

Questão: Como você implementaria as metodologias ágeis (Scrum, Kanban, XP e/ou Lean) e as abordagens tradicionais de desenvolvimento de software em um ambiente híbrido?

Alex Ramos

Para realizar a implementação das metodologias seria preciso primeiramente verificar o perfil de cada integrante do projeto, os requisitos para o desenvolvimento, sua complexidade e condições da realização do projeto (custos, tempo, recursos, ferramentas, métricas, padrões de qualidade etc.). Partindo disso, com a metodologia scrum os times contariam com ao menos um integrante de cada função destacada no texto de apoio visando uma maior eficiência dentro dos times, os ciclos de desenvolvimento seriam planejados de acordo com o tempo disponibilizado pela RDA e as funcionalidades que o sistema de controle precisa ter, decidindo quais dessas funcionalidades vão ser contempladas em cada ciclo, com o kanban seria possível organizar cada atividade a ser realizada durante cada ciclo. Com as documentações características das metodologias tradicionais, seria registrado formalmente o desempenho e eventos ocorridos durante o projeto para verificar possíveis mudanças no gerenciamento do projeto que possam ser necessárias, atestando também se o projeto teve sucesso em atingir os objetivos e os requisitos decididos pela RDA junto com as equipes do projeto, garantindo sua qualidade perante o esperado e no futuro servindo como base para projetos semelhantes.

Primeiramente escolhendo qual framework seria mais adequado para a proposta e complexidade do projeto, a partir disso durante o planejamento definir com será o acompanhamento do projeto, definindo as estratégias de medir o desempenho da equipe, do progresso e da qualidade do que será entregue, para isso seria possível a partir definir estratégias como a implementação do CI e CD tornando eficiente a verificação do código desenvolvido por toda a equipe além implementar-lo ao sistema, a partir de reuniões de alinhamentos frequentes seria possível debater junto a equipe o progresso de cada etapa de desenvolvimento, tornando a coleta de resultados obtidos durante o processo de acompanhamento e sua gerência no futuro. Para melhorar as estimativas a partir dos resultados do desempenho da equipe em ciclos anteriores de desenvolvimento será possível estimar com mais precisão o tempo, custos e recursos para os próximos ciclos, utilizando ferramentas de organização como Trello e Jira para controlar o andamento.

Para melhorar a eficiência da equipe do projeto, durante o gerenciamento as estratégias adequadas de comunicação, organização e controle da equipe deverão ser adotadas. Sendo assim, analisar quais ferramentas e métodos de comunicações serão eficiente levando em conta as condições da equipe, adotando uma frequência de reuniões adequada, promovendo a colaboração, exposição de visões e ideias, troca de feedbacks sobre o andamento do projeto, além de definir quais softwares poderão integrar ainda mais a equipe entre si. Na organização da equipe é preciso definir com clareza os objetivos, determinando em cada etapa os que vai ser feito e como será feito, organizando as atividades de forma que cada integrante possa trabalhar na sua área de conhecimento e também se desenvolver em outras partes do projeto, assim compreendendo o projeto como o todo. Por fim, também é preciso gerenciar as metodologias e práticas adotadas de acordo com o desempenho da equipe e do projeto, realizando as mudanças necessárias que vão possibilitar que equipe possa trabalhar melhor ao longo do projeto em um ambiente adequado.

Com o TDD cada parte do código do sistema de controle a ser desenvolvido no projeto teria as suas condições de aceitação (sua qualidade aprovada) definidas antes do processo de desenvolvimento de fato, guiando o desenvolvimento de forma assertiva aos objetivos de cada ciclo ou etapa de desenvolvimento. O BDD atuaria como uma forma de comunicação entre os responsáveis da RDA e junto ao TDD atuaria como uma estratégia de evitar retrabalho, fuga do escopo e da resolução que o projeto busca solucionar, formalizando partir de histórias de usuários para funcionalidades mais simples até planejamentos detalhados para as partes mais complexas. A integração contínua permitiria a visualização de cada progresso no desenvolvimento em um repositório e junto a entrega contínua, permitiria que os desenvolvedores realizem testes e verificações e implementassem ao sistema, caso aprovados, de forma contínua no sistema em desenvolvimento o que poderia tornar o sistema funcional bem mais cedo devido a necessidade da RDA. Por fim, a cultura de trabalho devops seria como motor para implementação de cada prática, o devops busca tornar o processo de desenvolvimento mais eficiente possível, com a comunicação entre integrantes e lideranças e integração interna das equipes e utilizando seus valores junto a um gerenciamento das equipes durante as etapas de adaptação as práticas, orientando e controlando seus desempenhos.

Alexandre

Levando em consideração elementos como tamanho, complexidade e estrutura do projeto para ponderar quais as técnicas e metodologias serão adequadas para a situação vigente, projetos pequenos e que dispensam uma maior formalidade precisarão de maior segurança, qualidade e agilidade em detrimento de um controle mais rígido dos processos, contudo em modelos híbridos seremos capazes de aliar elementos das metodologias tradicionais mais conhecidas, bem como das metodologias ágeis, uma implementação de uma metodologia chamada de SCRUM é

possível em cenário no qual se busca um trabalho incremental colaborativo, comunicação sem fronteiras, aprendizado contínua e responsabilidade compartilhadas, eventos como Sprints diárias são importantes para reunir os avanços, pendências e compartilhar também conhecimentos importantes, além da construção da sinergia organizacional mais assertiva, o projeto poderá sempre ser mutável com um estreito laço construído também pelo clientes pelos meios de comunicação estabelecidos.

