

Surfando no Scrum





Surfando no Scrum

Introdução



Prof. Adriano Martins

ESTE DOCUMENTO CONTÉM INFORMAÇÕES PROPRIETÁRIAS, PROTEGIDAS POR COPYRIGHT. TODOS OS DIREITOS RESERVADOS. NENHUMA PARTE DESTE DOCUMENTO PODE SER FOTOCOPIADA, REPRODUZIDA OU TRADUZIDA PARA OUTRO IDIOMA SEM CONSENTIMENTO DA PMG ACADEMY LTDA, BRASIL.

Os slides deste curso contém itens da Visual AGILExicon® , que é uma marca comercial da Innolution , LLC e Kenneth S. Rubin.

© Copyright 2012 - 2016, PMG Academy. Todos os direitos reservados.

www.pmgacademy.com

Design: By Freepik

Objetivo

O objetivo deste curso é ensinar as noções básicas do desenvolvimento da metodologia Scrum, e te dar uma base preparatória para uma das diversas certificações internacionais disponíveis, independentemente da base de conhecimento, seja do Scrum Alliance, Scrum.org ou EXIN para os níveis Foundation ou Master. Seja qual for o objetivo, é recomendável que você aprenda os conceitos e a abordagem que faremos deste fantástico método ágil de forma bem didática e objetiva.

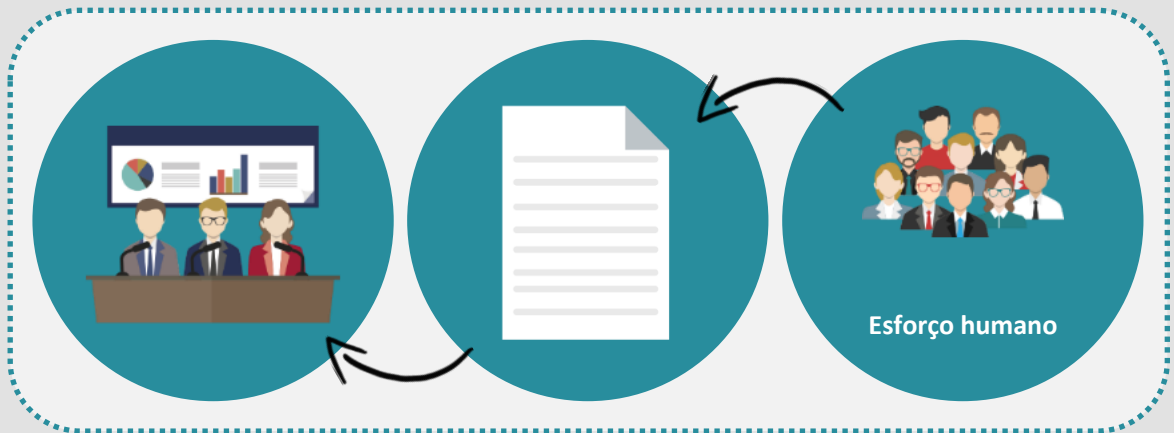
Você terá uma visão geral de alguns dados relativos ao gerenciamento de projetos e ao desenvolvimento Ágil. Depois, vamos surfar em alguns princípios básicos que você precisa entender antes de entrar na onda do Scrum, falando também sobre as estruturas da metodologia, os artefatos-chave e a implementação do Scrum no mundo real.

Recomendo que absorva todo este conteúdo antes de iniciar qualquer curso de Scrum, preparatório ou não.



No Começo de Tudo

Os projetos sempre existiram, mas nem sempre havia o seu gerenciamento, principalmente da forma que conhecemos hoje. Antigamente, algo era construído puramente com o esforço humano e posteriormente era documentado o que havia sido feito, mesmo que precariamente e sem os recursos que temos hoje em dia, através dos softwares e aplicativos.



Então, surgiu um problema: Com a existência de inúmeras pessoas trabalhando, a questão de como gerenciar grandes projetos era debatida. Sem a tecnologia apropriada na época, praticamente tudo era planejado com papel e caneta na mão e um amontoado de pessoas em salas de reuniões. Diante do caos, houve a necessidade de criar o papel de gerente de projetos, um “salvador”, aquele que colocaria ordem neste complexo emaranhado de processos, atividades, documentos e pessoas.

Proveniente da engenharia e construção civil, as disciplinas de gerenciamento de projetos eram repetidas e com pouca flexibilidade. Não estou insinuando que a flexibilidade seja boa, principalmente em excesso, pois a ideia central é gerenciar os projetos, as pessoas, o produto e fazer com que todos tenham o seu foco proposto no projeto. O desenvolvedor desenvolve, o cliente recebe o produto, o testador testa a solução, o gerente gerencia e o fornecedor contribui para o projeto, tudo encaixado e funcionando.

Entretanto, será que o método tradicional (*waterfall*) de gerente de projetos é o modelo mais adequado para os dias de hoje, principalmente para projetos na área de TI? É o que veremos mais a frente.

O Falso Sucesso

Não é raro vermos indicadores e gráficos mostrando a porcentagem de projetos de TI que fracassaram nas empresas, mesmo com a montagem de escritórios de projetos (PMO), metodologias, melhores práticas enraizadas na cultura da empresa, artefatos, profissionais certificados na área de gerenciamento de projetos, etc.

Você mesmo poderia parar por um momento e pensar na quantidade de projetos de TI na qual já trabalhou, seja como recurso, gestor ou até mesmo como cliente, que fracassou. Não é difícil encontrar números alarmantes, porém reais, de $\frac{1}{4}$ ou mais de projetos que fracassaram. Onde estamos errando se há tanta metodologia e práticas no mercado? Quando digo fracasso, quero me referir a fracasso em termos financeiros, prazos e de qualidade.



Do ponto de vista do gerente de projetos, não é bem assim, mas do ponto de vista do cliente, esses dados podem ser bem reais; aliás, se coloque novamente no papel do cliente e faça essa análise. Eu sei que não será difícil, pois um dia cliente aqui, outro como fornecedor lá.



Será que algumas vezes torcemos para que um cliente faça uma solicitação de mudança para aproveitarmos e aumentarmos o prazo para encobrir um mau planejamento do projeto?

Alguns lugares podem mostrar uma boa estatística, mas não é bom tomar como base números estatísticos, pois se temos 10 projetos, sendo 1 importante para o negócio e 9 sendo apenas projetos operacionais, e este único projeto estratégico falha e causa um prejuízo, então não seria uma boa mencionar em relatórios que 90% dos nossos projetos são concluídos com êxito. Isso é falso, mascara os resultados. Na prática, os projetos deveriam ser considerados finalizados com sucesso aos olhos do cliente.

Adotando o Gerenciamento de Projetos

O PMI, o maior e o principal instituto de gerenciamento de projetos do mundo, criado por 5 engenheiros em 1969, publicou um guia chamado PMBoK, contendo as áreas de conhecimento da gestão de projetos. Devido a forte ligação entre a engenharia e a TI, estas práticas foram facilmente e rapidamente aceitas dentro da área de TI.

