

Agilidade e DevOps: uma mistura perfeita

Hugo Bolina, Karini Rodrigues Magalhães, Luciana dos Santos Belo
Tecnólogo em Informática, Engenheira da computação, Cientista da computação
hugobolina@gmail.com, karini.magalhaes@gmail.com, lucyannasbelo@gmail.com

RESUMO

A forma como os softwares são construídos mudou. Depois de muito tempo sendo utilizada e com resultados quase sempre insatisfatórios, a abordagem tradicional no desenvolvimento de software está sendo preterida em função de novas metodologias. Afim de obter melhores resultados no desenvolvimento de software as empresas estão adotando os métodos ágeis. Ainda assim, a adoção dessa nova cultura não é simples e a implementação de metodologias ágeis, na maioria das vezes, deixa claro problemas que as organizações nem sabiam da existência ou que não davam a devida importância. Um problema quase sempre identificado é a falta de integração e sinergia entre setores essenciais da empresa. Como forma de integração e aumento dos resultados é apresentada a cultura DevOps. O trabalho busca, através de vasta bibliografia, os conceitos das metodologias ágeis e da cultura DevOps, propondo uma metodologia de implementação de ambas no ambiente organizacional, fundamentando a cultura DevOps na agilidade.

Palavras-chave: DevOps. Métodos ágeis. Kanban. Scrum. Lean.

1 INTRODUÇÃO

Um dos desafios das empresas de T.I. é manter a inovação tecnológica, diante de um ambiente cada vez mais competitivo, com frequentes mudanças nos softwares, executando em uma variedade de sistemas operacionais e dispositivos, todos conectados entre si (BRAGA, 2015). Diante desse cenário, as empresas vêm adotando alternativas ao

desenvolvimento de software tradicional. Os métodos ágeis ganharam força no mercado, pois são flexíveis, adaptativos e buscam entregar valor para o cliente de forma mais rápida.

Segundo Semedo (2012), as metodologias ágeis têm como objetivo a redução do ciclo de vida do software, fazendo com que a equipe desenvolva pequenas partes do produto, acelerando a entrega para o cliente e equilibrando a flexibilidade com a estabilidade. Em contrapartida, no ano de 2008, Patrick Debois levantou a questão que até então não era adotada pelas metodologias ágeis: os recursos de infraestrutura para garantia dos requisitos técnicos. Envolvendo a integração entre as equipes de desenvolvimento e operação e a automação de processos. Surgiu o conceito DevOps.

Existem empresas que utilizam as metodologias ágeis, porém não entregam o produto de forma rápida e com qualidade. As práticas DevOps podem auxiliar, juntamente com a adoção da metodologia ágil a aumentar a produtividade dos colaboradores, sempre visando a qualidade do produto. Sendo assim, este trabalho se justifica devido a semelhança de certos aspectos de ambas "metodologias", complementando uma a outra.

O objetivo é propor um modelo de processo que possa ser utilizado em uma equipe, de maneira que utilize dos melhores valores, princípios e práticas das metodologias ágeis, juntamente com o DevOps.

1.1. Metodologia

Foi realizada uma pesquisa em bases de periódicos, artigos, livros e revistas importantes da área de Engenharia de Software, visando elencar os conceitos e as práticas das metodologias ágeis e do DevOps, para serem utilizados no modelo proposto.

1.2. Organização do artigo

O artigo está organizado como segue: após os contextos acima, a seção 2 contém o referencial teórico. A seção 3 apresenta o modelo proposto. A seção 4 traz a análise e discussão dos resultados. A seção 5 descreve a conclusão do trabalho, limitações do estudo e recomendações de continuidade do mesmo.

2 REFERENCIAL TEÓRICO

As metodologias ágeis surgiram como alternativa às metodologias tradicionais de desenvolvimento apresentadas na época. Em 2001, 17 pessoas se reuniram para conversar. A partir desta conversa surgiu o chamado Manifesto for Agile Software Development (Manifesto para o desenvolvimento de software ágil), onde declararam os valores do movimento (MANIFESTO ÁGIL, 2001), que são:

- Indivíduos e interações mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

Desses valores, surgiram os 12 princípios do manifesto ágil. Várias metodologias ágeis foram surgindo, apoiando principalmente sobre os valores e princípios definidos pelo manifesto ágil.