A colaboração dentro da minha equipe deveria acontecer ligada aos processos envolvidos no projeto, alguma metodologia com o KanBan pode colaborar pois dividirá as tarefas planejadas, sendo executadas e outras pendentes, a comunicação entre os membros do projeto diminuirá riscos de falhas pois não existe competição e a responsabilidade aqui é compartilhada, é importante também que eu como sênior, saiba dividir as atribuições e alocar corretamente entre os diferentes colaboradores com base no nível de conhecimento e experiência de cada um, dessa forma o impacto sinérgico seria bastante proveitoso para o projeto com garantia de qualidade e eficiência para os clientes finais, acredito que um cronograma com as entregas contínuas (CD/CI) sejam elementos indispensáveis para o sucesso de qualquer natureza e espécie.

TDD, metodologia baseada em testes de código condiz no desenvolvimento do teste antes da codificação em si do programa e/ou software, a preocupação aqui está em modelar o comportamento do software esperado pelos usuários desses sistemas, alguns benefícios são detecção precoce de erros e vantagens em produtividade como eficiência e eficácia, já o BDD encontra-se baseada no corpo do projeto. CI/CD introduz, respectivamente, integração total e entrega contínua para os projetos, a integração total de trabalho é estabelecida por um repositório compartilhado na nuvem e sincronizado com todos os colaboradores envolvidos no desenvolvimento do projeto, já a entrega contínua traz a ideia de entregas parciais do produto ao cliente para fortalecer e credenciar o compromisso com o projeto, o mesmo atende melhor as mudanças relativas ao ambiente pois a característica principal aqui é o incremento, isto é, uma "contrução" no lugar de um produto entregue uma vez. Uma cultura Devops será implementada quando houver aliança entre operações e os processos de desenvolvimento, algumas características que deverão aparecer no processo serão a automação de processos, uma cultura de colaboração, quebra de barreiras departamentais, aspectos parciais de entrega dentro do projeto, comunicação e colaboração serão destaque para o Devops.

A adoção de uma framework de auxílio ao gerenciamento de projeto deverá ser destaque para planejar de maneira mais assertiva as atividades consoantes ao projeto, uma metodologia ágil também deve ser implantada em conjunto com aspectos formais tradicionais para introduzir as vantagens dos encontros contínuos de discussão, as famosas "daily" e elementos da metodologia KanBan usadas para

fomentar esse aspecto colaborativo e de construção de um código mais sincronizado e limpo para se trabalhar. As entregas parciais oferecidas nas Sprints reunirão os êxitos e falhas pendentes ainda no projeto, uma solução compartilhada poderá trazer uma qualidade maior para o projeto pelo declínio e atendimento aos elementos comunicativos, é importante também estabelecer métricas de desempenho e controle para mensurar de certa forma a execução e comparar com o que foi planejado inicialmente, testes de verificação e de validação são importantes ferramentas aqui para inspecionar e realizar o controle necessário para garantia de desenvolvimento de um software seguro e que gere valor para os usuários finais.

Antonio Marcelo

Quando se trata de um projeto de grande porte e prazo relativamente curto, é preciso pensar cada uma das etapas minuciosamente. Desde o planejamento, até a entrega para o cliente final. Existem técnicas para auxiliar o gestor do projeto que podem se adequar melhor em cada uma dessas etapas. O BPMN, por exemplo, é uma técnica tradicional que ajuda a visualizar o projeto como um todo, facilitando assim o planejamento. Já na fase de desenvolvimento, adotaria uma técnica ágil, como o SCRUM, por exemplo. Com esta metodologia, o gestor consegue implementar um fluxo bem definido de sprints de desenvolvimento e testes, com pequenas entregas ao cliente, para que ele possa aprovar ou não; caso aprovado, o fluxo se repete, avançando para o sprint seguinte. Caso contrário, o fluxo também se repete, porém corrigindo as falhas que o cliente reportou. Desta maneira o desenvolvimento vai avançando, com pequenas entregas, o que facilita na detecção de falhas no desenvolvimento e, conseqüentemente, evitando o desperdício de tempo, que é um fator muito importante para o cronograma do projeto.

Mais uma vez o ponto crucial é a comunicação: um bom gestor de projetos sabe manter a equipe focada, definindo metas alcançáveis para todos. Um bom gestor também reconhece e elogia (quando possível) o trabalho de cada membro do projeto. Atitudes como essas mantêm a equipe unida e aumenta o nível de um fator muito importante para o projeto como um todo: a motivação dos colaboradores. Manter cada um motivado pode ser a chave para um projeto onde a comunicação flui e tem uma alta eficácia e eficiência, realizando as entregas em tempo e com alta qualidade.

Primeiramente, é preciso que a que toda a equipe conheça bem essas práticas. Segundamente, TDD e BDD são melhores aproveitadas e mais facilmente adotadas se implementadas no início do projeto. Dito isto, o TDD é implementado criando uma rotina de testes antes mesmo de iniciar o desenvolvimento da aplicação/funcionalidade em si. Em seguida, o desenvolvimento é iniciado, baseando-se nos resultados dos testes (são feitos ajustes até que passe em todos os testes previamente criados). Na fase seguinte, o dev refatora o código para melhor

desempenho e legibilidade. Por outro lado, BDD é uma prática que precisa do projeto bem planejado, inclusive (e principalmente) o seu fluxo, pois todo o desenvolvimento será baseado no fluxo que o cliente deseja para o projeto. O comportamento final do projeto ser bem definido é primordial para esta prática. Já o CI/CD é uma prática que é melhor implementada não necessariamente só no início. Pode ser inserido no fluxo de desenvolvimento em qualquer estágio do projeto. Continuous integration e continuous delivery é um "protocolo" de testes em que o código é submetido, para que, como o nome já diz, seja testado e deployado automaticamente, proporcionando assim uma disponibilidade alta e entrega quase que instantânea. Geralmente é desenvolvido pela equipe de devops.

