

## **Analisando as técnicas ágeis lado a lado: XP, MSF e TDD**

Richard Francis da Cunha Junior<sup>1</sup>; 0000-0001-8479-6840  
Flavio Antonio Pires Luciano<sup>1</sup>; 0000-0001-5448-9831  
Gabriel Souza Alves Nascimento<sup>1</sup>; 0000-0003-0560-8659  
Lucas Cupertino da Silva Nascimento<sup>1</sup>; 0000-0002-8717-2694  
Venicio Siqueira Filho<sup>1</sup>; 0000-0002-8744-5023  
Sirlei Aparecida de Oliviera Bubnoff<sup>1</sup>; 0000-0002-5064-5742

1 – UniFOA, Centro Universitário de Volta Redonda, Volta Redonda, RJ.  
[202110223@unifoa.edu.br](mailto:202110223@unifoa.edu.br)

**Resumo:** Este relato de caso é uma proposta de trabalho gerado na disciplina de Engenharia e Desenvolvimento Ágil de *Software* utilizando a pesquisa como prática pedagógica, para que o discente possa gerar seu próprio conhecimento, bem como extensivo para a equipe. O estudo compara três técnicas ágeis de gestão de projetos: *Extreme Programming* (XP), *Microsoft Solutions Framework* (MSF) e *Test-Driven Development* (TDD). XP destaca-se pela ênfase na colaboração em equipe e práticas como o desenvolvimento em pares, promovendo ciclos rápidos de *feedback*. MSF oferece flexibilidade e pode ser adaptado a diversos tipos de projetos, não apenas de *software*, fornecendo um *framework* mais abrangente. TDD, por sua vez, orienta o desenvolvimento pelo teste, garantindo a qualidade do código ao exigir que os testes sejam escritos antes do código funcional. A comparação revela que, embora todas as técnicas compartilhem princípios ágeis, cada uma tem um foco distinto: XP na colaboração, MSF na flexibilidade e TDD na qualidade do código. A metodologia de utilizada neste trabalho foi a pesquisa bibliográfica em revistas científicas na *internet* e em livros relacionados à temática. O artigo conclui que a escolha da metodologia depende das necessidades específicas do projeto e da equipe, sugerindo uma possível integração dessas abordagens em diferentes contextos de desenvolvimento.

**Palavras-chave:** Metodologia ágil., *Extreme programming*. *Microsoft solutions framework*. *Test driven development*.

## INTRODUÇÃO

O mercado de *software* está em constante evolução, e as empresas precisam cada vez mais buscar por soluções inovadoras e de alta qualidade num tempo cada vez menor. Para enfrentar esse desafio, muitas equipes de desenvolvimento de *software* estão adotando metodologias ágeis, que se concentram em fornecer soluções flexíveis e adaptáveis aos problemas enfrentados pelos clientes e usuários. Neste trabalho, serão abordadas três metodologias ágeis diferentes: *eXtreme Programming* (XP), *Test-Driven Development* (TDD) e *Microsoft Solutions Framework* (MSF). Serão discutidas e analisadas como cada uma dessas metodologias aborda o procedimento para a criação de *software* de forma iterativa e incremental, com foco na entrega contínua de *software* funcional e de alta qualidade. Além disso, será explorado as principais características de cada metodologia e como elas podem ser aplicadas em diferentes contextos de desenvolvimento de *software*.

### 1. METODOLOGIA XP (*eXtreme Programming*)

De acordo com Nunes (2017) a metodologia ágil XP é um método de desenvolvimento de *software* que enfatiza o trabalho entre a equipe de desenvolvimento, a simplicidade do código, a comunicação constante com o cliente e a capacidade de se ajustar à mudança. O XP segue um processo iterativo e incremental, com foco na entrega contínua de *software* funcional e de alta qualidade. É uma metodologia popular em equipes de *software* ágil em todo o mundo.

O autor ainda relata que as principais práticas do XP incluem: programação em pares, onde duas pessoas escrevem o código juntas; teste automatizado, que auxilia na garantia que o *software* funcione corretamente; integração contínua (que permite que o *software* seja testado e integrado à medida que é desenvolvido); *design* limpo, que valoriza a simplicidade do código e facilita a manutenção e desenvolvimento do *software*.

Segundo Nunes (2017) outra prática importante do XP é a comunicação contínua com os clientes, o que permite que a equipe de desenvolvimento entenda melhor as necessidades do cliente e se adapte rapidamente às mudanças no projeto. Para

facilitar essa comunicação, o XP também utiliza histórias de usuários, que são descrições da funcionalidade do *software* do ponto de vista do cliente.

Em suma, a metodologia XP é aplicada por meio de práticas específicas que valorizam a colaboração, a simplicidade do código, a comunicação constante com os clientes e a capacidade de resposta às mudanças. Essas práticas ajudam as equipes de desenvolvimento a entregar *software* de alta qualidade de forma eficiente e adaptável.

