

Agile DMAIC: Um Método para Avaliar e Melhorar o Uso do Scrum em Projetos de Software

Thiago Ferraz V. da Cunha, Rossana M. C. Andrade*

¹Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)
Mestrado em Ciência da Computação (MDCC)
Universidade Federal do Ceará (UFC), Fortaleza – Ceará – Brasil

{thiagoferraz, rossana}@great.ufc.br

Abstract. *Scrum embraces continuous improvement and, naturally, tends to evolve its own implementation over time. However, it can be misused, delaying its benefits in productivity and quality. There should be then a periodic assessment on how its principles and practices are understood and addressed to avoid its misuse. Some maturity models allow measuring how agile practices have been adopted, however, these models are usually general what leads to more time to apply them in the project. Then, this article proposes the Agile DMAIC, a method that aims to identify and address issues in the use of Scrum, providing qualitative and quantitative measurements on how far its application is from the best possible implementation. The Agile DMAIC is applied in seven real projects for evaluation and their results are also discussed in this article.*

Resumo. *Scrum abrange a melhoria contínua e, naturalmente, tende a evoluir a sua própria implementação ao longo do tempo. No entanto, ele pode ser mal utilizado, atrasando seus benefícios de produtividade e qualidade. Para que isso não ocorra, é necessária a avaliação periódica de como seus princípios e práticas são compreendidos e implementados. Alguns modelos de maturidade permitem medir a forma de adoção de práticas ágeis, entretanto, esses modelos usualmente são generalistas e por isso muito extensos. Então, este artigo propõe o Agile DMAIC, um método que visa identificar e solucionar problemas no uso do Scrum, fornecendo medições qualitativas e quantitativas do quão distante a sua aplicação está de sua melhor implementação possível. Para avaliar o Agile DMAIC, ele é aplicado em sete projetos reais e seus resultados são também discutidos neste artigo.*

1. Introdução

O *Scrum* é um processo iterativo e incremental utilizado na gestão e desenvolvimento de projetos, que define uma estrutura base a partir de fundamentos, princípios, valores e práticas, e inclui orientações para a sua utilização [Schwaber 1995][Beedle et al. 2000]. A essa estrutura base são adicionadas práticas de engenharia e gestão para a instânciação do *Scrum* conforme o contexto e a necessidade de cada projeto [Rubin 2012].

A implementação do *Scrum*, portanto, é considerada única por projeto e sua adequada utilização depende de fatores como a experiência do time e do cliente, as competências desse time, a natureza do projeto, entre outros [Cohn 2009]. Esses fatores influenciam a forma na qual o *Scrum* deve ser instanciado para que seus benefícios sejam alcançados. Não é raro, todavia, casos em que o time do projeto sugere não mais utilizar o *Scrum*, devido a perdas de qualidade e produtividade durante a implantação ou nas primeiras iterações

*Bolsista de produtividade DT 2 (CNPq): 314021/2009-4

[Moe and Dingsøyr 2008][Balkanski 2008]. Conseqüentemente, em alguns desses casos, a culpa de tais resultados é associada a nova metodologia adotada e não ao mau uso da mesma [Crescêncio 2013]. Nesse cenário, o time comumente volta a utilizar a abordagem anterior e uma segunda tentativa de adoção do *Scrum* se torna improvável.

O sucesso da implantação e continuidade com o *Scrum* depende, portanto, de como cada projeto instancia essa metodologia e, além disso, de como essa instanciação é ajustada conforme o projeto evolui. Existem estratégias de adoção do *Scrum* para que os resultados iniciais sejam mais assertivos, por exemplo, selecionar um projeto piloto com uma equipe pequena, que possua um patrocinador engajado e cuja importância seja a necessária para motivar os envolvidos [Cohn 2009]. Entretanto, mesmo após uma implantação bem sucedida, os resultados do projeto podem evoluir lentamente ou até mesmo retroceder.

Nesse contexto, surge a necessidade de gerenciar a implementação do *Scrum*. Para a boa utilização de práticas ágeis, existem modelos de maturidade e *checklists* que possibilitam medir a forma de adoção dessas práticas e, se aplicados corretamente, servem como guia de como melhorar a agilidade nos projetos [Qumer et al. 2007][Sidky et al. 2007]. Entretanto, por serem usualmente generalistas, esses modelos são muito extensos, focando em práticas de várias metodologias ao mesmo tempo, o que dificulta quando o projeto já se utiliza do *Scrum* e deseja inicialmente melhorar o uso dessa metodologia. Além disso, para a avaliação do *Scrum*, há a necessidade de *checklists* mais específicos, que forneçam uma visão de forças e fraquezas no uso dos princípios e práticas ágeis do *Scrum*.

Este trabalho propõe então um método sistemático para avaliação e melhoria do uso do *Scrum* em projetos de *software*. Esse método é chamado *Agile DMAIC* e pode ser utilizado tanto na adoção do *Scrum* quanto na melhoria de sua implementação em projetos que já o utilizam há várias iterações. O *Agile DMAIC* é baseado em princípios e técnicas de gestão da qualidade e visa aumentar o alinhamento da gestão de projeto aos princípios ágeis, com foco na melhoria do uso de princípios e práticas do *Scrum*.

Para apresentar o *Agile DMAIC*, este artigo está estruturado em mais cinco seções. A Seção 2 introduz o *Scrum*, *Six Sigma* e *Lean Six Sigma*. Os trabalhos relacionados ao uso de modelos de maturidade e *checklists* para avaliação de práticas ágeis são discutidos na Seção 3. O *Agile DMAIC* é proposto na Seção 4 e os resultados de sua aplicação em sete projetos reais são discutidos na Seção 5. Finalmente, na Seção 6, o artigo é concluído com as principais contribuições e os trabalhos futuros.

2. Fundamentação Teórica

A adoção de práticas ágeis no desenvolvimento de *software* tem se tornado uma resposta a necessidade de se produzir mais rápido, um *software* com melhor qualidade e que satisfaça as necessidades do usuário. A comunidade ágil reconhece as vantagens do uso dessas práticas e o *Scrum* representa uma alternativa frente as metodologias tradicionais de desenvolvimento. Já a adoção do *Six Sigma* e *Lean Six Sigma* vem crescendo em organizações de Tecnologia da Informação, onde são utilizados na melhoria de processos de desenvolvimento, colaborando na realização dos objetivos de negócio dessas organizações [Pan et al. 2007]. Essas metodologias são detalhadas nas próximas Subseções.

2.1. *Scrum*

O *Scrum* é uma metodologia ágil para gestão de projetos, desenvolvimento e manutenção de produtos de *software*, que adota uma abordagem empírica e pode ser utilizado em

projetos complexos com, por exemplo, grande quantidade de produtos diferentes, uso simultâneo de várias tecnologias, elevado número de *stakeholders*, necessidade de conhecimentos técnicos especializados, muitas restrições tecnológicas e incertezas associadas [Schwaber 2007b][Schwaber 2007a]. O *Scrum* se baseia nos seguintes fundamentos:

- Transparência - todos os artefatos do projeto precisam estar visíveis aos envolvidos e as decisões e definições do projeto devem ser compartilhadas de forma efetiva;
- Inspeção - os artefatos são continuamente inspecionados e os problemas encontrados devem ser resolvidos o mais rápido possível; e
- Adaptação - a melhoria contínua deve ser executada em todas as frentes do projeto, com foco na melhoria da produtividade do time de desenvolvimento e da qualidade do produto.