Comunicação é a chave. Para melhorar o planejamento é preciso reuniões com o product owner, para que o resultado esperado seja muito bem definido. A comunicação é imprescindível para melhorar também as estimativas e o acompanhamento do projeto. Porém, com reuniões recorrentes com os membros das equipes de desenvolvimento. Frameworks como o SCRUM, por exemplo, também são bastante úteis, pois ajudam na entrega de um software em pequenos sprints, facilitando na detecção de desvios e falhas no desenvolvimento do projeto, auxiliando no acompanhamento e evitando a perda de tempo.

Arthur Conegundes

devido ao escopo do projeto se grande e bastante complexo é evidente que necessita de bastante documentação e processos legais/burocráticos, contudo, o prazo é curto e precisa ser desenvolvido rapidamente, além de necessitar de uma eficácia altíssima visto que é um sistema crítico. Tendo esses pontos em mente o início do projeto seria um pouco mais lento por conta da preparação e documentação do projeto (como o project charter e etc.) e em seguida, na parte do desenvolvimento das tecnologias de fato, eu utilizaria métodos como o kanban para deixar as tarefas de modo visual e intuitivo para que os membros do projeto pudessem acompanhar as tasks de maneira mais fácil e organizada. Elaboraria sprints com reuniões para acompanhar o andamento do projeto no modelo scrum, utilizaria do lean development em situações onde fosse possível eliminar etapas não tão importantes para economizar tempo e refatorar o código para deixá-lo mais eficiente e sempre testaria a integração de funcionalidades do código para que possíveis erros fossem apontados o quanto antes e consertá-los.

para garantir maior colaboração e eficiência eu iria propor primeiramente uma descrição em linguagem natural (aos moldes do BDD) para que tanto os clientes quanto o time pudessem ter uma visão clara do escopo do projeto e de sua função e do seus colegas. Após o primeiro passo me reuniria com meu time e dividiria equipes (em suas respectivas funções com base em suas habilidades em que se sentem mais seguros) e adotaria métodos de pontos para estimar com precisão a

eficiência de cada equipe e a dificuldade de cada task ao invés de simplesmente estimar um tempo X em semanas , dias etc. Utilizaria também do método kanban para que todos pudessem ter uma compreensão mais intuitiva da ordem das tarefas e do fluxo de trabalho. Com as equipes bem organizadas e com os objetivos em mente eu acredito que ao final desse processo todas as equipes podem se comunicar de forma melhor através de reuniões regulares para acompanhar o andamento do projeto.

Definiria primeiramente histórias de usuário em linguagem natural para deixar mais evidente os intuitos de cada funcionalidade que o produto deve apresentar devido ao fato de ser muito complexo em termos de técnicos de engenharia (por exemplo: como o usuário interage com os controles do mecha, que tipo de sensores e quais alertas o piloto recebe de acordo com um erro que o mecha apresente etc.) em seguida estudaria com a equipe os requisitos que o software deve atender , quais tecnologias usariamos e partiria para o desenvolvimento com base em testes contínuos e integração de funcionalidades constantemente(por exemplo: testaria o sistema motor do mecha para ver se o software está interagindo com os comandos do piloto de forma funcional, em seguida testaria se o software consegue operar o sistema de armas do mecha , etc.) apontando a dificuldade de cada uma das tasks e adaptando as equipes para trabalhar em cima dos principais problemas até que , no final de toda a etapa de testes , nós conseguíssemos integrar essas funcionalidades e gerar um produto funcional.

acredito que para planejar melhor um time e estimar o andamento dela seria ideal usar , aliada à outras metodologias ágeis ,a pontuação de complexidade de tasks(planning poker) , pois não só é possível com base no histórico da equipe estimar um tempo de finalização de uma task mas também é possível acompanhar o desenvolvimento daquela equipe em pontos fortes e fracos(feedbacks), portanto , conhecendo a maturidade de cada equipe é possível manejá-las de tal forma que elas sejam mais eficientes com determinados tipos de tasks e aumente a produtividade, influenciando positivamente no andamento do projeto.

Breno Santos

O produto a ser desenvolvido em questão é um sistema cujo funcionamento é crítico para as operações realizadas no planeta, e o prazo de entrega é curto, portanto, cabe o uso de metodologias de desenvolvimento ágil como Scrum, que lida fortemente com o fator de tempo como parâmetro para a realização de entregas de modo dinâmico, já que ocorre, através das Sprints, revisões do que já foi feito e do que será realizado em seguida. É de suma importância realizar o desenvolvimento orientado a testes, de modo que ao longo da construção do sistema, as funcionalidades entregues sejam efetivamente validadas até mesmo pelos usuários, já que trata-se de um produto extremamente crítico. Portanto cabe, também, a adoção da XP, que

tem como seu forte a otimização da parte técnica do desenvolvimento de software, atributo necessário para acelerar e tornar mais otimizados os processos de construção do sistema dos Mech Suits.

