

# A utilização de metodologias, práticas e métricas ágeis no processo de desenvolvimento de software na região sul do Brasil: Uma revisão sistemática

Ana Paula Fabrini<sup>1</sup>, André Ricardo Zavan<sup>1</sup>

<sup>1</sup>Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas – Instituto Federal do Paraná (IFPR) – Campus de Paranavaí  
87703-536 – Paranavaí – PR – Brasil

ana\_paulafabrini@hotmail.com, andrezavan@ifpr.edu.br

**Abstract.** *This Paper intend to investigate through a systematic literature review, which of the following methodologies, practices and metrics are supported in agile software development projects in the south region , are used more frequently and the challenges and contributions arising from the use. The methodologies adopted were the bibliographical research and the systematic literature review.*

**Resumo.** *O presente trabalho tem como objetivo investigar por meio de uma revisão sistemática de literatura, quais as metodologias, práticas e métricas adotadas em projetos ágeis de desenvolvimento de software na região sul do Brasil, são utilizados com maior frequência e os desafios e contribuições decorrentes da utilização. As metodologias adotadas foram a pesquisa bibliográfica e a revisão sistemática de literatura.*

## 1. Introdução

Métodos ágeis são uma alternativa à gestão tradicional de projetos. Conforme Chaos Report (Standish Group, 2015) indica, apenas 29% dos projetos de *software* foram bem sucedidos, onde um dos fatores apontados para o sucesso de tais projetos foi a utilização da metodologia ágil para o desenvolvimento e gerenciamento do projeto.

A utilização de ciclos iterativos e entregas incrementais que buscam a entrega rápida e com o *software* funcionando estão presentes em metodologias ágeis como: FDD (*Feature Driven Development*), DSDM (*Dynamic System Development Methodology*), ASD (*Adaptative Software Development*), *Crystal*, XP (*eXtreme Programming*) e Scrum (Somerville, 2011).

Este trabalho tem como foco a identificação entre as diferentes metodologias ágeis adotadas, suas respectivas práticas e métricas que estão sendo utilizados no processo de desenvolvimento de *software*, através da metodologia da revisão sistemática de literatura, apresentando a realização do teste piloto desenvolvido, que tem como objetivo avaliar os critérios e garantir a qualidade da revisão.

## 2. Objetivo

O objetivo geral deste trabalho é investigar por meio de uma revisão sistemática de literatura, quais as metodologias, práticas e métricas adotadas em projetos ágeis de

desenvolvimento de *software* na região sul do Brasil, são utilizados com maior frequência e os desafios e contribuições decorrentes da utilização.

Para viabilizar o alcance do objetivo geral, foram definidos os seguintes objetivos específicos:

- Coletar estudos publicados em anais de congressos, revistas científicas, simpósios, seminários ou dissertações de graduação, especialização, mestrado e doutorado, no período de 2010 a 2016 referentes a região sul do Brasil;
- Estabelecer critérios para a seleção dos artigos coletados;
- Analisar os artigos conforme os critérios de seleção;
- Apresentar a metanálise obtida.

### 3. Justificativa e Motivação

Do ponto de vista acadêmico, o presente trabalho utiliza métodos da engenharia de software empírica, apresentando as metodologias ágeis de desenvolvimento de software, suas práticas e as métricas para medição do desenvolvimento de software. Portanto, este trabalho busca investigar e reunir de forma sistemática, utilizando a revisão sistemática de literatura, as metodologias ágeis, práticas e métricas utilizadas na região sul do Brasil. Permitindo identificar, quais metodologias ágeis, bem como suas práticas e por fim, as métricas utilizadas na região sul e os desafios e benefícios da utilização.

### 4. Referencial Teórico

Nesta seção são apresentados os conceitos de Metodologias ágeis, Práticas e Métricas.

#### 4.1. Metodologias ágeis

Conforme Pressman (2011), o *software* é desenvolvido ou passa por um processo de engenharia, ele não é fabricado no sentido clássico. Portanto, o *software* a ser desenvolvido necessita da adoção de práticas da engenharia de *software*.