2.1. Scrum

Scrum é um framework de desenvolvimento de software onde as pessoas podem resolver problemas complexos e adaptativos de forma empírica, entregando produtos com mais valor agregado no ponto de vista do cliente. Pode ser empregado vários processos ou técnicas, sendo adaptável às necessidades de cada projeto e equipe. Isso faz do Scrum um framework leve, simples de entender e extremamente difícil de dominar (SCHWABER; SUTHERLAND, 2016).

Segundo o Scrum Guide (2016), a Sprint é o core do Scrum. Uma Sprint é um time-boxed fixo de um mês ou menos, dependendo da necessidade da organização. Além da Sprint, o Scrum prescreve quatro eventos formais: a reunião de planejamento da Sprint, as reuniões diárias, a reunião de revisão da Sprint e a retrospectiva da Sprint. Além dos eventos o Scrum possui papéis definidos, sendo eles o *Scrum Master*, o *Product Owner* e o Time Scrum. São definidos também os artefatos, sendo eles o *Product Backlog*, o *Sprint Backlog* e o Incremento.

2.2. Lean

O sistema de produção Lean surgiu da necessidade de melhoria do processo de produção das empresas automobilísticas japonesas, para manter a concorrência com as indústrias americanas. A solução deste problema resultou no que conhecemos como Sistema de Produção Lean ou Sistema Toyota de Produção (LIKER, 2006).

O foco do Lean é a eliminação de desperdício através do mapeamento do fluxo de trabalho que agrega valor ao produto e a melhoria contínua do processo de trabalho.

2.3. Método Kanban

Kanban é um método utilizado para a definição, gerenciamento e melhoramento dos serviços entregues. O método Kanban é baseado na visibilidade e conhecimento da quantidade de trabalho através do fluxo de trabalho (ANDERSON, 2016).

No método Kanban, as tarefas ou itens de trabalho são representadas através de cartões, que são fixados em um quadro. O quadro é dividido em fases que representam os fluxos de trabalho da organização ou da equipe (SILVA; SANTOS; NETO, 2012).

2.4. DevOps

A ideia proposta por DevOps teve origem em 2008, quando Patrick Debois publicou o artigo *“Agile and operations infrastructure: how infra-gile are you”* onde demonstrou como a infraestrutura poderia ser tratada de forma ágil, respondendo rapidamente às mudanças de forma similar as metodologias de desenvolvimento ágil de software (BRAGA, 2015).

DevOps é mais que uma metodologia ou processo, é uma forma de mentalidade que evoluiu devido aos esforços em tentar aproximar as equipes de desenvolvimento de software e operação (MAHANTA et.al, 2016).

Para Walls (2013), DevOps é um movimento cultural combinado com práticas de desenvolvimento de software que permite o desenvolvimento do produto mais rapidamente. Além disso, DevOps acaba com o conflito de interesses entre as equipes de desenvolvimento e operação com algumas características principais, que são: abertura da comunicação, alinhamento de incentivo e responsabilidade, respeito e verdade.

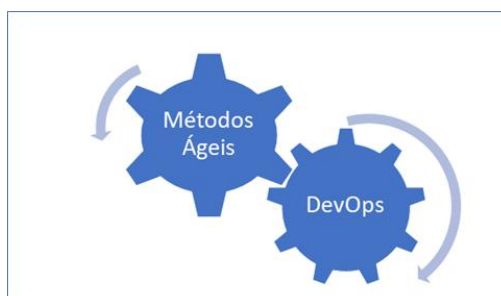
De acordo com Hutterman (2012), DevOps é similar ao movimento ágil no que tange os aspectos de relacionamento interpessoal. O coração do movimento DevOps não pode ser resumido somente em uma nova abordagem, processos e ferramentas. DevOps permite que especialistas de várias disciplinas se relacionem e reúnam suas experiências individuais.

