Um *Survey* sobre Métodos Ágeis e o Pós-Agilismo

Tainá Leal[1], Gleison Santos[2]

[1] Departamento de Informática Aplicada – DIA/CCET
[2] Programa de Pós-Graduação em Informática – PPGI
UNIRIO – Universidade Federal do Estado do Rio de Janeiro
Av. Pasteur 458, Urca – CEP 22290-240 – Rio de Janeiro – RJ – Brasil
{taina.leal, gleison.santos}@uniriotec.br

Abstract. *Contexto:* Após mais de uma década de experiência com os métodos ágeis, críticos do movimento promovem um novo conceito de desenvolvimento ágil, onde os profissionais não estão mais presos a uma só método e sua implementação não mais segue estritamente o que descrevem os livros e guias. *Objetivo*: Esse trabalho propõe analisar os fatores positivos e negativos observados pelos profissionais de TI após a implementação dos métodos ágeis, e identificar de que maneira os métodos e práticas ágeis estão sendo aplicados. *Método*: Foi executado um *survey* descritivo interseccional com foco em profissionais de TI com experiência em métodos ágeis. *Resultados*: A maioria dos profissionais da amostra, após experiências com os métodos ágeis, são capazes de detectar o que funciona e descartar/alterar o que não funciona. A maioria dos respondentes não se prende a um único método. *Conclusões*: O objetivo foi alcançado, uma vez que foi possível identificar, no contexto da amostra, como o uso dos métodos ágeis estão evoluindo para um pensamento dito Pós-Agilista.

Keywords: Métodos ágeis, Pós-agilismo, Práticas ágeis, Processos de software.

1 Introdução

Após mais de uma década de experiência com os métodos ágeis, os profissionais desenvolvedores de software ainda sentem dificuldade em escolher um método que melhor se adequem às suas necessidades. Henninger *et al*. [1] atentavam para a importância de escolher conscientemente um método baseado nas circustâncias de cada projeto e equipe. Uma das estratégias apresentada pelos autores foi saber equilibrar processos de metodologias leves (ágeis) e de metodologias pesadas (tradicionais) para tratar problemas com diferentes ferramentas, linguagens, domínios de aplicações e níveis de experiência entre os desenvolvedores.

Contudo, o decorrer da década foi representado por desenvolvedores cada vez mais afastados das metodologias clássicas e tolerantes apenas às novas metodologias ágeis. Um nicho comercial foi identificado pela indústria, que rapidamente começou a vender treinamentos, certificados, livros e guias de aperfeiçoamento das novas técnicas. Críticos do movimento comparam a adoção dos métodos ágeis com o surgir

de uma nova religião, onde seus seguidores deixam de refletir os princípios da ideologia e encaram suas práticas como regras a serem seguidas [2].

A insatisfação com o modelo de marketing dos métodos ágeis e a percepção de alguns pontos fracos geraram descontentamento por parte de uma parcela dos profissionais da área [3]. O termo Pós-Agilismo foi então criado para registrar uma era onde os desenvolvedores passam a ser mais céticos e cuidadosos com a mudança das práticas internas de seus projetos. A proposta é que os profissionais analisem, testem e utilizem as técnicas e práticas que se adequem aos seus projetos, mesmo que elas sejam combinações de métodos tradicionais e ágeis [4].

Com essa nova tendência vem a importância de uma análise mais específica das práticas ágeis individuais e não tanto do método ágil como um todo. *Kurapati et al*. [5] fizeram uma extensa revisão da literatura com foco em métodos ágeis e descobriram que a grande maioria dos estudos e pesquisas existentes não dão ênfase às práticas utilizadas pelos desenvolvedores, geralmente documentando somente o método usada. Com isso, há uma necessidade na indústria em saber quais práticas particularmente estão sendo implementadas, quais são rejeitadas, quais são consideradas eficazes e quais têm um baixo nível de aprovação [5].

Esse trabalho propõe analisar os fatores positivos e negativos observados pelos profissionais de TI após a implementação dos métodos ágeis, e identificar de que maneira esses métodos e práticas estão sendo aplicados. Dessa forma, espera-se identificar, por meio de um *survey*, se o uso dos métodos ágeis está evoluindo para um pensamento dito Pós-Agilista.

Além desta introdução, este artigo está estruturado da seguinte forma: a Seção 2 apresenta uma revisão da literatura sobre Métodos Ágeis e Pós-Agilismo, a Seção 3 apresenta a metodologia da pesquisa utilizada, a Seção 4 contém as discussões dos resultados do *survey*, por fim, a Seção 5 apresenta as considerações finais.

2 Métodos Ágeis e Pós-Agilismo

Os conceitos primordiais dos métodos ágeis transparecem o reconhecimento do profissional como recurso mais valioso para um projeto e a intenção de dinamizar seu desenvolvimento, deixando-o mais flexível a mudanças e documentações. Em geral, o projeto que segue um método ágil é caracterizado como: gradual, com ciclos de desenvolvimento rápidos e pequenas entregas; cooperativo, com participação ativa do cliente; simples, sendo de fácil aprendizado e modificações; adaptável, estando preparado para mudanças de última hora [6].

Dentre os métodos ágeis disponíveis, os mais conhecidos são: Extreme Programming (XP) [7], Lean Software Development [8], Feature-Driven Development (FDD) [9], Scrum [10] e Adaptive Software Development [11]. Cada método abrange diversas práticas, sendo algumas únicas de um método específico ou compartilhadas entre diferentes métodos. A lista de práticas ágeis varia também entre diferentes autores. Ver, por exemplo, os trabalhos de Abrantes e Travassos [13], Williams [14], Kurapati *et al*. [5] e Jalali e Wohlin [15], ou a análise comparativa entre eles feita por Costinhas [12].

O alto número de alternativas de implementação de práticas e métodos ágeis pode desorientar a equipe em sua decisão. A ausência de uma análise prévia com a finalidade de compreender a fundo quais os problemas e obstáculos a serem enfrentados pelo time e a falta de imersão de todos os funcionários aos princípios fundamentais do método podem levar a uma escolha equivocada, o que resultará em fracasso e descontentamento da equipe adotante [16].

