

# AgilAC: Um *framework* para avaliação de Métodos Ágeis

Marcelo Benites Gonçalves<sup>1</sup>

Leandro Magalhães de Oliveira<sup>2</sup>

Maria Istela Cagnin<sup>3</sup>

*Faculdade de Computação  
Universidade Federal do Mato Grosso do Sul  
Campo Grande-MS, Brasil*

---

## Resumo

Os métodos ágeis priorizam os fatores humanos e a capacidade de entrega como fatores de sucesso. Essa abordagem tem ganhado popularidade em muitos ambientes reais de desenvolvimento de software. Considerando as diversas propostas de métodos ágeis, não foram identificadas na literatura técnicas e ferramentas que apoiem de forma eficaz a verificação de agilidade de métodos de desenvolvimento durante atividades de concepção, construção e adaptação. Desta forma, este trabalho propõe um *framework* que dá suporte à análise de agilidade em métodos de desenvolvimento, denominado AgilAC, em que é possível analisar a conformidade do método às características mais amplamente defendidas para a família dos métodos ágeis. Adicionalmente, para analisar a usabilidade do *framework*, a relevância dos seus resultados e ilustrar o processo de condução da análise, foi realizada a análise de agilidade de alguns métodos de desenvolvimento de software, em especial, o Scrum e uma versão adaptada do Scrum ao CMMI.

*Palavras-chave:* método ágeis, análise de agilidade, *framework*.

---

## Abstract

The agile methods give priority to human factors and to the delivery ability as success factors. This approach has gained popularity in many real environments of software development. Considering the various agile methods proposals, were not identified in the literature techniques and tools that effectively support the agility verification of development methods in activities of design, construction and adaptation. Thus, this paper proposes a framework, called AgilAC, that gives support to the agility assessment in development methods. With AgilAC is possible to analyze the method's compliance with the most widely defended characteristics for the family of agile methods. Additionally, to analyze the usability of the framework, the relevance of its results and to illustrate the process of conducting the analysis, the analysis of agility was conducted for some software development methods, particularly to Scrum and to an Scrum's adapted version of the CMMI.

*Keywords:* agile methods, agility assessment, framework.

---

<sup>1</sup> Email: marcelobtg@gmail.com

<sup>2</sup> Email: leandro.oliveira.ms@gmail.com

<sup>3</sup> Email: istela@facom.ufms.br

## 1 Introdução

Métodos ágeis surgiram como uma proposta para o desenvolvimento de software, que valoriza as relações interpessoais e o aumento na capacidade de entrega como elementos determinantes do sucesso nos projetos [2]. Essa nova abordagem de desenvolvimento propõe um nível mínimo de sistematização de suas atividades para que se ofereça apenas o suficiente para apoiar no desenvolvimento do software [11], sustentando de forma eficaz e equilibrada a produtividade e o andamento dos projetos [8].

Dada sua configuração menos burocrática e restritiva, os métodos ágeis se mostraram adequados em cenários onde modelos tradicionais são considerados ineficientes [12] como, por exemplo, em organizações pequenas com baixo poder de investimento ou em projetos com requisitos dinâmicos e equipes reduzidas.

Contudo, a evolução dos métodos ágeis revelou problemas de padronização e caracterização. Atualmente, são muitas as propostas ágeis existentes e não há um padrão comum entre elas. O resultado é a falta de consenso e o surgimento de métodos propostos como ágeis mas que contemplam apenas parte da filosofia de desenvolvimento ágil [9].

Desta forma, a fim de aumentar a confiabilidade da metodologia<sup>1</sup> ágil, é importante o estudo de mecanismos que avaliem estes métodos em relação ao que se espera deles, provendo assim um suporte adequado para atividades de definição, adaptação e implantação dessa família de métodos. Foram identificadas, em um processo de revisão sistemática e análise previamente conduzido [6], abordagens existentes na literatura nesta direção que são brevemente descritas na Seção 2.

Observou-se em Benites e Cagnin [6] que os critérios de análise de agilidade utilizados pelas abordagens estudadas não levam em consideração o nível de burocracia e a sistematização que possam comprometer a agilidade em relação ao conjunto total de atividades do método; não é verificado o relacionamento entre a estrutura do método em relação a seus objetivos individuais de aplicação; e a apresentação dos resultados de análise das propostas não reporta um diagnóstico conciso em relação à agilidade do método de desenvolvimento sob análise.

Neste contexto, este trabalho propõe um *framework* para dar suporte às atividades de análise de agilidade em métodos de desenvolvimento de software, denominado AgilAC<sup>2</sup>. Esta proposta visa estabelecer uma abordagem que guie a análise de agilidade em métodos de desenvolvimento preenchendo algumas das lacunas identificadas nas abordagens já existentes para avaliação de métodos ágeis, com o compromisso de prover uma avaliação estruturada, abrangente e evolutiva para esta família de métodos.

O artigo está estruturado da seguinte forma: na Seção 2 são descritas brevemente algumas abordagens já existentes para avaliação de métodos ágeis e uma discussão de suas principais carências e pontos fortes, que colaboraram para a definição do *framework* AgilAC; na Seção 3 o *framework* é descrito e, em paralelo, são apresentados os resultados da sua submissão ao Scrum [22] e a uma versão adaptada do Scrum [21], indicada neste trabalho por Scrum', que se utiliza das práticas genéricas do CMMI (*Capacity Maturity Model Integration*)[4] para auxiliar na institucionalização de suas práticas ágeis, adicionalmente é apresentado um resumo da aplicação do *framework* em outros dois métodos ágeis – XP [5] e OpenUP [14] na

<sup>1</sup> O termo metodologia foi adotado neste artigo com o significado de um conjunto de métodos.

<sup>2</sup> O nome do *framework* proposto reúne o termo “Ágil” como prefixo, se referindo ao propósito central de avaliação do *framework* e o sufixo “AC” significando “avaliação baseada em níveis de cobertura”.

Seção 4 são apresentados os resultados gerais obtidos com o uso e a construção do *framework*, principais contribuições, dificuldades encontradas e indicações de trabalhos futuros.

