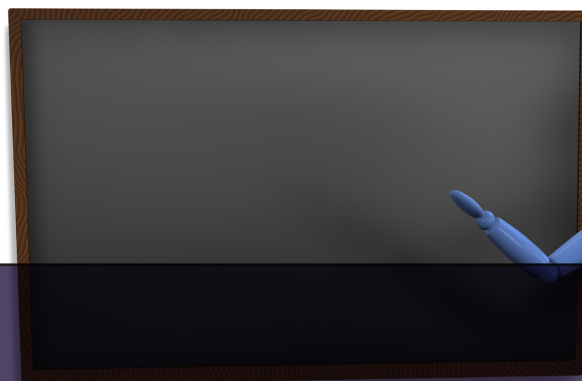


Nesta seção você encontra artigos voltados para as práticas e métodos ágeis.



Scrum e o gerenciamento de projetos

O Scrum e a sua relação de aliança com o gerenciamento de projetos tradicional

De que se trata o artigo?

Apresentar de forma prática como um mesmo projeto pode ser beneficiado com a aplicação das técnicas ágeis de gerenciamento de projetos, unidas e aliadas às tradicionais, de forma a se complementarem mutuamente nos aspectos de papéis e responsabilidades do Scrum, com os papéis e responsabilidades do gerente de projetos, e sem haver conflitos de interesse.

Em que situação o tema é útil?

Este artigo retrata soluções possíveis para projetos de desenvolvimento de sistemas que apresentam indefinições sobre qual abordagem de gerencia-

mento de projetos utilizar, ou que tenham problemas de definição de papéis, gerando conflitos entre as responsabilidades do Scrum e do gerente de projetos.

Resumo:

A aliança do Scrum com o gerenciamento de projetos tradicional é possível e pode ser muito mais fácil do que se imagina. Este artigo mostra como unir as duas abordagens gerando benefícios para um projeto de desenvolvimento de sistemas, e acabando com os conflitos de papéis, responsabilidades e interesses.



Fábio Cruz

fabiorcruz@gmail.com

Graduado na área de TI e PMP certificado com mais de 17 anos de experiência profissional, atuando sempre na área de desenvolvimento de sistemas, sendo os últimos 10 dedicados à liderança de equipes e à coordenação de projetos. Atualmente gerente de projetos na empresa catarinense Paradigma Business Solutions, voluntário no PMI Chapter de Santa Catarina, colaborador na Scrum.org e autor do www.fabiorcruz.com, um blog especializado em gerenciamento de projetos.

Projetos são todo trabalho que possui uma ou mais atividades bem definidas, que precisam ser completadas em um período determinado por uma data inicial e uma data final. Para que os projetos sejam bem entendidos, executados e entregues com qualidade é necessário que haja no mínimo um planejamento prévio, uma alocação de profissionais qualificados, uma direção e um controle eficiente, além de uma experiência para antecipar possíveis problemas com o intuito de evitá-los.

Isso tudo junto e aplicado a um projeto, chama-se gerenciamento de projetos.

As principais abordagens de gerenciamento de projetos utilizadas em grande escala atualmente são a tradicional e a ágil. A primeira é identificada por determinar uma sequência de passos a serem completados formando processos, e a segunda é mais vista como um conjunto de pequenas tarefas, ao invés de um processo completo. No Brasil, a mais divulgada nos dias de hoje dentre as tradicionais é a contida no Guia PMBOK®

(*Project Management Body Of Knowledge*), do PMI (*Project Management Institute*), e entre as ágeis temos o Scrum.

O Scrum é um *framework* para gerenciamento ágil de projetos, onde o seu conteúdo é relativamente simples, porém a sua aplicação é complexa. O *framework* do Scrum é um conjunto composto por pequenas equipes, que executam eventos com duração fixa e ciclos iterativos, sempre apoiados por artefatos e regras que servem para unir todos estes componentes.