Utilizaria o framework Scrum para estruturar as atividades e parametrizar os períodos de trabalho, mas para isso, seria necessário sondar os membros da equipe para se situar do nível de maturidade técnica dos mesmos, e assim estudar o melhor modelo de gestão para administrar o time. Entendendo como que funcionam: seus ritmos, pontos fortes e pontos fracos, seria possível atribuir funções e determinar as tarefas de cada um, sempre tendo a comunicação como fator chave para o sucesso da gestão. Adotaria a cultura DevOps, pois um sistema que trabalha de modo colaborativo e dinâmico, agrega mais valor do que um cujo resultado final é apenas a soma de suas partes individuais. Seriam também implementadas práticas para assegurar o trabalho realizado pela equipe, com acompanhamentos contínuos e validações com várias faces: tanto dirigidos por testes quanto pelo comportamento dos usuários. (TDD e BDD). Graças a isso, poderia tranquilamente realizar a entrega contínua das funcionalidades do sistema dos Mech Suits de modo gradual e seguro, garantindo a eficácia e eficiência do desenvolvimento do projeto. A adoção de ferramentas para tornar palpável a visualização do progresso é imprescindível, por isso seria indispensável o uso de plataformas como Trello ou Jira, para tornar concreto para mim e toda a equipe o trabalho já realizado. A chave para uma boa gestão é a comunicação, e há várias formas (verbais e não verbais, abstratas e concretas) de realizá-la, é necessário se situar em relação aos componentes envolvidos no processo: pessoas, metodologias, ferramentas, cultura, entre outras partes que compõem o contexto.

As práticas de TDD seriam extremamente importantes para o desenvolvimento do projeto para que houvesse a efetivação das entregas de modo contínuo, tendo sua validação realizada, além dos testes, pelo ponto de vista de diferentes usuários, com uma pluralidade de agentes (BDD). Graças à essa validação dinâmica e recorrente, seria possível realizar a integração contínua das funcionalidades já entregues. Para todo o processo ser potencialmente otimizado, cabe a adoção da cultura DevOps, que busca integrar em um só corpo a comunicação e trabalho colaborativo dos times de desenvolvimento e operacional (infraestrutura), assim, diminuindo barreiras burocráticas e o retrabalho de algumas partes, tornando mais fluido o processo de conceber o produto final.

Quando se trata do estabelecimento de estimativas e acompanhamentos do projeto, vale considerar o uso do Scrum, framework que busca estruturar o processo de construção do produto em Sprints, intervalos de tempo para desenvolver funcionalidades, revisar os trabalhos já feitos e os que serão concebidos futuramente, sendo assim, estabelecendo estimativas e realizando o devido acompanhamento. Para fazer com que essa organização deixasse de ser abstrata, eu utilizaria

ferramentas de organização e visualização, para assim gerir tais metodologias de modo estruturado. Um bom exemplo de ferramenta seria o Jira, que me permitiria organizar e tornar mais estruturado e demonstrativo o caminhar do projeto.

Caio Elias Rabelo Pinheiro

Levando em consideração o tamanho do projeto e a importância é extremamente necessário que o processo tenha uma alta documentação para que os desenvolvedores possam trabalhar melhor entre si e caso haja mudança na equipe não haja problemas, além disso um pouco de rigidez é necessário em grandes projetos, ao mesmo tempo a depender da maturidade da equipe é interessante o uso de kanban, que dá mais liberdade para a equipe e pode ajudar com prazos apertados, já que cada desenvolvedor pode escolher uma certa parte do projeto que julgar mais confortável, agilizando o processo, do método scrum podemos utilizar a cultura dos encontros, tanto para manter a equipe informada do estágio atual do projeto e de mudanças no rumo, já que temos testadores na equipe e os feedbacks deles são de suma importância dado o propósito do projeto, segurança e testagem devem ser os pilares do desenvolvimento de um software como esse, por isso a utilização de características de metodologias tradicionais, mas sem perder a dinâmica e objetividade de abordagens ágeis

Para um melhor gerenciamento muitas técnicas podem ser utilizadas, acredito que uma das principais seria o sistema de pontuação de equipes existente nas metodologias ágeis, com ele é possível medir a capacidade e eficiência do grupo e de forma mais certa delegar tarefas, já que o BDD seria adotado acredito também que as equipes teriam instruções bem claras do que deve ser feito, o que colabora com uma comunicação clara e eficiente, outro ponto essencial é o trabalho da equipe de testes, que devem passar feedbacks claros e construtivos para a equipe, mantendo assim a qualidade do software bem alta, tudo isso juntamente com um ambiente de trabalho confortável possibilitaria que o projeto fosse concluído sem maiores problemas.

Já que o projeto tem como principal foco o uso prático do sistema acredito que o BDD encaixaria melhor, levando em consideração que os usuários tenham em mente como deve funcionar a utilização das mecânicas e ter demandas específicas no uso do software então seguindo os feedbacks e instruções deles os desenvolvedores conseguem seguir bem o projeto, a utilização de CI nesse projeto é de grande importância para garantir a funcionalidade da aplicação sem que o código quebre, além de manter uma versão mais atualizada sempre, por fim acredito que DevOps seja indispensável nesse projeto pela diversidade da equipe e importância do projeto, para unir todas as sub-equipes e conseguir gerar um resultado melhor já que deve haver uma colaboração e comunicação melhor entre todo o time.

Para o planejamento acredito que o kanban e as dailys são indispensáveis, eles ajudam a saber quem deve ficar com cada parte do projeto além de sempre se manterem atualizados, para as estimativas um ponto que pode ser utilizado uma junção de dois sistemas um deles é o de pontuação de tarefas, que ajuda a medir não só a complexidade mas também o tempo que cada atividade pode levar, o outro são a medição do tempo através de sprints, com isso seria mais fácil classificar cada fase do projeto em termos de tempo e complexidade, para o acompanhamento do projeto as reuniões continuam sendo um ponto forte, mantendo toda a equipe atualizada e a parte do que está acontecendo, e como dito em quase todas as respostas até aqui, o trabalho da equipe de testagens é primordial para o entendimento do que está acontecendo no projeto, eles ditam a qualidade do software e permitem que o projeto ande.