O *Scrum* identifica três papéis: o *Product Owner*, que representa o cliente e é responsável pela gestão do escopo; o *Scrum Master*, que é o líder-servidor do time e responsável por garantir que o *Scrum* seja compreendido e seguido, e o Time de Desenvolvimento, que realiza o desenvolvimento propriamente dito. Os principais artefatos e eventos do *Scrum* são descritos a seguir:

- *Product Backlog* - conjunto completo dos requisitos conhecidos do produto, com os itens priorizados pelo *Product Owner*;
- *Sprint* - janela de tempo que corresponde a uma iteração;
- *Sprint Backlog* - contém as atividades planejadas pelo Time de Desenvolvimento para atender aos itens priorizados do *Product Backlog*;
- *Sprint Planning* - reunião de planejamento da *Sprint* realizada pelo Time de Desenvolvimento para a execução da *Sprint*;
- *Daily Scrum* - reunião envolvendo o time e o *Scrum Master* onde cada participante informa o que fez, o que fará e quais são os impedimentos que estão atrapalhando a execução das atividades;
- Revisão da *Sprint* - reunião em que o *Product Owner* revisa os *softwares* ou produtos entregues e aprova ou não a *Sprint*; e
- Retrospectiva da *Sprint* - reunião em que o time reporta os pontos positivos e negativos da *Sprint* passada, identificando as ações necessárias para que os problemas encontrados sejam evitados nas *Sprints* futuras.

Embora existam muitas informações e treinamentos sobre o *Scrum*, a adoção dessa metodologia é particular ao projeto e à organização e, dessa forma, cada implantação torna-se um desafio único e que não deve ser subestimado. Surge, portanto, a necessidade de gerenciar adequadamente essa adoção de acordo com as necessidades de cada projeto, bem como gerenciar a utilização do *Scrum* a médio e longo prazo.

2.2. Six Sigma e Lean Six Sigma

O *Six Sigma* é uma metodologia quantitativa que objetiva aumentar o desempenho e a lucratividade das empresas. Criado em meados de 1980, o *Six Sigma* é amplamente utilizado por seus resultados comprovados e é considerado como “a metodologia de qualidade do século XXI” [Werkema 2012]. Essa metodologia melhora os resultados financeiros e de qualidade das empresas a partir da identificação e remoção das causas de defeitos e da diminuição da variabilidade de produtos, processos e serviços.

O *Six Sigma* utiliza um conjunto de métodos e ferramentas de gestão de qualidade, incluindo métodos estatísticos, para realizar a análise detalhada dos processos, a fim de descobrir as causas dos defeitos. Isso é conseguido através do uso de duas submetodologias,

o *DMAIC* e *DMADV*. O primeiro é utilizado para a melhoria de processos existentes, enquanto o segundo é utilizado para o desenvolvimento de novos processos ou produtos em níveis de qualidade *Six Sigma*. As fases dessas submetodologias são sumarizadas na Tabela 1 [George 2003]. O *Six Sigma* cria uma infraestrutura de pessoas que são especialistas nesses métodos. Cada projeto *Six Sigma* segue uma sequência definida de passos e tem metas quantificadas de redução de custos financeiros (e aumento do lucro).

Tabela 1. Sub-metodologias DMAIC e DMADV, adaptado de [George 2003].

	Fase	Descrição
DMAIC	Definir (Define)	Definir os objetivos do projeto e de clientes (interno e externo)
	Medir (Measure)	Medir o processo para determinar o desempenho atual
	Analisar (Analyze)	Analisar e determinar as causas de defeitos
	Melhorar (Improve)	Melhorar o processo a partir da eliminação de defeitos
	Controlar (Control)	Controlar o desempenho do processo após implantação de melhorias
DMADV	Definir (Define)	Definir os objetivos do projeto e de clientes (interno e externo)
	Medir (Measure)	Determinar as especificações do cliente
	Analisar (Analyze)	Analisar o processo quanto ao atendimento das necessidades do cliente
	Projetar (Design)	Realizar o projeto do processo com foco nas necessidades do cliente
	Verificar (Verify)	Verificar se o desempenho e a capacidade do novo processo atendem às necessidades do cliente

O *Lean Six Sigma (LSS)* é uma metodologia para definição e melhoria de produtos, processos e serviços com foco na redução de defeitos ou falhas, na eliminação da variação e dos desperdícios, priorizando, de forma planejada e objetiva, a obtenção de resultados de qualidade e financeiros [George 2003]. Essa metodologia consiste na integração dos princípios e técnicas do *Lean* [Poppendieck 2005] e do *Six Sigma* e potencializa os benefícios dessas duas metodologias, aliando os ganhos em velocidade do *Lean* aos ganhos em qualidade do *Six Sigma*.

O *LSS* identifica os seguintes papéis: Liderança Executiva, que é a alta gerência responsável pela visão estratégica da organização, a qual será utilizada para a implantação e execução do *LSS*; *Champions*, que assumem a responsabilidade pela implantação da metodologia na organização; *Master Black Belts*, que atuam como orientadores internos; *Black Belts*, que aplicam a metodologia em projetos específicos e *Green Belts*, que se ocupam da execução dos projetos *LSS*, juntamente com suas outras responsabilidades, sob a orientação de *Black Belts*.

3. Trabalhos Relacionados

Nesta Seção são analisados os trabalhos relacionados à adoção, melhoria e avaliação de práticas ágeis, modelos de maturidade e *checklists*, bem como sobre o uso do *DMAIC* em projetos de *software*.

Sidky *et al.* (2007) propõem o *Agile Adoption Framework (AAF)* como um processo para adoção de práticas ágeis. O *AAF* é composto de dois componentes, o Índice de Medição Ágil, que identifica o potencial ágil alcançável por projeto e organização; e o Processo de 4-Estágios, que auxilia a avaliar se o projeto e a organização estão aptos a avançar na adoção de práticas ágeis e a identificar quais práticas devem ser adotadas. Para a avaliação de agilidade são utilizados aproximadamente 300 indicadores, que são questões do tipo escala de classificação (uma única classificação é selecionada para a questão, entre uma sequência gradual de classificações possíveis). Esses indicadores são agrupados por público alvo (desenvolvedores, gerentes e avaliadores da qualidade) e, por observação e entrevistas, são classificados para medir o atendimento das características relacionadas aos princípios e

práticas ágeis. Se por um lado, essa quantidade de indicadores torna o método abrangente, por outro lado, o torna extenso, demandando esforço significativo em sua utilização. Além disso, O AAF não se restringe a uma metodologia ágil específica e, no entanto, não há restrições em utilizá-lo, por exemplo, com o *Scrum*.