## 2 Abordagens de Avaliação de Métodos Ágeis

Com o objetivo de investigar o “estado da arte” das abordagens para avaliação de agilidade, foi conduzido anteriormente um processo de revisão sistemática [6]. Este estudo averiguou as principais abordagens de avaliação de agilidade e, de forma complementar, realizou uma meta-síntese que comparou e identificou as principais contribuições e carências existentes nestas propostas. A seguir são brevemente descritas as principais abordagens analisadas e suas carências que motivaram a definição do *framework* proposto neste trabalho.

Qumer e Henderson-Sellers [16, 17] propõem um *framework*, denominado 4-DAT, que avalia a agilidade sob quatro dimensões ou perspectivas de análise: 1) Verificação do escopo do método, que verifica itens como tamanho da equipe, tipo de código, estilo de desenvolvimento, cultura de negócios, tamanho do projeto e mecanismos de abstração; 2) Caracterização de agilidade de forma quantitativa, relacionando fases e práticas do método com características ágeis (flexibilidade, rapidez, parcimônia (*leanness*), aprendizado e responsividade); 3) Verificação do atendimento aos valores ágeis, levando em consideração quatro valores ágeis providos pelo Manifesto Ágil e dois valores estabelecidos pelos próprios autores; e 4) Caracterização do processo de software, que analisa dois componentes principais do processo de software do método (processo de engenharia de produto e gerenciamento de processo).

O 4-DAT apresenta uma estrutura mais adequada à comparação do que para a avaliação de métodos ágeis. Na dimensão 2, de análise quantitativa, os autores não abordam um estudo aprofundado que justifique a eficácia ou precisão da proposta de tratamento matemático realizada. O que acaba por permitir casos em que métodos com práticas excessivas gerem falsos positivos, pois cada fase ou prática são analisadas individualmente, não sendo possível quantificar a inter-relação ou influência das fases ou práticas em relação ao método. Uma das carências da dimensão quantitativa do 4-DAT é ao tratar métodos puramente não prescritivos que não determinam atividades, sendo essencialmente constituídos por valores e regras.

Abramhamsson *et al.* [1] propõem um *framework* para análise e comparação de métodos ágeis que analisa como o método atende aos aspectos: fases de desenvolvimento, gestão de projeto, provisão de guia de aplicação com regras concretas e aplicáveis, abrangência de aplicação do método, existência de evidências e estudos empíricos sobre o método, baseados em um trabalho anterior [2] em que realiza uma comparação de metodologias ágeis.

Na abordagem de Abramhamsson *et al.* [1] foi verificado um direcionamento maior ao melhoramento e comparação entre métodos ágeis, tendo entre seus principais objetivos identificar diferenças e similaridades entre métodos ágeis, não estipulando ou verificando adequadamente as características exclusivas para esta família de métodos. Os autores adotam um modelo mais genérico de avaliação e comparação, sendo assim sua contribuição é importante para verificar principalmente a evolução dos métodos ágeis sob a perspectiva dos aspectos gerais do desenvolvimento de software.

Conboy [9] estabeleceu um *framework* para análise de agilidade que percorre diversas disciplinas de desenvolvimento de software adaptadas ao contexto ágil, criando um grupo de conceitos para métodos ágeis em desenvolvimento de software

(criação, proação, reação e aprendizado). O autor faz a avaliação das atividades do método ágil sob esses conceitos e dá exemplos de como adaptar as atividades para alcançar mais agilidade nestes conceitos.

Contudo, o trabalho desenvolvido por Conboy não se atém à avaliação de critérios para métodos ágeis, seu *framework* delimita disciplinas de desenvolvimento sob a ótica ágil e faz sugestões de melhoramentos mas não determina um critério de avaliação que possa gerar um resultado preciso para o avaliador.

Taromirad e Ramsin [23], após um breve estudo de alguns *frameworks* de análise e comparação de métodos ágeis, definiram seu próprio *framework* – CEFAM (*Comprehensive Evaluation Framework for Agile Methodologies*) [24] para análise de métodos ágeis, que propõe cobrir as deficiências encontradas nos *frameworks* estudados pelos autores. O CEFAM segue uma estrutura de análise hierárquica em que os principais critérios de avaliação são: processos, linguagem de modelagem, agilidade, utilização e critérios de contexto múltiplo.

Foi identificado que o CEFAM apresenta muitas variações nos domínios de valor estabelecidos para os resultados dos critérios avaliados, o que dificulta a compreensão do resultado geral da análise. Outro item a ser considerado é o fato de que o CEFAM inclui em seu conjunto de critérios de análise alguns elementos não amplamente reconhecidos para a metodologia ágil, como por exemplo a avaliação do tratamento dado pelo método em relação a linguagem de modelagem dos projetos e itens baseados na estrutura do XP [5, 24] que é um método ágil específico. Como a avaliação não diferencia estes elementos menos gerais, há o risco de se obter resultados não abrangentes.

Outros trabalhos não foram analisados pois tratam de: (i) propostas de avaliação de métodos específicos como o Nokia Test [20] que avalia equipes de desenvolvimento baseado no Scrum ou o XP-EF que avalia a aderência de processo às práticas do XP [26], (ii) técnicas de análise empírica, como análise de desempenho de equipes em projetos [13]; (iii) ferramentas para análise de sub-áreas do contexto ágil como por exemplo a avaliação de agilidade exclusivamente aplicada a equipes ágeis [7].

Em linhas gerais, além das carências já discutidas anteriormente, nenhuma das abordagens analisadas atinge adequadamente aos objetivos de: gerar como saída um diagnóstico conciso, expressivo e unificado sobre a agilidade de cada método analisado como relatórios resumidos de análise; observar as características mais amplamente admitidas na literatura da metodologia ágil; relacionar as características do método aos seus objetivos de aplicação; mensurar o nível de sistematização de processos e atividades, que não podem ser insuficientes ou excessivos demais a ponto de comprometer a agilidade do método dentro do seu contexto de aplicação.

### 3 AgilAC: *Framework* para Avaliação de Agilidade

Entre os desafios existentes no estabelecimento de um mecanismo para a análise de métodos ágeis estão: a heterogeneidade entre métodos já consolidados como ágeis; a baixa especificidade nas descrições de padronização já consolidadas como, por exemplo, o manifesto ágil [10]; e a subjetividade inerente a certos aspectos dos métodos ágeis, principalmente os aspectos baseados em valores e princípios, que tornam difícil estabelecer modelos imunes a julgamentos subjetivos por parte do avaliador.