Um dos membros originais da Agile Alliance, Dave Thomas, previu em apenas um mês após a criação do manifesto que a essência do método ágil iria facilmente se perder em meio a tanto alvoroço da indústria e a rapidez com que foram-se criando produtos levando o nome Ágil como marca. Nas palavras de Thomas [17], é triste ver o que se originou para ser um grupo de valores independentes acabar se desvalorizando tão rapidamente por interpretações errôneas.

Em meados de 2006, o termo Pós-Agilismo, tendo autoria atribuída simultaneamente à Jonathan Kohl e Jason Gorman [18], foi começando a aparecer em discussões entre membros da comunidade de informática. Segundo Kohl [19], Pós-Agilismo é uma nova era onde desenvolvedores e profissionais de T.I. são livres para usar as práticas que funcionam e descartar as que não funcionam, sem a obrigação de seguir todas as regras de um *framework*. A inspiração para a criação do termo foi o movimento modernista e pós-modernista, visto na arquitetura e no meio artístico. Ágil se equipara ao modernismo pela sua visão progressiva, porém com muitas regras em torno de seus valores e ferramentas. Enquanto no pós-modernismo, equiparado ao Pós-Agilismo, a percepção de falhas do movimento anterior ocasionou na junção ou alteração de seus processos, possibilitando uma criação mais livre [19]. Os valores e temas observados pelo pós-agilismo incluem [32]: liberdade, mistura de processos, pragmatismo, anti-dogmatismo, aceitar mudanças (em processos de software também), criação de valor, visão e empatia.

A análise de alguns relatos [20], [21], [22] é um indício de que o Pós-Agilismo pode já ser uma realidade entre os departamentos de desenvolvimento. Block [21] e Marchenko e Abrahamsson [22] destacam a escolha das equipes em adaptar ou desconsiderar uma ou outra prática específica, ou combinar práticas oriundas de diferentes *frameworks*. Alguns praticantes questionam a definição de Pós-Agilismo argumentando que a adaptação é um dos principais valores do método ágil, portanto não se pode afirmar que essa característica é original do "novo" método. Como contra-argumento, os Pós-Agilistas distinguem entre ser "ágil" e ser "Ágil", tendo no primeiro a capacidade de se adaptar à situações sociais ou comerciais, e no segundo a adoção de um *framework* como um dogma, seguindo as práticas fielmente [23].

3. Metodologia de Pesquisa

Definição do Objetivo da Pesquisa e Método a ser Utilizado – O objetivo da pesquisa é analisar por meio de um *survey* se as equipes de desenvolvimento estão de fato combinando diferentes métodos e se há adaptação das práticas utilizadas para melhor servir os projetos. Essas informações ajudarão a compreender melhor o estado da arte dos métodos ágeis e auxiliarão na tentativa de evidenciar, ou não, a existência

do Pós-Agilismo como uma nova direção no ambiente de processos de software, no universo da amostra pesquisada.

A escolha pelo método de pesquisa *survey* descritivo interseccional [24] se deu pelo fato da pesquisa se propor a descobrir a distribuição de certos traços e atributos da amostra, não sendo a intenção, no momento, tentar explicar o porquê dessa distribuição. As informações obtidas por meio de questionário destinado aos profissionais de TI com experiência em métodos ágeis possibilitam compreender o estado atual do uso desses métodos. A partir da análise qualitativa dos dados coletados pretende-se responder a questão central deste trabalho, entendendo o progresso na utilização dos métodos ágeis e a evolução de seu uso. A escolha pelo método *survey* se dá pela combinação de suas características principais com o contexto do trabalho em questão.

Elaboração e Estruturação do Questionário – Uma preocupação durante a criação do questionário foi minimizar o número de perguntas e estruturá-las de forma a poupar esforço e tempo dos respondentes durante a coleta de dados. Caso a disposição das perguntas não fosse otimizada, o alto número de variáveis definidas na etapa de planejamento poderia comprometer o envolvimento dos respondentes, que ou optariam por não responder a pesquisa ou perderiam a consistência de suas respostas, passando a selecionar opções a esmo a fim de concluir o questionário rapidamente. Pensando nisso, o questionário foi dividido em quatro partes, descritas a seguir.

Parte 1: Identificação e caracterização do pesquisado. Composta de nove perguntas que ajudam a caracterizar o perfil do respondente e garantir que sua participação na pesquisa agrega valor aos resultados coletados. A anonimidade dos dados foi garantida na apresentação da pesquisa. Foram definidas questões para identificar o nível de experiência do respondente com desenvolvimento de software e métodos ágeis. Essas informações são úteis na fase de análise para identificar correlações entre determinados grupos e suas posturas em relação aos métodos e práticas. Foi incluída também uma pergunta sobre a maneira como o respondente utiliza os métodos ágeis em seus projetos, podendo assim identificar a proporção de profissionais da amostra que costumam seguir um único método e os que mesclam práticas de diversos métodos. Esse dado é essencial na tentativa de verificar a existência do pós-agilismo entre a amostra estudada.

Parte 2: Conhecimento sobre alguns métodolos ágeis e experiência com os métodos citados. Foi perguntado o nível de conhecimento dos respondentes sobre nove diferentes métodolo. A seleção desses métodolos se deu com base na revisão de literatura, onde se observou uma frequência maior de citações desses métodos nos artigos de Wang *et al*. [25], Abrahamsson *et al*. [6] e Dingsøyr *et al*. [26].

Outras questões registram o(s) método(s) utilizado(s) por cada profissional no momento da execução da pesquisa e o grau de satisfação com sua aplicação. Essas informações complementam os dados coletados na terceira e quarta parte da pesquisa, permitindo relacionar os pontos positivos e negativos com cada método.