Vode e Sutherland (2008) propõem o *Nokia Test* para a avaliação do uso do *Scrum* em projetos de *software*. Esse *checklist* foi elaborado em 2005 e, inicialmente, focava no desenvolvimento ágil. Em 2007, ele foi ajustado para avaliar as práticas do *Scrum* e, em 2008, Jeff Sutherland, cocriador do *Scrum*, estendeu o mesmo de forma a fornecer uma pontuação do uso do *Scrum*. O *Nokia Test* possui 50 itens e foca na avaliação do uso de artefatos e papéis do *Scrum*. Entretanto, ele é pouco abrangente e não avalia, por exemplo, parte dos princípios e valores, em relação aos eventos, avalia apenas a *Sprint* e, sobre os papéis, não há itens a respeito do *Scrum Master*.

Kniberg (2009) propõe o *CRISP Scrum-Checklist* como guia para os primeiros passos com o *Scrum* e para avaliar a implementação dessa metodologia nos projetos. Esse *checklist* foi elaborado em 2008 e tem como principal público-alvo as equipes iniciantes no *Scrum*. O *CRISP Scrum-Checklist* possui 78 itens e foca na avaliação de eventos e papéis. Esses itens são organizados por seções, cada uma associada implicitamente a uma prioridade. Ao contrário do *Nokia Test*, o *CRISP Scrum-Checklist* é bastante abrangente. Entretanto, por incluir a avaliação de práticas recomendadas mas não obrigatórias ele se torna desnecessariamente extenso.

Bezerra *et al.* (2007) propõem o *MiniDMAIC* como uma simplificação do *DMAIC* para tratar causas e resolução de problemas em projetos de *software*. Essa simplificação tem como objetivo reduzir a duração do *DMAIC* e, por conseguinte, o seu custo, associando-o ao tratamento de riscos dos projetos. O *MiniDMAIC* é uma simplificação do *DMAIC* voltada ao desenvolvimento de *software*, entretanto, seu foco é o tratamento de problemas gerais, como desvios na produtividade, prazo, densidade de projetos, entre outros.

Cunha *et al.* (2011) apresentam uma abordagem de integração do *Scrum* ao *Lean Six Sigma (SLeSS)*. Essa abordagem de gestão de projetos e melhoria de processos é utilizada no desenvolvimento de customizações de *software* para celulares. O *SLeSS* define mecanismos para a integração dessas metodologias e um deles é a aplicação de princípios e técnicas do *Lean Six Sigma* para melhorar o uso do *Scrum*. Entretanto, a ausência de documentação e o uso do *DMAIC* sem simplificações torna esse mecanismo sobre a avaliação e melhoria do *Scrum* de difícil reuso e extenso.

O *Agile DMAIC*, método proposto neste trabalho, é resultado da continuação da pesquisa apresentada em [Cunha *et al.* 2011] e tem como objetivo aprimorar a abordagem *SLeSS* com foco na avaliação das práticas e princípios do *Scrum*. Em relação aos outros trabalhos apresentados, o *Agile DMAIC* pretende ser menos extenso, como o *MiniDMAIC* [Bezerra *et al.* 2007], mas sendo específico para avaliação e melhoria do *Scrum*. Vale ressaltar que ele pode ser utilizado em conjunto com o *MiniDMAIC*, mas possuem objetivos distintos.

4. Agile DMAIC

Nesta Seção apresentamos o *Agile DMAIC*, que é um método sistemático proposto neste trabalho para avaliar e melhorar o uso do *Scrum* em projetos de *software*. Esse método pode ser utilizado tanto na adoção do *Scrum* quanto na melhoria de sua implementação. Os objetivos do *Agile DMAIC* podem ser resumidos a seguir:

- ✓ Possibilitar a gestão quantitativa da implementação do *Scrum*;
- ✓ Avaliar o uso de princípios e práticas do *Scrum* por projeto e a nível institucional; e
- ✓ Identificar e tratar as causas de problemas para a evolução da implementação do *Scrum*, considerando o contexto e a necessidade dos projetos e clientes.

O *Agile DMAIC* utiliza princípios e técnicas do *Lean Six Sigma* e foca na execução, de forma iterativa e incremental, de uma versão simplificada do *DMAIC* [Tayntor 2007]. A execução do *Agile DMAIC* é realizada em paralelo à execução de *Sprints* do *Scrum* conforme apresentado na Figura 1. À medida que problemas no uso do *Scrum* são identificados e tratados, as melhorias são implantadas a cada *Sprint* do *Agile DMAIC*.

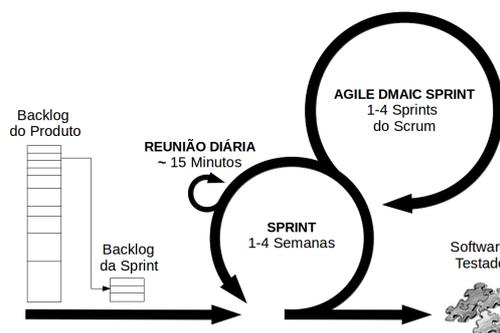


Figura 1. Execução Iterativa e Incremental do *Agile DMAIC*

Enquanto as *Sprints* do *Scrum* tem tamanho de 1 a 4 semanas [Schwaber and Sutherland 2011], as *Sprints* de execução do *Agile DMAIC* são de 1 a 4 vezes o tamanho de uma *Sprint* do *Scrum*. Esse tamanho foi assim definido com base nos experimentos realizados no estudo de caso deste trabalho. Por exemplo, em um projeto cuja *Sprint* do *Scrum* é de 2 semanas, uma *Sprint* do *Agile DMAIC* pode ser de 2 a 8 semanas, de forma que no máximo a cada 4 *Sprints* do *Scrum*, problemas no uso dessa metodologia sejam identificados, tratados e melhorias sejam implantadas com seus respectivos planos de controle. Também, por analogia, assim como o objetivo principal de uma *Sprint* do *Scrum* é a entrega de *software* testado, o objetivo de uma *Sprint* do *Agile DMAIC* é a implantação de melhorias no uso de princípios e práticas do *Scrum*.

Uma visão geral do *Agile DMAIC* é apresentada na Figura 2 e os detalhes são apresentados no decorrer desta Seção. O *Agile DMAIC* é composto de três componentes:

- Índice de Medição do Nível do *Scrum* (*IMeNS*), que avalia a adoção das práticas do *Scrum* por projeto e a nível institucional;
- *Checklist* de Avaliação do *Scrum* (*CAS*), utilizado para coletar a percepção das equipes quanto ao uso de princípios e práticas do *Scrum*; e
- *DMAIC* [Tayntor 2007], uma submetodologia do *Lean Six Sigma* [George 2003] voltada a melhoria de processos. No método proposto neste trabalho, o *DMAIC* é utilizado para a definição de metas, identificação e análise de causas raízes, implantação e controle de melhorias para possibilitar a gestão quantitativa da implementação do *Scrum*.