Neste contexto, este é o primeiro artigo de uma série que irá abordar a aplicação do *framework* do Scrum apoiado por abordagens tradicionais, em projetos reais de desenvolvimento de sistemas. Os principais assuntos aqui apresentados serão os papéis e responsabilidades do Scrum, aliado e complementado por papéis e responsabilidades existentes no gerenciamento de projetos tradicional, além de algumas maneiras de solucionar problemas de conflitos entre eles.

Scrum

Scrum é um *framework* para gerenciamento de projetos ágeis muito utilizado na área de desenvolvimento de software. Uma das principais características do Scrum é permitir o desenvolvimento de produtos complexos de uma forma incremental e iterativa, contribuindo para decompor um grande produto complexo, em pequenos subprodutos mais simples, mais rápidos e mais fáceis de serem desenvolvidos e entregues. No Scrum estas iterações são chamadas de Sprints, e uma característica marcante de sua estrutura e aplicação é o controle sobre os trabalhos realizados, e a possibilidade de correção e adaptação rápida, permitindo que a equipe altere sua forma de agir ou de pensar o mais breve possível, evitando que problemas ou erros causem danos maiores ao projeto.

O Scrum possui vários eventos definidos para acontecer em determinados momentos da execução do projeto, e o mais importante deles é a Sprint, que é também a principal responsável pelo sucesso da aplicação deste *framework*, concentrando alguns dos pontos mais positivos e de maior destaque do Scrum.

O conceito da Sprint permite a divisão de um grande projeto que possui uma data de início e uma data de entrega final, em vários outros pequenos projetos, com várias datas de início e fim bem definidas. Esta divisão possibilita que as tarefas de controle sejam realizadas em cima de partes menores, mais claras e mais visíveis, além do término total de pequenas partes do sistema ao final de cada um destes pequenos “projetos”.

Em resumo, a Sprint é uma caixa fechada que possui uma data determinada para iniciar e terminar, possuindo atividades bem definidas para serem cumpridas neste tempo limitado, sendo justamente esta caixa fechada a maior qualidade da aplicação do Scrum.

O conceito de “caixa” permite o controle total do que está contido dentro dela, possibilitando a percepção de erros ou desvios mais rapidamente pelos envolvidos, e consequentemente a reação a estes desvios é mais rápida, evitando que os problemas causem estragos grandes e mitigando a possibilidade de danos irreparáveis. Porém, muitos questionamentos ainda são feitos quanto à eficiência e eficácia do Scrum em grandes projetos,

ou como ele aborda e trata as documentações de apoio ao desenvolvimento, e principalmente como o gerente de projeto, ou o gerenciamento de projetos tradicional se encaixa no meio destes conceitos ágeis que o Scrum prega.

Neste contexto, o objetivo deste artigo será justamente entender um pouco mais sobre estes questionamentos, sugerir como respondê-los durante um ciclo de desenvolvimento de sistemas, e mostrar que o gerenciamento de projetos tradicional pode complementar e fortalecer o Scrum, assim como o Scrum pode complementar e fortalecer o gerenciamento de projetos tradicional. Eles não precisam ser excludentes, e nem tão pouco concorrentes.

Existem várias abordagens difundidas e bem aceitas atualmente no mercado de gerenciamento de projetos tradicional, tais como o Guia PMBOK®, PRINCE2, IPMA entre outras. No entanto, não iremos comparar o Scrum a nenhuma destas abordagens específicas neste momento. O objetivo aqui será tratar o gerenciamento de projetos como uma área de conhecimento geral, relacionando algumas das práticas e papéis comuns à maioria das abordagens tradicionais, com as práticas e papéis descritos pelo *framework* do Scrum.

Papéis e responsabilidades no Scrum

Em qualquer tipo de projeto, e seja qual for a abordagem de gerenciamento, o primeiro ponto a ser tratado é um dos mais importantes e influenciadores para o sucesso de um projeto, é a identificação correta dos papéis e responsabilidades necessárias para a condução do projeto. Como complementação, é de fundamental importância a divulgação clara destas definições para toda a equipe envolvida, levando em consideração os times da empresa executante e da empresa contratada.