Parte 3: Identificação de pontos positivos e negativos da adoção do método ágil. Questões foram definidas em decorrência dos pontos negativos observados nos relatos de experiência das equipes da Dataprev [20], Nokia [22] e IASTA [21]. Da mesma

forma, questões correspondem aos pontos positivos observados após a adoção do método ágil foram definidos com base nos benefícios descritos nestes trabalhos.

Parte 4: Identificação de práticas utilizadas e percepção sobre elas. Nessa seção são abordadas as práticas ágeis citadas em [20], [21], [22], no trabalho de Shashank e Darse [27] e na comparação realizada por Costinhas [12]. O critério de seleção do conjunto de práticas a serem analisadas foi a reunião das práticas mais mencionadas nesses artigos. Assim, no escopo deste trabalho, as seguintes práticas foram consideradas: quadros de controle de tarefas, backlog do produto, duração pré-estabelecida das sprints, definição de histórias e divisão de tarefas em subtarefas, gráfico burndown, planning poker, scrum of scrums, reuniões de planejamento, entregas curtas e frequentes, reuniões diárias, reuniões de retrospectivas, quadro kanban, programação em pares, padronização do código, testes unitários, integração contínua, refatoração frequente do código, TDD (*Test Driven Development*), BDD (*Behavior Driven Development*).

A escala definida para o registro dessas variáveis foi: "Não utilizo/utilizei essa prática", "Utilizo/utilizei sem adaptações. Considero benéfica para o projeto", "Utilizo/utilizei sem adaptações. Tenho ressalvas quanto ao seu benefício", "Utilizo/utilizei com adaptações. Considero benéfica para o projeto" e "Utilizo/utilizei com adaptações. Tenho ressalvas quanto ao seu benefício". A vantagem desse tipo de estruturação das opções é o fato de se poder inferir três características sobre cada prática em uma única pergunta: se é ou não utilizada, se é ou não modificada para melhor uso no projeto e se é considerada benéfica ou irrelevante para o desenvolvimento.

Instrumento de Distribuição do Questionário - A construção do questionário, feito no Google Forms, incluiu uma apresentação inicial, contendo o objetivo da pesquisa, o foco do trabalho, a garantia de anonimato dos dados coletados e informações para contato em caso de necessidade de maiores esclarecimentos. As perguntas foram configuradas para que não fosse possível o envio de respostas em branco, a não ser em campos adicionais de texto livre, disponíveis para os respondentes que sintam a necessidade de conceder maiores informações sobre suas respostas.

Com a finalidade de reduzir a ocorrência dos itens destacados na desvantagem do instrumento auto-administrado, a formatação do questionário – incluindo ordem das questões, espaçamento entre as perguntas, formatos das respostas e as instruções – foi foco de atenção durante a criação da pesquisa. Também foram incluídos campos de nome e e-mail, numa tentativa de validar as respostas dos pesquisados, garantindo uma resposta por participante.

Amostra - O tempo reduzido disponível para a distribuição, execução e análise da pesquisa, a ausência de recursos financeiros e materiais e a necessidade de um processo de amostragem acessível e descomplicado foram os principais motivos para a adoção de uma amostra não probabilística. Para melhor contornar os problemas decorridos da utilização de uma amostra não probabilística, o critério utilizado nesse trabalho foi a amostragem por julgamento. Neste formato são escolhidos membros da população que são boas fontes de informação sobre o assunto pesquisado e que

possam representar da melhor maneira possível, dentro do contexto de uma amostra não probabilística, a população estudada.

Como uma maneira de garantir a amostragem por julgamento, as perguntas que abordam o tempo de experiência do pesquisado com desenvolvimento de software, conhecimento a respeito dos métodos averiguados, tempo de experiência com métodos ágeis e o número de projetos que participou onde foram adotados métodos ágeis, atuam como uma ferramenta de filtragem na seleção dos elementos que serão incluídos na amostra.

O questionário foi distribuído através de e-mails e compartilhamento na rede social Facebook. Uma requisição de participação, juntamente com o link para a pesquisa, foi enviada por e-mail através de uma lista de contatos composta por antigos e atuais colegas de trabalho e alunos da Escola de Informática Aplicada da UNIRIO. Como complemento, foi divulgado em grupos no Facebook de alunos da Escola de Informática Aplicada da UNIRIO e em grupos de desenvolvedores que trabalham com Python e com a ferramenta Ruby on Rails. Devido às caracteristicas de praticidade e agilidade dessas duas linguagens, foi previsto o alto número de profissionais com experiência em métodos ágeis nesses grupos.

Após a filtragem determinada para assegurar o critério usado na amostragem por julgamento, foram excluídos dois registros que não se enquadravam no perfil da população pesquisada. Os casos eliminados não apresentavam experiência com desenvolvimento de software e com métodos ágeis. Sendo assim a amostra total considerada nesta pesquisa foi de 58 profissionais.

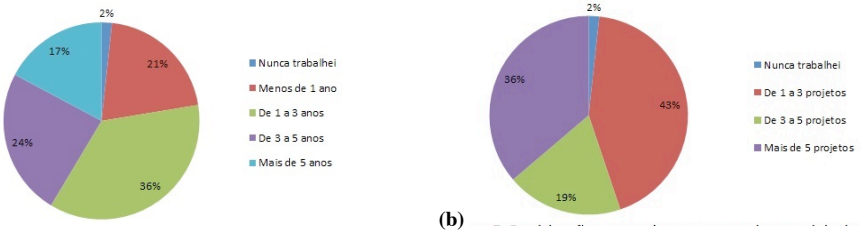
4. Análise dos Resultados

A análise dos resultados foi feita utilizando principalmente o cálculo da distribuição de frequências absoluta, quantidade de vezes que um valor aparece na amostra, e relativa, razão entre a frequência absoluta e o tamanho da amostra [28]. Também foram realizadas análises de relação entre variáveis, com o intuito de encontrar padrões nas respostas de determinados grupos de respondentes.

Perfil da Amostra - A maioria dos participantes da pesquisa declarou ter mais de cinco anos de experiência em desenvolvimento de software (57%) e ocupar cargos Sênior ou de Coordenador/Gerente (57%). Apesar do alto nível de experiência na área por parte dos pesquisados, apenas 17% da amostra declararam ter mais de cinco anos de experiência com métodos ágeis, como visto na Figura 1(a).