O *Agile DMAIC* se baseia em coletar periodicamente a percepção da equipe do projeto (*Product Owner*, *Scrum Master* e *Time*) em relação ao uso do *Scrum* e, a partir dessa percepção, identificar possíveis problemas em sua implementação. O resultado da aplicação do *CAS* é utilizado como insumo na identificação de problemas e, além desse *checklist*, as

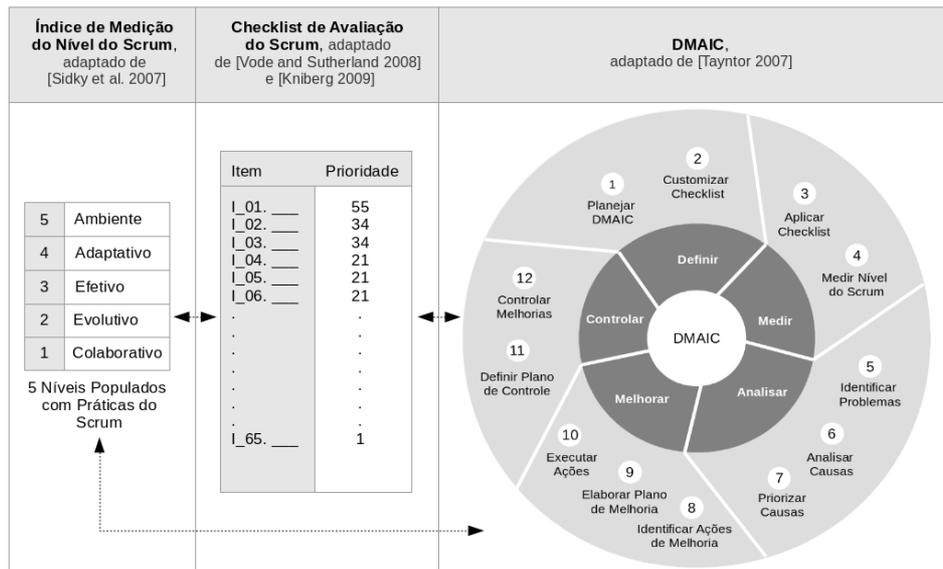


Figura 2. Visão Geral do *Agile DMAIC*

Reuniões de Revisão e Retrospectivas também podem ser fontes de identificação de problemas. A partir da aplicação do CAS e da análise da coleta é possível classificar o nível do *Scrum*, utilizando o *IMeNS*, e identificar onde a equipe pode atuar para melhorar seus resultados com essa metodologia. Uma vez identificados os problemas, é necessário analisar as causas, definir e implantar ações de melhoria, antecipando os benefícios do *Scrum*, como a melhoria na produtividade do time e na qualidade do produto [Dybå and Dingsøyr 2008].

Dessa forma, o *Agile DMAIC* é utilizado como ferramenta para a definição de uma meta de melhoria no uso do *Scrum*, na análise de seu estado atual, na identificação e implantação de melhorias e no controle das mesmas para que o uso dessa metodologia continue em constante evolução.

4.1. Índice de Medição do Nível do Scrum (IMeNS)

O *IMeNS* é utilizado para classificar os projetos quanto ao nível de adoção de princípios e práticas do *Scrum*. Esse índice é uma especialização do Índice de Medição Ágil especificado no *Agile Adoption Framework (AAF)* [Sidky et al. 2007], a partir de sua simplificação ao remover as práticas e conceitos não relacionados ao *Scrum*. Como exemplo, o *Test Driven Development* [Beck 2002] e a Programação em Pares [Beck and Andres 2004] foram removidos do mapeamento proposto em [Sidky et al. 2007] pois, embora não haja restrições em usá-los com o *Scrum*, não são especificados como práticas dessa metodologia. Dessa forma, o *Agile DMAIC* adaptou apenas o Índice de Medição Ágil definido por Sidky *et al.* (2007) e não se utilizou, por exemplo, dos indicadores e demais componentes do *AAF*.

Os níveis definidos no *IMeNS* são, do 1 ao 5 (veja Figura 2), respectivamente: Colaborativo, promove a comunicação e colaboração entre os envolvidos do projeto; Evolutivo, foca na entrega contínua de *software*; Efetivo, objetiva o desenvolvimento de *software* testado e de alta qualidade, realizado de forma eficiente e efetiva; Adaptativo, estabelece as práticas que atuam para agilizar as respostas às mudanças; e Ambiente, que promove um ambiente altamente produtivo. Esses níveis são acessados a partir de práticas associadas a princípios ágeis do *Scrum* cujo mapeamento é apresentado na Tabela 2. Como exemplo, para acessar o nível 2 é necessário que o projeto atenda as práticas do nível 1 (P1 a P6) e

do nível 2 (P7 a P9). Da mesma forma, para acessar ao nível 5 é necessário que o projeto atenda a todas as 17 práticas mapeadas nessa Tabela.

Tabela 2. Os 5 Níveis do IMeNS Populados com Práticas do Scrum, Adaptado de [Sidky et al. 2007]

	Princípios Ágeis do Scrum				
	<i>Abraçar mudanças para entregar valor ao cliente</i>	<i>Planejar e entregar software com frequência</i>	<i>Focar nas pessoas</i>	<i>Excelência técnica</i>	<i>Colaboração com o cliente</i>
Nível 5 Ambiente	[P16] Iterações orientadas ao cliente, [P17] Feedbacks contínuos de sua satisfação				
Nível 4 Adaptativo		[P12] Entregas menores e frequentes, [P13] Planejamento adaptativo		[P14] Monitorar o progresso diário	[P15] Cliente prontamente acessível
Nível 3 Efetivo			[P10] Times auto-organizáveis, [P11] Comunicação face a face		
Nível 2 Evolutivo		[P7] Entregas contínuas, [P8] Planejamentos em níveis distintos		[P9] Monitorar o status da iteração	
Nível 1 Colaborativo	[P1] Reflexão e ajuste do processo	[P2] Planejamento colaborativo	[P3] Equipes colaborativas, [P4] responsáveis e motivadas	[P5] Tarefas executadas de forma voluntária	[P6] Compromisso do cliente no trabalho com a equipe

4.2. Checklist de Avaliação do Scrum (CAS)

O CAS¹ foi elaborado a partir do *Nokia Test* [Vode and Sutherland 2008] e do *CRISP Scrum-Checklist* [Kniberg 2009] com o intuito de consolidar o melhor desses *checklists* e solucionar os pontos de melhoria dos mesmos. Esse novo *checklist* é aplicado à equipe do projeto (*Product Owner*, *Scrum Master* e o Time) para obter, a partir da percepção dessa equipe, as forças e fraquezas da atual implementação do *Scrum*.