Para o desenvolvimento de *software*, é necessário a utilização de métodos e ferramentas durante o seu processo de desenvolvimento, chamado de modelo de processo. O modelo de processo a ser escolhido pela equipe de desenvolvimento, é baseado na natureza do projeto e nas ferramentas a serem utilizadas. Os modelos de processo promovem uma organização nas atividades a serem desenvolvidas.

Nos últimos dez anos, as metodologias ágeis surgem motivadas pela necessidade de buscar alternativas para os modelos tradicionais de desenvolvimento de projetos de *software*. Conforme os princípios do Manifesto ágil (Beck *et al*, 2001), as metodologias ágeis possuem como princípios em comum, a interação e a entrega rápida do *software* funcionando.

Diferente das metodologias tradicionais, que são estruturadas em fases separadas e delimitadas, nas metodologias ágeis as fases de concepção e implementação interagem durante todo o processo de desenvolvimento, promovendo assim uma interação entre os profissionais responsáveis pelo desenvolvimento do *software*.

A seguir serão abordadas as principais metodologias que se enquadram na coleção das metodologias ágeis.

#### 4.1.1 Feature Driven Development

Conforme Palmer e Felsing (2002), essa metodologia define duas fases compostas por cinco processos. A primeira denominada concepção e planejamento é constituída pelos processos de desenvolver um modelo abrangente, construir a lista de funcionalidades e planejar uma funcionalidade. Já a segunda fase, ou seja, a etapa de construção, é composta pelos processos detalhar por funcionalidade e construir por funcionalidade. Os autores apresentam as práticas existentes nesta metodologia, conforme a seguir: Modelo de objetos de domínio, UML em cores, Desenvolvimento por funcionalidades, Posse de código, Time de funcionalidades, Inspeções, Gerência de configuração, Integração constante e Visibilidade de progressos e resultados.

#### 4.1.2. Dynamic System Development Methodology

Dynamic System Development Methodology (DSDM), é uma metodologia de desenvolvimento voltada a ajustar as funcionalidades de acordo com um tempo fixo pré estabelecido para atender o prazo estipulado. Consiste em três fases sequenciais: Pré-projeto, Projeto e Pós-Projeto e que possui cinco fases distintas: estudo da viabilidade, estudo de negócio, modelo de iteração funcional, projeto e construção de iteração e implementação.

Conforme Sbrocco (2012), a estrutura do DSDM baseia-se em princípios ou conjunto de técnicas, que são considerados boas práticas de utilização dessa metodologia, resumidamente são: Participação ativa dos usuários, Abordagem cooperativa, Equipes com poder de decisão, Entregas contínuas, Desenvolvimento iterativo e incremental, *Feedback*, Alterações reversíveis, Fixar os requisitos essenciais, Teste em todo o ciclo de vida, *Timeboxing*, MoSCoW e Prototipagem.

#### 4.1.3. Adaptative Software Development

De acordo com Sbrocco (2012), o ASD, surgiu da necessidade de uma implementação mais rápida, para isso permite dividir o *software* em partes e evita que algumas fases tomem um tempo maior que o necessário. Possui um conjunto de propriedades que caracterizam o processo, conforme Abrahamsson (2002) são: Foco na missão, Desenvolvimento Baseado em componentes, Iteração, Prazos pré-fixados, Tolerância a mudanças e Orientado a riscos.

#### 4.1.4. Crystal

De acordo com Sbrocco (2012), a metodologia *Crystal*, consistem em uma família de metodologias de desenvolvimento para atender diferentes tipos e tamanhos de projetos. Comos os cristais, essa metodologia é dividida em cores, que representam a intensidade, ou seja, quanto mais escura a cor , maior é a complexidade do projeto. Conforme Santos *et. al.*, (2017), essa família de metodologia é baseada nas seguintes práticas: *Staging*, Edição e revisão, Monitoramento, Paralelismo e fluxo, Inspeções de usuários, *Workshops* refletivos, *Work products*, Padrões e Ferramentas.