As Figuras 1(a) e 1(b) apontam que 2% da amostra (1 participante do total de 58) não possuem experiência com métodos ágeis. De acordo com o critério de corte definido, o pesquisado não se encaixaria no perfil estudado, porém a decisão de manter seu registro na amostra veio após a análise de suas respostas, pois ele indicou que havia participado de uma tentativa de implementação de Scrum, tendo sido interrompida sem sucesso. Após análise da relação entre as variáveis "Tempo de trabalho com desenvolvimento de software" e "Tempo aproximado de experiência com métodos ágeis", foi observado que 31% dos participantes assinalaram o mesmo intervalo de tempo de experiência nas duas variáveis, tendo o restante da amostra declarado menos tempo de experiência com métodos ágeis do que tempo de trabalho

na área de desenvolvimento. Essas informações indicam que pelo menos 69% puderam participar de projetos onde outros tipos de métodos foram usados, facilitando assim a comparação e identificação dos pontos positivos e negativos da adoção do método ágil.

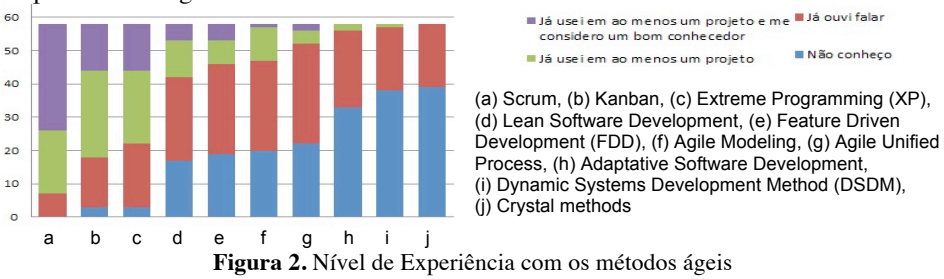


(a) **(b)**

Figura 1. (a) Tempo de experiência com métodos ágeis e **(b)** Participações em projetos com uso de método ágil.

Experiência com as Métodos Ágeis - Ao serem indagados sobre a maneira como os métodos ágeis são utilizadas em seus projetos, 74% da amostra responderam misturar práticas de métodos diferentes, enquanto 26% declaram seguir apenas um método. Essa predominância pela combinação de métodos diferentes e suas práticas, entre a amostra pesquisada, é um grande indício da real evolução no uso dos métodos ágeis, sugerindo que o desenvolvedor não sente mais necessidade em seguir a "doutrina" de um único método.

A Figura 2 apresenta a distribuição da frequência, por nível decrescente de conhecimento e experiência, das dez métodos ágeis incluídas na pesquisa. É possivel distinguir os métodos mais populares entre a amostra, verificando que o único método ágil listado onde 100% dos participantes têm conhecimento sobre sua existência é o Scrum. Do total de 58 profissionais, sete já ouviram falar em Scrum, 19 já usaram em ao menos um projeto e 32 já usaram em projeto e se consideram bons entendedores do método. Kanban e Extreme Programming vêm em seguida na ordem de popularidade, possuindo o mesmo valor de respondentes que declaram ser bom conhecedores, com uma pequena vantagem para o Kanban em número de respondentes que já usaram em ao menos um projeto. O método Crystal Methods aparece em último lugar na ordenação, tendo sido o único da lista não utilizado por nenhum dos pesquisados em nenhum projeto. Além disso, foi o método ágil menos conhecido, com 39 respondentes alegando não terem conhecimento sobre ela.



(a) Scrum, (b) Kanban, (c) Extreme Programming (XP), (d) Lean Software Development, (e) Feature Driven Development (FDD), (f) Agile Modeling, (g) Agile Unified Process, (h) Adaptative Software Development, (i) Dynamic Systems Development Method (DSDM), (j) Crystal methods

Figura 2. Nível de Experiência com os métodos ágeis

Em relação aos ágeis utilizados nos projetos correntes na época da execução do questionário, o resultado é consistente com as informações da Figura 2. Os mais utilizados no momento da pesquisa foram Scrum (47 respondentes), XP (20), Kanban (16) e Lean (5). As demais em conjunto foram citadas 12 vezes.

Após análise dos dados dos pesquisados que afirmaram seguir um único método em seus projetos (15 de 58 respondentes) foi constatado que todos utilizam Scrum, sendo que um participante alterna entre Scrum e Extreme Programming. Enquanto entre os respondentes que costumam misturar práticas de métodos diferentes, foi observado uma maior predominância na combinação de práticas de Scrum e XP e práticas de Scrum, XP e Kanban.

Em relação ao nível de satisfação com os métodos usados, é evidente o contentamento da maioria dos pesquisados: 15% dos participantes estão muito satisfeitos com o(s) método(s) adotado(s) e 66% se consideram satisfeitos, indicando um alto grau de aprovação dos métodos utilizados entre a amostra. Contudo, 19% enxergam espaço para melhora no uso e implementação das práticas ágeis.

Com a finalidade de identificar alguma diferença de opinião entre profissionais que misturam diferentes métodos e profissionais que seguem somente uma, o nível de satisfação foi medido separadamente em cada grupo. As Figuras 3(a) e 3(b) comprovam que a proporção das opiniões dos dois grupos são equivalentes, revelando que ambos em sua maioria aprovam os métodos seguidos.