Visando estabelecer um mecanismo de análise onde os desafios supracitados são minimizados tanto quanto possível, o AgilAC segue um processo de análise organizado por etapas oferecendo ao avaliador a possibilidade de observar o método sob diferentes perspectivas de análise. Sua divisão conceitual para a avaliação dos

critérios esperados para métodos ágeis é baseada na divisão por áreas de conhecimento em gerenciamento de projetos do PMBOK [15], que oferece uma organização em alto nível de todo o conhecimento relacionado ao gerenciamento de projetos.

Contudo, é importante ressaltar que o embasamento no PMBOK se limita à estrutura para divisão dos critérios de avaliação em áreas de conhecimento coesas de gerenciamento, ou seja, as áreas de conhecimento do PMBOK são utilizadas para atingir uma boa organização dos critérios de avaliação que representam as características aceitas para métodos ágeis. Entretanto, o AgilAC não tem entre seus objetivos avaliar explicitamente a aderência de métodos ágeis ao PMBOK.

Os métodos ágeis não seguem um padrão na descrição dos seus elementos. Devido a este fato, o avaliador deverá entender a estrutura do método e observar todos os elementos que a compõe (por exemplo, tarefas, atividades, práticas, valores, etc).

### 3.1. Etapas do Framework

A aplicação do AgilAC a um método de desenvolvimento segue uma sequência de três etapas.

A etapa 1 verifica como os elementos do método atendem aos critérios de agilidade estipulados pelo AgilAC. Na etapa 2 o método é observado sob uma visão global de avaliação que verifica seu alinhamento aos valores ágeis do Manifesto Ágil e sintetiza um nível de agilidade para o método. Ao final, na etapa 3 são descritas as instruções e são fornecidos os ativos para a produção do relatório padronizado de saída, em que os resultados da análise são resumidos e organizados formando um diagnóstico final relativo à agilidade do método.

Além dos itens exclusivos de cada etapa do *framework*, foi considerado um conjunto geral de itens com o propósito de manter a estrutura do AgilAC aplicável: (i) descrever de forma inteligível, coerente e aplicável o processo de avaliação; (ii) minimizar redundâncias entre os critérios avaliativos que constituem o *framework*; e (iii) ser capaz de orientar o avaliador sob como utilizar os critérios de avaliação para observar o método.

A descrição detalhada de cada etapa do *framework* é apresentada nas subseções a seguir. A escolha, para ilustração da análise, do Scrum [22] se dá ao fato do método ser amplamente difundido e aceito, possibilitando a avaliação do *framework* proposto. Já o Scrum' [21] se trata de uma versão adaptada do Scrum original ao CMMI e, desta forma, facilita a observação de sensibilidade dos resultados de análise do AgilAC. De forma complementar são brevemente descritos resultados obtidos com a análise de outros dois métodos – o XP e o OpenUP.

Também é importante destacar que os autores do Scrum' construíram esta adaptação com foco na institucionalização das práticas ágeis por meio das práticas genéricas do CMMI e da consequente otimização de esforços empregados, verificando em projetos piloto, aplicados em uma organização aderente ao nível 5 do CMMI, melhoras significantes de desempenho com esta abordagem. Desta forma espera-se que esta versão adaptada não apresente grandes variações na agilidade em relação ao Scrum original. Este é exatamente o motivo pelo qual ela foi adotada pois permite a verificação do nível de sensibilidade mais apurado do *framework*, ou seja, se fosse escolhido um método ou adaptação claramente não ágil, mesmo com uma baixa sensibilidade à agilidade dos métodos, o *framework* poderia identificar perdas na agilidade da versão adaptada em relação à original.

A semântica atribuída para “método” é de extrema importância, pois é necessário

estabelecer o que conceitualmente os métodos ágeis devem prover. Rumbaugh [18] define o termo “método” como sendo um guia que oferece a descrição de um caminho para se atingir um determinado resultado esperado, desta forma fica evidente a necessidade do método conter a definição dos objetivos e resultados esperados e a descrição dos meios para atingir satisfatoriamente esses objetivos.

Assim, antes mesmo de verificar se um método de desenvolvimento está alinhado aos imperativos ágeis, o avaliador, por meio de uma primeira leitura geral das especificações do método, deve verificar se o método define seus objetivos de aplicação (por exemplo, escopo de projeto, tipo de organização) e se provê recursos adequados para atingir estes objetivos, ou seja, se descreve como atingir seus objetivos (por exemplo, com processos, práticas, atividades ou tarefas).

### *3.2. Etapa 1: Analisar a Cobertura do Método em relação a Critérios Ágeis*

Na etapa 1 do AgilAC as características aceitas para métodos ágeis são organizadas como critérios de avaliação que estão agrupados de acordo com as áreas de conhecimento do PMBOK. O objetivo desta etapa é verificar a agilidade do método por meio do levantamento de níveis de cobertura que representam a capacidade do método analisado em atender aos critérios de avaliação estipulados.

Os critérios de avaliação do AgilAC representam as características mais amplamente aceitas para métodos ágeis e são baseados em duas fontes principais: 1) o trabalho de Abrantes e Travassos [3], que realiza uma revisão sistemática e levanta as características aceitas para métodos ágeis mais comumente defendidas pelos pesquisadores verificadas por nível de incidência e, desta forma, reflete adequadamente o que é usualmente considerado como elementos indispensáveis para métodos ágeis; e 2) os valores e princípios ágeis, disponíveis no manifesto ágil [10], que representa um esforço inicial para a formalização dos métodos ágeis sendo hoje amplamente aceito pela comunidade ágil [16].

Para aplicar o framework nesta etapa (Tabela 1), o avaliador deve verificar a aderência do método aos critérios de avaliação de agilidade estipulados seguindo os passos: 1) indicar o nível de cobertura do método sob análise em relação a cada critério; 2) calcular a incidência de cada tipo de nível de cobertura que é dada pela porcentagem de critérios que tiveram o nível específico de cobertura (Nível 0, Nível 1 ou Nível 2) em relação ao total de 20 critérios avaliados; 3) ao final o avaliador tem três porcentagens que indicam a incidência de cada nível na avaliação (NA, AT e AE, conforme ilustrado na Tabela 1) que servirão para a formação de um diagnóstico final de agilidade produzido na etapa 3.