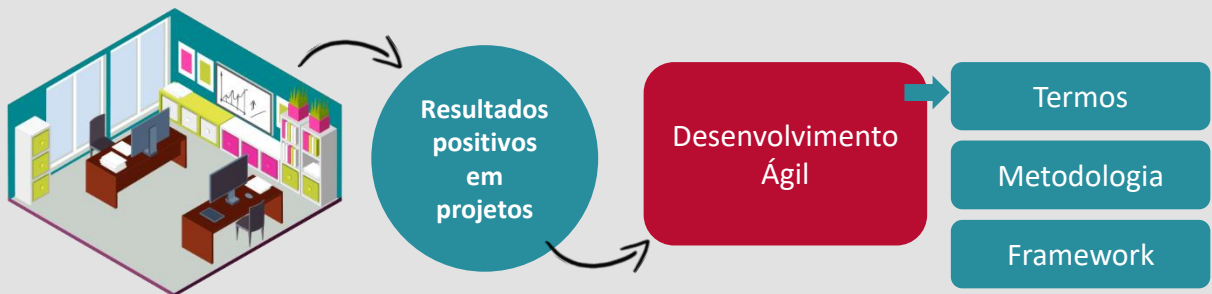
Naturalmente, devido à dinâmica das áreas de TI, principalmente no setor de desenvolvimento de software, foi percebido um descompasso e desalinhamento quando comparados ao ciclo de vida de Desenvolvimento de Software. Consequentemente, o desempenho dos projetos não eram como o esperado, mesmo contando com profissionais altamente qualificados e certificados pelo PMP®, que representam hoje a certificação de maior prestígio e respeito no gerenciamento de projetos mundial, não pela sua dificuldade em ser aprovado no exame, mas pelo seu conteúdo e por suas práticas, que são aceitas e extremamente funcionais.

O PMBoK é fantástico, é completo, mas ao longo do tempo percebeu-se que há alguns passos adicionais que não estavam sendo contemplados neste guia. Desenvolvedores se questionavam como poderiam obter melhores resultados, e as respostas sempre os direcionava à metodologia tradicional do guia do PMBoK, mas olhos atentos já enxergavam o desenvolvimento ágil como algo mais agregador.



Framework e Metodologia

Empresas começaram a observar os resultados positivos em projetos gerados pelo desenvolvimento ágil e se interessaram nos principais termos mencionados, a metodologia e o framework. A metodologia é o estudo do método, nela são descritos os passos da execução do processo, de forma organizada, iterativa e repetitiva. Porém, em contrapartida, o Scrum também é considerado um framework.



Framework



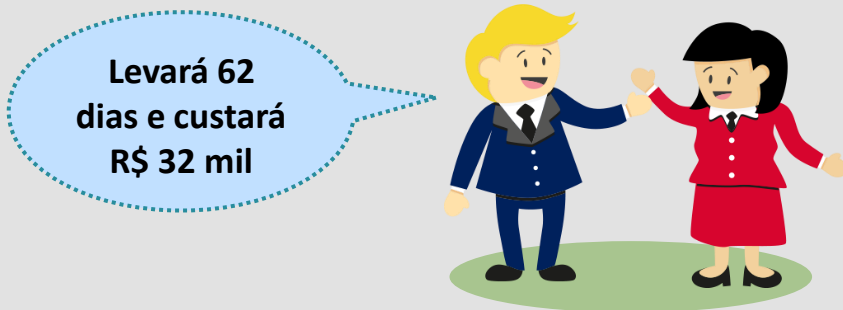
É estruturado, abrangente e normalmente mais flexível que a metodologia. Flexibilidade esta que nos controla, não nos deixa desviar dos objetivos e nos puxa sempre para o que nos interessa.

O framework SCRUM nos permite trabalhar com outras ferramentas, tais como Crystal, XP, DSDM, FDD e outras práticas ágeis. Da mesma forma que o PMBoK nos informa como os projetos e processos de todas as áreas devem ser cumpridos, isso também acontece em um framework ágil, assim que é apresentada esta variedade de ferramentas ágeis. Mesmo sendo passos formalizados, ou seja, outras metodologias ágeis que vão além do Scrum, é possível combiná-las e utilizá-las de acordo com a necessidade.

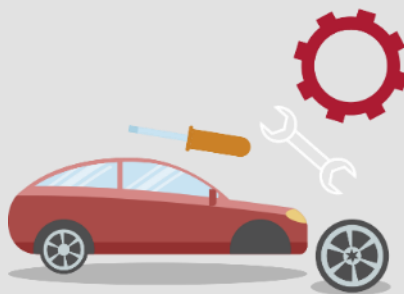
Não interessa qual metodologia utilizamos, o fato é que todas vão buscar atender a Qualidade através do conhecido triângulo de ferro, composto pelo Prazo, Custo e Escopo.

Estimativas Mascaradas

Como base em um escopo, devemos estimar o prazo e o custo de forma eficiente. No gerenciamento de projetos tradicional, o correto é sermos precisos no planejamento e nas estimativas, dizendo ao cliente que o projeto levará 62 dias e que custará R\$ 32 mil para ser concluído, e 62 dias depois, o cliente pagará exatamente R\$ 32 mil.



Garanto, não faremos estimativas realistas de primeira, pois alcançamos isso depois de muita prática, e talvez precisamos recorrer às probabilidades. Projetos de desenvolvimento de software de TI nunca são iguais, os requisitos mudam muito e por isso é difícil definir uma estimativa. Faça um teste você mesmo: Imagine quanto tempo você levaria para realizar uma nova atividade, como por exemplo, trocar o pneu de um carro que acabou de comprar. Pergunte às pessoas à sua volta e você ouvirá 15, 20 minutos, ou algo perto disso. Porém, nunca ouvimos 13 minutos e 32 segundos, ou 16 minutos e 47 segundos, algo mais exato.



Agora, troque o pneu do seu carro umas 30 vezes e estime novamente. Com certeza, a sua probabilidade inicial será substituída por uma estimativa mais realista. Este é um ponto essencial na estimativa, não tenha dúvida!

Estimativas Mascaradas

Em se tratando de estimativas, quantas vezes o cliente já te pediu para cortar custos ou diminuir o prazo? Normalmente, a documentação e os testes são os primeiros a serem deixado de lado. Sabendo disso, os projetos seguintes são estimados com um prazo e um custo maior, já esperando pelo corte, e os próximos projetos seguem a mesma rotina, proporcionando um ciclo vicioso.



Para piorar a situação, os recursos do projeto também aumentam as suas estimativas por não conhecerem exatamente quanto tempo levarão para concluir certas atividades. Com certeza, mais cedo ou mais tarde os projetos sofrerão cortes, então, os analistas e desenvolvedores, sabendo disso, inserem essa “gordura” em suas estimativas. O que não faltam são técnicas para estimar, como por exemplo, métodos de ponto de função. Entretanto, talvez falte algo muito importante: a experiência, para sair da probabilidade e ir às estimativas reais. Em outras palavras, nesses casos precisamos trocar muitos pneus!