3 APRESENTAÇÃO DO MODELO

De acordo com FJELDSTAD (2017), metodologias ágeis e DevOps, muito mais do que mudanças de tecnologias, são mudanças de culturas. Mudanças de comportamento mental a que nos acostumamos em TI. É de extrema importância a assimilação e aceitação do processo por toda a organização.

Como método ágil, foi adotado o Scrum, visto que é um dos métodos mais praticados entre as organizações. A Figura 1, apresenta como se dará o funcionamento do modelo, tendo o Scrum trabalhando em conjunto com as práticas DevOps, de modo que um complemente o outro.

Figura 1 - Métodos ágeis + DevOps



Fonte: Elaborada pelo autor, 2017.

Visto que será trabalhado com o Scrum, utilizamos os papéis definidos nessa metodologia, acrescentando o time de operação:

- *Product Owner*: Responsável pela representação do cliente na equipe (podendo ser o próprio cliente). Responsável pela visão, priorização e aceite do produto.
- *Scrum Master*: Facilita a execução das atividades e remove os impedimentos, contribuindo pela entrega de valor e metas propostas.
- *Time de desenvolvimento*: Deve ser uma equipe multidisciplinar, garantindo a entrega como acordado com o cliente. Deve ser auto gerenciado e ter habilidades suficientes para atingir as metas.
- *Time de operação*: Responsável pela infraestrutura que irá suportar o produto. Deve garantir o funcionamento nos ambientes estabelecidos pela organização e monitorar a aplicação no ambiente de produção.

O processo de desenvolvimento tem início a partir da elaboração da visão do produto e da montagem do *product backlog*. O desenvolvimento é realizado através de ciclos iterativos (*sprints*), com duração de 2 semanas. No início de cada ciclo será realizada a reunião de planejamento da Sprint. Após essa definição, a *sprint* é inicializado e a equipe se compromete a entregar a meta definida.

No decorrer da *sprint*, é realizada a *Daily Meeting*, uma reunião entre os integrantes do time de desenvolvimento e operações diariamente.

Durante o desenvolvimento do produto, um processo automatizado realiza a integração contínua do código. A cada *commit* realizado no repositório central serão executados os testes unitários e uma revisão de código, visando garantir a qualidade do produto e fornecendo um feedback contínuo para a equipe. Se o código produzido passar por esse processo, então ele será integrado com o restante e um *build* será produzido.

Quando o *build* é gerado, ele é disponibilizado no ambiente de homologação para a execução de testes automatizados. Toda uma *suíte* de testes automatizados deve ser executada sobre o *build*. Ao fim da execução dos testes, caso não seja encontrado nenhum problema a *build* é promovido a candidato para ser disponibilizado no ambiente de produção ao fim da Sprint.

Ao fim da Sprint acontecem as cerimônias de encerramento. Na *Sprint Review*, as funcionalidades são apresentadas para o *Product Owner* e o mesmo deverá dar a tarefa como concluída, caso ela respeite todos os critérios de aceitação definidos. Esse fato

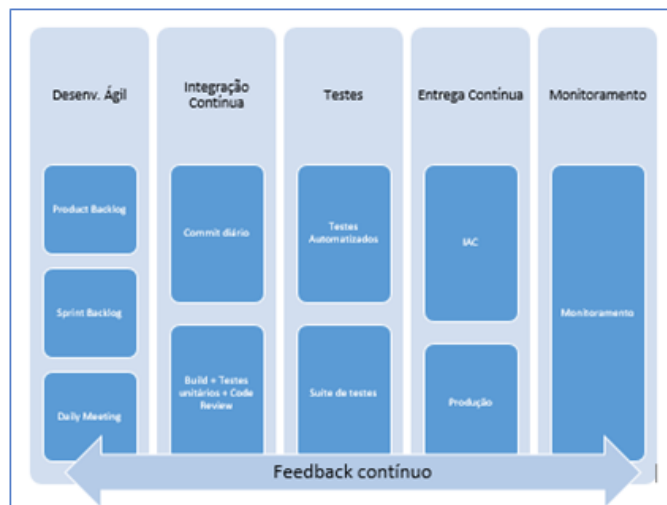
oficializa o último *build* produzido como *build* oficial a ser liberado em produção, através da entrega contínua.