A seguir são descritos os três possíveis níveis de cobertura a serem admitidos pelo avaliador na análise do método ágil em relação aos critérios de avaliação de agilidade do AgilAC:

- **Nível 0:** representado pelo símbolo ○, este nível de cobertura deve ser adotado quando não há elementos no método que possam atender às exigências determinadas pelo critério verificado. Por exemplo, o Scrum não atende o critério “Emergência”, pois grande parte do projeto é planejado nas fases iniciais.
- **Nível 1:** representado pelo símbolo ●, este nível de cobertura deve ser adotado quando há elementos no método que possam atender, de maneira satisfatória e econômica, dados os objetivos de aplicação do método e as exigências determinadas pelo critério verificado. Não sendo possível estabelecer um subgrupo dentro deste grupo de elementos que também possa atender satisfatoriamente ao critério determinado. Por exemplo, o Scrum atende ao critério “Desenvolvimento

- Iterativo” através de *sprints* (nome dado as iterações do Scrum).
- **Nível 2:** representado pelo símbolo ●+, este nível de cobertura deve ser adotado quando há elementos no método que possam atender, de maneira satisfatória, dados os objetivos de aplicação, às exigências determinadas pelo critério verificado porém o conjunto geral de elementos é excessivamente burocrático ou sistematizado de forma que possa ser otimizado. Ou seja, um subconjunto menor e mais simplificado do conjunto de elementos do método também atendem de forma satisfatória ao critério verificado considerando os objetivos de aplicação do método e sem prejudicar a cobertura dos demais critérios. Por exemplo, o Scrum’ que contém um número excessivo de atividades de prevenção e monitoramento de riscos. Dessa forma, ele atende o critério “Convergência”, mas realiza uma quantidade excessiva de atividades.

Para indicar os níveis de cobertura do método sob análise em relação aos conjuntos de critérios de avaliação de agilidade do *framework*, o avaliador deve considerar que a satisfação de cada critério é diretamente dependente da compatibilidade entre os fins, que se tratam dos cenários de aplicação especificados pelo método (por exemplo, escopo de projeto, tipo de organização que pode utilizar o método), bem como os meios determinados pelo método sob análise para se atingir a esses fins (como por exemplo práticas, processos, atividades e tarefas). Na Tabela 1 são apresentados os níveis de cobertura do Scrum e do Scrum’ em relação aos critérios de avaliação de agilidade.

Tabela 1. Cobertura do Método Ágil em relação às Características Esperadas para Métodos Ágeis

Área de Conhecimento	Crítérios de Agilidade	Scrum	Scrum’
<b>Integração:</b> reúne as atividades das demais áreas de conhecimento com ações integradoras para gerenciar com sucesso as expectativas das partes interessadas e atender aos requisitos	<b>Desenvolvimento Iterativo:</b> o desenvolvimento da aplicação envolve vários ciclos curtos? <b>Princípio Ágil Relacionado –</b> “Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado”.	●	●
	<b>Desenvolvimento Incremental:</b> o método determina que o desenvolvimento do sistema se baseie em pequenos <i>releases</i> de versões funcionais do software, com ciclos rápidos de desenvolvimento, onde o sistema é partido em incrementos que podem ser desenvolvidos em paralelo em tempos diferentes? <b>Princípio Ágil Relacionado –</b> “Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo”.	●	●
<b>Escopo:</b> deve assegurar que o projeto inclui todo o trabalho necessário, e apenas o necessário, para terminar o projeto com sucesso.	<b>Modularidade:</b> o método propõe a modularização das atividades de seu processo de desenvolvimento de forma que haja flexibilidade suficiente para adicionar ou remover módulos de atividades quando necessário?	○	●
<b>Tempo:</b> trata dos elementos relativos ao controle do tempo dispendido na execução do projeto	<b>Emergência:</b> os processos, princípios, estruturas de trabalho são reconhecidos durante o projeto ao invés de serem pré-determinados? Permite-se que tecnologia e os requisitos emergjam ao longo do ciclo de vida do produto e a fase inicial de levantamento seja minimizada?	●	○
	<b>Restrição de Prazo:</b> o método estabelece de alguma forma limites de tempo para cada iteração programada de modo previsível? Há ações para controle e tratamento destes prazos?	●	●

<p><b>Custos:</b> está ligada às estimativas, aos orçamentos e ao controle dos custos envolvidos no projeto, de modo que o projeto possa ser executado com custos estabelecidos e controlados</p>	<p><b>Parcimônia, enxutez ou leveza (<i>Jeanness</i>):</b> o método determina o mínimo necessário de atividades para mitigar riscos e alcançar metas? O método promove remoção de todas as atividades dispensáveis ao processo de desenvolvimento? <b>Princípio Ágil Relacionado</b> – “Simplicidade – a arte de maximizar a quantidade de trabalho não realizado é essencial”.</p>	<p>●</p>	<p>○</p>
<p><b>Qualidade:</b> determina políticas de qualidade, objetivos e responsabilidades, de modo que o projeto siga parâmetros de qualidade pré-estabelecidos, satisfazendo às necessidades para as quais foi empreendido</p>	<p><b>Testes:</b> o método trata a degradação da qualidade devido a entregas muito curtas? O método descreve alguma política de testes do produto ao longo do ciclo de vida do projeto?</p>	<p>●</p>	<p>●</p>
	<p><b>Realimentação:</b> As equipes são capazes de procurar e receber continuamente realimentação (<i>feedback</i> de satisfação) de modo mais freqüente e com mais rapidez?</p>	<p>●</p>	<p>●</p>
<p><b>Recursos Humanos:</b> o gerenciamento dos recursos humanos do projeto inclui a organização e gerenciamento da equipe do projeto.</p>	<p><b>Reflexão/Introspecção:</b> o método estabelece ações de autoanálise de desempenho nas equipes? <b>Princípio Ágil Relacionado</b> – “Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.”</p>	<p>●</p>	<p>●</p>
	<p><b>Equipe Capacitada:</b> o método determina regras relacionadas a capacitação técnica dos membros das equipes? <b>Princípio Ágil Relacionado</b> – “Continua atenção à excelência técnica e bom design aumenta a agilidade.”</p>	<p>●</p>	<p>●</p>
	<p><b>Equipe Motivada:</b> há políticas de motivação dos membros das equipes? <b>Princípio Ágil Relacionado</b> – “Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho”.</p>	<p>●</p>	<p>●</p>
	<p><b>Equipes Pequenas:</b> o método determina um número reduzido de membros por equipe e do número de equipes do projeto?</p>	<p>●</p>	<p>●</p>
	<p><b>Equipes Auto-organizáveis:</b> o método provê uma estrutura de atividades que permita que as equipes tenham autonomia para auto organização? <b>Princípio Ágil Relacionado</b> – “As melhores arquiteturas, requisitos e <i>designs</i> emergem de equipes auto-organizáveis”</p>	<p>●</p>	<p>●</p>
	<p><b>Equipe Colaborativa e Comunicativa:</b> o método determina a participação ativa e bom relacionamento entre os membros da equipe, como fator determinante de sucesso de sua aplicação? <b>Princípios Ágeis Relacionados</b> – “Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto”; “O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face”.</p>	<p>●</p>	<p>●</p>
	<p><b>Equipes Locais:</b> o método é sensível à localização da equipe, estipulando que os membros de uma equipe devam trabalhar no mesmo ambiente (por exemplo na mesma sala ou em salas adjacentes)?</p>	<p>●</p>	<p>●</p>
	<p><b>Equipe Colaborativa:</b> o método define ações para tornar o ambiente de trabalho propício à equipes colaborativas?</p>	<p>●</p>	<p>●</p>