O CAS é formado de 65 itens (I_01 a I_65) distribuídos em treze seções, ordenadas de forma decrescente quanto a prioridade das mesmas em relação aos benefícios do *Scrum*. Cada item possui uma priorização (1 a 55, utilizando a sequência de *Fibonacci*, sendo 1 a menor e 55 a maior prioridade) e essa priorização foi predefinida com base na experimentação do *Agile DMAIC* nos projetos reais utilizados como estudo de caso neste trabalho e pode ser customizada pela equipe. A Figura 3 apresenta algumas seções e itens do CAS. Por exemplo, a seção *Planejamento do Sprint* possui 8 itens (I_04 a I_11). O primeiro item dessa seção verifica se há planejamento a cada iteração e possui prioridade 34 (P_34).

O CAS foca em avaliar como são utilizados os eventos e artefatos do *Scrum*. Entretanto também abrange bem os papéis, fundamentos, valores e princípios dessa metodologia. Para uma comparação entre os *checklists* analisados neste trabalho, os gráficos de abrangência de práticas do *Scrum* são apresentados na Figura 4 para o *Nokia Test*, *CRISP Scrum-*

¹[http://www.great.ufc.br/CAS/Checklist_de_Avaliacao_do_Scrum\(CAS\).pdf](http://www.great.ufc.br/CAS/Checklist_de_Avaliacao_do_Scrum(CAS).pdf)

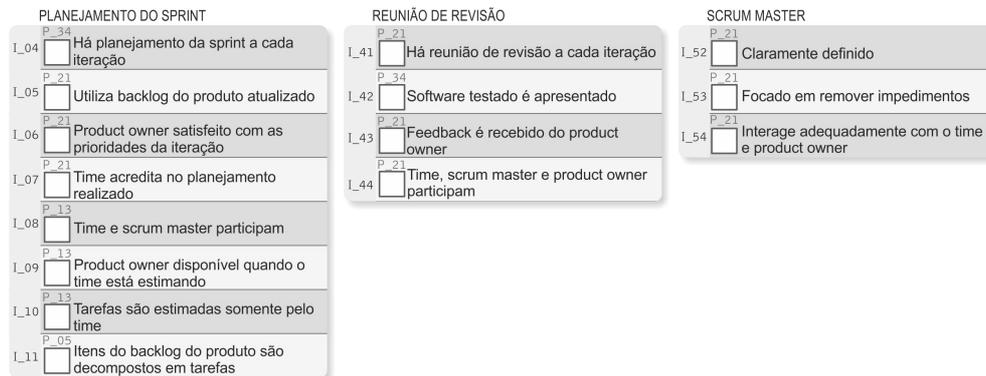


Figura 3. Algumas Seções e Itens do Checklist de Avaliação do Scrum (CAS)

Checklist e o CAS, respectivamente. O *Nokia Test* possui itens a respeito do *Backlog* do Produto, *Sprint*, Plano de Lançamentos, Gráfico de *Burndown*, *Product Owner*, Time, Fundamentos, Valores e Princípios. Entretanto, algumas práticas do *Scrum* não são abrangidas por esse checklist, como exemplo, não há itens sobre a Retrospectiva, Reunião de Revisão, *Scrum Master*, entre outros. Já o *CRISP Scrum-Checklist* não abrange diretamente apenas um dos fundamentos (inspeção) e o *CAS* abrange todas as categorias identificadas.

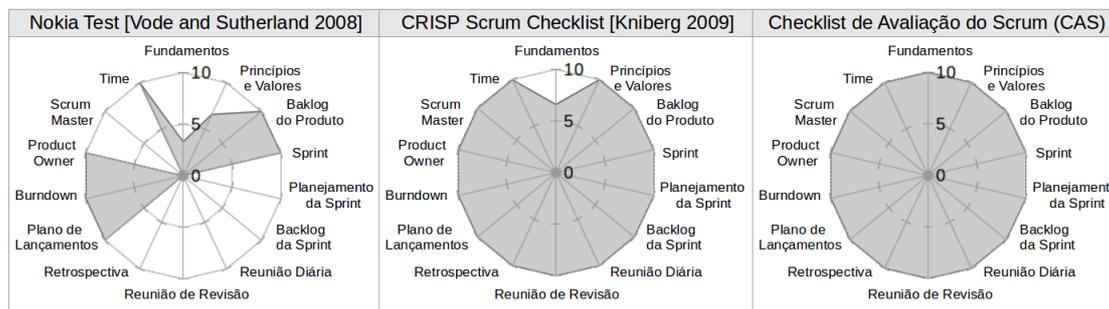


Figura 4. Abrangência dos Checklists Quanto às Práticas Ágeis do Scrum

A utilização do *CAS*, em conjunto com o *IMeNS*, possibilita classificar o nível do *Scrum* e serve como guia na adoção e melhoria do uso dessa metodologia. O atendimento das práticas do *IMeNS* é verificado a partir da aplicação do *CAS* à equipe do projeto e do mapeamento de práticas do *IMeNS* aos itens do *CAS*, conforme apresentado na Tabela 3. Como exemplo, a prática *Feedbacks contínuos da satisfação do cliente* (P17) é verificada a partir do atendimento do projeto aos itens do *CAS* do I_38 ao I_44.

4.3. DMAIC

O *Agile DMAIC* propõe uma simplificação do *DMAIC* [Tayntor 2007] para a avaliação e melhoria do uso do *Scrum* em projetos de *software*. Nessa simplificação, o *CAS* é utilizado para a coleta da percepção da equipe e o nível do *Scrum* é calculado a partir dessa percepção e da priorização de itens do *CAS*, customizado pela equipe. As fases e passos do *DMAIC* utilizado pelo *Agile DMAIC* são apresentados na Tabela 4.

A customização dos itens e da priorização do *CAS* é realizada de acordo com o contexto do projeto e cliente e visa identificar o máximo nível possível do *Scrum* para o projeto, isso porque podem existir restrições impostas pelo negócio, cliente, ou pela organização que

Tabela 3. Mapeamento de Práticas do IMeNS aos Itens do CAS

	Prática <-> Itens
Nível 5 Ambiente	P17 <-> I_38 a I_44
	P16 <-> I_02, I_05, I_06, I_32, I_33, I_49
Nível 4 Adaptativo	P15 <-> I_06, I_09, I_38 a I_40, I_43
	P14 <-> I_16, I_18, I_30, I_37, I_57, I_59 a I_65
	P13 <-> I_04, I_18, I_36, I_49, I_55
	P12 <-> I_01, I_12, I_13, I_19, I_20, I_22, I_23
Nível 3 Efetivo	P11 <-> I_08, I_09, I_40, I_41, I_43, I_44, I_48, I_54, 59, I_60, I_62
	P10 <-> I_25 a I_29, I_34, I_35, I_51 a I_54, I_56, I_58
Nível 2 Evolutivo	P9 <-> I_16, I_18, I_24, I_30, I_37, I_63, I_65
	P8 <-> I_04, I_05, I_09, I_11, I_17, I_18, I_21, I_24, I_33, I_35, I_49, I_50, I_55
	P7 <-> I_01, I_13, I_19, I_20
Nível 1 Colaborativo	P6 <-> I_09, I_38 a I_40
	P5 <-> I_04, I_07, I_08, I_10, I_11, I_17, I_24, I_35, I_56, I_58
	P4 <-> I_07, I_48, I_52 a I_54
	P3 <-> I_08, I_31, I_35, I_44, I_48, I_52 a I_54
	P2 <-> I_17, I_35, I_37, I_49 a I_51, I_55 a I_57
	P1 <-> I_03, I_14, I_15, I_41 a I_47, I_50