O mais complicado é saber que serão trocados 30 pneus, porém de carros, tamanhos, formas, ferramentas e locais diferentes.

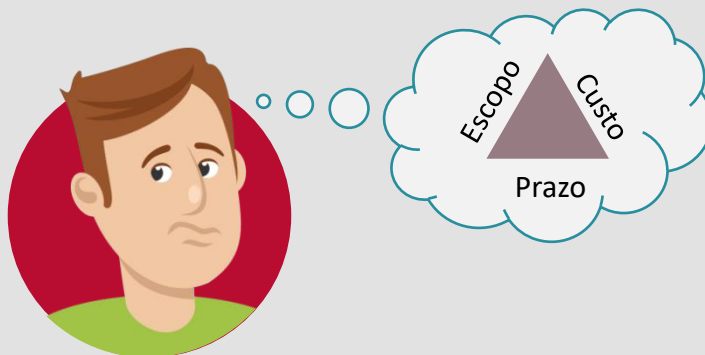
Eliminando as Incógnitas

Você pode até minimizar o problema da estimativa no momento do primeiro projeto, porém, percebe-se que nos próximos o problema persiste. Sempre há um esforço muito grande em manter o escopo, entregar no prazo e dentro do orçamento, mas os clientes acabam sempre obtendo os mesmos resultados.

A razão disso nunca funcionar é bem simples. Planejar com cada vez mais detalhes nunca resolverá o problema da maioria dos projetos, porque estamos lidando com projetos tradicionais, ou seja, é praticamente impossível definir todos os requisitos de um projeto da qual não se conhece. Cada projeto de TI é diferente um do outro.

A grande questão é que precisamos trabalhar para minimizar os riscos das possíveis incógnitas em projetos. No desenvolvimento ágil, há uma série de ferramentas que são bem diferentes do Guia do PMBoK.

O ágil nos faz pensar diferente em relação ao triângulo Escopo, Custo e Prazo...



Ou seja, ele nos ajuda a questionar sobre o que podemos entregar em uma determinada quantidade de tempo, ao invés de planejar o projeto todo.



É mais ou menos assim: A TI desenvolve passo a passo, entrega as partes e o cliente percebe o valor de acordo com as entregas. O triângulo é o mesmo, mas a forma de lidar com cada uma dos três pilares é o que será diferente.

Eliminando as Incógnitas

Quantas vezes a entrega de um projeto é totalmente diferente do que foi especificado pelo cliente?



Junto disso, vivenciamos o problema de comunicação requerido em projetos tradicionais, a documentação através de uma lista infinita de requisitos, e no final das contas, percebemos que não era isso que o cliente queria.

E assim, aumenta cada vez mais a distância entre a TI e as áreas de negócio, e nunca desenvolveremos um projeto que realmente entregue valor real ao negócio, ou seja, o cliente já deduz que você vai falhar de qualquer jeito, e simplesmente não há forma de reverter isso.

Quando Usar o Scrum?

Se as questões relatadas sobre as estimativas, excesso de documentação, falta de planejamento, distância, problemas de comunicação é algo que venha a incomodar sua empresa ou principalmente a sua área de TI, então vale a pena pensar em uma possível implementação da metodologia Scrum. Mas, se a TI e as áreas de negócios conhecem bem o que está sendo entregue, tem total domínio da tecnologia, as mudanças são relativamente baixas e as partes interessadas constantemente estão de acordo com os requisitos, é prudente avaliar se realmente há a necessidade de migrar de uma metodologia tradicional para ágil.

Caso seja preciso migrar de um ambiente complicado e complexo para um ambiente onde se busca alinhamento de diferentes opiniões das partes interessadas e onde os diferentes requisitos sejam entendidos e aceitos com total confiança e tranquilidade, o caminho é o Scrum, principalmente quando há uma variedade de culturas diferentes.



Fica muito claro que as metodologias ágeis são, na sua grande maioria, os responsáveis pela melhoria de vários aspectos, não só na entrega do produto, mas também na mudança cultural, na motivação da equipe, comunicação e foco no que é importante.

Ao adotar o Scrum como sua metodologia de gerenciamento de projetos, esteja ciente que para o Scrum, o importante é o resultado, o software funcionando e não necessariamente a documentação. O que importa é a interação acima de processos e ferramentas. O Scrum dará sempre mais valor para a colaboração com o cliente do que para a negociação de contratos. Então, você vai responder às mudanças mais do que seguir um plano.

Scrum para Todos?



Atenção: o Scrum não é recomendado em todas as ocasiões, em todas as empresas e para todas as pessoas. Por exemplo, se os requisitos são sempre bem entendidos e nunca mudam e, além disso, as partes interessadas e clientes sempre estão de acordo com estes requisitos, estes são fatores para manter o gerenciamento tradicional, principalmente se em seu ambiente não há muitas mudanças e os critérios de sucesso são conhecidos.

Se a sua equipe está bem alocada em projetos, talvez não vale a pena mudar a cultura da qual já está funcionando. Normalmente, empresas que tem uma equipe de TI muito grande e que não é possível de quebrá-las em grupos de três a nove membros, pode não ser possível alcançar resultados com o Scrum, principalmente quando há muitos fornecedores, recursos externos etc., pois no Scrum a comunicação deve ser justa, diária e constante.

Outro fator que deve ser levado em consideração antes de migrar para um ambiente ágil é verificar se há alguém disposto e disponível para assumir o papel do dono do produto, como um Product Owner, e se pelo menos ele estará apto a assumir responsabilidades delicadas, como por exemplo, enfrentar patrocinadores ou metodologias adversas.

A ideia central na opção de utilizar o Scrum é compreender se o modelo atual que usamos é aderente, traz resultados, já foi executado diversas vezes, é conhecido por todos. Também se não geram, normalmente, riscos ou atrasos e, principalmente, se as entregas são bem conhecidas, pois se tudo isso é positivo, não há muitos motivos para migrar para os métodos Scrum, pois o objetivo dele é gerar valor para a organização. Se isso já ocorre, por que mexer em time que está ganhando?

Ciclos do Scrum

Projetos tradicionais e o Scrum têm os mesmos objetivos. Gerenciar projetos. Ambos focam no planejamento, no desenvolvimento, testes, etc. Nada muda no conceito. A diferença predominante está nos ciclos.

Imagine um projeto que estima levar um ano para ser concluído. No **projeto tradicional**, investimos, por exemplo:

- 1 mês planejando
- 1 mês especificando
- 1 mês fazendo a análise
- 6 meses desenvolvendo
- 1 mês implantando
- E, os últimos 2 meses, testando e documentando



Desta forma, o cliente terá algo funcionando apenas daqui a 12 meses.