	<b>Princípio Ágil Relacionado</b> – “Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto”.		
<b>Comunicação:</b> assegura a gestão das informações do projeto de maneira apropriada	<b>Equipe Comunicativa:</b> quando permite membros distribuídos, o método prevê ações para diminuir os impactos na comunicação? <b>Princípio Ágil Relacionado</b> – “O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face”.	●	●
<b>Riscos:</b> monitoramento e controle de riscos de um projeto.	<b>Convergência:</b> o método determina explicitamente ações de ataque efetivo a riscos? a cada iteração os riscos são atacados de forma pró-ativa?	●	● <sub>+</sub>
	<b>Adaptabilidade:</b> o método é tolerante a mudanças? descreve ações para atender e reagir a mudanças de última hora tratando situações, requisitos, prioridades ou riscos não previstos inicialmente ? prevê respostas eficientes com o surgimento da necessidade de mudanças nos requisitos e no ambiente? <b>Princípio Ágil Relacionado</b> – “Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente”.	●	○
<b>Aquisições:</b> o gerenciamento das aquisições do projeto abrange a administração da negociação do produto com o cliente.	<b>Cooperatividade:</b> o método determina a participação ativa e bom relacionamento do cliente com a equipe de desenvolvimento como fator determinante de sucesso de sua aplicação? <b>Princípio Ágil Relacionado</b> - “Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente”.	●	●
	<b>Cobertura (%)</b>	<b>Scrum</b>	<b>Scrum'</b>
	<b>Nível 0 – Total não satisfatoriamente atendido (NA)</b>	5%	15%
	<b>Nível 1 – Total satisfatoriamente atendido e otimizado (AT)</b>	95%	80%
	<b>Nível 2 – Total atendido com Excedente (AE)</b>	0%	5%

Durante a aplicação dessa etapa foram analisadas as características gerais dos métodos escolhidos de acordo com os critérios da avaliação. A cada critério foi atribuído um nível de acordo com as características dos métodos. No Scrum observou-se 95% dos critérios satisfatoriamente atendidos, como espera-se de um método ágil consolidado. Scrum' por sua vez, devido às suas modificações, apresentou 80% de cobertura satisfatória dos critérios, enquanto 20% dos critérios não são atendidos ou são atendidos com uma quantidade excessiva de atividades.

A Tabela 2 resume os problemas encontrados em relação à agilidade do método indicando pontos críticos e, quando possível, indicando possibilidades de melhoria. O avaliador deve descrever justificativas tão detalhadas quanto possível para todos os critérios onde a cobertura alcançou os níveis 0 (NA) ou 2 (AE).

Onde a cobertura teve Nível 0 o avaliador pode, no campo de “razões”, sugerir adaptações, novos processos, atividades ou tarefas para efetivar a cobertura do critério. Onde a cobertura teve Nível 2 o avaliador deve descrever onde verificou excesso na determinação de esforços e, em caráter opcional, fornecer um possível subconjunto do conjunto de elementos propostos pelo método que também atenda ao critério de avaliação de agilidade, porém de uma forma mais econômica que demande menos esforços.

Durante o preenchimento da Tabela 2 sobre os métodos Scrum e Scrum', foi possível comparar quais características dos métodos analisados não atingiu o Nível 1 durante a aplicação da primeira etapa e quais as razões levaram-no a não atingir esse

nível. Através dessas informações é possível identificar quais motivos fizeram com que um método perdesse agilidade e, dessa forma, modificá-lo para obter o nível de agilidade desejada. Durante esta etapa da análise foi verificado que os métodos avaliados são, de acordo com o *framework*, ágeis apesar de haver elementos que poderão ser modificados para aumentar a agilidade desses métodos caso seja desejado.

Tabela 2. Análise global de complexidade ágil

Pontos Críticos à Agilidade do Método		
Scrum (critérios não atendidos)	Critério	Razões
	<b>Modularidade</b>	O Scrum é dividido em <i>sprints</i> . Antes da realização de cada <i>sprint</i> existe uma reunião de planejamento e durante a mesma são definidas atividades que serão realizadas e resultados que são esperados durante esse ciclo. Por esse motivo, existe uma diminuição da flexibilidade de modificação das atividades.
Scrum (critérios atendidos em excesso)	Critério	Razões
	--	--
Scrum (critérios não atendidos)	Critério	Razões
	<b>Emergência</b>	O Scrum faz grande parte do projeto do software na fase inicial.
	<b>Parcimônia (leanness)</b>	Foram inseridas atividades para gerar uma melhor e mais completa documentação. O Scrum utiliza um <i>checklist</i> para verificar se as histórias de usuários contém as informações esperadas.
Scrum (critérios atendidos em excesso)	Critério	Razões
	<b>Convergência</b>	Gestão de riscos fortemente adotada.

### 3.3. Etapa 2: Analisar a Aderência em relação aos Valores Ágeis

Os valores especificados pelo Manifesto Ágil como “Valores Ágeis” apenas indicam tendências que os métodos ágeis devem seguir de forma não restritiva. Por exemplo, o valor ágil “Software em funcionamento ao invés de documentação abrangente” indica que disponibilizar versões funcionais do software deve ser mais valorizado que realizar uma extensa documentação, mas não exclui a importância da documentação nem afirma que ela deve ser desconsiderada.