#### 4.1.5. eXtreme Programming

Conforme Beck (2000), o XP é uma metodologia ágil voltada para equipes pequenas e

médias, que possuem no seu desenvolvimento requisitos vagos ou que estão mudando frequentemente. Possui uma abordagem de desenvolvimento orientado a objetos como paradigma completo. O processo do XP é composto pelas atividades: planejamento, projeto, codificação e testes.

Teles (2014), apresenta as seguintes práticas do XP: Cliente presente, Jogo do planejamento, *Stand Up Meeting*, Programação em par, Desenvolvimento guiado pelos testes, *Refactoring*, Código coletivo, Código padronizado, *Design* simples, Metáfora, Ritmo sustentável, Integração contínua e *Releases* curtos.

#### 4.1.6. Scrum

Conforme Sbrocco (2012), o Scrum estabelece um conjunto de práticas e regras que devem ser cumpridas pela equipe. O Scrum emprega os eventos com duração fixa para criar regularidade, chamados de *Sprints*. Entre os elementos do Scrum que têm duração fixa, temos a *Sprint Planning Meeting* (Reunião de Planejamento da Versão para Entrega), *Sprint*, *Daily Scrum* (Reunião Diária), *Sprint Review Meeting* (Revisão da Sprint) e *Sprint Retrospective* (Retrospectiva da Sprint).

Ao final de uma *Sprint*, o Time apresenta as funcionalidades já implementadas na reunião chamada *Sprint Review Meeting*. Em seguida ocorre o *Sprint Retrospective*, também uma reunião, que visa identificar o que está funcionando corretamente e o que pode ser melhorado. Por fim, encerra-se a *Sprint* e se reinicia o ciclo. O Scrum possui um artefato chamado *Burndown Chart*, que constitui um gráfico utilizado para a representação do progresso de desenvolvimento do trabalho realizado em comparação com o trabalho total planejado.

#### 4.2. Medição e Métricas

Uma das prioridades no desenvolvimento ágil é satisfazer o cliente com entrega de valor (Beck *et al*, 2001). Portanto, a metodologia ágil exige um ciclo de inspeção, adaptação e melhoria. Com a medição possibilita-se o acompanhamento mais preciso das atividades do processo.

Conforme Pressman (2011), medições podem ser aplicadas ao processo de desenvolvimento de *software* com a intenção de melhoria contínua. A medição pode ser usada pelos desenvolvedores como um instrumento de avaliação de qualidade dos artefatos e auxiliar na tomada de decisões durante o desenvolvimento.

### 5. Metodologia

Inicialmente essa pesquisa foi dividida em três etapas: (1) Revisão bibliográfica, (2) Coleta e Seleção dos estudos, e (3) Análise dos estudos selecionados. Na primeira etapa, foi realizada uma revisão bibliográfica sobre os seguintes temas: Engenharia de Software, Metodologias ágeis, Práticas e Métricas, onde neste trabalho foi aplicado os conceitos de revisão sistemática de literatura.

A revisão sistemática de literatura tem como objetivo identificar, selecionar, avaliar, interpretar e sumarizar estudos disponíveis relevantes para um tópico de pesquisa (Felizardo *et al*, 2017). Uma revisão sistemática é conduzida por meio de um processo que envolve três fases: Planejamento, Condução e Publicação dos Resultados.

A fase do Planejamento consiste principalmente na especificação e na avaliação do protocolo (teste piloto) a ser seguido para execução da Revisão Sistemática de Literatura. O protocolo de uma Revisão Sistemática de Literatura envolve a definição das questões de pesquisa, a estratégia de busca, a seleção das fontes de pesquisa e da *string* de busca, dos critérios de inclusão e de exclusão de trabalhos. A próxima e segunda fase, a de Condução, consiste na investigação em fontes definidas na fase anterior. O estudo e a classificação dos trabalhos encontrados devem ser feitos, guiados pelos critérios de inclusão e de exclusão definidos anteriormente. Por fim, a terceira fase é a de Análise dos Resultados, consiste na coleta e na organização dos dados extraídos dos artigos selecionados.