Em projetos Scrum funciona da seguinte forma:

- No primeiro mês é feito o planejamento, análise, desenvolvimento, implantação e teste apenas de alguns requisitos. Como resultado, o cliente já receberá algo funcionando;
- No segundo mês, é feito o planejamento, análise, desenvolvimento, implantação e teste apenas de mais alguns requisitos. Consequentemente, o cliente terá mais um pedaço da solução funcionando;
- No terceiro mês, repetimos exatamente a mesma coisa, até completarmos os doze meses previstos inicialmente.

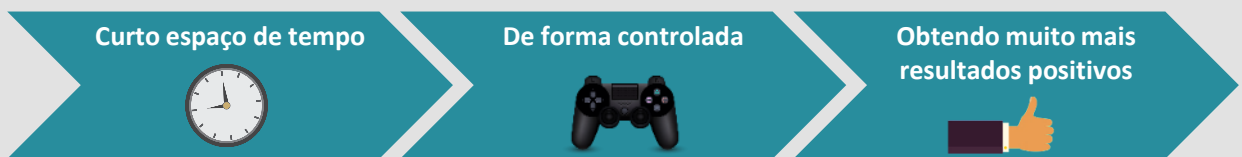
Em cada um destes ciclos, o cliente já recebe algo funcionando, pois a equipe de projetos especifica, desenha, desenvolve e já testa rapidamente. Diante disso, o cliente pode avaliar constantemente se é isso que ele quer, se está gostando ou não, enquanto o projeto avança, ciclo após ciclo.



Diferencial do Scrum

Para obter o resultado de um projeto de software através da metodologia tradicional, ou seja, um software rodando e funcionando, devemos aguardar todo o processo formal, começando pela primeira fase do ciclo, ou seja, o planejamento, depois a análise, o desenvolvimento e o teste. Porém, este processo pode ser muito longo e corremos o risco também de termos um produto diferente do que estava descrito no escopo inicial.

No Scrum, este processo acontece em...



Ele faz isso passando por todas as quatro fases em uma série de ciclos curtos menores, em pedaços pequenos e iterativos, ou seja, repetitivos. Desta forma, na metodologia ágil, quebramos um projeto grande em pedaços menores e passamos várias vezes pelo mesmo ciclo de desenvolvimento.

Uma tremenda vantagem no Scrum é que são feitas priorizações dos itens no início de cada ciclo, e como resultado, será entregue aquilo que é importante, prioritário e de missão crítica! Ou seja, é possível entregar, logo nas primeiras semanas, algo no qual o cliente já pode trabalhar.

Outro fator que nos impede de trabalhar sem desperdício é o modelo Trabalho em Andamento ou Progresso em Andamento (*WIP – Work in Progress*) utilizado na metodologia tradicional. Por exemplo, imagine três recursos que precisam entregar três itens de um projeto, e cada um é responsável por cada entrega. Se as três entregas forem concluídas daqui apenas seis meses, muito provavelmente o cliente não ficará satisfeito. Mas, se ao invés disso, um ou dois recursos ajudarem o primeiro recurso a entregar completamente sua entrega em apenas um mês, o cliente poderá experimentar a solução, sem desperdício ou perda de produtividade. Se ele gostar, vai para o próximo ciclo; caso contrário, cancela o projeto sem muitas perdas.

Princípios do Scrum

Em meados de 1990, Schwaber e Sutherland se juntaram para manifestar os princípios do desenvolvimento ágil de software, sendo considerados posteriormente como os criadores do Scrum. Resumidamente, estes doze princípios são:



“Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de Valor”.



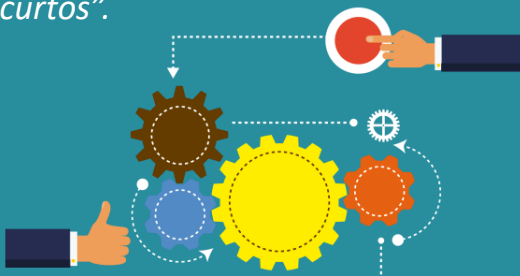
“Aceitar mudanças e requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas”.



Princípios do Scrum



“Entregar software funcionando com frequência, na escala de semanas até meses, com preferência para os períodos mais curtos”.



“Pessoas relacionadas a negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto”.



“Construir projetos ao redor de indivíduos motivados, dando a eles o ambiente e suporte necessários e confiando que farão o trabalho”.



O método mais eficiente e eficaz de transmitir informações para, e por dentro de, um time de desenvolvimento é através da conversa cara a cara.



Princípios do Scrum



“Software funcional é a medida primária de progresso”.



“Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter indefinidamente passos constantes”.



Contínua atenção à excelência técnica e ao bom design aumenta a agilidade.



Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feita.



Princípios do Scrum



“Os melhores requisitos, designs e arquiteturas emergem de times auto organizáveis”.



“Em intervalos regulares, o time reflete em como ficar mais efetivo e então ajusta e otimiza seu comportamento de acordo”.



Projetos que Atrasam



Um dos motivos dos projetos atrasarem é porque normalmente a equipe acaba assumindo mais do que um projeto ao mesmo tempo, e assim, acabam não focando em nenhum projeto específico. Acabam se ocupando com problemas técnicos do dia a dia, com os projetos já em produção, resolvendo incidentes, etc.

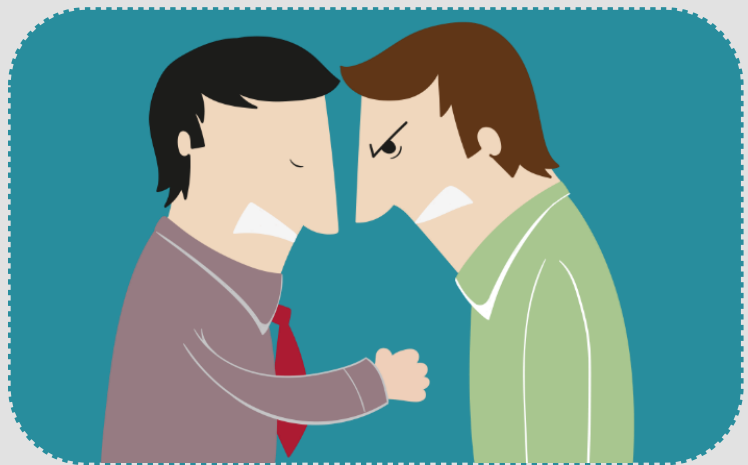
É fácil ouvirmos dos recursos, em reuniões de status de projetos tradicionais, que uma atividade está quase concluída, 99% pronta, praticamente acabada e assim por diante. Além dessas reuniões serem improdutivas e demorarem muito, ajudam ainda mais no atraso dos projetos. As atividades acabam sendo atribuídas isoladamente, e com isso, não se estimula a colaboração, como acontecem em projetos Scrum, onde um recurso pode assumir atividades de outros colaboradores tranquilamente. Ao trabalharmos isoladamente, não haverá motivos para concluirmos uma atividade antecipadamente, ao contrário de projetos ágeis, que o foco está na entrega ao cliente, todos unidos para o mesmo objetivo. Em uma semana de projeto pode acontecer muita coisa, algo pode sair dos trilhos e causar grandes atrasos. Porém, se fizermos reuniões diárias, este risco é minimizado.