Carlos Kaynan

Em um ambiente híbrido, normalmente o software é grande e complexo devido à implementação de metodologias ágeis e tradicionais, onde se preza a velocidade de entrega do projeto, a segurança, a documentação e acompanhamento do projeto pelo cliente. A princípio, abordaria o uso do Scrum para destrinchar e dividir o projeto com a equipe, a vista de receber Sprints diários ou semanais, dependendo da complexibilidade do projeto, podendo também fazer uso do kanban se preciso, para ter uma ideia de metas mais "visual" sobre o projeto. Posteriormente utilizaria de abordagens tradicionais para fins de documentar e regularizar o projeto, e claro sempre tendo no planejamento o acompanhamento do projeto pelo cliente, utilizando de métodos como "história do projeto" onde o cliente conta como ele utilizaria o software de modo meio informal, no intuito de melhorar a compreensão da equipe em relação a como o software irá funcionar.

Uma prática que estou familiarizado e que funciona muito bem, é o uso do "Trello" como ferramenta de organização, visão de projeto, metas, datas, entre várias outras vantagens que essa ferramenta dispõe. Podendo gerenciar a equipe da melhor forma possível, otimizando o trabalho em equipe, a colaboração, e consequentemente a eficiência no desenvolvimento do projeto. Outra prática que me agrada e com certeza agregaria muito valor a equipe, é a cultura DevOps, com automatização de serviços e infraestruturas, metas e códigos otimizados, eu teria um time mais eficiente, colaborativo, e eficaz.

Em um cenário onde eu escolheria as práticas adotadas para desenvolvimento de um projeto, a princípio utilizaria a prática do TDD onde o "problema" ou "teste" daria erro (não haveria código), o membro da equipe faria a solução básica para passar por esse "problema" e posteriormente otimizaria o código. Eu não utilizaria do método BDD, apesar de muito bom, não me agrada muito o teste de cada "situação" que o software abordará, eu acho o método anterior mais eficaz. A depender do projeto, eu

utilizaria o CI/CD, principalmente o CD, pois o método de automatizar os testes e ainda ter acompanhamento do cliente, ao meu ver, é muito eficaz, produtivo e agrega muito valor ao projeto. E em relação à cultura DevOps, eu acho de extrema importância cada vez mais as empresas adotarem essa cultura, pois a colaboração de cada membro do time, com metas, foco, gerenciamento de equipe, automação de serviços, são algumas das várias boas práticas e benefícios que a DevOps agrega ao time.

A princípio, me agrada muito o método de "pontuação do projeto", onde a equipe dá uma pontuação em relação à complexibilidade do projeto a fim de estimar um tempo limite para entrega do mesmo, definir as metas e decidir alguma das metodologias ágeis que se encaixe melhor para aquele determinado projeto. Por exemplo, o cliente quer um software de cadastro muito simples, a equipe iria dar uma pontuação a esse projeto e escolheria por exemplo o Scrum como metodologia, pelo fato de ser simples, daria Sprints diárias, sem necessidade de usar algumas práticas como TDD ou BDD. Outro exemplo seria um software muito complexo e com uma grande quantidade de dados, com o método de pontuação, o time poderia optar por um ambiente híbrido, somado com a prática do CD para testes e acompanhamento do projeto pelo cliente.

Cauê Marinho Gomes de Souza

Notando-se que o cenário de desenvolvimento do projeto se apresenta como "híbrido", fica mais apropriado a adoção de metodologias que atendam às expectativas de agilidade e entrega com alta qualidade, porém que venham a abranger pessoas com diversos níveis de conhecimento na mesma. Dito isso, há a necessidade da implementação de metodologias práticas com características de abordagens tradicionais, tais como a formatação de uma documentação bem feita para um melhor controle do andamento do projeto, aliadas às metodologias pré-estabelecidas no mercado como o SCRUM/KANBAN. A escolha pelo conjunto das duas metodologias se dá pelos fatores que elas têm para agregar no processo de desenvolvimento do produto. Enquanto o SCRUM, trabalha em torno de Sprints, normalmente diárias, as quais possibilitam uma maior adaptação da equipe aos desafios e necessidades do projeto, o KANBAN preza por uma melhor visibilidade do andamento do projeto no geral, se apresentando por meio de tabelas sobre a situação atual do desenvolvimento do produto, dando assim um "NORTE" para os membros da equipe responsável pelo desenvolvimento. Vale ressaltar que seria interessante uma apresentação prévia dos modelos a serem aplicados no processo de desenvolvimento do produto, visto que o nível de conhecimento da equipe pode ser variado e não constante. Dessa forma, garantiria-se uma maior acurácia no seguimento do planejamento do projeto.

Creio que para um bom gerenciamento da colaboração de toda a equipe, se faz necessário atributos essenciais aos gestores no geral, não somente coligados à engenharia de software. Primeiramente, se faz necessária a compreensão do ambiente em que será desenvolvido o projeto, bem como os participantes da equipe a ser comandada. Verificar os interesses e o comportamento dos membros, vai se valer tão importante quanto o nível técnico de cada um. A partir disso, entendendo as diversas perspectivas dentro da equipe, pode ser mais fácil a divisão de tarefas, uma vez que as capacidades de cada um precisam ser atendidas. Ademais, um espírito de liderança, capaz de lidar com diversas situações, deve ser mostrado a fim de sempre incentivar a equipe a lidar com obstáculos e novos desafios que sempre hão de aparecer no meio do processo de desenvolvimento do projeto. Abrindo o escopo para fora das características do Sênior, também é indispensável uma boa apresentação dos objetivos a serem alcançados no projeto do software. Desde um bom entendimento das necessidades do cliente(o qual pode ser atingido tendo em vista a implantação de uma técnica de BDD) , até a formatação de um código limpo(Começando por uma linguagem natural a fim de serem definidas estruturas básicas e essenciais), tudo isso se faz importante na comunicação da equipe e, consequentemente, na colaboração entre os membros, aumentando a eficiência e eficácia do projeto.