A entrega contínua fica a cargo do time de operação, que produz os scripts para automação da infraestrutura do ambiente, fazendo com que a entrega em produção não tenha interferência humana. Esse time também é responsável por manter o produto estável no ambiente e realizar o monitoramento do produto visando alta disponibilidade.

Com a *build* oficialmente em produção, é necessário executar o constante monitoramento do ambiente e das funcionalidades, para que seja possível fornecer *feedback* e atuação de forma assertiva e proativa em relação a problemas encontrados e como as novas implementações estão impactando os usuários do sistema.

A Figura 2 exemplifica todo o processo de desenvolvimento descrito nos parágrafos anteriores.

Figura 2 - Processo de desenvolvimento



Fonte: Elaborada pelo autor, 2017.

4 RESULTADOS E DISCUSSÃO

Foi proposto uma metodologia baseada nos valores dos métodos ágeis apresentados anteriormente, assim como melhores práticas DevOps, simples o suficiente para que qualquer empresa de software possa conseguir implantar a metodologia e validar se a mesma gera ou não benefícios para seu negócio.

O motivo de utilizar DevOps com método ágil se deve para demonstrar que o DevOps é a concretização da metodologia ágil, pois possui o objetivo de disponibilizar o software em produção de forma rápida e estável, com a integração de processos e ferramentas que automatizam processos e potencializam o ciclo de desenvolvimento (COMPUTER WORLD, 2016; ARAÚJO, 2013). A seguir são apresentadas as práticas das metodologias ágeis e de DevOps que foram apresentadas na metodologia proposta junto com as justificativas destas escolhas.

- Planejamento Sprint: Para o planejamento será necessário realizar uma reunião para definir o que irá e o que não irá para a Sprint. Deve-se considerar as possíveis tarefas não planejadas. Nesta reunião é necessária a presença dos responsáveis pelo desenvolvimento, pelas operações, infraestrutura e o time de suporte, para assim discutir as funcionalidades do produto e seus recursos operacionais, assim todos os passos do projeto são pensados conjuntamente (BETTELEY; SKELTON, 2017; GAEA, 2017).
 - *Backlog*: Além das histórias e tarefas de desenvolvimento, deve-se ter histórias e tarefas de funcionalidades e requisitos não funcionais, assim o *backlog* possuirá um conjunto completo de tarefas, épicos e narrativas que ajudará na entrega e continuidade do produto (BETTELEY; SKELTON, 2017).
 - *Sprints*: Betteley e Skelton (2017) consideram que a melhor opção de tempo para Sprints são de duas semanas, pois é a melhor opção para o desenvolvimento de novos recursos e para melhorar a implantação e os testes, se durassem mais tempo, seria difícil manter o foco e atualizada a lista de *feedback* contínuo. Se fosse mais curto, as reuniões e cerimônias passariam a ocupar a maior parte do tempo reservado para o Sprint, seria então o tempo ideal para que o time fique focado e determinado a terminar o que começaram.
- Reuniões diárias:
 - Gestão a vista, Kanban: com a gestão a vista é possível que, também, as pessoas de operações consigam acompanhar os estágios de desenvolvimento e serem mais proativos na identificação de possíveis gargalos, podendo assim antecipar ações para garantir o prazo das entregas, pois terá uma visão clara e objetiva das atividades que envolverão a equipe de desenvolvimento e operações, gerando a integração através de atividades entre as equipes (ARAÚJO, 2013). Assim é possível ter uma melhoria colaborativa através do entendimento compartilhado de teorias sobre o

trabalho, fluxo, processo e risco, onde a própria equipe sugere melhorias no que pode ser feito para agilizar o trabalho (ANDERSON, 2010).