Após as definições das questões de pesquisa, a estratégia de busca, as fontes de pesquisa, a *string* de busca e dos critérios de seleção, deu-se início a avaliação do protocolo, ou seja, o teste piloto, cujo objetivo é verificar a viabilidade da revisão. Conforme Felizardo *et al* (2017), os critérios de seleção devem ser avaliados por meio de um teste piloto, para isso, os critérios de inclusão e exclusão são aplicados. O objetivo é refinar os critérios e treinar os revisores nos procedimentos da revisão.

### 5.1. Questão de Pesquisa

Esse trabalho possui como questões de pesquisa (Q1) Quais são as práticas ágeis, metodologias e métricas utilizados no processo de desenvolvimento de *software*? (Q2) Quais as dificuldades encontradas e contribuições tais metodologias, práticas e métricas utilizadas trazem ao desenvolvimento de *software*?

### 5.2. Busca automática e coleta de estudos

Para a busca automática, foram selecionados mecanismos de indexação relevantes e nacionais. O processo de busca inicia-se com a avaliação das questões de pesquisa e a formulação da *string* de busca. Portanto, a busca automática foi realizada por meio de uma *string* que agrupou os termos referentes ao tema e as questões de pesquisa. A *string* utilizada foi: (“metodologia ágil” OR "agile methodology" AND "práticas" OR "practices" AND "métricas" OR "metrics" AND "paraná" OR "santa catarina" OR "rio grande do sul"). Já a busca manual, foi a pesquisa manual em cada repositório ou biblioteca digital. Os resultados preliminares são apresentados a seguir na tabela abaixo.

**Tabela 1. Resultados da Busca Automática e Manual**

Local Pesquisado	Resultados
Google Acadêmico ( <a href="https://scholar.google.com.br/">https://scholar.google.com.br/</a> )	196
Banco de Teses e Dissertações Capes( <a href="http://capesdw.capes.gov.br/banco-teses/#/">http://capesdw.capes.gov.br/banco-teses/#/</a> )	5
Repositorio da UFRGS ( <a href="http://www.lume.ufrgs.br">http://www.lume.ufrgs.br</a> )	4
Repositório da UFSC ( <a href="https://repositorio.ufsc.br">https://repositorio.ufsc.br</a> )	8
Repositório UTFPR ( <a href="http://repositorio.roca.utfpr.edu.br">http://repositorio.roca.utfpr.edu.br</a> )	4
Biblioteca Digital de Teses e Dissertações PUCRS ( <a href="http://tede2.pucrs.br">http://tede2.pucrs.br</a> )	7
Acervo Digital da UFPR ( <a href="http://acervodigital.ufpr.br/">http://acervodigital.ufpr.br/</a> )	3
Biblioteca Digital UEM ( <a href="http://nou-rau.uem.br/nou-rau/">http://nou-rau.uem.br/nou-rau/</a> )	2
Biblioteca Digital UEL( <a href="http://www.bibliotecadigital.uel.br/teses_dissertacoes.php">http://www.bibliotecadigital.uel.br/teses_dissertacoes.php</a> )	3
<b>Total</b>	<b>232</b>

### 5.3. Critérios e seleção dos estudos

Foi realizada a verificação dos títulos e resumos presentes nos estudos para identificar os estudos relevantes e evitar redundâncias. Quanto a fonte de busca Google Acadêmico, ao utilizar a *string* de busca foram retornados 196 (cento e noventa e seis) resultados, sendo extraídos para análise: título, palavras-chaves e resumo e que após a leitura destes itens, 18 (dezoito) estudos foram selecionados. Ao final, no total foram identificados 54 (cinquenta e quatro) estudos potencialmente relevantes.

O próximo passo foi a aplicação dos critérios de inclusão e exclusão. Os critérios de inclusão foram: (1) Estudos publicados em anais de congressos, revistas científicas, simpósios, seminários ou monografias de graduação, especialização e dissertações de mestrado e doutorado; (2) Atender às questões de pesquisa e ao tema proposto; (3) Deve corresponder ao período delimitado de 2010 até 2016; (4) Devem ser produzidos na região sul do Brasil; Já os critérios de exclusão foram: (1) O estudo já tenha sido coletado por outra fonte; (2) O estudo não atende os requisitos para inclusão. Após a aplicação dos critérios de inclusão e exclusão, 4 (quatro) estudos foram considerados relevantes, descritos no quadro abaixo.