O Time Scrum é definido a partir de três papéis bem claros, o Scrum Master, o Product Owner e o Time, onde suas principais responsabilidades são:

- Scrum Master: É o responsável por garantir que o Time siga as regras do Scrum, e por remover os impedimentos existentes para que o Time não tenha problemas em atingir a sua meta;
- Product Owner: É o responsável por gerenciar o escopo do projeto, e por garantir o valor do trabalho gerado pelo Time;
- Time: É o responsável por executar o desenvolvimento, e transformar o escopo definido para o projeto em um sistema pronto e disponível para entrega ao cliente. As equipes que utilizam o Scrum gostam de intitular este trabalho de “mão na massa”, porque é efetivamente composto pelas tarefas de construção do sistema, e envolve todos os trabalhos técnicos de desenvolvimento.

O termo descontraído “mão na massa” evidencia que o Scrum é bem direcionado e voltado para a etapa de desenvolvimento. Os papéis do Scrum parecem atender muito bem a necessidade de construir um incremento do sistema, onde de maneira resumida o Product Owner define o que é preciso ser construído, e qual a ordem de construção. O Time entra no processo entendendo as necessidades trazidas pelo Product Owner e definindo como realizará os trabalhos

necessários para atingir a meta do projeto, além de também ser o responsável por determinar a sua própria capacidade e velocidade de trabalho. Por fim, o Scrum Master assume a responsabilidade de contribuir e garantir que o Time siga as regras do Scrum durante os trabalhos de desenvolvimento, além de ser a pessoa mais indicada para remover obstáculos, orientar a resolução de problemas que atrapalhem a evolução dos trabalhos do Time e fazer com que o trabalho flua sem interrupções e com a produtividade mais alta possível dentro dos padrões estabelecidos pelo próprio Time.

Apesar de parecer bastante simples, é aqui que começam alguns dos problemas ou questionamentos referentes a papéis e responsabilidades.

Quando se pensa nos trabalhos específicos de desenvolvimento com o objetivo de entregar uma parte do sistema, os papéis do Scrum parecem atender muito bem a esta necessidade, porém, é preciso refletir e pensar sobre a existência de outros trabalhos que precisam ser realizados.

É comum que antes, durante e após a etapa de desenvolvimento, diversas outras tarefas sejam concluídas em paralelo contribuindo para que um projeto seja bem sucedido, e influenciando para que o ambiente externo ao Time Scrum não afete os seus trabalhos, e nem tão pouco os trabalhos do Time afete o ambiente externo. Sendo este o primeiro ponto que o gerenciamento tradicional pode complementar o Scrum e dividir a responsabilidade de algumas tarefas, e onde os papéis no Scrum começam a ser relacionados a outros existentes na abordagem tradicional.

O gerente de projeto no apoio

Um dos papéis mais polemizados quando o assunto é Scrum, é o tradicional gerente de projetos. Primeiro porque este papel específico não aparece nas definições de papéis do Scrum, e segundo por que o Scrum defende que o Time deve ser auto-gerenciável, o que resumidamente é interpretado como não havendo a necessidade de ter um gerente de projetos na equipe.

Considerando o Time Scrum conforme definição, e aplicando esta formação de time em projetos de desenvolvimento de software, temos duas situações que se evidenciam quase que naturalmente:

- Situação A: Um pequeno projeto, com poucos recursos, um prazo curto e pouco dinheiro envolvido tem mais chances de ter sucesso, e funcionar bem, atingindo o objetivo proposto, sem um gerente de projeto envolvido em nenhuma etapa, desde que o Scrum seja bem aplicado;
- Situação B: Um projeto médio ou grande, com dezenas de pessoas, uma duração maior que um ano, com aquisições de máquinas ou equipamentos caros e alguns milhões investidos, tem menos chances de alcançar o sucesso, principalmente por gerar uma grande insegurança e até uma desconfiança no poder de auto-gerenciamento do Time Scrum. Isso geralmente se dá, especialmente, por questões que frequentemente acompanham projetos com as características mencionadas neste item, tais como: “Quem cuidará dos custos do projeto?”, “Quem definirá os recursos necessários e envolvidos com

todo o projeto?”, “Quem negociará mudanças no custo e no prazo com o cliente?”, “Quem tratará do gerenciamento das aquisições?”, “Quem gerenciará os riscos?”, “Quem observará a estratégia da organização em relação à estratégia do projeto?”, “Quem desenvolverá os recursos humanos do projeto?” ou “Quem fará o gerenciamento dos *stakeholders*?”.

Por refletir sobre estas duas situações, e ter dificuldade na resposta destas questões é que vários profissionais da área de sistemas, e até de outros setores, são descrentes quanto ao funcionamento do Scrum em grandes projetos. Realmente olhando apenas para as situações apresentadas acima, e considerando apenas o Time Scrum assumindo todas as responsabilidades descritas, aliada a ausência de um gerente de projeto, fica mesmo fácil ter esta impressão e acreditar que somente pequenos projetos funcionam com Scrum.

Por isso a proposta desta série de artigos é justamente mostrar que o Scrum funciona tanto em projetos pequenos e curtos, como em projetos grandes e longos, levando em consideração equipes, custos, prazos e complexidade. Sendo que para isso basta o suporte de algumas outras abordagens que podem apoiar o Scrum, e principalmente dividir certas responsabilidades entre os papéis do Scrum e os papéis mais tradicionais, que neste caso agirão como complementares uns dos outros.

Ainda observando as duas situações citadas, alguns grupos podem responder rapidamente que o próprio Time Scrum poderá ser responsável por todas as atividades contidas tanto na situação A, quanto na situação B. Porém, caso isso seja realmente feito, surge pelo menos uma nova questão tão importante quanto as outras: “Se o Time Scrum ficará responsável por todas as tarefas de gerenciamento que não se encaixam nas conhecidas como ‘mão na massa’, quem efetivamente desenvolverá e entregará o produto do projeto no prazo estimado?”, ou “É possível que o Time Scrum gerencie todas as áreas que precisam de gestão, inclusive se auto-gerencie durante o desenvolvimento, e consiga ser produtivo, mantendo a qualidade e entregando o sistema proposto?”.

Este último questionamento, em conjunto com os exemplificados na situação B, formam um grupo de perguntas que são difíceis de serem respondidas tendo à disposição apenas os papéis existentes no Scrum. Por isso dá-se início a reflexão sobre a importância e contribuição do tradicional papel do gerente de projetos, complementando e ajudando o Time Scrum.

Micro e macro-gerenciamento

Para este artigo, o macro-gerenciamento é aquele em que a equipe de execução não atua e não é responsável diretamente, e que na prática está circulando ao redor dos trabalhos de quem coloca “a mão na massa”, passando na maioria do tempo despercebidamente por esta equipe. Quando bem realizado, não será notado e não influenciará diretamente na produtividade, porém indiretamente provocará uma influência positiva e aumentará a produtividade do Time Scrum.

Já o micro-gerenciamento é justamente o trabalho de auto-gestão realizado pelo próprio Time para desenvolver o produto

e entregar uma versão do sistema ao cliente. Este por sua vez influencia diretamente o trabalho do Time, além de ser responsável direto pela produtividade da equipe de execução.

Pegando especificamente o gerenciamento de custos como exemplo, inclusive considerando os fatores que são influenciados ou influenciam os custos, já temos uma grande área que pode ser assumida pelo gerente de projetos, e que podemos denominar aqui como macro-gerenciamento. Primeiro por ser uma área de gestão, que apesar de influenciar a execução, não precisa ser de responsabilidade da equipe que executa, deixando-a livre deste controle para dedicar-se ao desenvolvimento do produto.