- Integração contínua: segundo Braga (2015) a cada mudança no software, um conjunto de testes automatizados deve ser executado, encontrando falhas em alguns dos testes, assume-se que o software falhou e detecta-se o erro de forma mais eficiente permitindo assim que os defeitos possam ser descobertos mais cedo gerando menos prejuízo.
- Testes contínuos e automatizados: deve-se testar o código em cada *commit* para garantir que não "quebrou" a base de código (BETTELEY; SKELTON, 2017)
- Entrega contínua: a entrega contínua consegue colocar os times de desenvolvimento e de operações juntos, em uma mesma equipe (BETTELEY; SKELTON, 2017).
 - *Deploy* automatizado: garante o fornecimento de aplicações com maior rapidez e mais assertividade, permitindo que o time possa dedicar-se a outras etapas do projeto (GAEA, 2017).
 - Infraestrutura como código: o objetivo é fazer *deploy* não só de aplicações, mas de infraestrutura de forma rápida e controlada, equipe para de administrar e passa a desenvolver a infra (CARVALHO, 2017), permite então construir ambientes consistentes, confiáveis e repetíveis de forma automática gerando a base para se formar um pipeline de implantação confiável (BRAGA, 2015).
 - Gestão de configuração: Tempestini (2016) cita que controle de versões é importante e precisam se manter organizados e automatizados, ajudando a garantir um processo de *deploy* seguro e com baixa incidência de bugs.
- Está repetindo, então automatize: Automatizar processos e metodologias é o mesmo que otimizar o tempo, pois elimina retrabalho, os riscos nos processos (menos falhas humanas), diminui energia gasta pelo time, aumenta a qualidade de trabalho do time e aumenta o controle do trabalho (CENTRAL SERVER BLOG).
- *Feedback* contínuo: o *feedback* é ampliado entre os times e assim aumenta a comunicação entre os envolvidos no projeto, permitindo uma redução no tempo de entrega, melhora na qualidade do trabalho final e, por consequência, os custos de execução de cada fase (CENTRAL SERVER BLOG, 2017)
- Monitoramento: a utilização de ferramentas para monitoramento permite que alertas sejam disparados para o time sobre eventuais problemas que possam ocorrer em

produção ou no processo de implantação, permitindo que o problema seja investigado e resolvido o mais rápido possível (BRAGA, 2015; SATO, 2013).

5 CONCLUSÃO

Pôde-se perceber que nas metodologias ágeis o foco é a entrega de valor e o cliente é parte integrante e tem papel fundamental no desenvolvimento das soluções. Na cultura DevOps o cliente também tem um papel fundamental, já que todo o processo visa fornecer de forma ágil e livre de erros o produto final ao usuário. Apesar de terem surgido em momentos distintos, as culturas se complementam.

Foi escolhido o Scrum para trabalhar, juntamente com o DevOps. Utilizando-se desses dois “métodos” foi proposto um modelo de processo para ser aplicado em uma organização, tendo as cerimônias e papéis do Scrum, trabalhando em conjunto com as práticas abordadas no DevOps.

O modelo proposto é uma iniciativa de formalizar a utilização do método ágil, juntamente com as práticas DevOps, visando extrair o melhor de ambos e potencializar a produtividade, integração e qualidade do produto e da equipe.

Tendo em vista que o modelo não pôde ser aplicado em uma equipe, alguns trabalhos futuros são sugeridos:

- Validação do modelo de processo proposto em uma equipe;
- Definição de métricas de qualidade e desenvolvimento da equipe, buscando a melhoria contínua.
- Avaliação das melhorias elencadas devido a utilização do modelo proposto.

REFERÊNCIAS

ANDERSON, D.J. **Kanban: Successful Evolutionary Change for Your Technology Business**. 1ª ed. Sequim: Blue Hole Press, 2010.

ANDERSON, David J. CARMICHAEL, Andy. **Essential Kanban Condensed**. Lean Kanban University Press. 2016. 1ª Ed.

ARAÚJO, Daniel P. **Expandindo métodos ágeis com DevOps**. Engenharia de Software Magazine 60, Rio de Janeiro, 2013. Disponível em: <<http://www.devmedia.com.br/expandindo-metodos-ageis-com-devops/28508>> Acesso em: 07 out. 2017.

BETTELEY, James. SKELTON, Matthew. **Merging Agile and DevOps**. Disponível em <<https://www.infoq.com/articles/merging-devops-agile>> Acesso em: 16 out. 2017.

BRAGA, Felipe Antonio Motta. **Um panorama sobre o uso de práticas DevOps nas indústrias de software**. 2015. 123f. Dissertação (Mestrado em Ciência da Computação). Universidade Federal de Pernambuco, Recife, 2015.