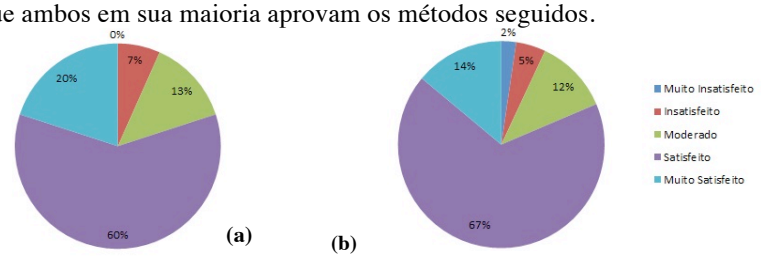


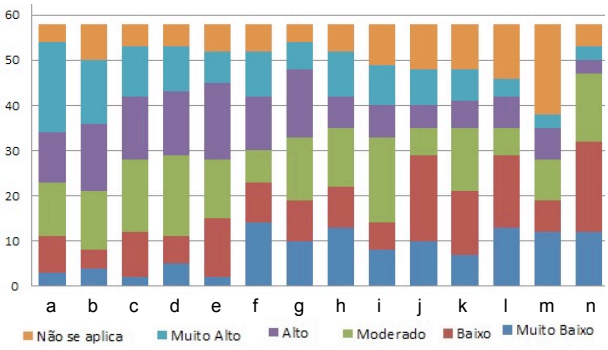
Figura 3. Satisfação com **(a)** uso de um método ágil e com **(b)** mistura de métodos

Os respondentes puderam expressar, em uma pergunta aberta, os principais motivos que justificam o nível de satisfação com os métodos utilizados em seus projetos. Dentre as diversas justificativas, entre os profissionais *satisfeitos*, as mais citadas foram: (i) Entregas rápidas; (ii) Maior controle das atividades da equipe; (iii) Maior integração da equipe nas atividades; (iv) Maior qualidade do produto entregue; Maior flexibilidade à mudanças. Por parte dos respondentes que se declararam *insatisfeitos*, as justificativas principais foram: (i) Falta de regularidade com o padrão do método prejudica a execução do projeto; (ii) Feedback frequente do cliente resulta em muitas alterações de última hora, atrasando a entrega; (iii) Método ágil funciona melhor em equipes experientes, sendo identificado grande dificuldade em seu uso por parte de profissionais inexperientes; (iv) Pouca documentação dificulta o desenvolvimento e a gerência do projeto.

Pontos Positivos e Negativos da Adoção dos Métodos Ágeis - As variáveis que englobam os fatores que podem vir a dificultar a adoção dos métodos ágeis foram classificadas por nível de impacto nos projetos. O gráfico da Figura 4 distribui o resultado de cada variável, ordenando pelos fatores de maior impacto observado pelos respondentes. Os fatores que costumam gerar mais obstáculos na implementação de um método ágil são condizentes com os relatos de experiência pesquisados. A dificuldade da equipe da IASTA em encontrar um *Product Owner* responsável por cada produto, dificultando na resolução do escopo e feedback do produto e o impasse na estimação das tarefas por *sprint* foram as maiores queixas no relato de Block [21].

Em similar, a divisão de manutenção da Dataprev/PB manifestou insatisfação com a má priorização das atividades (*Backlog* mal desenvolvido) e a falta de suporte por parte dos fornecedores de requisitos da equipe [20]. No caso da Nokia, o descontentamento se referia aos diferentes níveis de motivação entre os profissionais da equipe e a dificuldade em mudar a cultura da empresa para se adaptar aos princípios do novo método ágil implantado [22].

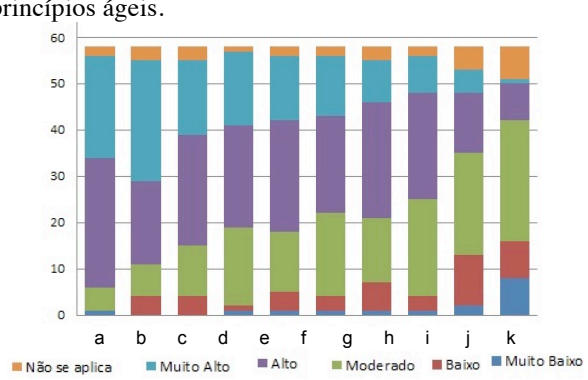
De fato, a mudança na cultura da empresa foi um fator recorrente mencionado pelos criadores do Manifesto Ágil. A imersão à filosofia e aos princípios dos métodos ágeis por todos os funcionários envolvidos são tidos como pré-requisitos para o sucesso da implementação de um método ágil [16] [29] [30], justificando a sua posição em segundo lugar na classificação de impacto feita pelos pesquisados.



(a) falta de colaboração do cliente, (b) mudança na cultura da empresa, (c) prazos mal estabelecidos, (d) diferentes níveis de motivação entre os profissionais, (e) backlog mal definido), (f) falta de interação e trabalho colaborativo entre os profissionais, (g) falta de liderança do Agile Coach, (h) documentação inadequada, (i) treinamento inadequado, (j) foco na entrega rápida resulta em produtos de menor qualidade, (k) mudanças imprevistas na equipe, (l) número alto de integrantes na equipe, (m) localização desfavorável dos membros da equipe, (n) longa curva de aprendizado.

Figura 4. Fatores que dificultam a adoção do método ágil

Em relação às variáveis que identificam os fatores positivos decorrentes da adoção dos métodos ágeis, também ordenadas por ordem de maior impacto na Figura 5, é possível perceber que os pontos que receberam maior marcação de fato refletem os princípios ágeis.



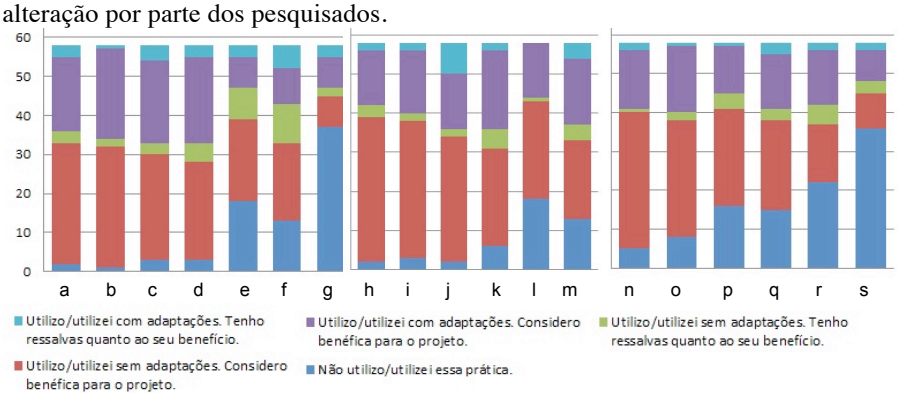
(a) maior comunicação entre os integrantes da equipe
(b) feedback frequente do cliente ajuda a direcionar melhor o projeto
(c) maior satisfação por parte do cliente
(d) maior produtividade,
(e) melhor planejamento,
(f) capacidade de lidar com imprevistos,
(g) maior satisfação por parte da gerência
(h) maior qualidade do conteúdo produzido
(i) menos ocorrência de bugs
(j) custo reduzido

