa entre os modelos Cascata, RUP e Ágil, abordando suas características, vantagens, limitações e contextos de aplicação. São analisados critérios como estrutura, flexibilidade, documentação e participação do cliente. A análise evidencia a tendência de adoção de modelos híbridos, que conciliam controle e adaptabilidade, auxiliando organizações na escolha estratégica mais apropriada ao seu contexto.

Palavras-chave: Processo de Software; Modelo Cascata; RUP; Metodologias Ágeis; Engenharia de Software.

1. INTRODUÇÃO

A definição de um modelo de processo de software é uma etapa estratégica no planejamento e condução de projetos de desenvolvimento. Modelos como Cascata, RUP e Ágil representam abordagens distintas para organizar o ciclo de vida do software, variando em termos de estrutura, documentação, flexibilidade e participação do cliente. A escolha inadequada pode comprometer prazos, qualidade e orçamento.

No cenário atual, onde as demandas por entregas rápidas e adaptáveis crescem, compreender as diferenças entre essas abordagens se tornou essencial. Este trabalho tem como objetivo revisar e comparar os modelos de processo de software Cascata, RUP e Ágil, destacando suas aplicações conforme o perfil do projeto.

2. FUNDAMENTAÇÃO TEÓRICA

O modelo Cascata é sequencial e tradicional, sendo eficaz quando os requisitos são bem definidos desde o início (SOMMERVILLE, 2011). Sua rigidez dificulta adaptações posteriores.

O RUP, segundo Kruchten (2004), adota uma abordagem iterativa e incremental, com disciplinas bem definidas e forte documentação, sendo indicado para projetos de médio e grande

¹Discente da Ciência da Computação, IFSULDEMINAS – Campus Muzambinho. E-mail: yasminppasqua16@gmail.com.

²Orientador, IFSULDEMINAS – Campus Muzambinho. E-mail: paulo.santos@muz.if sulde minas.edu.br.

porte.

Já as metodologias ágeis, como o Scrum, priorizam colaboração, resposta à mudança e entregas frequentes (BECK et al., 2001). Boehm e Turner (2016) enfatizam o equilíbrio entre agilidade e disciplina, sugerindo modelos híbridos como alternativa eficaz.

3. MATERIAL E MÉTODOS

Trata-se de uma revisão bibliográfica baseada em artigos científicos e técnicos publicados entre 2015 e 2024, disponíveis em bases como IEEE Xplore, ACM Digital Library, Scopus e Google Scholar.

Descritores de busca utilizados: ("software process model" AND "comparison"), ("modelo de processo de software" AND "Cascata" OR "RUP" OR "Ágil").

Os critérios de comparação foram: estrutura do processo, flexibilidade, documentação, participação do cliente, controle de qualidade e adaptabilidade.

4. RESULTADOS E DISCUSSÃO

A análise revelou que o modelo Cascata é mais adequado para projetos com requisitos bem definidos e ambientes estáveis, oferecendo controle e previsibilidade, mas apresentando dificuldades de adaptação a mudanças.

O RUP se destaca por sua formalidade e robustez em projetos complexos, proporcionando equilíbrio entre planejamento e iteração, embora demande maior esforço documental e capacitação da equipe.

Por sua vez, o modelo Ágil é altamente flexível, ideal para projetos com requisitos dinâmicos e entregas contínuas, mas pode carecer de controle formal em ambientes regulamentados. Conforme Thamizhiniyan et al. (2023), o modelo Cascata é adequado em ambientes estáveis. O RUP equilibra planejamento e iteração, exigindo capacitação da equipe. Já o Ágil, segundo Yahya e Maidin (2022), é mais eficaz em cenários dinâmicos, mas pode carecer de controle formal.

O uso de modelos híbridos, como propõem Boehm e Turner (2016), combina a disciplina de abordagens tradicionais com a flexibilidade ágil. Isso permite maior alinhamento com os objetivos organizacionais, especialmente em setores que demandam inovação e rastreabilidade.

5. CONCLUSÃO

Os modelos analisados apresentam características distintas que devem ser consideradas conforme o tipo e a complexidade do projeto. O modelo Cascata é eficaz para contextos estáveis, o RUP é indicado para projetos robustos e documentados, enquanto o Ágil atende melhor cenários dinâmicos e colaborativos.

A literatura evidencia a tendência de integração entre modelos, formando abordagens híbridas. Essa combinação permite equilibrar agilidade, controle e qualidade, sendo especialmente útil em organizações que precisam se adaptar sem perder consistência.

Implicações práticas: gestores devem considerar fatores como tipo de cliente, ritmo de mudanças e necessidade de controle ao escolher o modelo.

Trabalhos futuros: recomenda-se explorar o impacto da Inteligência Artificial na automação de fases dos modelos e investigar a eficácia de modelos híbridos em startups de base tecnológica.

REFERÊNCIAS

BECK, Kent et al. Manifesto Ágil: princípios e valores. 2001. Disponível em: <https://agilemanifesto.org/>. Acesso em: 1 jun. 2024.

BOEHM, Barry; TURNER, Richard. *Balancing agility and discipline: a guide for the perplexed*. 2. ed. Boston: Addison-Wesley, 2016.

KRUTCHEN, Philippe. *The Rational Unified Process: an introduction*. 3. ed. Boston: Addison-Wesley, 2004.

PRESSMAN, Roger S. *Engenharia de software*. 8. ed. São Paulo: McGraw Hill, 2016.

SOMMERVILLE, Ian. *Engenharia de software*. 9. ed. São Paulo: Pearson, 2011.

THAMIZHINIYAN, V. et al. Transition from Waterfall to Agile Methodology: an action research study. *IEEE Xplore*, 2023. Disponível em: <https://ieeexplore.ieee.org/document/10021888>. Acesso em: 1 jun. 2024.

WANGENHEIM, Christiane Gresse von et al. Combinação de abordagens ágeis e tradicionais na melhoria de processos de software. *Revista Brasileira de Computação Aplicada*, Passo Fundo, v. 9, n. 2, p. 33–49, jul./dez. 2017. Disponível em: <https://seer.upf.br/index.php/rbca/article/view/7590>. Acesso em: 1 jun. 2024.

YAHYA, Norzariyah; MAIDIN, Siti Sarah. The Waterfall Model with Agile Scrum as the Hybrid Agile Model for the Software Engineering Team. *ResearchGate*, 2022. Disponível em: <https://www.researchgate.net/publication/366370758>. Acesso em: 1 jun. 2024.