**Quadro 1. Estudos incluídos**

Título	Autor
Métodos ágeis na concepção e manutenção de sistemas ligados à infraestrutura	Fernanda Lanção Morais Martins
DESIGN E PRÁTICAS ÁGEIS: Aplicação de Filosofia e Princípios Ágeis no Desenvolvimento de Modelos Tridimensionais para Jogos Digitais	Alessandro Peixoto de Lima
Aplicação de Métricas de Software com Scrum	Mateus Luiz Gamba, Ana Cláudia Garcia Barbosa
Aprimorando a gerência e o desenvolvimento de software com metodologias ágeis	Mauricio Andrezza Sganderla, Guilherme Lacerda, Vinicius Gadis Ribeiro, Sidnei Renato Silveira

## 6. Resultados Gerais

Foram extraídos dos trabalhos incluídos, o título, o ano da publicação, os autores, o tipo (artigo ou dissertação), a natureza, a metodologia ágil utilizada, as práticas ágeis, as métricas, as possíveis contribuições e dificuldades apresentadas. Entre os 4 (quatro) estudos incluídos estão, 2 (dois) artigos, uma monografia de graduação e uma de especialização. Quanto à natureza todos são denominados estudos de caso.

### 6.1. Metodologias Utilizadas Encontradas

Essa seção apresenta as metodologias ágeis utilizadas de acordo com os trabalhos incluídos. A metodologia ágil mais citada foi a Scrum, citada duas vezes e também a utilização da combinação das metodologias Scrum e XP no mesmo projeto, citado em dois estudos.

## 6.2. Práticas Ágeis Utilizadas Encontradas

As práticas ágeis foram sintetizadas separadamente de acordo com a metodologia da qual as práticas pertencem. Dentre as práticas mais citadas da metodologia Scrum foram *Product Backlog*, *Daily Scrum*, *Sprint Retrospective* e *Sprint Planning*, todas elas citadas duas vezes cada. As práticas *Sprint Review*, *Sprint Planning* e *Sprint Backlog*, foram citadas uma vez em cada trabalho. As práticas pertencentes a metodologia XP citadas nos estudos, foram Programação em Par, citada duas vezes e Refatoração, citada uma vez.

## 6.3. Métricas Utilizadas Encontradas

As métricas foram citadas uma vez cada nos estudos selecionados, sendo as : *Project Velocity* (quanto de *software* o time consegue entregar por iteração), *Ideal Day* (quantidade de trabalho que um profissional consegue concluir em um dia de trabalho), *Planning Poker* (utilizado para estimar valor para um item do *Product Backlog*), Pontos de Função (usado para medir o tamanho funcional do *software*), Custo (Retorno de Investimento - ROI), Escopo (Entrega de Conteúdo), Qualidade (Iterações Testadas e Entregues), Tempo (Tempo de Ciclo) e *Lead Time*.

## 6.4. Desafios e Contribuições Encontrados nos Estudos incluídos

Os benefícios citados nos estudos foram: aumento da confiança e comunicação das equipes com a adoção das metodologias ágeis, a adoção de práticas ágeis favoreceram o desenvolvimento do projeto, bem como a adoção de métricas combinando a metodologia ágil permitiram a realização de estimativas mais confiáveis. Conforme o estudo de Martins (2013), a autora apresenta que, a prática ágil programação em par juntamente com a prática reunião diária ajudaram o time a melhorar sua comunicação e a resolver possíveis bloqueios em tarefas. Os benefícios das métricas é citado no estudo de Lima (2015), no qual o autor relata que o uso de métricas para mensurar os resultados oferece condições de melhor prever resultados futuros.