Primeiramente julgaria ser necessário o entendimento por parte da equipe acerca dos requisitos exigidos pelo cliente na entrega do produto final, e isso se define pelas funcionalidades a serem oferecidas pelo software. Tal entendimento pode ser dado por meio do uso da técnica das Histórias do usuário, uma vez que há o contato direto com o cliente, e são compreendidas diversas necessidades a serem atendidas e implementadas no software final. Creio que as práticas de BDD seriam mais interessantes, visto que uma ótima interação software-usuário e as funcionalidades sendo atendidas de forma correta e coesa são o que agregam valor ao produto final. Ademais, a utilização de CI se encaixaria bem, uma vez que por se tratar de um contexto de bom funcionamento do software na "civilização", há a necessidade de uma atualização contínua no projeto a fim de evitar possíveis erros e transtornos na utilização e manutenção do produto. Além disso, tendo em vista que o trabalho a ser realizado se dá em um ambiente híbrido, se faz necessária uma cultura DEVOPS tendo como objetivo uma maior integração das partes colaboradoras, gerando uma maior fluidez na execução das tarefas.

Como já abordado no processo de implementação, a utilização da metodologia do Kanban seria de fundamental importância uma vez que a mesma dá conta de dar uma maior visualização para a equipe de como está o andamento do projeto. Dessa forma é possível ter uma maior organização das tarefas e realizar uma previsão acerca do que ainda precisa ser feito. No entanto, também se faria necessária a adoção de estratégias como a atribuição de pontos de estimativa, os quais seriam

definidos pelos desenvolvedores em uma lógica de dificuldade e complexidade estimada a partir das histórias do usuário coletadas previamente. Dessa forma, a equipe poderia gerir melhor o tempo e o esforço a serem gastos na produção do software, e também ajudaria no gerenciamento de tarefas, a fim de serem cumpridos todos os requisitos previamente postados no planejamento inicial do software.

Demetriu Gabriel

Implementaria as metodologias ágeis baseado fortemente na documentação bem feita das abordagens tradicionais e nos desafios da equipe. Pois, na documentação estaria claro como as demandas das tasks estariam chegando para os desenvolvedores, sendo um sênior, além das demandas das tasks e como elas chegam ao time, é possível conseguir métricas e conclusões de hard skill e soft skill, como: verificar a maturidade do time, colaboração entre o time e outros stakeholders relacionados ao projeto. Além disso como o projeto está com o tempo apertado e alta prioridade, poderíamos utilizar o Lean como metodologia, devido que com ele podemos construir um MVP para e fazer uma melhor manutenção no código incrementando com o tempo, entrega rápida com qualidade de software e validação do cliente, relacionando essa metodologia ágil com as abordagens tradicionais de software, para obter melhor gerenciamento, contribuição e colaboração entre o time usano DevOps para otimizar tasks e trabalhar a colaboração entre os stakeholders.

utilizando metodologias ágeis, práticas de desenvolvimento com abordagens tradicionais, mas antes iria fazer uma leitura da equipe e stakeholders para determinar o grau de maturidade, colaboração e como o trabalho é feito para ter uma base. Com equipes com uma maturidade menor talvez ocorram problemas de complexidade relacionadas ao projeto, documentação, como gestão e acompanhamento, para ajudar usaria uma prática de desenvolvimento como BDD e DevOps relacionado com CI\CD para melhorar a relação a colaboração da equipe e o comportamento do cliente e do software com sua validação, juntamente com DevOps otimizando as tasks e o BDD em como essas tasks estão chegando ao time, reduzindo retrabalho, otimizando código e atendendo as dores do cliente, Para completar para gerenciar estimativas, acompanhamento, uma metodologia ágil como Scrum com seu Scrum Master, relacionando as Sprints e dailys para bom acompanhamento e no final de cada Sprint revisar o que foi feito e o que pode ser melhorado, colabora para melhor eficiência e colaboração entre equipe, resultando num melhor ambiente de trabalho. um software com entrega rápida, contínua, compartilhada, com melhor qualidade de software e aproveitando rápido do feedback do cliente e dos stakeholders e aumento da maturidade da equipe.

Para promover a adoção dessas práticas de desenvolvimento, o TDD adotaria em relacionado ao BDD, porque, o TDD como os testes são escritos antes, e depois seria

necessário escrever um código mínimo para passar nesse teste, ele pode ser revisado ou estaticamente ou dinamicamente para melhorar a eficiência, com a finalidade de obter uma melhor qualidade de software, eficiência e um melhor entendimento do problema, melhorando a rapidez para escrever a solução, usando o CI para manutenção do código contínua e de forma compartilhada, porém com o BDD o comportamento do usuário estaria relacionado ao comportamento do software, resultando numa melhor validação do software e feedback mais rápido do cliente, colaborando para melhor colaboração entre setores do time, e como o BDD vem a relação com o CD e DevOps para melhor otimização com a extensão do CI junto para aumentar a colaboração e melhor gerenciamento com eficiência das suas atividades.

Uma das maneiras de melhorar a estratégia da equipe é identificar a maturidade dela, a colaboração entre esses integrantes com isso podemos utilizar metodologias como Scrum para definir suas metas e tasks, com seu modelo de sprint e dailys para no final a revisão desse sprint e obter resultados do que pode ser melhorado, resultando em um maior acompanhamento do projeto, ou com o Kanban com seu quadro sobre as tasks e em que modo estão como: fazer, fazendo e pronto. Tendo reuniões para alinhar o planejamento e estimativas e acompanhamento para o projeto relacionando com a cultura DevOps, CI\CD para melhorar as operações e colaboração entre o time, resultando numa maior eficiência na entrega de um software e melhorando e acompanhando o projeto.