Figura 5. Fatores que facilitam a adoção do método ágil

A alta colaboração do cliente, previsto na filosofia ágil de desenvolvimento, é responsável por dois fatores de grande impacto: melhor direcionamento do projeto, devido ao feedback constante e maior satisfação do cliente. Contudo, também é um dos fatores que podem dificultar a adoção de métodos ágeis, caso o suporte dado pelo

cliente não seja adequado (ver Figura 4). Essas informações sugerem que a participação do cliente deve receber profunda atenção por parte da equipe, podendo ser vantajoso definir como as reuniões e encontros com os clientes podem ser conduzidos de maneira a otimizar a contribuição do cliente.

Considerações Sobre as Práticas - O conjunto de práticas analisadas é apresentado na Figura 6. É facilmente perceptível que a maioria das práticas utilizadas são consideradas benéficas pelos respondentes, sejam elas adaptadas ou não. No entanto, todas possuem alguma parcela de rejeição por uma porção dos profissionais, mesmo que pequena. Também pode-se perceber que todas as práticas sofrem algum tipo de alteração por parte dos pesquisados.



(a) quadros de controle de tarefas, (b) backlog do produto, (c) duração pré-estabelecida das sprints, (d) definição de histórias e divisão de tarefas em subtarefas, (e) gráfico burndown, (f) planning poker, (g) scrum of scrums, (h) reuniões de planejamento, (i) entregas curtas e frequentes, (j) reuniões diárias, (k) reuniões de retrospectivas, (l) quadro kanban, (m) programação em pares, (n) padronização do código, (o) testes unitários, (p) integração contínua, (q) refatoração frequente do código, (r) TDD (*Test Driven Development*), (s) BDD (*Behavior Driven Development*)

Figura 6. Considerações sobre as práticas

As 5 práticas mais frequentes em cada tipo de resposta são ilustradas na Figura 7. A prática menos usada pela amostra, 64% dos respondentes nunca a utilizaram, foi a Scrum of Scrums, justificável pela limitação de sua utilização em equipes grandes. Logo em seguida vem a prática de *Behavior Driven Development*, uma das técnicas orientadas a testes, nunca utilizada por 62% da amostra. Dentre as práticas de testes, é observável a preferência pela técnica de testes unitários.

Em contrapartida, a prática mais utilizada é a criação do *Backlog* do produto - ainda que com adaptações – seguida de quadro de controle de tarefas, reuniões de planejamento e reuniões diárias. As quatro práticas mais utilizadas são de características de gerência de projeto, indicando que apesar da filosofia ágil incentivar a maior atenção ao desenvolvimento, os profissionais ainda percebem a importância das etapas de planejamento.

Muitos respondentes enviaram comentários pertinentes sobre sua opiniões a respeito das práticas e o que os levaram a adaptar certas técnicas. Quanto à necessidade de alteração de certas práticas, um profissional opinou:

> *"Cada projeto e time possuem características diferentes que exigem adaptações em quase todas as práticas. Por exemplo, times mais experientes e introsados não precisam quebrar histórias em tarefas ou usar planning poker para fazer estimativas, pois nesses casos o custo/benefício é pequeno. De maneira*

geral, times mais inexperientes ou recém-formados precisam adotar as práticas mais "by-the-book" e ir adaptando/descartando práticas conforme elas se tornarem obsoletas/desnecessárias.” (Coordenador/Gerente, mais de cinco anos de experiência em desenvolvimento de software)

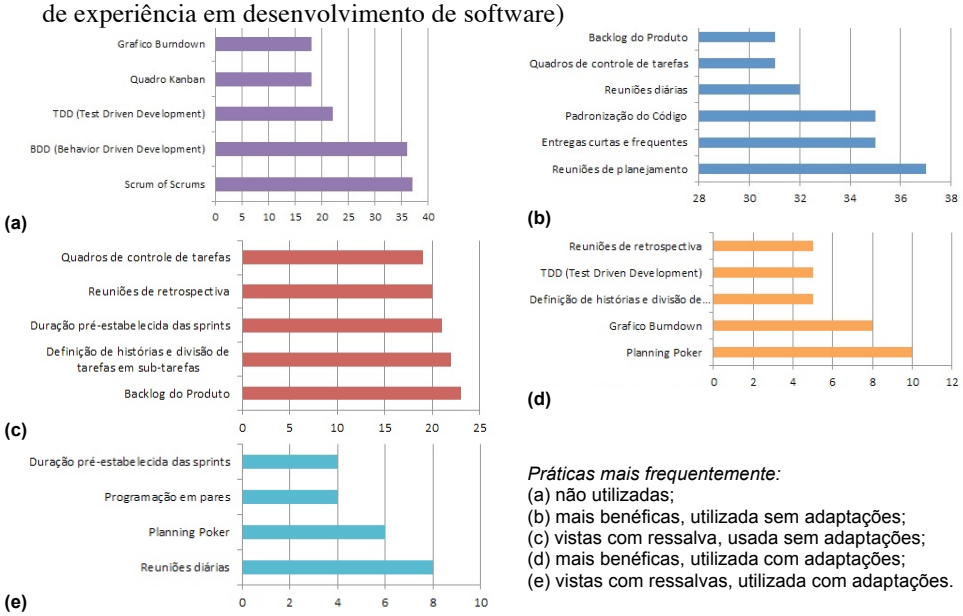


Figura 7. Cinco práticas mais frequentes em cada tipo de resposta

Um dos respondentes afirmou que a necessidade de adaptar a “Duração pré-estabelecida das *sprints*” partiu da frequente mudança de prioridades dos itens no *backlog*. A flexibilidade do tempo da *sprint* permitia que a equipe se adaptasse melhor às mudanças. Outro pesquisado reportou a adição de mini contratos de desenvolvimentos durante a definição das histórias e divisão em subtarefas. Essa adaptação “obrigava todos a trabalhar com baixas margens de erro”, já que qualquer falha identificaria facilmente o responsável.