Os projetos, que de alguma maneira já estão atrasados, tendem a sacrificar as últimas fases principalmente as fases de teste e documentação. As últimas fases sempre são penalizadas quando se trabalha com projetos tradicionais. Não resta mais tempo para concluir o projeto, então a entrega ao cliente vem com muitos problemas e defeitos, seja porque não testaram ou porque se apressaram demais, testando superficialmente.

Projetos que Atrasam

Com o Scrum, as atividades tem a duração de apenas um dia, então, diariamente o projeto progride, nos dando visibilidade do seu andamento. Não se espera uma semana para obter o retrato real do progresso do projeto. Isso significa que temos a chance de sabermos se vamos ter problemas amanhã e não só na semana que vem.

Outro fator comum que ajuda a atrasar os projetos é a altura da voz de quem grita. Quantas vezes vimos clientes, gestores das áreas de negócios ou até mesmo os líderes e gerentes de projetos, pressionados pelos clientes, terem uma voz mais ativa do que outros; em outras palavras, nesses casos, quem grita mais alto tem maior prioridade. Isso é um erro e não é assim que funciona no Scrum, pois o cliente não só participa das prioridades, como toda a equipe, gestores e partes interessadas. O Scrum tenta garantir este modelo ao máximo (porém, infelizmente sabemos que estamos lidando com seres humanos).



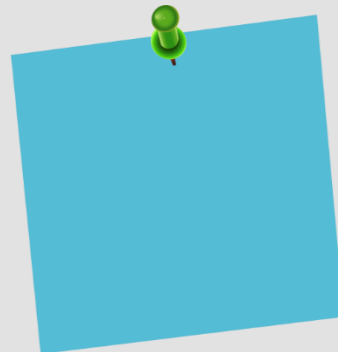
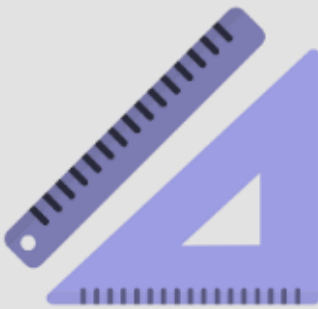
Existem atividades que podem ser concluídas em poucas horas. Em projetos tradicionais é dado um período até razoavelmente grande, por exemplo, uma atividade de 3 horas que deve ser completada dentro de um mês. Naturalmente, protelamos estas tarefas e deixamos para a última hora em virtude de outras prioridades. E como resultado, corremos o risco de atrasar, principalmente se esta atividade fizer parte do caminho críticos do projeto. O Scrum minimiza este problema com as reuniões e execuções diárias, ou seja, todos os dias os membros do time devem entregar algo, completar suas tarefas e relatar diariamente à equipe do seu progresso.

Projetos que Atrasam

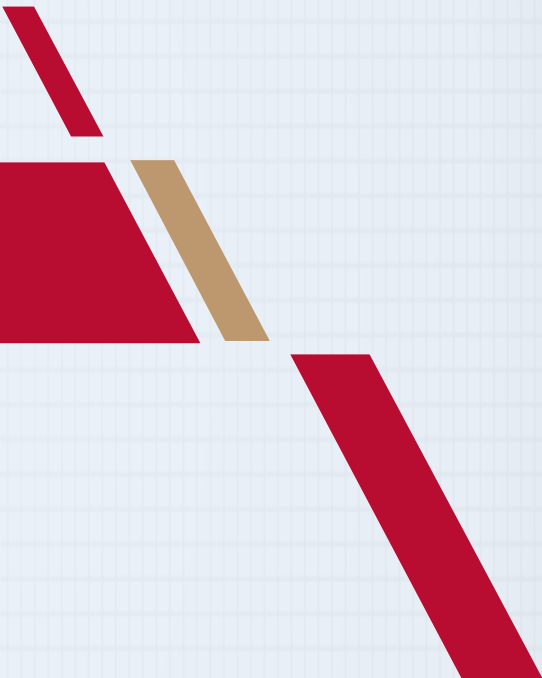
Rédeas curtas? Não diria isso, pois a equipe assume um compromisso de ser assim e ponto final! O resultado positivo convence neste modelo.



Soma-se a isso, o uso das métricas, não para medir o que as pessoas estão fazendo, mas sim, para direcioná-las. Considera o movimento dos post-its como uma ferramenta poderosa para demonstrar e direcionar a equipe. Outro fator interessante que leva o projeto ao atraso é que as pessoas não trabalham oito horas por dia.



Trabalhamos menos do que isso – normalmente seis horas – pois, no restante do dia estão lendo e-mails, conversando, indo ao banheiro, fazendo qualquer outra coisa, menos trabalhando. Se estimarmos uma tarefa de 40 horas que pode ser executada em uma semana, ou seja, oito horas por dia, é fato que ela vai atrasar. O ideal seria então em torno de sete dias para completar estas mesmas 40 horas.

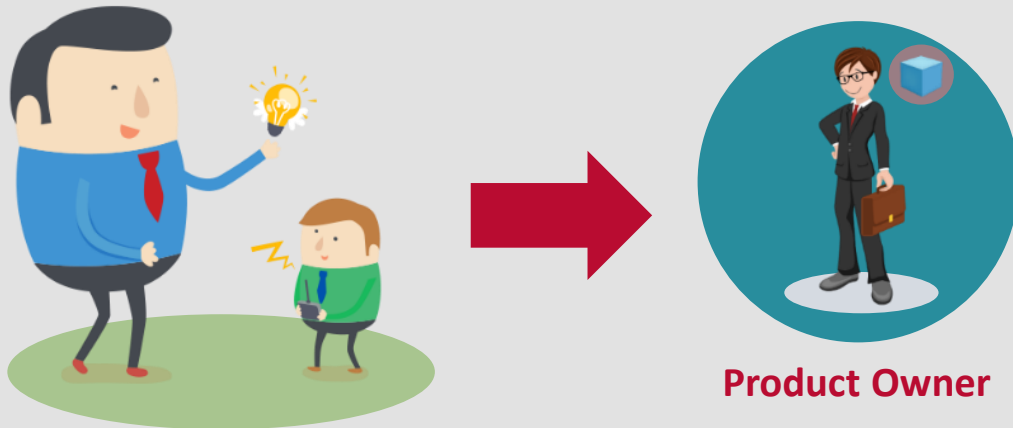


Surfando no Scrum

Conceitos Básicos

Na Prática

Antes de entrar nos conceitos do Scrum, é bom entendermos de uma forma geral como acontece a entrega de um projeto, da sua concepção ao teste e aceitação final do cliente. Normalmente, alguém tem uma ideia para uma solução ou para resolver um problema e é submetido para aprovação.