Diego Rafael Gomes de França

Na questão do Scrum, eu implementaria de maneira em de entender como estão os conhecimentos da minha equipe em relação ao projeto, e nas sprints eu debateria em relação ao que tem que ser entregue, alinhando quais são as necessidades urgentes propostas pelo stakeholder, quais são os impedimentos de cada um dos integrantes. Como se trata de um ambiente híbrido, eu usaria o Kanban para entender quais são as tarefas com um maior importância e definiria no quadro de tarefas as prioridades buscando a conclusão. Já usando a Lean, eu deixaria as tarefas que não são tão urgentes em segundo plano, buscando resolver elas quando não houvesse uma tarefa crítica a se resolver. E por último, usando o XP, eu aumentaria a análise do código, criaria vários meios de fazer os testes, entraria sempre em contato com os analistas de negócios e as pessoas que irão usar o projeto final para ver se aquilo que estamos criando é o proposto e funcionável.

Definiria as demandas menos complexas para membros com menos experiência e que ainda precisam de supervisão de alguém para aprender ainda mais e em caso de dúvida, pediria ajuda a alguém que tenha uma senioridade menor, buscando também o crescimento dos membros mais novos em relação a isso, fazendo com que eles entendam a complexidade e possam avançar em outras demandas de maior nível. Através também da cultura DevOps, buscaria trazer de maneira

construtiva a interação da equipe no escopo do projeto. Também gerenciaria por meio das dailys buscar com que haja uma interação entre os membros, buscando mostrar com que a ajuda mútua se torne importante tanto para o avanço da equipe e dos membros. Buscaria de maneira constante conversar com os Stakeholders, mostrando o avanço do projeto, as funcionalidades completas, entendendo de maneira que o que eu desenvolvi está alinhado com o processo de negócio posto entre as partes e buscando melhorias com meu time, fazendo 1:1, comentando sobre possíveis soluções, resolução de problemas e buscando melhoria.

Eu promoveria a adoção das práticas de TDD mostrando na equipe como a prática do Teste (foi meio redundante) para os engenheiros de software, é algo importante para o produto ser efetivo, buscando entender que é necessário ter um alinhamento na questão de quando for desenvolver, buscando uma alternativa de otimizar o código e buscando refatorar sempre que necessário e possível, já que um dos princípios principais do TDD, seria isso, promoveria também que a equipe de testadores buscassem fazer o teste com qualidade, de maneira automatizada para um melhor desempenho na hora da entrega, já a questão do BDD, entraria também juntamente com o TDD com os analistas de negócio, fazendo com que haja entendimento do que está sendo criado, tendo uma colaboração mútua com o produto final, ouvindo cada parte e trazendo um rascunho do escopo do projeto que está sendo desenvolvido. Já no CI/CD, eu traria como alternativa a documentação de tudo o que foi proposto a ser desenvolvido, como as features inseridas (testadas e aprovadas) no sistema, através de sistemas de versionamento. Também, ao implementar uma cultura DevOps eu buscaria uma maneira de obter uma cooperação mútua entre as equipes (seja engenheiro de software, testador) trazendo meios que essas práticas se alinhem com o time e tragam os resultados esperados tanto pela equipe que está desenvolvendo quando pelos usuários finais.

No planejamento, a estratégia que eu usaria era criar Sprints, e cada dia, antes do dia da sprint, eu faria dailys com os membros do time, buscando entender como eles estão em relação ao que tem que ser entregue, se eles estão entendendo qual é o escopo do projeto, quais são suas dúvidas, suas necessidades de melhoria, como eles podem melhorar, se possuem algum impedimento em relação a isso. Cada dia, eu buscaria a alternativa de fazer com que eles entendam os prazos para o progresso do projeto através disso, eu usaria o Planning Poker para estimar os projetos de forma eficaz, com um debate com todos os membros, eu discutiria quais são as demandas necessárias para o progresso do projeto, calcularia esses Story Points de maneira democrática, distribuindo as tarefas mais complexas para os membros com maior senioridade, e tarefas menos complexas para membros do time que não possuem tanta experiência em lidar com as demandas. Buscaria também contribuir com cada um, mostrando a maneira que eles atuam usando o framework ágil de forma adequada, e também usaria ferramentas de projetos (como Jira, Trello), para

organizar de maneira efetiva a contribuição das demandas de cada membro no projeto.

Eduardo Dias de Oliveira Teles

Por se tratar de um projeto de extrema importância, e por serem produtos que impactam diretamente na segurança do usuário final, é importante que o projeto seja bem planejado e documentado, para atender as expectativas do usuário final. Pra isso, eu utilizaria uma abordagem tradicional no início do projeto, mais na fase de planejamento, e depois disso utilizaria o scrum na fase de desenvolvimento, para entregas contínuas e de fácil adaptabilidade, e o kanban para poder visualizar o desenvolver do projeto mais claramente, através de quadros de objetivos.

A própria cultura DevOps é uma grande aliada nessa questão da colaboração, pois tenta unir todas as equipes que tem uma participação no projeto, isso já é um grande ponto de partida para a eficácia e eficiência do projeto. Porém um outro ponto importante seria utilizar a métrica de pontuação citada na questão anterior para motivar a equipe. Quando um colaborador recebe atividades que não consegue cumprir, isso acaba o desmotivando. Por isso, sabendo o rendimento da equipe e sabendo a dificuldade de cada atividade através das pontuações, posso designar atividades que a equipe vá conseguir concluir, os motivando a continuar entregando cada vez mais.