Tabela 4. Fases e Passos do DMAIC Simplificado pelo Agile DMAIC

	Passo	Descrição
Definir	1 Planejar DMAIC	Elaborar plano de execução do DMAIC, especificando a meta de melhoria do nível do Scrum por projeto e a nível institucional, o escopo de atuação (projetos alvo), o responsável (Green Belt) pelo DMAIC e demais envolvidos
	2 Customizar Checklist	Definir a frequência de aplicação do CAS e customizá-lo (revisão de itens e priorização) de acordo com o projeto e cliente. Por exemplo, por necessidade do negócio, o cliente define que o software só seja implantado em seu ambiente ao final do projeto e, dessa forma, o item I_13 do CAS deixa de ser prioridade
Medir	3 Aplicar Checklist	O CAS é aplicado à equipe do projeto (Product Owner, Scrum Master e o Time). Seções do CAS completamente atendidas podem ser excluídas em aplicações seguintes até que haja indicativo de regressão
	4 Medir Nível do Scrum	A coleta é validada e itens com menos de 25% e mais de 75% de pontuação, em relação ao tamanho da equipe, são definidos como não atendidos e atendidos, respectivamente. Itens fora desses intervalos precisam ser revisados em conjunto com a equipe. A classificação do nível do Scrum é acessada pelos mapeamentos das Tabelas 2 e 3. O indicador do nível do Scrum é a razão da soma de priorizações de itens atendidos pela soma de todas as priorizações, multiplicada por 10, ou seja, $10 * \frac{\sum P(\text{atendidos})}{\sum P}$
Analisar	5 Identificar Problemas	Os itens não atendidos pelo projeto são considerados problemas e são priorizados conforme customizados pela equipe no CAS
	6 Analisar Causas	De posse da coleta e de problemas já identificados, é possível discernir entre causas relacionadas ao entendimento da equipe, desconhecimento de princípios e práticas, falha na comunicação ou demais causas, as quais podem ser identificadas com uso de técnicas do Lean Six Sigma
	7 Priorizar Causas	As causas são priorizadas a partir das análises realizadas, da priorização do CAS e dos mapeamentos de níveis e práticas das Tabelas 2 e 3
Melhorar	8 Identificar Ações de Melhoria	Uma vez identificadas as causas raízes dos problemas, a definição de ações de melhoria é realizada com a equipe
	9 Elaborar Plano de Melhoria	O plano de melhoria estabelece a priorização das ações, os prazos e responsáveis pela execução e acompanhamento dessas ações
	10 Executar Ações	As ações são executadas conforme estabelecido no plano de melhoria
Controlar	11 Definir Plano de Controle	Avaliar riscos, definir gatilhos, ações, responsáveis e prazos, para assegurar que as melhorias já implantadas sejam duradouras
	12 Controlar Melhorias	Controlar o projeto conforme definido no plano de controle

dificultem ou inviabilizem o uso de algumas práticas do *Scrum*. Essas alterações no *DMAIC* objetivam definir um indicador de desempenho, o nível do *Scrum*, para gerir a implementação do *Scrum* a partir do *DMAIC*.

Enquanto o *DMAIC* é tradicionalmente implementado a partir do modelo em cascata (o progresso de uma fase para a próxima se dá de uma forma puramente sequencial), o *Agile DMAIC* se propõe a utilizá-lo de forma iterativa e incremental, possibilitando que a duração de seu ciclo seja reduzida para 1 a 16 semanas (1 a 4 *Sprints* do *Scrum*), potencializando a antecipação da implantação de melhorias em relação ao uso tradicional do *DMAIC*.

5. Resultados

Os projetos reais utilizados como estudo de caso deste trabalho são desenvolvidos em laboratórios de P&D&I do Grupo de Redes de Computadores, Engenharia de *Software* e Sistemas (GREat)¹ da Universidade Federal do Ceará (UFC). Um resumo da caracterização dos projetos e de suas principais dificuldades é apresentado na Tabela 5. Esses projetos são voltados à Pesquisa (P), Desenvolvimento (D) e Manutenção (M) de aplicações para *smartphones*. Antes do *Agile DMAIC*, os projetos *App1*, *App2*, *App3* e *App7* já utilizavam o *Scrum*, enquanto os demais passaram a adotá-lo a partir do *Agile DMAIC*, com exceção do projeto *App5* que se decidiu por não utilizá-lo. A inclusão do projeto *App5* nesse estudo de caso é realizada para verificar que o *Agile DMAIC* consegue diferenciar entre os projetos que usam o *Scrum* e aqueles que não o utilizam.

Tabela 5. Resumo da Caracterização dos Projetos

Projeto	Tipo	Duração	Equipe	Complexidade	Principais Dificuldades
App1	D	9 meses	8	Alta	Comunicação com o cliente
App2	P&D	9 meses	7	Alta	Comunicação com cliente, complexidade da tecnologia, frequentes mudanças de escopo
App3	P&D	9 meses	8	Alta	Comunicação com o cliente e complexidade da tecnologia
App4	D&M	2 anos	6	Média	Comunicação com o cliente, dificuldade de concepção do produto, frequentes mudanças de prioridade e escopo
App5	P&M	2 anos	26	Alta	Frequentes mudanças de escopo e prazo, grande dependência de informações do produto, alta rotatividade da equipe
App6	P&D	6 meses	6	Alta	Complexidade técnica da pesquisa e do <i>design</i> de interfaces (grande quantidade de dados de entrada)
App7	P&D	9 meses	10	Alta	Comunicação com o cliente, complexidade da tecnologia

A adoção do *Agile DMAIC* foi antecedida por treinamentos teóricos e práticos aos principais envolvidos. Nesses treinamentos, os fundamentos, princípios, valores, papéis, artefatos e eventos do *Scrum* foram apresentados e discutidos, e algumas práticas foram realizadas para a fixação dos novos conceitos. Uma visão da linha de tempo com os marcos de início dos projetos, realização de treinamentos, adoção do *Scrum* e *Agile DMAIC* é apresentada na Figura 5. Como exemplo, o projeto *App6* iniciou em Janeiro, participou do Treinamento de *Scrum*, adotou o *Scrum* e *Agile DMAIC* em Fevereiro e realizou suas avaliações em Fevereiro, Março e Abril de 2014.