A prática de *Planning Poker*, uma das cinco mais vistas com ressalva, tendo sida adaptada ou não (Figura 7(e) e 7(c)), recebeu comentários parecidos dos respondentes, a respeito de seu uso e benefício. Um dos participantes alegou que os números de complexidade geralmente têm significados diferentes para cada profissional, o que o faz ter ressalvas quanto à prática.

Em relação às reuniões, alguns participantes argumentaram que as reuniões não têm frequência ou dias certos para sua realização. Um dos respondentes afirmou que sua equipe combinou as reuniões de planejamento e de retrospectiva em um só dia e eliminou as reuniões diárias por julgar improdutivo. Outro participante apontou a importância do preparo de cada integrante para as reuniões.

A respeito das “Entregas curtas e frequentes”, outro respondente ressaltou que essa prática por si só não agrega muitos benefícios se a entrega não vier atrelada ao teste e aceite do cliente. As práticas de testes são vistas por alguns profissionais da amostra como necessárias apenas em certas circunstancias.

Um dos participantes confessou utilizar testes "quando dá tempo", e outro afirmou:

> *"(...)As vezes (em raras ocasiões) o custo de criar o teste é maior que o benefício (...). Em geral, quanto mais permanente o código (um código que será executado durante muito tempo), mais vale a pena escrever testes, pois o retorno do investimento nesses casos é certo."* (Coordenador/Gerente, mais de cinco anos de experiência em desenvolvimento de software)

O resultado da pesquisa evidencia que a tendência entre a amostra é misturar métodos e adaptar as práticas de acordo com o objetivo e situação de cada projeto. A experiência com os métodos permite julgar o que funciona e o que não funciona, o que justifica a opinião de muitos profissionais que afirmam que os métodos ágeis trazem melhor resultados em equipes experientes. Seguir um método único ao pé da letra não é realista na indústria de desenvolvimento de software, no sentido de que os diferentes tipos de empresas, equipes, projetos, clientes e produtos existentes não devem ser tratados da mesma maneira. Sendo assim é importante que profissionais estejam desapegados dos livros e manuais dos métodos e consigam tomar decisões por si só. Mais detalhes sobre o método de pesquisa e os resultados deste trabalho podem ser obtidos em [31].

6. Considerações Finais

O objetivo inicial desse trabalho foi identificar a existência dos princípios do Pós-Agilismo nas equipes de desenvolvimento de software. Com isso optou-se pela realização de uma pesquisa tipo *survey* que pretendia apontar as dificuldades e benefícios encontrados com a adoção de um método ágil, definir como os métodos estão sendo implementadas, discernir as práticas mais e menos utilizadas, mencionando também o nível de produtividade de cada prática no dia a dia da equipe.

O resultado final coletado indica que a maioria dos profissionais da amostra, após experiencias com os métodos, são capazes de detectar o que funciona e descartar/alterar o que não funciona. Concluímos também que a maioria dos respondentes não se prendem a um único método. Entende-se que com este resultado o objetivo inicial do trabalho foi alcançado, uma vez que foi possível identificar, no contexto da amostra, como o uso das métodos ágeis estão evoluindo para um pensamento dito Pós-Agilista. Devido à limitações de recursos, a pesquisa não pode ser realizada em uma amostra probabilística, o que impediu a generalização dos dados analisados. O pouco tempo reservado para a execução da pesquisa também foi um obstáculo encontrado, visto que caso existisse mais tempo para essa etapa seria possível conseguir uma amostra maior.

Essas informações podem incentivar futuros pesquisadores e/ou desenvolvedores a reunir e catalogar as práticas que mais beneficiam cada tipo de equipe, em cada tipo de projeto, identificando suas adaptações e os motivos para tais. Apesar de ser impraticável definir um único processo, metodologia e conjunto de práticas ideais a serem seguidos em cada tipo de situação, a conscientização de que projetos diferentes possuem demandas diferentes já é uma evolução no estado da arte dos métodos ágeis.

Durante a execução desse trabalho foi possível detectar algumas propostas para trabalhos futuros na área, como: estudo mais aprofundado das práticas individuais que trazem mais ganhos para os projetos; identificação dos tipos de adaptações mais

comuns feitas nas práticas, analisando as motivações para tais modificações; catálogo classificando as melhores combinações métodos/tipos de projeto, métodos/tipos de equipe e práticas/tipos de projeto, práticas/tipos de equipe; estudo sobre *Software Craftsmanship*. Uma outra vertente da evolução dos métodos ágeis, paralela ao Pós-Agilismo.

Acredita-se que os dados coletados da pesquisa foram muito ricos em informação, resultado do grande empenho por parte dos pesquisados que colaboraram ao fornecer esclarecimentos pertinentes. Um outro possível trabalho futuro é o reaproveitamento desses dados em uma análise mais aprofundada, com a utilização de conhecimentos estatíticos mais complexos e específicos, capazes de extrair melhores resultados e verificar mais relações entre variáveis.

Agradecimentos. Os autores agradecem à FAPERJ pelo auxílio financeiro (projeto E-26/110.438/2014).