Após alguém aprovar essa ideia e considerar algo importante para a empresa, um responsável é alocado para conduzir essa frente, normalmente o Product Owner. Há então uma reunião de *kickoff*, ou seja, um pontapé inicial no projeto, para formalizá-lo, tradicionalmente acompanhado pelo patrocinador do projeto, o Scrum Master, a equipe de desenvolvimento ou outras partes interessadas. Neste ponto é descrito a visão do projeto, os riscos, premissas etc., ou seja, é definido o escopo de alto nível do projeto.

Com base no escopo, damos início ao projeto, começando pelas estimativas e detalhamento dos requisitos do projeto através de diversas técnicas do Scrum, priorizando as entregas, dando peso nos requisitos mais importantes e aprovando com os interessados. Podemos utilizar quadros brancos para planejar e demonstrar o andamento do projeto, e além disso, utilizar estas ferramentas diariamente para atualizar o andamento do projeto.

Até aqui nada muda de um projeto Scrum para o Tradicional, porém a forma que vamos conduzir as fases é que vai fazer toda diferença. Queremos desenvolver o projeto de forma e já entregar algo aos clientes, pelo menos no primeiro mês. Entregar algumas partes do todo, testado e funcionando.

Estrutura do Scrum

A Metodologia Scrum é bem estruturada, apesar de simples e extremamente funcional. O seu modelo contempla papéis, artefatos e cerimônias. Ela diz que devem existir apenas três funções, ou seja, três papéis que irão compor o Time Scrum:



Scrum Master



Product Owner



Desenvolvedores

Basicamente, existem três artefatos: o Backlog do Produto, que é um plano geral para o projeto, que contém uma lista ordenada dos itens – como os recursos – que podemos adicionar ao produto, chamamos de Meta de Produto; o Backlog da Sprint, um plano de curto prazo para o projeto que será implementado em uma única Sprint. Esse Backlog contém a Meta da Sprint, uma lista dos itens do Backlog do Produto selecionados para a Sprint, com uma série de atividades que criamos para decompor os itens; e por fim, o Incremento, que é a última versão de um produto criado para o cliente. Esse Incremento leva como compromisso a “Definição de Pronto”.

Existem quatro reuniões, conhecidas também como cerimônias que devem ser realizadas ao longo de um projeto Scrum, tais como:



Planejamento da Sprint



Reunião Diária

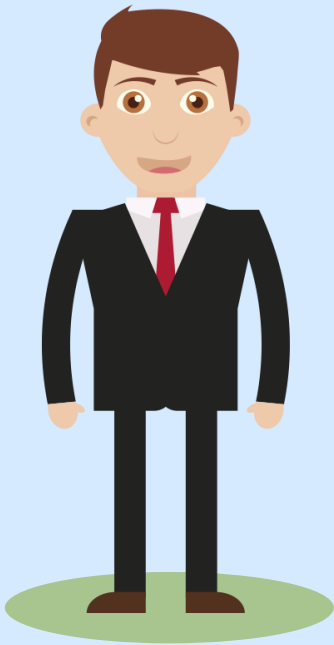
Estrutura do Scrum



Revisão



Retrospectiva
da Sprint



Antes de iniciar um projeto Scrum, é ideal que se desenvolva um documento de Meta do Produto, contendo as premissas, a prioridade deste projeto, as necessidades de negócio, as justificativas do projeto, e a Definição do “Pronto”, ou seja, devemos saber que algo está concluído, independentemente da perspectiva, seja dos clientes ou do Time Scrum.

Papéis do Scrum

Qualquer pessoa, independentemente da sua posição em uma estrutura hierárquica, pode assumir diferentes papéis no Scrum. Em projetos Scrum, ninguém está no comando como em projetos tradicionais através do Gerente de Projetos. As funções são destruídas através dos processos e de diferentes três tipos de funções.



Scrum Master



Product Owner



Desenvolvedores

O Scrum Master é uma função que segue uma linha semelhante a de um Gerente de Projetos em projetos tradicionais. No entanto, eles instruem os recursos a seguirem a metodologia, porém as semelhanças param por aí. O Scrum Master age como um facilitador e tenta ajudar a equipe de desenvolvimento produzir resultados desejados aos negócios demonstrando exatamente como é o Scrum, coordenando reuniões e construindo um time de alto desempenho, servindo como um agente na mudança cultural, e na aceitação das práticas do Scrum. Ele atua principalmente como um guia, um *coach*, um treinador.

O outro papel do Scrum é o Dono do Produto, ou Product Owner. Atua como o patrocinador, representando os interesses dos *stakeholders* (partes interessadas), aquele que ajuda a definir os requisitos iniciais, chamados de Backlog do Produto, suas responsabilidades também são de definir as prioridades, servindo como um porta-voz do cliente e responsável pelo retorno financeiro do projeto.

E por fim, a equipe que desenvolve o trabalho do projeto, os Desenvolvedores; são estes que colocam a mão na massa, são as pessoas de competências técnicas. São aquelas pessoas que transformam o Backlog do Produto em funcionalidades, realizando de fato a entrega do produto funcionando e testado.

Estes três papéis compõem o Time Scrum!

Definição do “Pronto”

Em projetos Scrum é extremamente importante definirmos alguns conceitos, como por exemplo, a definição do “pronto” como os critérios de sucesso de uma rodada no ciclo do Scrum. Esta rodada que no Scrum nós chamamos de Sprint.

Então, é importante definir antes, o que é considerado pronto em cada pequeno pedaço do projeto, em cada ciclo, pois aos olhos de um Time de Desenvolvimento, o produto está pronto quando finalizado; para a equipe de teste, quando testado; para o Product Owner, quando implantado e para o cliente, quando funcional, mas, mesmo que o pronto seja entendido, haverá nuances e detalhes que realmente deve ser negociado.

Definir o pronto evitará desgastes, o “disse que me disse”, ruídos e outros problemas. Deve haver uma negociação destes critérios e supõe-se que isso seja feito junto com o cliente, por intermédio do Product Owner. E óbvio, essa definição tem que acontecer no início do projeto.

Existe um segredo aqui, pois devemos definir, separadamente, os critérios do pronto e os critérios de aceitação. Os critérios de aceitação são estabelecidos através dos requisitos dos clientes, das quais chamamos de histórias e, os critérios do pronto, são aplicados a todas as histórias dos usuários em uma Sprint.



Macro Etapas do Scrum

Como sabemos, um projeto Scrum trabalha com ciclos curtos, ou seja, desenvolvemos, especificamos e testamos tudo em um único ciclo e em uma única Sprint. Em cada ciclo, ou cada início de uma Sprint, faremos uma reunião de Planejamento, onde toda equipe se reúne para dar uma estimativa de alto nível dos itens do Backlog.

Dimensionar em alto nível cada item do Backlog é simplesmente informar se ele é pequeno, médio ou grande, simples assim! Com o resultado, podemos agora descobrir o tempo que levará para ser concluído cada item, pois ao final, o objetivo já é entregar algo em que o cliente possa utilizar.