Em contrapartida, o Time Scrum, ficando livre do gerenciamento dos custos, pode se dedicar exclusivamente ao gerenciamento das atividades de desenvolvimento, e das decisões de como realizará o trabalho necessário para entregar o produto final do projeto, incluindo as estimativas destes trabalhos, sendo estas tarefas específicas denominadas aqui como micro-gerenciamento.

Uma observação importante é que o macro não é mais nem menos importante que o micro-gerenciamento, é apenas uma separação que tem o objetivo de diferenciar grupos de atividades que podem ser realizadas por um conjunto distinto de pessoas, focando justamente em otimizar os trabalhos de cada profissional e perfil envolvido, maximizando a produtividade e os ganhos de toda a equipe do projeto.

Com esta divisão de esforços e responsabilidades é possível enxergar uma colaboração entre a abordagem tradicional e a ágil, e entender que a equipe do projeto pode ser composta pelo Time Scrum (ágil) e por um gerente de projetos (tradicional).

Esta aliança se destaca ainda mais quando observamos que as tarefas de responsabilidade do micro-gerenciamento não ocupam o tempo do gerente de projetos, deixando-o afastado do Time Scrum o suficiente para que ele não interfira nas atividades realizadas dentro da Sprint, e nem tão pouco nas decisões do Time. Isto possibilita que o Time Scrum se auto-gerencie sem desgastes externos, além de reforçar e manter uma das regras mais fortes do Scrum que é o auto-gerenciamento do Time de desenvolvimento.

Em contrapartida, esta forma de trabalho permite que o gerente de projetos possa se dedicar mais fortemente às atividades do macro-gerenciamento, estas que não ocuparão nenhum dos papéis definidos pelo Scrum. Com isso é possível que o Product Owner, o Scrum Master e o Time trabalhem focados exclusivamente em suas responsabilidades, explorando ao máximo as atividades determinadas pelo Scrum, maximizando sua produção diária e o desenvolvimento do objetivo da Sprint.

0 Time multidisciplinar e o gerenciamento tradicional

Para o Scrum o Time deve ser multidisciplinar, ou seja, o próprio Time e seus integrantes devem ser responsáveis por realizar todas as tarefas necessárias para entregar o trabalho do projeto. Não devendo haver distinção de cargos ou funções, títulos ou senioridades e muito menos áreas determinadas ou

específicas de atuação. No Scrum todos os integrantes do Time são conhecidos como desenvolvedores.

Esta regra, aliada à regra que diz que nenhum item definido para ser realizado deve ter um dono, e que todas as tarefas a serem realizadas são de responsabilidade de todo o Time, definem uma forma de trabalhar bem característica do Scrum. Esta característica pode ir contra outras abordagens de gerenciamento de projetos, ou de boas práticas de desenvolvimento de sistemas, que defendem como positivo ter especialistas em determinadas áreas, como por exemplo: banco de dados, qualidade ou análise.

Em outras palavras, para o Scrum, o Time de desenvolvimento deve ser composto por profissionais que consigam realizar atividades de implementação de código, prototipação de telas, ou designer de *interfaces*, banco de dados, arquiteturas, imagens, integrações entre sistemas, entre outras habilidades que possam ser necessárias para a execução dos trabalhos e entrega do projeto.

Sem sombra de dúvida, esta é uma regra básica e importante para que o Scrum possa ser aplicado e traga bons resultados, porém pode haver algumas variações dependendo da forma com que a empresa organiza, distribui e até contrata seus profissionais. Por exemplo, geralmente os profissionais especializados em banco de dados só trabalham nesta área, e raramente fazem outras atividades como análise ou até mesmo programação. Então como inserir um DBA Oracle certificado e especializado em um Time Scrum seguindo a regra de que todos devem ser desenvolvedores?