CARVALHO, Guto. **O que é DevOps afinal?** 2013. Disponível em: <<http://gutocarvalho.net/octopress/2013/03/16/o-que-e-um-devops-afinal/>> Acesso em 16 out. 2017.

CENTRAL SERVER BLOG. **5 Motivos para usar a cultura DevOps sempre!** Disponível em: <<https://www.centralserver.com.br/blog/5-motivos-para-usar-cultura-devops-sempre>>. Acessado em: 16 out. 2017

COMPUTER WORLD. **Três mudanças necessárias para abraçar o conceito de DevOps**, 8 jul. 2016. Disponível em: <<http://computerworld.com.br/tres-mudancas-necessarias-para-abracar-o-conceito-de-devops>>. Acesso em: 01 out. 2017.

FJELDSTAD, Sverre ; LUNDQVIST, Martin ; OLESEN, Peter . **From waterfall to agile: How a public agency launched new digital services**. Disponível em: <<https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/from-waterfall-to-agile-how-a-public-agency-launched-new-digital-services>>. Acesso em: 29 out. 2017.

GAEA. Disponível em: <<https://www.gaea.com.br>> Acesso em: 16 out. 2017.

HÜTTERMANN, Michael. **DevOps for Developers**. 1ª ed. New York: Apress, 2012.

LIKER, Jeffrey K. Meier, David. **The Toyota Way FieldBook**. A Practical Guide for Implementing Toyota's 4Ps. McGraw-Hill (2006).

MAHANTA, Prabal. et al. **DevOps Culture and its impact on Cloud Delivery and Software Development**. 2016 International conference on Advances in Computing, Communication & Automation (ICACCA) (Spring), Dehradun, 2016, pp. 1-6.

MANIFESTO ÁGIL. 2001. Disponível em: <<http://agilemanifesto.org/>>. Acesso em: 18 ago. 2017.

SATO, Danilo. **DevOps na prática:** entrega de software confiável e automatizada. 1ª ed. São Paulo: Casa do Código, 2013.

SCHWABER, Ken; SUTHERLAND, Jeff. **Guia do Scrum** - Um guia definitivo para o Scrum: As regras do Jogo. 2016. Disponível em: <<http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Portuguese-Brazilian.pdf>>. Acesso em: 21 ago. 2017.

SEMEDO, Maria João Moreno. **Ganhos de produtividade de sucesso de Metodologias Ágeis vs Metodologias em Cascata no desenvolvimento de projectos de software.** Março 2012. Dissertação (Mestrado em Engenharia Informática e Sistemas de Informação) – Universidade Lusófona de Humanidades e Tecnologias. Lisboa. 2012.

SILVA, Diogo Vinícius de S. SANTOS, F. Alan de O. NETO, Pedro Santos. **Os benefícios do uso de Kanban na gerência de projetos de manutenção de software.** VIII Simpósio Brasileiro de Sistemas de Informação - Trilhas Técnicas. 2012. São Paulo.

TEMPESTINI, Lucas. **Como as APIs ajudam em DevOps?** 2016. Disponível em: <<https://sensedia.com/blog/apis/como-as-apis-ajudam-em-devops/>>. Acesso em 16 out. 2017.

WALLS, Mandi. **Building a DevOps Culture.** O'Reilly Media Inc. United States of America. 1ª Ed. 15 abr. 2013.

2017

Autorização de Divulgação de Artigo Técnico

AUTORIZAÇÃO DE PUBLICAÇÃO

☐ AUTORIZO A PUBLICAÇÃO DO ARTIGO TÉCNICO NA INTERNET, JORNAIS E REVISTAS TÉCNICAS EDITADAS PELO IETEC.

☐ **NÃO** AUTORIZO A PUBLICAÇÃO OU DIVULGAÇÃO DO ARTIGO TÉCNICO.

BELO HORIZONTE, ____/____/____

CURSO: _____

SEMESTRE/ANO: _____

TURMA: _____

TÍTULO DO ARTIGO:

NOME DO AUTOR (LEGÍVEL)

ASSINATURA