Sobre os desafios encontrados o mais citado foi a resistência inicial das equipes, citados em dois estudos. Conforme o estudo analisado de Sganderla *et al.*, (2016), os autores relataram dificuldades nas práticas de Refatoração e Programação em par, onde, a prática de Programação em par foi a que mais gerou discussão, pois os desenvolvedores achavam que teriam um rendimento maior sozinhos do que dois programadores juntos em uma máquina. No entanto, após vários experimentos, a equipe conseguiu adaptar-se com a prática e visualizou os benefícios da sua utilização.

## 7. Conclusão

Através da realização do teste piloto e, portanto com base nos benefícios e desafios encontrados pelos estudos incluídos, percebeu-se que a utilização de metodologias ágeis, práticas ágeis e métricas, colaboram com a melhoria do processo de desenvolvimento de *software*. As metodologias ágeis combinadas com a utilização de suas práticas, trouxeram as equipes mais confiança e agilidade. Já as métricas, contribuíram com a detecção de problemas e assim permitiram a constante melhoria da qualidade do *software*. Portanto, concluído a realização do teste piloto, pretende-se

realizar a conclusão da segunda e terceira fases da pesquisa e assim a conclusão da revisão sistemática de literatura.

## Referências

- Abrahamsson, Pekka; Salo, Outi. Agile Software Development Methods-Review and Analysis. Espoo 2002. VTT Publications.
- Beck, K. Extreme Programming Explained: Embrace Change, 2000. Addison-Wesley.
- Beck, K. et al. Manifesto for agile software development, 2001. Disponível em: <<http://www.agilemanifesto.org>>. Acesso em: 01 mai. 2017.
- Felizardo, Katia Romero et al. Revisão sistemática da literatura em Engenharia de Software: teoria e prática. 1.ed. Rio de Janeiro: Elsevier, 2017.
- Gamba, Mateus L.; Barbosa, Ana Cláudia G. Aplicação de Métricas de Software com Scrum. 2010. Disponível em: <<http://goo.gl/Dr1K6X>>. Acesso em: 20 ago. 2017.
- Lima, Alessandro P. DESIGN E PRÁTICAS ÁGEIS: Aplicação de Filosofia e Princípios Ágeis no Desenvolvimento de Modelos Tridimensionais para Jogos Digitais. 2015. 190 f. Dissertação (Pós-graduação em Design)- Universidade Federal do Rio Grande do Sul. Porto Alegre, 2015.
- Martins, Fernanda L.M. Métodos ágeis na concepção e manutenção de sistemas ligados à infraestrutura. 2013. 69f. Dissertação (Graduação em Ciência da Computação)-Universidade Federal do Rio Grande do Sul. Porto Alegre, 2013.
- Palmer, Stephen R.; Felsing, John M. A Practical Guide to Feature Driven Development. Prentice Hall, 2002.
- Pressman, Roger. Engenharia de Software: Uma abordagem profissional. 7ª ed. Artmed: Porto Alegre, 2011.
- Santos, Jorge; Quintiliano, Leandro; Barbosa, Lilian; Silva, Paulo R. H.; Giraldes, Wanderley Jr. Métodos Ágeis. Disponível em : <<http://goo.gl/tqk6Jm>> Acesso em: 02 jun. 2017.
- Sbrocco, J. Metodologias ágeis: engenharia de software sob medida. 1ª ed. São Paulo, Érica. 2012.
- Sganderla, M. A. et al. Aprimorando a Gerência e o Desenvolvimento de Software com Metodologias Ágeis. RCT-Revista de Ciência e Tecnologia, v.2, n.2, 2016. Disponível em : <<https://goo.gl/o4z3Ks>>. Acesso em: 20 ago. 2017.
- Standish Group: The Chaos Report 2015. Disponível em: <<https://www.standishgroup.com>>. Acesso em: 03 abr. 2017.
- Sommerville, Ian; Bosnic, Ivan; Gonçalves, Kalinga G. Engenharia de Software. 9ª ed. Pearson Prentice Hall: São Paulo, 2011.
- Teles, V. Extreme Programming. Disponível em: <<http://bit.ly/1QBchTo>>. Acesso em: 05 abr. 2017.