## 2. METODOLOGIA TDD (*Test-Driven Development*)

Segundo Borges (2015) o TDD é uma metodologia ágil de desenvolvimento de *software* que enfatiza a escrita de testes automatizados antes de escrever o código do *software*. A seguir estão as etapas básicas para o desenvolvimento de software usando a metodologia TDD, conforme orientações do autor acima:

- **Escrever um teste:** A primeira etapa do TDD é escrever um teste que falhe. Este teste deve validar uma funcionalidade específica que você deseja adicionar ao seu *software*;
- **Executar o teste:** Execute o teste que você acabou de escrever. Como você ainda não escreveu nenhum código, o teste deve falhar;
- **Escrever o código:** Escreva o código que faz o teste passar. Escreva o mínimo possível de código para que o teste passe;
- **Executar todos os testes:** Execute todos os testes que você escreveu até agora para garantir que o código que você acabou de escrever não tenha quebrado nenhum teste existente;
- **Refatorar o código:** Refatore o código que você acabou de escrever para garantir que ele esteja limpo e bem organizado;
- **Repetir:** Repita todo o processo novamente para adicionar a próxima funcionalidade.

## 3. METODOLOGIA MSF (*Microsoft Solutions Framework*)

O MSF conforme descrito por Paschino (2002) é uma metodologia ágil de desenvolvimento de *software* que enfatiza a entrega de *software* com rapidez, qualidade e eficiência. A metodologia MSF se desenvolve em ciclos iterativos e incrementais. Ela foi criada pela *Microsoft* para ajudar a gerenciar projetos de desenvolvimento de *software* baseados em suas tecnologias.

Possuindo três fases principais, sendo elas de planejamento, desenvolvimento e estabilização.

- **Planejamento:** A equipe de desenvolvimento define nesta fase o escopo do projeto, identifica os requisitos e estabelece metas e objetivos. É importante ter em mente que a metodologia MSF enfatiza a personalização e a adaptação às necessidades específicas do projeto;
- **Desenvolvimento:** Nesta fase, a equipe de desenvolvimento trabalha em ciclos iterativos e incrementais para desenvolver o *software*. A equipe deve se comunicar regularmente e garante ainda que os requisitos do projeto sejam atendidos. Além disso, a equipe deve usar ferramentas e processos para garantir a qualidade do *software*;
- **Estabilização:** Por fim, a equipe de desenvolvimento realiza testes, correções de *bugs* e prepara o *software* para lançamento. É importante garantir que ele atenda aos requisitos do projeto e que esteja pronto para ser lançado.

## RELATO DA EXPERIÊNCIA

*Microsoft Solutions Framework* (MSF) foi utilizado para estruturar o ciclo de vida do projeto, incluindo a definição de fases, atividades e entregas.

A equipe adotou o MSF para a organização do trabalho, dividindo o projeto em fases claras: concepção, planejamento, desenvolvimento, estabilização e implementação. Cada fase tinha suas atividades e entregas bem definidas, o que facilitou a gestão de expectativas com os *stakeholders*.

*Extreme Programming* (XP) foi escolhido para promover o desenvolvimento ágil e adaptativo, permitindo entregas incrementais e frequentes de valor para o cliente.

A equipe implementou práticas XP como iterações curtas (de duas semanas), integração contínua, *feedback* constante do cliente e programação em par. Isso garantiu que o *software* estivesse sempre em um estado funcional e pronto para ser entregue.

*Test Driven Development (TDD)* foi adotado para assegurar a qualidade do código e facilitar a manutenção e evolução do *software*.

O desenvolvimento de cada funcionalidade foi conduzido através da escrita de testes unitários antes da implementação do código. Isso resultou em um código mais limpo, modular e com menos defeitos, além de proporcionar uma maior confiança durante as refatorações.

O MSF, XP e TDD são metodologias que se diferem em diversos pontos no desenvolvimento de *software*, no quadro 01 abaixo apresenta-se um comparativo com algumas vantagens e desvantagens de cada uma das técnicas:

Quadro 01: Quadro Comparativo entre as Metodologias

	Metodologias		
	XP	MSF	TDD
Vantagens	<ol style="list-style-type: none"> <li>1. Promove um processo de desenvolvimento de <i>software</i> colaborativo, permitindo que os desenvolvedores trabalhem juntos para entregar <i>software</i> de alta qualidade;</li> <li>2. Ajuda a garantir que o código seja testado desde o início do desenvolvimento, aumentando a qualidade do <i>software</i> e minimizando erros de regressão;</li> <li>3. Ajuda a reduzir o tempo e os custos de desenvolvimento do <i>software</i>, permitindo que as equipes entreguem <i>software</i> em ciclos curtos e frequentes;</li> <li>4. Ajuda a minimizar o número de <i>bugs</i> através do desenvolvimento de testes automatizados e revisões de código regulares.</li> </ol>	<ol style="list-style-type: none"> <li>1. É uma metodologia flexível que pode ser adaptada às necessidades de qualquer projeto;</li> <li>2. Ajuda a garantir que as necessidades do cliente sejam atendidas através do foco na colaboração com o cliente e nos requisitos do cliente;</li> <li>3. Promove um ciclo rápido de <i>feedback</i>, permitindo que os desenvolvedores detectem e corrijam problemas rapidamente;</li> <li>4. Ajuda a reduzir o tempo e os custos de desenvolvimento do <i>software</i>, permitindo que as equipes entreguem <i>software</i> em ciclos curtos e frequentes.</li> </ol>	<ol style="list-style-type: none"> <li>1. Ajuda a garantir que o código seja testado desde o início do desenvolvimento, o que aumenta a qualidade do <i>software</i>;</li> <li>2. Promove um ciclo rápido de <i>feedback</i>, permitindo aos desenvolvedores detectar erros mais rapidamente;</li> <li>3. Ajuda a melhorar a documentação do código e a manutenibilidade do <i>software</i>;</li> <li>4. Promove uma maior confiança no código, uma vez que cada funcionalidade é testada com precisão antes de ser implementada.</li> </ol>
Desvantagens	<ol style="list-style-type: none"> <li>1. Pode ser difícil de implementar em projetos grandes e complexos;</li> <li>2. Requer uma equipe altamente experiente e dedicada para implementar com sucesso;</li> <li>3. Pode ser difícil de entender e implementar para desenvolvedores iniciantes;</li> <li>4. Pode ser mais difícil de integrar com outras metodologias ágeis, como o Scrum.</li> </ol>	<ol style="list-style-type: none"> <li>1. Requer uma equipe experiente em metodologias ágeis para implementar com sucesso;</li> <li>2. Pode ser que em projetos menores, seja menos produtivo ou menos complexo;</li> <li>3. Pode exigir investimento significativo em infraestrutura e ferramentas de <i>software</i>;</li> <li>4. Pode ser menos adequada para projetos com requisitos muito específicos ou técnicos.</li> </ol>	<ol style="list-style-type: none"> <li>1. Em alguns projetos de desenvolvimento de <i>software</i>, pode não ser eficaz.;</li> <li>2. Pode ser difícil para os desenvolvedores iniciantes entenderem como TDD funciona;</li> <li>3. Pode levar mais tempo para escrever código porque é necessário escrever testes antes de implementar a funcionalidade;</li> <li>4. Aumento no tempo de desenvolvimento.</li> </ol>

Conclusão da Equipe	XP enfatiza a colaboração e a comunicação da equipe, e pode ser uma boa opção para projetos com necessidades específicas de <i>software</i> , mas pode ser difícil de implementar em equipes inexperientes ou em projetos com requisitos altamente específicos.	MSF é uma metodologia ágil que enfatiza a personalização e a adaptação às necessidades específicas do projeto, mas pode exigir investimento significativo em ferramentas e infraestrutura de desenvolvimento de <i>software</i> .	TDD é uma abordagem que enfatiza a escrita de testes antes do código, promovendo alta qualidade e redução de erros, mas pode ter uma curva de aprendizado íngreme e ser difícil de implementar em projetos grandes e complexos.
---------------------	---	---	---

Fonte: Desenvolvido pelos autores (2023)

## CONSIDERAÇÕES FINAIS

Não é possível afirmar qual é a melhor metodologia ágil entre *Test Driven Development* (TDD), *Extreme Programming* (XP) e *Microsoft Solutions Framework* (MSF), pois cada uma tem suas vantagens e desvantagens e sua escolha dependerá do contexto e das necessidades específicas do projeto, bem como cada empresa precisa verificar estas ponderações para fazer a escolha adequada dentro da cultura empresarial. Portanto, é importante avaliar cuidadosamente as necessidades do projeto e as habilidades da equipe antes de decidir qual metodologia ágil é a melhor escolha para o seu projeto. Além disso, muitas vezes as metodologias ágeis são combinadas e modificadas de acordo com os requisitos exclusivos do projeto, criando assim uma abordagem personalizada e eficaz.

## REFERÊNCIAS

- BORGES, Eduardo N. **Conceitos e Benefícios do *Test Driven Development* por Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)**. 2015. – Porto Alegre – RS – Brazil Disponível em: [https://www.researchgate.net/profile/Eduardo-Borges-6/publication/268241623\\_Conceitos\\_e\\_Beneficios\\_do\\_Test\\_Driven\\_Development/links/551416540cf283ee0834a480/Conceitos-e-Beneficios-do-Test-Driven-Development.pdf](https://www.researchgate.net/profile/Eduardo-Borges-6/publication/268241623_Conceitos_e_Beneficios_do_Test_Driven_Development/links/551416540cf283ee0834a480/Conceitos-e-Beneficios-do-Test-Driven-Development.pdf). Acesso em: 27 fev. 2023.
- NUNES, Rodrigo Dantas. **A Implantação das metodologias ágeis de desenvolvimento de *software* Scrum e *extreme programming* (XP): uma alternativa para pequenas empresas do setor de tecnologia da informação**. 2017. Disponível em: <http://www.forscience.ifmg.edu.br/forscience/index.php/forscience/article/view/117> Acesso em: 23 fev. 2023.
- PASCHINO, Enzo *et al.* **MFS Project Management Discipline**. 2002. Disponível em: <http://www.download.microsoft.com/download/b>. Acesso em: 07 abr. 2023