Neste contexto a etapa 2 propõe uma visão unificada dos valores e prioridades do método sob análise, por meio da análise de sua aderência aos valores do Manifesto Ágil. Ao final é gerado um índice geral de aderência aos valores ágeis (**AV**) que contribui na produção do relatório de saída da etapa 3. Com a aplicação desta etapa também é possível alcançar elementos do método sob análise que não foram observados na etapa 1, por não estarem relacionados ao cumprimento direto dos critérios de avaliação de agilidade do *framework*.

Na etapa 2, o avaliador deve verificar a aderência do método aos valores ágeis seguindo os passos: 1) verificar em cada valor ágil os elementos do método que contribuem positivamente (“Valorizam mais...”) ou negativamente (“em relação a...”) para tornar o método alinhado aos valores ágeis, adicionando uma descrição sucinta de quais são esses elementos e como é dada sua contribuição; 2) após levantar

os dois grupos de elementos o avaliador deve estabelecer se o método é ou não aderente ao valor sob análise considerando não somente a quantidade de elementos como também seu nível de influência na utilização do método; 3) ao final deve ser feito o cálculo da aderência representada pela porcentagem de valores em que o método conseguiu ser aderente em relação ao total dos valores (**AV**). A Tabela 3 apresenta o esquema de análise a ser adotado nesta etapa.

Tabela 3. Análise de Aderência do Método aos Valores Ágeis

	Valoriza mais...	Em relação a...	É aderente ao valor? (Sim/Não)
	Indivíduos e interações	Processos e ferramentas	
<b>Scrum</b>	<i>Daily Scrum</i> ; Clientes envolvidos com o desenvolvimento; Reunião de planejamento de <i>sprints</i> ;	---	SIM
<b>Scrum'</b>	<i>Daily Scrum</i> ; Clientes envolvidos com o desenvolvimento; Reunião de planejamento de <i>sprints</i> ;	---	SIM
	Software em funcionamento	Documentação abrangente	
<b>Scrum</b>	Pequenos <i>Releases</i> ; Produção de pouca documentação;	---	SIM
<b>Scrum'</b>	Pequenos <i>Releases</i> ;	Documentação mais detalhada, mas não exagerada;	SIM
	Colaboração com o cliente	Negociação de contratos	
<b>Scrum</b>	Participação ativa do cliente no desenvolvimento do software;	---	SIM
<b>Scrum'</b>	Participação ativa do cliente no desenvolvimento do software;	---	SIM
	Responder a mudanças	Seguir um plano	
<b>Scrum</b>	Mudanças são bem vindas mesmo que tardias.	---	SIM
<b>Scrum'</b>	---	Grande parte do projeto definido nas fases iniciais.	NÃO
	<b>Aderência (AV)</b>	<b>Scrum</b>	100%
		<b>Scrum'</b>	75%

Para aplicar esta etapa nos métodos escolhidos foi preciso associar algumas práticas dos mesmos com os valores ágeis. Após essa associação, foi necessário verificar quais características eram mais importantes e dessa forma averiguar se o método proposto era ou não aderente ao valor ágil. Assim como a etapa anterior, essa etapa é de fácil aplicação e leva em consideração o conhecimento do avaliador para definir os pontos importantes e, dessa forma, definir a aderência ao valor ágil. Durante esta etapa foi possível perceber que os elementos do Scrum são aderentes aos valores ágeis, ao contrário do Scrum', que contém elementos não aderentes.

### 3.4. Etapa 3: Gerar Relatório de Análise de Agilidade

A etapa 3 finaliza a análise de agilidade, integrando os resultados das etapas anteriores e retornando conclusões de análise e um diagnóstico de agilidade. Os passos incluídas nesta etapa são: (i) gerar um diagnóstico geral de agilidade que sintetize e relacione as impressões levantadas nas etapas de análise anteriores utilizando para isso o algoritmo apresentado na Figura 1; (ii) reunir todos os resultados obtidos nas etapas anteriores em um único documento – o relatório de análise de agilidade.

Ao final desta etapa, o relatório gerado deve conter: os índices de cobertura e de aderência aos valores ágeis levantados respectivamente nas etapas 1 e 2; um dos diagnósticos finais de agilidade dispostos na Figura 2; os resultados da etapa 1 (Tabelas 1 e 2); e os resultados da etapa 2 (Tabela 3).

O diagnóstico final de agilidade resumido para o método formaliza uma saída sintética e coesa com a análise conduzida. Este diagnóstico é gerado pelo relacionamento dos índices obtidos nas etapas anteriores. Considerando os índices produzidos na etapa 1, foi considerada a necessidade do método analisado possuir a maior parte dos critérios verificados corretamente atendidos – AT (corresponde ao Nível 1 de cobertura descrito na Subseção 3.2). No que se refere ao índice **AV** levantado na etapa 2, com base nas análises realizadas para diversos modelos e na porcentagem de implementação de resultados esperados proposta no método de avaliação MA-MPS[19], foi determinado que, no índice final da análise de aderência, deve ser observada uma aderência mínima de 85% aos valores ágeis. Desta forma, a geração do diagnóstico deve seguir o algoritmo da Figura 1 e cada parecer é descrito conforme pode ser observado na Figura 2.