Criaria casos testes e histórias de usuário na fase de planejamento do projeto, junto com o Stakeholder, que iriam garantir que as funcionalidades das máquinas iriam ser cumpridas. Depois disso, utilizando metodologias ágeis, iríamos começar a desenvolver a máquina da forma mais rápida, que passe no teste, e mostre o comportamento esperado. Depois disso iríamos "lapidando" o produto para que fique mais otimizado e pronto para entrega. Isso iria prevenir que possíveis falhas ocorressem no futuro, onde os custos seriam mais altos. O Ci/CD também é importante para entregas constantes, e para que as partes desenvolvidas sejam testadas em conjunto, ou seja, o tempo inteiro, quando os testes tivessem sendo realizados, eles iriam ser realizados juntos, para ver se estão funcionando em conjunto, e também, uma entrega automatizada para que quando o produto fosse testado, já pudesse ser implementado rapidamente para o usuário poder utilizar.

Eu utilizaria uma métrica de pontuação, onde na fase de planejamento, a equipe em conjunto definiria a pontuação de cada funcionalidade do produto, utilizando o nível de dificuldade como parâmetro, e com isso, durante a execução do projeto, eu teria como saber o progresso de cada sprint, vendo a pontuação que a equipe conseguiu atingir, e assim, poder designar melhor quais funcionalidades seriam aplicadas, e quais seriam focadas, e também poder identificar se está havendo alguma dificuldade em alguma funcionalidade.

Evandro de Souza Santos Junior

Visto que , o prazo é curto , é necessário boas práticas para o bom andamento do projeto , como estão com um prazo curto para entrega , usar os sprints do Scrum se torna essencial pois vai dividir em ciclos curtos , os processos que tem que ser feito e no final de cada sprint é entregue algo funcional para o cliente , até chegar ao produto final. Ademais , estão enfrentando problema quanto a organização , eu aconselharia a utilizarem o quadro kanban também , listando o que é para fazer, o que está sendo feito e o que foi concluído, aumentando assim a eficiência e deixando mais visual a organização do projeto. Uma outra prática que adotaria seria a história do usuário, melhorando assim a eficiência da equipe na codificação para não haver desperdício de tempo , visto que estão com um prazo curto, codificando algo para depois ter que refazer do início, isto vai diminuir o desperdício de tempo no decorrer do projeto. Além disso, é necessário também uma documentação de qualidade, para que as mudanças no decorrer do projeto não sejam um problema pois se trata de um projeto grande, portanto, a utilização das metodologias ágeis , tornam o projeto flexível e eficaz , buscando a entrega do produto com qualidade e velocidade e assim , respeitando pilares fundamentais como testes, segurança e documentação para a entrega deste software com qualidade e êxito.

Para a melhoria desses fatores, o Kanban é essencial, por deixar mais visual o que tem que fazer , o que está sendo feito e o que já foi concluído , isto aumenta a organização, o planejamento , além de deixar claro para os componentes do projeto como está o andamento do mesmo. Além disso, reuniões diárias é essencial para saber como está o andamento das atividades que estão em execução , feedbacks entre os que estão envolvidos no projeto, isso fará com que prazos sejam cumpridos e o planejamento seja bem estabelecido. Outro fator importante é dividir o que tem que ser feito em pequenas tarefas , para que não haja sobrecarga em nenhum componente e atrase o cumprimento do prazo porém, nenhum componente pode deve está sem tarefa a serem executadas , este monitoramento aumenta eficiência da equipe e faz com que sempre tenha um bom acompanhamento e o projeto seja feito no prazo estabelecido.

A utilização de uma cultura DevOps no projeto, é essencial, isto aumentará a comunicação sem fronteiras entre equipes envolvidas no projeto, as práticas do CI/CD como a automatização e testes irá aumentar a eficiência da equipe , um repositório organizado e bem estruturado para garantir uma boa documentação. Além disso , práticas do Kanban para uma organização visual para sempre está listado o que há de fazer e o que tem de ser feito , uma ferramenta que utilizaria seria o Notion para a equipe está sempre acompanhando o decorrer do projeto, assim aumentando assim a eficácia da equipe. Além de sempre fazer o monitoramento das tarefas atribuídas, se alguém está com dificuldade, para assim ajudar ou solicitar que alguém ajude, se todos estão com uma tarefa, como está indo a execução da

mesma, verificar que ninguém esteja sobrecarregado , isso fará com que o gerenciamento seja eficiente e não ocorra atraso na entrega do projeto.

Como prioridade , é necessário a utilização de testes no decorrer do projeto , utilizando a abordagem TDD, isto faz com que os desenvolvedores não esqueçam de realizar os testes, pois iriam fazer primeiro os testes para depois começar a codificação para passar por esses teste, depois que passa , são feitas melhorias no código e estes testes tem que ser automatizados para eliminar desperdício de tempo no projeto. Quanto as práticas do BDD, é de extrema importância , entender o usuário por meio de seus comportamentos, para assim fazer implementações baseadas no feedbacks e demandas que são ditas pelo usuário , isso aumenta a eficiência na equipe e aumenta a velocidade na entrega do produto final por tá em constante comunicação com o usuário e codificando já com implementações demandadas. Já quanto a CI e CD a integração contínua e a entrega contínua é fundamental, ter um repositório compartilhado entre os desenvolvedores , o qual as mudanças são documentadas lá e testes automatizados são realizados antes de cada lançamento no mesmo , é importante pois alerta no início o problema e não quando já está em avançado um parte do desenvolvimento , ademais o deploy automatizado da tempo a equipe e as mudanlas já são feitas ao o produto que está sendo usado pelo usuário. As práticas de DevOps, colaboram com a boa comunicação entre equipe de desenvolvimento e a de operações, utilizam as práticas de CI/CD , isto faz com que tenha uma quebra de fronteira entre diferentes setores do projeto, além de práticas de documentação.