Como exemplo do uso do *Agile DMAIC*, os diagramas de Causa e Efeito e Pareto do projeto *App1* para o mês de Janeiro são apresentados na Figura 6. Esses diagramas foram elaborados na fase *Analisar* do *DMAIC* (passos 5, 6 e 7 da Tabela 4) a partir da coleta com o *CAS*, aplicado via *Web* à equipe. O diagrama de Causa e Efeito pode ser utilizado para discutir os principais problemas e causas do atual nível do *Scrum* com a equipe e a

¹<http://great.ufc.br/>

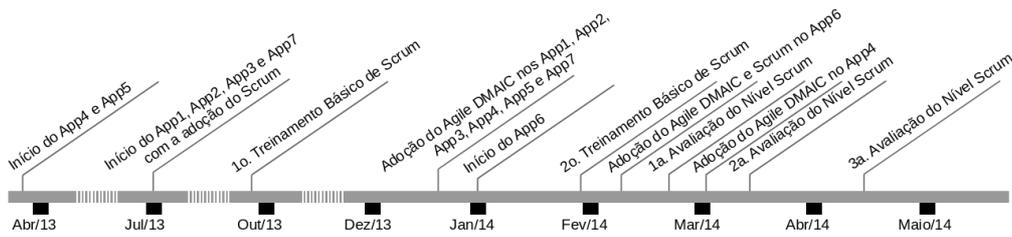


Figura 5. Linha de Tempo dos Treinamentos, da Adoção do Scrum e Agile DMAIC nos Projetos

visão dos principais ofensores, priorizados de acordo com a customização do CAS, pode ser obtida com o uso do diagrama de Pareto. Com esses diagramas do projeto *App1*, por exemplo, é possível identificar que essa equipe deve priorizar a realização das Reuniões de Revisão, a elaboração do Plano de Lançamentos utilizando a Velocidade do time, realizar os Testes de Aceitação a cada iteração, identificar e solucionar as causas do Time está sendo interrompido e controlado de fora, bem como passar a priorizar o *Product Backlog* por valor de negócio.

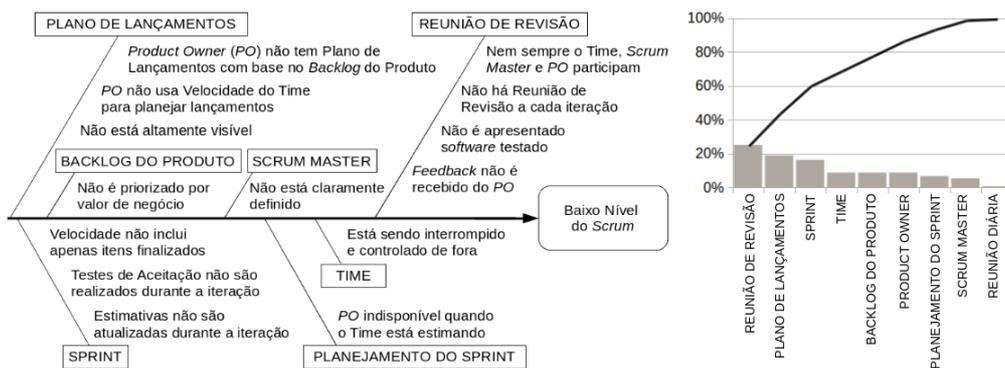


Figura 6. Diagramas de Causa e Efeito e Pareto do Projeto App1 em Janeiro/14

O gráfico com o resultado consolidado das avaliações do projeto *App1* é apresentado na Figura 7. De acordo com a análise da equipe, esse projeto obteve impacto em Março por falta de prioridade do *Product Owner*. Embora tenham sido identificadas e implementadas ações de melhoria como a definição de um *Scrum Master*, a realização de Reuniões de Retrospectiva e a melhor execução de atividades de planejamento (priorização do *Product Backlog*, alocação e estimativas realizadas pelo time), o indicador foi impactado pela ausência desse *Product Owner* nas atividades do projeto.

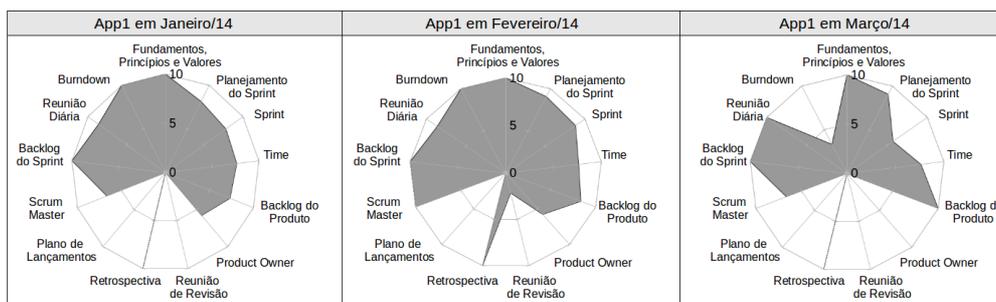


Figura 7. Resultado das Avaliações do Projeto App1

O gráfico de *Boxplot* dos projetos é apresentado na Figura 8. O projeto *App5* obteve o nível mais baixo por não usar o *Scrum* e, quanto aos demais, as análises das equipes confirmaram que o novo método refletiu a realidade do uso do *Scrum* nos meses avaliados. A variação desse indicador do nível nem sempre foi positiva em todos os meses. Entretanto, em análises detalhadas, os problemas identificados com o uso do método e que refletiram em uma variação negativa também foram comprovados pelas equipes. O projeto *App6* obteve maior variação (positiva) a partir de ações de melhoria identificadas com o *Agile DMAIC*, como a necessidade de definições mais claras dos papéis (*Scrum Master* e *Product Owner*) e a resolução de falhas de comunicação (entendimento da equipe), bem como a melhoria no uso de práticas e artefatos.

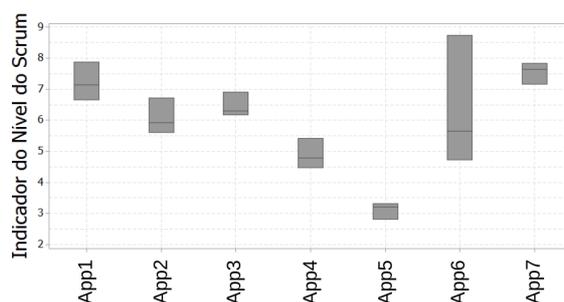


Figura 8. Boxplot dos Projetos

O gráfico de controle dos projetos que utilizam o *Scrum* é apresentado na Figura 9. Esses projetos apresentaram uma média de 6,4 para o indicador do nível do *Scrum* e ficaram abaixo da meta estabelecida ou limite de especificação inferior (8,0), que implica em uma baixa capacidade atual do processo. Essa meta foi definida a partir da análise da primeira coleta dos projetos, referente a Janeiro/14, e da avaliação dos principais ofensores no uso do *Scrum* nesses projetos. Entretanto, mesmo nessa fase inicial, as equipes concordaram que o *Agile DMAIC* foi coerente em suas avaliações e que possibilitou a identificação e resolução de problemas no uso do *Scrum*.