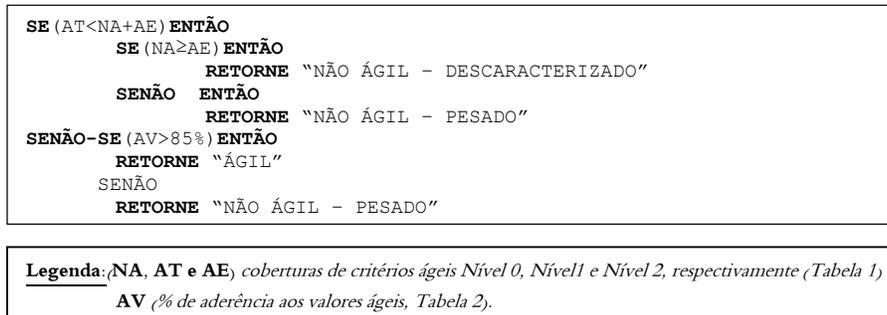


Fig. 1. Algoritmo para indicar o diagnóstico final de agilidade do método

Não Ágil (Descaracterizado)	Ágil	Não Ágil (Pesado)
<ul style="list-style-type: none"> <li>o método não pode ser considerado ágil pois é incompleto no atendimento aos critérios mínimos considerados para métodos ágeis (denominado descaracterizado). A reestruturação do método deve considerar o aprimoramento de seus processos para atender aos pontos deficientes verificados durante a análise. Para sua melhoria é importante observar sugestões e outras observações no pontos críticos observados aos critérios não atendidos (Tabela 3).</li> </ul>	<ul style="list-style-type: none"> <li>O método pode ser considerado ágil pois descreve satisfatoriamente elementos necessários ao desenvolvimento de software, é aderente aos valores ágeis e cobre de forma adequada os critérios mais comumente defendidos para métodos ágeis na literatura. Pode ser considerado para sua melhoria a análise de correção dos pontos críticos identificados, se houver (Tabela 3).</li> </ul>	<ul style="list-style-type: none"> <li>O método não pode ser considerado ágil pois determina o emprego de mais esforços que o mínimo necessário para o atendimento aos critérios observados (denominado pesado). Desta forma recursos podem ser desperdiçados comprometendo a agilidade geral do método. Pode ser realizada uma adaptação seguindo as simplificações sugeridas na relação de pontos críticos a agilidade referentes aos critérios atendidos em excesso (Tabela 3).</li> </ul>

Fig. 2. Diagnóstico final da agilidade do método

### 3.5. Outros resultados: análise de agilidade do OpenUP e XP

Assim como ilustrado para o Scrum e Scrum', foi realizada a aplicação do AgilAC a outros métodos incluindo o XP (*Extreme Programming*) e o OpenUP. A Tabela 4 apresenta um resumo dos resultados obtidos das análises. Métodos ágeis foram priorizados para serem submetidos ao AgilAC pois métodos tradicionais, por exemplo, são facilmente identificáveis como não ágeis e o objetivo maior é que o *framework* tenha a sensibilidade de identificar pequenas diferenças na agilidade entre métodos construídos ou adaptados sob o intuito de serem ágeis.

O método Scrum, foi avaliado como ágil atendendo a quase todos os critérios estabelecidos pelo *framework*. Já segundo a avaliação realizada sobre o Scrum', devido às modificações, manteve parte de suas características ágeis, mas os elementos adicionais inseridos modificaram a configuração original tornando o método insatisfatoriamente responsivo a mudanças, e dessa forma foi diagnosticado como não ágil pelo *framework*.

Os métodos XP e OpenUp também foram classificados como ágeis, sendo que o OpenUP apresentou uma porcentagem maior no nível 2 de cobertura (AE) em relação aos demais métodos analisados. A razão disso pode estar relacionada a origem prescritiva do método OpenUP, no caso, considerado uma instância do RUP (Rational Unified Process) [14] com características de métodos ágeis.

Tabela 4. Resumo de resultados de análises realizadas

Método	Etapa1 (NA)	Etapa1 (AT)	Etapa1 (AE)	Etapa2 (Aderência)	Avaliado como
Scrum	5%	95%	0%	100%	Ágil
Scrum'	15%	80%	5%	75%	Não Ágil-Pesado
XP	0%	95%	5%	100%	Ágil
OpenUP	0%	85%	15%	100%	Ágil

## 4 Conclusões

O AgilAC se mostrou um mecanismo de avaliação de agilidade abrangente ao admitir um conjunto de características e valores para métodos ágeis já amplamente defendidos na literatura, não se limitando assim a ter algum método ágil específico como parâmetro de análise. Os resultados do uso do *framework* na avaliação dos quatro métodos ágeis indicam que o AgilAC consegue produzir um bom diagnóstico em relação a aderência dos métodos aos imperativos ágeis como também agrupa pontos críticos de melhoria de forma organizada e útil quando se pretende dar mais prioridade a estes pontos para melhoria da agilidade do método.

Em relação as carências identificadas nas abordagens para análise de agilidade em métodos de desenvolvimento, o AgilAC apresenta uma estrutura adaptável e evolutiva pois sua organização inclui a observação dos aspectos gerenciais e dos valores gerais do método. Adaptações no *framework* podem ser convenientes em situações como avaliar métodos sob um contexto menos abrangente ou avaliar um método ágil específico. Assim, é possível realizar adaptações específicas no AgilAC, caso se verifique a necessidade de tratamento mais pormenorizado de algum critério por meio de sua expansão em novos critérios mais específicos sem desestruturar o corpo de análise de cobertura.

Da mesma forma é possível inserir alterações de modo a evoluir o *framework* para atualizá-lo ao se verificar possibilidades e aprimoramento nos valores ou critérios admitidos para métodos ágeis ou para manter o *framework* alinhado com o amadurecimento dos métodos ágeis gerando novas versões do AgilAC que atendam às novas realidades de usos e expectativas para esta família de métodos.

A despeito de considerar na definição do *framework* AgilAC a redução da subjetividade e independência de valores pessoais do avaliador, há uma considerável carga de subjetividade que é inerente ao processo de análise dadas a impossibilidade de quantificar a grande maioria dos critérios de avaliação, a pluralidade de métodos ágeis e cenários de aplicação a que se propõem. Desta forma, verifica-se que o êxito na avaliação ainda depende significativamente da aplicação do *framework* por avaliadores comprometidos com a imparcialidade e que tenham um bom domínio do assunto.

Contudo, estas dificuldades encontradas não invalidam a contribuição do *framework* nas atividades de análise a que se propõe inicialmente. Espera-se que, com o amadurecimento dos métodos ágeis, o estabelecimento de padrões mais delineados e corroborados pela prática nas organizações, seja possível diminuir gradualmente a subjetividade tornando o AgilAC mais preciso em versões posteriores.