Na prática será bem difícil implementar este conceito na maioria das empresas, e mais uma vez teremos o papel do gerente de projetos atuando como facilitador deste tipo de adaptação. Na maioria das vezes, o que mais acontece nas organizações atualmente é um profissional DBA (*Data Base Administrator*), por exemplo, ser realmente especializado e não exclusivo a apenas um projeto, mas sim compartilhado entre várias equipes que necessitam de suas habilidades referentes a banco de dados. Sendo assim, o gerente de projetos que estará dando apoio aos Times Scrum, poderá gerenciar este profissional DBA, alocando-o quando necessário dentro dos Times para executar tarefas específicas de banco de dados, e que com a regra da auto-gestão e do micro-gerenciamento foram solicitadas pelo próprio Time Scrum.

Uma característica do Scrum, é que o Time é composto por um grupo de profissionais que são considerados apenas pelo título de desenvolvedores. Esta definição não é uma falha do Scrum, e em muitos casos poderá ser uma qualidade deste *framework* ágil, no entanto, dependendo do caso, poderá ser uma fraqueza, pois é muito difícil considerar que na prática teremos sempre profissionais que fazem tudo, ou então acreditarmos que em todos os projetos existam profissionais especialistas em tudo. Na prática, isso é quase impossível nos dias de hoje, ainda mais quando falamos de especialistas, tecnologias avançadas, muito novas ou até mesmo antigas.

Sendo assim, mesmo o Time devendo ser capaz de executar todas as tarefas necessárias para realizar o trabalho do

projeto, teremos situações em que precisaremos de especialistas, tais como DBAs, programadores especializados em alta plataforma como Cobol para uma integração com um sistema Java, ou alterações em um módulo SAP, ou analistas de testes especializados em testes específicos com ferramentas próprias para este fim, entre outras dezenas de exemplos que poderíamos mencionar.

Em um cenário similar ao destacado acima, a teoria do Scrum funcionará com uma certa dificuldade, e precisará de apoio externo para gerenciar a alocação de profissionais que não serão fixos ao Time, e que muitas vezes serão compartilhados entre vários Times distintos em diferentes momentos.

Na visão do Scrum, a necessidade de um profissional especializado pode ser encarada como um impedimento, e neste caso o Scrum Master seria o mais indicado para remover este impedimento, o que para este caso significaria a alocação do profissional. Entretanto, esta tarefa pode não ser tão simples, podendo envolver uma negociação com outras equipes, uma contratação temporária ou até mesmo definitiva, uma subcontratação de outra empresa ou fábrica de software, ou seja, voltamos ao questionamento de custos que foi abordado anteriormente, demonstrando novamente que seria mais interessante deixar a cabo do papel do gerente de projeto gerenciar este tipo necessidade. Lembrando que esta ação liberaria o Scrum Master para manter o Time trabalhando em suas atividades diárias, e permitindo que ele remova impedimentos mais ligados às tarefas menores.

Conclusão

Quando apenas um olhar, com apenas um ângulo de visão, observa uma mesma imagem por diversas vezes, o resultado pode ser sempre a mesma fotografia. No entanto, quando um mesmo olhar altera sensivelmente o seu ângulo de visão, a mesma imagem observada pode gerar diferentes fotografias.

Esta analogia pretende mostrar que quando se olha apenas para a abordagem do Scrum e se analisa os seus pontos positivos e negativos, ora pode-se entender que é a única e melhor aplicação a fazer, ora pode-se ver uma aplicação ruim e ineficiente. A mesma análise pode ser feita com as abordagens tradicionais, onde as chances de se obter um dos dois resultados ilustrados acima são consideráveis.

Por isso devemos provocar uma reflexão simples de que o Scrum não é melhor e nem tão pouco pior do que uma abordagem tradicional de gerenciamento de projetos, e a utilização de uma ou outra não deve ser encarada como a única solução para todos os problemas.

Um ponto extremamente positivo que o Scrum trouxe aos projetos, de uma forma mais simplista e de melhor aplicação e assimilação, foi justamente a adaptação constante e mais breve possível aos desvios de planejamento e execução, o que pode levar a uma melhoria contínua muito eficiente.