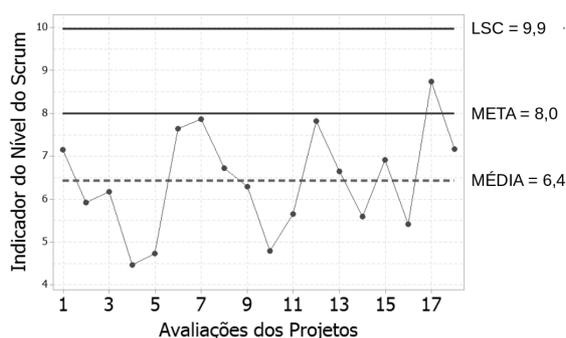


Figura 9. Controle dos Projetos que Utilizam o Scrum

Do ponto de vista dos projetos, o método contribuiu para a melhoria do uso dos princípios e práticas do *Scrum*. A partir dessa melhoria, há a expectativa que as ações já implantadas influenciem positivamente na produtividade da equipe e qualidade dos produtos [Layman et al. 2004][Dybå and Dingsøy 2008]. Entretanto, ainda há a necessidade de mais avaliações com o *Agile DMAIC* nesses projetos para apresentar dados que correlacionem a evolução do nível do *Scrum* com a melhoria da produtividade, qualidade e satisfação do cliente.

6. Conclusão e Trabalhos Futuros

Este artigo propôs o *Agile DMAIC*, um método para avaliação e melhoria do uso do *Scrum*, o qual foi aplicado em projetos reais, de complexidades média a alta, voltados à pesquisa, desenvolvimento e manutenção de *software*. A análise dos resultados do novo método, realizada em conjunto com as equipes, possibilitou identificar que, mesmo com certo tempo e experiência com o *Scrum*, essas equipes necessitavam avaliar de que forma estavam utilizando os princípios e práticas dessa metodologia.

Essa avaliação periódica de forças e fraquezas da implementação do *Scrum*, a partir da percepção da equipe, possibilitou identificar e tratar os principais problemas no uso dessa metodologia. Dessa forma, diante dos modelos de maturidade e *checklists* de avaliação de práticas ágeis analisados neste trabalho, o *Agile DMAIC* se mostrou uma alternativa simples, de fácil aplicação e capaz de produzir bons resultados mesmo no curto prazo.

Como trabalhos futuros, pretende-se aprimorar o *Agile DMAIC* a partir da simplificação de seus componentes. Os mapeamentos de níveis em práticas ágeis e de práticas em itens de seu *checklist* podem ser revisados. Além disso, os passos do *DMAIC* podem ser documentados mais detalhadamente. Uma automação do *Agile DMAIC* pode ser proposta a partir da implementação do *checklist*, mapeamentos e do *DMAIC* com o uso de ferramentas de *workflow*. Além disso, pretende-se avaliar os custos-benefícios da adoção desse novo método a médio e longo prazo, por exemplo, a relação entre o uso do mesmo e a melhoria na produtividade, qualidade e satisfação do cliente, e, por fim, verificar o seu comportamento em outros projetos, com diferentes perfis em relação aos já avaliados, bem como em outras organizações.

Referências

- Balkanski, P. (2008). Team dysfunctions and scrum. Disponível em: <http://www.scrumalliance.org/community/articles/2008/october/team-dysfunctions-and-scrum>. Acesso em: 07 Abril de 2014.
- Beck (2002). *Test Driven Development: By Example*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Beck, K. and Andres, C. (2004). *Extreme Programming Explained: Embrace Change (2Nd Edition)*. Addison-Wesley Professional.
- Beedle, M., Devos, M., Sharon, Y., Schwaber, K., and Sutherland, J. (2000). Scrum: An extension pattern language for hyperproductive software development.
- Bezerra, C. I. M., Coelho, C. C., Gonçalves, F. M. G. S., Pires, C. G. S., and Telles, Gabriela, A. A. B. (2007). Minidmaic: Uma abordagem para análise e resolução de causas em projetos de desenvolvimento de software. VIII Simpósio Brasileiro de Qualidade de Software.
- Cohn, M. (2009). *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley Professional. ISBN 978-0321579362.
- Crescêncio, S. (2013). Disfunções do scrum. Disponível em: <http://oncast.com.br/blog/?p=1515>. Acesso em: 07 Abril de 2014.
- Cunha, T., Dantas, V., and Andrade, R. M. C. (2011). Sless: A scrum and lean six sigma integration approach for the development of software customization for mobile phones. In *Software Engineering (SBES), 2011 25th Brazilian Symposium on*, pages 283 –292.

- Dybå, T. and Dingsøy, T. (2008). Empirical studies of agile software development: A systematic review. *Inf. Softw. Technol.*, 50:833–859.
- George, M. L. (2003). *Lean Six Sigma For Service: How to Use Lean Speed and Six Sigma Quality to Improve Services and Transactions*. McGraw-Hill.
- Kniberg, H. (2009). Scrum checklist – version 2.0. Disponível em: <http://blog.crisp.se/2009/08/14/henrikkniberg/1250265360000>. Acesso em: 14 Abril de 2014.
- Layman, L., Williams, L. A., and Cunningham, L. (2004). Exploring extreme programming in context: An industrial case study. In *Agile Development Conference*, pages 32–41.
- Moe, N. and Dingsøy, T. (2008). Scrum and team effectiveness: Theory and practice. In Abrahamsson, P., Baskerville, R., Conboy, K., Fitzgerald, B., Morgan, L., and Wang, X., editors, *Agile Processes in Software Engineering and Extreme Programming*, volume 9 of *Lecture Notes in Business Information Processing*, pages 11–20. Springer Berlin Heidelberg.
- Pan, Z., Park, H., Baik, J., and Choi, H. (2007). A six sigma framework for software process improvements and its implementation. In *Software Engineering Conference, 2007. APSEC 2007. 14th Asia-Pacific*, pages 446–453.
- Poppendieck, M. (2005). Principles of lean thinking.
- Qumer, A., Henderson-sellers, B., and McBride, T. (2007). Agile adoption and improvement model.
- Rubin, K. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional, 1st edition.
- Schwaber, K. (1995). Scrum development process. In *Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA)*, pages 117–134.
- Schwaber, K. (2007a). *Agile Project Management with Scrum*. Microsoft Press.
- Schwaber, K. (2007b). *The Enterprise and Scrum*. Microsoft Press.
- Schwaber, K. and Sutherland, J. (2011). Guia do scrum, um guia definitivo para o scrum: As regras do jogo. Disponível em: <http://www.scrum.org/Portals/0/Documents/ScrumGuides/ScrumGuide-PortugueseBR.pdf#zoom=100>. Acesso em: 23 Abril 2014.
- Sidky, A., Arthur, J., and Bohner, S. (2007). A disciplined approach to adopting agile practices: the agile adoption framework. *Innovations in Systems and Software Engineering*, 3(3):203–216.
- Tayntor, C. B. (2007). *Six Sigma Software Development; 2nd ed.* Taylor & Francis Ltd, Hoboken, NJ.
- Vode, B. and Sutherland, J. (2008). Scrum but test. Disponível em: <http://antoine.vernois.net/scrumbut/?page=test&lang=en>. Acesso em: 14 Abril de 2014.
- Werkema, C. (2012). *CRIANDO A CULTURA LEAN SEIS SIGMA*. ENGENHARIA. Elsevier.