No dia a dia do Scrum, começamos com uma reunião diária do Scrum, entre 10 e 15 minutos para avaliarmos três questões importantes:



O importante é que a equipe relate estas questões à própria equipe, preferencialmente de pé.

E claro, depois de desenvolvermos este pedaço do projeto, teremos um tempo para fazer a revisão do que entregamos e adequar se necessário. Podemos fazer estas adequações novamente, aproveitando as janelas de liberações definidas pela empresa.

Essa revisão da Sprint é muito importante no ciclo do Scrum, pois será neste momento que haverá a validação das funcionalidades construídas para liberação de um pacote ao cliente. Fazendo isso, podemos garantir que a entrega está atendendo as expectativas iniciais dos clientes. Isso é feito quando há a revisão de tudo que foi construído, verificando se tudo foi realizado conforme o esperado pelos requisitos descritos nas histórias. Além da revisão, há também a reunião de retrospectiva, que tem o objetivo de avaliar os seus processos, ou seja, o que funcionou ou não em termos do Scrum.

Requisitos do Projeto

Os requisitos dos produtos que os clientes especificam são chamados de Backlog, ou seja, uma lista de características, tecnologia e funções que deve compor a entrega de um produto. Em algumas literaturas encontramos uma divisão interessante do Backlog, que sugere a quebra deste Backlog em dois níveis, para facilitar o desenvolvimento e o entendimento, tal como, primeiro, o Backlog do Produto que representa todos os requisitos de um projeto.

Um Backlog do produto nunca estará completo, e não pode ser comparado com documentos que contém requisitos ou especificações do produto, já que ele está sempre mudando e estará sujeito à expansão, graças às mudanças da visão que teremos do produto.

E depois temos o Backlog da Sprint, a menor divisão de um projeto, ou seja, é a parte do Backlog do Produto. Lembra-se que um projeto é dividido por Sprints menores? Então, normalmente o tamanho de uma Sprint é de um mês e a cada Sprint concluída adiciona-se um incremento ao produto.

Os desenvolvedores são responsáveis pelo Backlog da Sprint, então somente eles podem fazer algum tipo de alteração nesse Backlog. Entretanto, isso não quer dizer que ninguém mais está envolvido. A Meta da Sprint, por exemplo, é criada por todo o Time Scrum, e enquanto os desenvolvedores decidem sobre o número de itens no Backlog da Sprint, o PO decide a prioridade e consequentemente o que ficará pronto na Sprint.



Histórias do Usuário

O Backlog é escrito em épicos e dividido em histórias. É mais fácil escrever os requisitos desta maneira, e por isso que cada Sprint começa com uma Reunião de Planejamento, durante o qual as Histórias de Usuários são consideradas.

As histórias são escritas geralmente pelo Product Owner, através do ponto de vista do usuário final. Uma história é curta e clara, e narrada pelo usuário demonstrando o que ele realmente deseja. Com base nestas histórias é possível então criar os critérios de aceitação. Ou seja, se a descrição da História do Usuário diz que:



Com este exemplo, é fácil relacionarmos pelo menos três critérios, ou seja, as histórias só estarão satisfeitas se atendermos a estes critérios de aceitação: Disponibilizar uma relação de produtos para o cliente; Avaliar a situação financeira do cliente; Mostrar na tela do sistema o método de pagamento disponibilizado conforme situação do cliente.

O importante não é definir todas as histórias e muito menos os épicos de uma só vez e no início do projeto. Como o Scrum é iterativo, faz-se então a definição das Histórias dos usuários assim que se avança em cada Sprint, e por isso é importante focar na história atual e na próxima Sprint, e não em todas. Isso gera outro ganho, pois fica mais fácil priorizar as histórias e definir o que realmente é importante.

Histórias do Usuário

Para definir o que é importante para o cliente, pode ser utilizado uma ferramenta que ajude a definir o que deve ter, o que deveria ter, o que poderia ter e o que não terá. Como resultado, será necessário separar o que deve conter em suas histórias.

A história de usuário não deve ser confundida com os Casos de Usos. Assim como os Casos de Usos, a história de usuário também é utilizada para estimar um projeto, porém não é tão detalhada, sendo criada em reuniões de planejamento com a finalidade de estimar o tempo para a entrega de uma versão, ou seja, com baixo nível de detalhes. Um Caso de Uso conduz ao desenvolvimento da solução e a história do usuário conduz a aceitação do cliente, ou seja, a partir do que foi escrito, isso servirá como critério de aceitação do produto.

Estimativas

Existem diferentes técnicas numéricas para estimar os itens do Backlog. Tais como para estimar as histórias, usamos o que chamamos de pontos de histórias, ou seja, o peso de cada estória, o tamanho total da história de um usuário, considerando o risco, a quantidade de esforço exigido e o nível de complexidade. Geralmente é expressa nas mesmas unidades utilizadas para a estimativa da velocidade da Sprint. Não é uma técnica exata, porém é utilizada como modelo comparativo.



HISTÓRIA	IMPORTÂNCIA
História 35	Valor de importância 50 (mais importante/prioritária)
História 4	Valor de importância 30
História 89	Valor de importância 20
História 13	Valor de importância 10 (menos importante/prioritária)

Usar a comparação para estimar, é partir do pressuposto que uma determinada história tem, por exemplo, 20 pontos, que histórias menores tem 10, e as maiores têm 30.

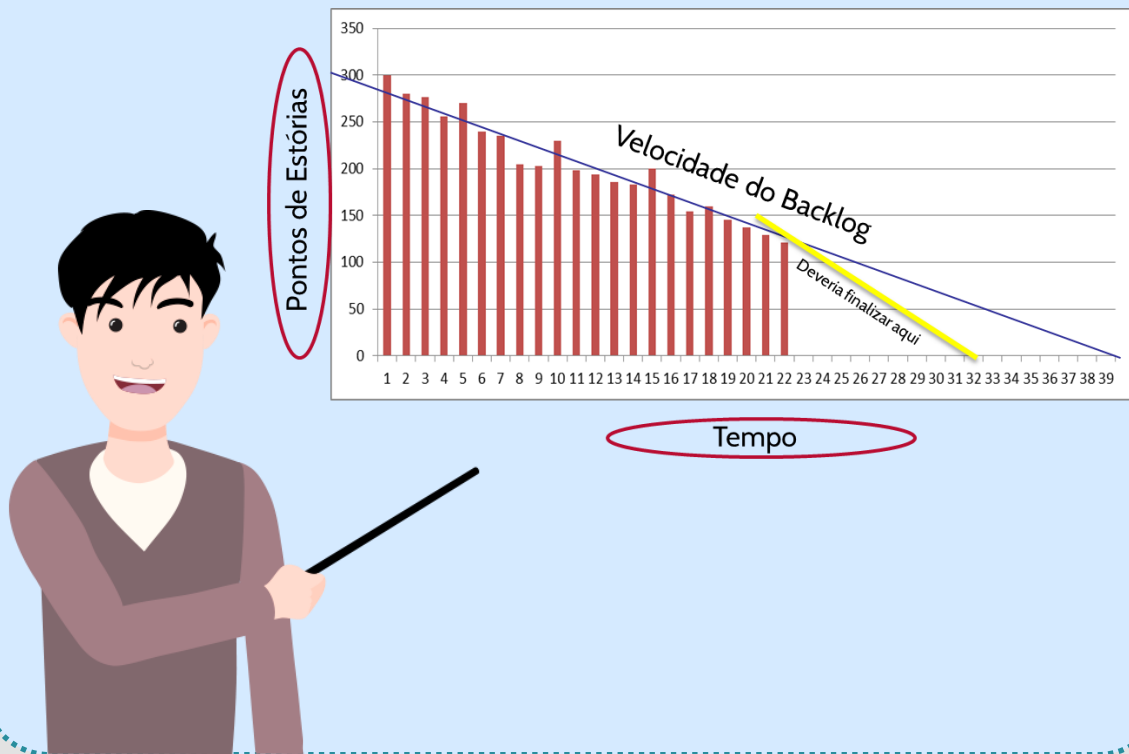
Algumas vezes são utilizadas em conjunto técnicas como a Sequência Fibonacci e o Planning Poker.