Referências

1. HENNINGER, S; IVATURI, A; NULI, K; THIRUNAVUKKARAS, A; Supporting Adaptable Methodologies to Meet Evolving Project Needs. Extreme Programming and Agile Methods - XP/Agile Universe 2002: Second XP Universe and First Agile Universe Conference Chicago, IL, USA, p. 33-44. 2002.
2. CARDONA, J. Agile Liturgy, 2014, http://www.thoughtworks.com/insights/blog/agile-liturgy. Acessado em 06 de Outubro de 2014.
3. FERNANDEZ, D.; FERNANDEZ, J. Agile Project Management – Agilism versus traditional approaches. The Journal of Computer Information Systems, Winter 2008/2009. 49(2), pg 10-17. 2009.
4. HIBBSs, C.; JEWETT, S.; SULLIVAN, M. The Art of Lean Software Development: A Practical and Incremental Approach, 1ª Edição, O'Reilly Media, Inc., CA. 2009.
5. KURAPATI, N; MANYAN, V. S. C; PETERSEN, K. Agile Software Development Practice Adoption Survey. In: Agile Processes in Software Engineering and Extreme Programming. Springer Berlin Heidelberg, 2012. p. 16-30.
6. ABRAHAMSSON, P.; SALO, O.; RONKAINEN, J.; and WARSTA, J. Agile software development methods: Review and Analysis. Espoo, Finlândia: Technical Research Centre of Finland, VTT Publications 478. 2002.
7. BECK, K. Extreme Programming Explained. Addison Wesley, Reading, MA. 1999
8. CHARETTE, R. N. Foundations of Lean Development: The Lean Development Manager's Guide. Vol. 2, The Foundations Series on Risk Management (CD). Spotsylvania, Va.: ITABHI Corporation. 2002
9. COAD, P.; PALMER, S. Feature-Driven Development. Prentice Hall, Englewood Cliffs, NJ. 2002
10. SCHWABER, K., BEEDLE, A., 2002. Agile Software Development with SCRUM. Prentice-Hall, Upper Saddle River, NJ
11. HIGHSMITH, J., 2002. Agile software development ecosystems, Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
12. COSTINHAS, C. E. Um Estudo de Caso Sobre Adoção de Práticas Ágeis Em Um Ambiente Tradicional. Monografia de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro. 2013

13. ABRANTES, J. F.; TRAVASSOS, G. H. Common agile practices in software processes. In: Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on. IEEE, 2011. p. 355-358
14. WILLIAMS, L. Agile software development methodologies and practices. Advances in Computers, v. 80, p. 1-44, 2010
15. JALALI, S.; WOHLIN, C. Agile practices in global software engineering-A systematic map. In: Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on. IEEE, 2010. p. 45-54
16. ABRAHAMSSON, P.; WARSTA, J.; SIPONEN, M.T.; RONKAINEN, J. New Directions on Agile Methods: A comparative analysis. Proceedings. 25th International Conference on Software Engineering. 2003. p. 244-254
17. BINSTOCK, A. Interview with Alan Kay, 2012. http://www.drdobbs.com/architecture-and-design/interview-with-alan-kay/240003442 Acessado em 8 de Agosto de 2014
18. SKORKIN, A. Are you actually a Post-Agilist? 14 de Outubro de 2008. http://www.skorks.com/2008/10/are-you-actually-a-post-agilist/. Acessado em 18/05/2014.
19. KOHL, J. Post-Agilism Frequently Asked Questions. 28 de Abril de 2007. http://www.kohl.ca/2007/post-agilism-frequently-asked-questions/ Acessado em 1 de Setembro de 2014.
20. RAMOS, A. L. B. M.; LUNA DE LIMA, T. L.; CUNHA, L. M. R. V.; TAVARES, R. L. A. Práticas Ágeis Aplicadas a um Processo de Manutenção de Software: Um Relato de Experiência. Simpósio Brasileiro de Qualidade de Software, Salvador. 2013. p 342 – 349.
21. BLOCK, M. Evolving to Agile: A story of agile adoption at a small SaaS company. Agile Conference 2011.
22. MARCHENKO, A; ABRAHAMSSON, P., Scrum in a Multiproject Environment: An Ethnographically-Inspired Case Study on the Adoption Challenges. Agile, 2008. AGILE '08. Conference , vol., no., pp.15,26, 4-8 Aug. 2008
23. MYER, T. Professional CodeIgniter. John Wiley & Sons. 2008.
24. BABBIE, E. Métodos de Pesquisas de Survey. Tradução Guilherme Cezarino. Belo Horizonte: Ed. UFMG, 1999
25. WANG, X.; CONBOY, K.; CAWLEY, O.; "Leagile" Software Development: An Experience Report Analysis of the Application of Lean Approaches in Agile Software Development. The Journal of Systems and Software. 2010
26. DINGSØYR, T.; NERUR, S.; BALIJEPALLY, V.; NILS, B. M. A decade of agile methodologies: Towards explaining agile software development. Journal of Systems and Software, Volume 85, Issue 6, Junho de 2012. p 1213-1221.
27. Shashank, S.P., Darse, D.H.P.: Finding common denominators for agile practices: A systematic literature review. Master's thesis, Blekinge Institute of Technology, Sweden (2011)
28. KENNEY, J. F.; KEEPING, E. S. Mathematics of Statistics, Part 1 (em). 3rd. ed. Princeton, NJ: Van Nostrand Reinhold, 1962.
29. BINSTOCK, A. Dave Thomas Interview: The Corruption of Agile; Ruby and Elixir; Katas and More, 2014. http://www.drdobbs.com/architecture-and-design/dave-thomas-interview-the-corruption-of/240166688?pgno=2 . Acessado em 10 de Agosto de 2014
30. TURNER, J. When Agile Goes Bad, 18 Nov. 2008. Disponível em: http://www.cio.com/article/2432171/agile-development/when-agile-projects-go-bad.html Acessado em: 13 Ago. 2014.
31. LEAL, T. C. O. A. Pós-Agilismo – Um Estudo Sobre o Legado das Metodologias Ágeis Para os Processos De Software. Monografia de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro. 2014
32. POST-AGILISM BLOG. Post-Agilism, 2015. http://www.postagilism.com/?page_id=2. Acessado em 22 de Fevereiro de 2015.