É com esta visão de adaptabilidade que devemos encarar nossos projetos reais, e termos em mente que nenhuma metodologia, *framework* ou abordagem é perfeita e funciona na plenitude para todas as situações, principalmente em um mercado rápido, feroz e mutante como é o setor de tecnologia. Entretanto, se os profissionais que atuam em funções gerenciais, lembrarem que o seu principal objetivo, e também o objetivo de todas as abordagens de gerenciamento de projetos, sendo ágeis ou tradicionais, é o mesmo e que se resume a entregar um projeto com sucesso, não há porque pensarmos que o Scrum não pode trabalhar ao lado de uma abordagem tradicional, e vice e versa.

Portanto, quando surgir uma questão sobre qual abordagem é a melhor ou qual deve ser usada em uma determinada situação, reflita sobre a possibilidade de união entre elas, somando os pontos positivos, subtraindo os pontos negativos, e obtendo como resultado final o sucesso de forma mais controlada, fácil e segura.

Referências

Introdução ao Scrum, blog FabioCruz.com

www.fabiorcruz.wordpress.com/outros/introducao

Introdução ao PMBOK®, blog FabioCruz.com

www.fabiorcruz.wordpress.com/pmbok%20%ae/introducao

Scrum Guide 2011, escrito por Ken Schwaber e Jeff Sutherland

www.scrum.org/scrumguides

Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto.

Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:

www.devmedia.com.br/esmag/feedback



Nesta seção você encontra artigos voltados para as práticas e métodos ágeis.



Fundamentos das Metodologias Ágeis

Conheça em detalhes os princípios dos métodos ágeis



André Abe Vicente

andvicente@gmail.com

Mestre em Ciências da Computação e Matemática Computacional (LabES - Laboratório de Engenharia de Software) pela Universidade de São Paulo (USP) - 2010. Bacharel em Informática pela Universidade Estadual do Oeste do Paraná (Unioeste - Campus Cascavel). Têm experiência na área de Engenharia de Software desde 2005 e na área de Qualidade de Software e Testes desde 2007. Além disso, é entusiasta do uso de práticas de desenvolvimento e condução de projeto utilizadas em métodos ágeis. Como Analista de Testes, atua no planejamento, automação e execução de testes com apoio de ferramentas de teste e integração contínua.

Para enfrentar problemas com prazos e complexidade de métodos tradicionais da engenharia de software, diversos métodos ágeis de desenvolvimento estão sendo utilizados em projetos de software. Estes métodos possuem como principal objetivo a satisfação do cliente, preocupando-se com a entrega incremental de software desde as etapas iniciais de desenvolvimento, produtos de trabalho de engenharia de software minimizados e simplicidade global no desenvolvimento.

Métodos ágeis foram desenvolvidos para beneficiarem a entrega rápida de

De que se trata o artigo?

Nesse artigo são descritas de forma geral a origem e as motivações para a utilização de métodos ágeis e também detalhes sobre os principais métodos. Os métodos ágeis são descritos principalmente sob o ponto de vista do fluxo de atividades e práticas conduzidas durante o desenvolvimento de software.

Em que situação o tema é útil?

Métodos ágeis foram desenvolvidos para beneficiarem a entrega rápida de código que agregue valor ao cliente por meio do desenvolvimento em pequenos ciclos. Para atingir esse objetivo, esses métodos são focados na contínua interação entre desenvolvedores e clientes, que garantem que o software atenda as necessidades de mudança dos requisitos do cliente.

Resumo:

Neste artigo serão descritos os métodos ágeis de desenvolvimento, principalmente do ponto de vista de valores e princípios comuns a esses métodos, assim como suas práticas individuais. Para isso, optou-se por descrever com maiores detalhes o método XP, Scrum e Lean, por esses serem os métodos mais aplicados no contexto atual da área de desenvolvimento ágil, além de terem relação com a utilização de métricas de software seja por meio de suas práticas ou pelo seu processo e papéis envolvidos no projeto de software.