Como trabalhos futuros podem ser citados: (i) estender o *AgilAC* para torná-lo um guia de avaliação mais completo, adicionando uma etapa de preparação para análise, no caso de estudos mais aprofundados de métodos, que inclua grupos de avaliação com o gerenciamento de papéis e atividades; (ii) como uma forma de atenuar a dificuldade de análise devido a heterogeneidade dos elementos utilizados na definição dos métodos ágeis (por exemplo práticas, processos, tarefas e valores) é possível estudar a criação de um processo para mapear todos os elementos do método a ser analisado em uma estrutura coerente de itens observáveis, possibilitando assim aumentar a precisão dos resultados retornados pelo AgilAC, pois se a análise utilizar como entrada sempre uma mesma estrutura de método é possível melhorar os passos de análise dividindo de forma mais coerente os elementos a serem analisados; (iii) abordar ferramentas de análise empírica e de personalização do *framework* para que organizações possam realizar avaliações mais direcionadas; (iv) conduzir estudos de

validação e aprimoramento do AgilAC, incluindo atividades como a realização de diversas avaliações de agilidade do mesmo método com avaliadores diferentes para confrontar relatórios de saída e o estudo comparativo de resultados para grupos de métodos com características específicas; (v) desenvolver uma ferramenta computacional que dê apoio à aplicação do *framework*; e (vi) construir um banco histórico de avaliações que possa representar uma base genérica de avaliação para os interessados.

## Referências

- [1] Abrahamsson, P., J. Warsta, M. Siponen, e J. Ronkainen, “New Directions on Agile Methods: A Comparative Analysis,” 2003. IEEE. pp. 244–254.
- [2] Abrahamsson, P., O. Salo, J. Ronkainen, e J. Warsta, “Agile Software Development Methods: Review and Analysis,” 2002.
- [3] Abrantes, J.F., e G.H. Travassos, “Uma Revisão Sistemática: Caracterização de Métodos Ágeis de Desenvolvimento de Software”. Relatório Técnico nº ???, PESC–Programa de Engenharia de Sistemas e Computação COPPE / UFRJ. 2007.
- [4] Ahern, D., J. Armstrong, A. Clouse, J. Ferguson, W. Hayes, e K. Nidifèr, “CMMI scampism distilled: appraisals for process improvement,” First. s.l. : Addison-Wesley Professional, 2005.
- [5] Beck, K. e C. Andres, “Extreme Programming explained: embrace change,” 2. s.l. : Upper Saddle River: Addison-Wesley, 2005.
- [6] Benites, M. e M. I. Cagnin, “Técnicas para Avaliação de Agilidade em Métodos de Desenvolvimento: Uma Revisão Sistemática,” Relatório Técnico. Facom/UFMS. 2011, URL: [www.dct.ufms.br/index.php?section=report](http://www.dct.ufms.br/index.php?section=report).
- [7] Cohn, M., “Comparative Agility” disponível URL://comparativeagility.com/files/ADP08\_AssessingYourAgility.pdf
- [8] Cockburn, A., “Agile Software Development,” 2. s.l. : Pearson Education India, 2002.
- [9] Conboy, K., “Toward a conceptual framework of agile methods: a study of agility in different disciplines,” s.l. : ACM Press, pp. 37–44, 2004.
- [10] Highsmith, J. e M. Fowler, “The Agile Manifesto,” Software Development Magazine, Vol. 9, pp. 29–30, 2001.
- [11] Highsmith, J., “Agile software development ecosystems,” s.l. : Addison-Wesley Longman Publishing Co., Inc., 2002.
- [12] Khoshroo, B.M. e H. Rashidi, “Towards a Framework for Agile Management Based on Chaos and Complex System Theories,” 2009. pp. 291–292, 2009.
- [13] Lappo, P.; Andrew, H. C.T. (2004). Assessing Agility. Extreme Programming and Agile Processes in Software Engineering Journal, v. 3092, pp. 331–338. Springer.
- [14] OpenUP – OPEN UNIFIED PROCESS. EPF – Eclipse Process Framework Project. URL: [http://www.eclipse.org/downloads/download.php?file=/technology/epf/OpenUP/library/OpenUP\\_library\\_PT\\_1.0\\_20070801.zip/2011](http://www.eclipse.org/downloads/download.php?file=/technology/epf/OpenUP/library/OpenUP_library_PT_1.0_20070801.zip/2011).
- [15] PMI, “A Guide to the Project Management Body of Knowledge (PMBOK Guide): An American National Standard ANSI/PMI,” 4. s.l. : Project Management Institute, 2008.
- [16] Qumer, A. e B. Henderson-Sellers, “A framework to support the evaluation, adoption and improvement of agile methods in practice” s.l. : Elsevier Science Inc., The Journal of Systems and Software, Vol. 81, pp. 1899–1919, 2008.
- [17] Qumer, A. e B. Henderson-Sellers, “Crystallization of agility back to basics,” Proceedings of the First International Conference on Software and Data Technologies, INSTICC Press, Vol. 2 , pp121–126, 2006.
- [18] Rumbaugh, J., “What is a Method?” s.l. : SIGS Publication Incorporation, Journal of Object Oriented Programming, Vol. 8, pp. 10–10. , 1995.
- [19] SOFTEX – Associação para a Promoção da Excelência do Software Brasileiro – SOFTEX. MPS.BR – Guia de Avaliação MA–MPS, junho 2011. URL: [www.softex.br/mpsbr/\\_guias/default.asp](http://www.softex.br/mpsbr/_guias/default.asp). Acesso em Agosto/2011b.
- [20] Sutherland, J., “Nokia Test”, (2009), URL: <http://jeffsutherland.com/nokiatest.pdf>
- [21] Sutherland, J., C. R. Jakobsen e K. Johnson, “Scrum and CMMI Level 5: The Magic Potion for Code Warriors,” p. 466, 2008.
- [22] Sutherland, J. e K. Schwaber, “The Scrum Guide – A body of Knowledge,” URL: <http://www.scrum.org/scrumguides>, 2010.

- [23] Taromirad, M. e R. Ramsin, "An Appraisal of Existing Evaluation Frameworks for Agile Methodologies," s.l. : 15th IEEE International Conference and Workshop on the Engineering of Computer Based Systems, pp. 418-427, 2008.
- [24] Taromirad, M. e R. Ramsin, "CEFAM: Comprehensive Evaluation Framework for Agile Methodologies," s.l. : 32nd Annual IEEE Software Engineering Workshop, IEEE. pp. 195-204, 2008.
- [25] Teles, V. M., "Um estudo de caso da adoção das práticas e valores do extreme programming," Dissertação de Mestrado IM-NCE, UFRJ. 2005.
- [26] Williams, L.; W. Krebs, L. Layman, A.I. Anton, P. Abrahamsson, "Toward a framework for evaluating extreme programming", IEE Seminar Digests, pp. 11-20, n° 920,(2004)