O Planning Poker é um jogo de baralho que ajuda a definir os pesos das histórias, onde cada participante determina um Peso para a história e ao final há um consenso dos resultados. O importante é que o tamanho tenha base nos esforços e não no tempo da conclusão de uma história.

0 1/2 1 2

A partir do momento que um membro do time estima o ponto da estória, com bases comparativas, principalmente de algo que já foi feito no passado, este membro então expõe sua opinião e faz-se a votação porque foi escolhido aquela pontuação, e caso não cheguem a um consenso, faz uma nova rodada do jogo até conseguirem chegar em um denominador comum.

Gráfico de Burndown



Um gráfico de Burndown é construído partindo do pressuposto que as Sprints duram 30 dias. Na vertical são representados os Pontos de História dos Usuários e na horizontal o tempo. Ele é formado em barras, sendo que cada barra representa um dia. Assim que o gráfico avança no tempo, as barras no gráfico diminuem. Por exemplo: imagine que em um certo dia tenhamos 300 pontos de história, e no dia seguinte 288, pois concluímos 12 pontos de história no dia anterior. No terceiro dia, temos 276, pois concluímos mais 12 pontos de história no dia anterior e assim por diante. Desta maneira, descemos no gráfico até zerarmos, demonstrando o que falta ser entregue, até o momento que ele chegue a zero ponto no ultimo dia da Sprint.

Se traçarmos uma linha reta na diagonal, do ponto mais alto – da quantidade total dos pontos de história de usuários, normalmente à esquerda e acima - até o ponto mais baixo do gráfico, à direita – no último dia da Sprint, poderemos observar a tendência das entregas, ou seja, a velocidade do time para concluir as histórias.

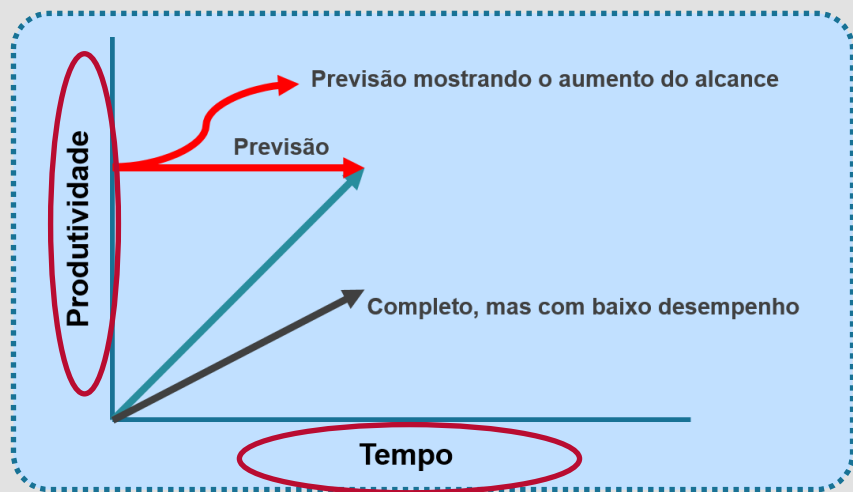
Gráfico de Burndown

Naturalmente, assim que o tempo avança, a equipe pode se deparar com mudanças e descobrir que haverá menos tempo do que o previsto para concluir certas histórias, por isso não é incomum termos picos, acima desta linha diagonal, demonstrando que tivemos mais esforço do que o previsto inicialmente.

Com as barras acima desta linha reta diagonal, podemos observar então o quanto estamos deixando de desenvolver e o quanto de histórias estão sendo acumuladas, principalmente se fizermos esta linha diagonal com base no topo de cada barra e o final da diagonal finalizar depois dos 30 dias.

O correto, quando há um atraso, não é aumentar o prazo de conclusão; por isso, normalmente a solução para minimizar o atraso da Sprint inteira é retirar as histórias desta Sprint atual e inseri-las no Backlog do Produto para serem concluídas nas próximas Sprints.

Gráfico de Burnup



Este gráfico demonstra a produtividade, já que ao contrário do Burndown, que à medida que o gráfico de barras vai diminuindo, é sinal que estamos evoluindo, no Burnup o indicador de andamento é para cima. O gráfico de Burndown nos dá uma visibilidade apenas de uma Sprint de 30 dias, e o gráfico de Burnup nos fornece informações, não só da produtividade, mas também dos pontos de histórias que estão sendo produzidos, até chegamos ao total previsto de histórias.

Neste gráfico é colocada na vertical a quantidade total de histórias, ou seja, a definição da sua produtividade, e é informado na horizontal o tempo previsto total do projeto. Podemos então traçar uma interseção entre os dois pontos e acompanharmos o andamento do projeto. Com isso, será possível visualizarmos antecipadamente se conseguiremos atingir os resultados propostos.

Se no andamento das entregas das histórias percebermos que o projeto vai atrasar ou teremos histórias sem serem concluídas, não podemos simplesmente reduzir as histórias e deixarmos para a próxima Sprint, pois aqui estamos visualizando o projeto como um todo, incluindo todas as Sprints, embora não demonstrada no gráfico.

Este gráfico sinaliza também a quantidade de características e se as histórias foram adicionadas pelo Product Owner de acordo com a linha de base inicial. Neste gráfico, fica evidente o que é mudança de escopo.

Processos Empíricos

Diferentemente da metodologia ágil, no Scrum não são utilizados gráficos de Gantt para mensurar o trabalho. Isso é feito com outras técnicas e ferramentas mais eficientes. No Scrum, não há Gerentes de Projetos e nem atribuições dos recursos, ao contrário, as próprias pessoas trabalham em colaboração uns com os outros, onde cada integrante nesta estrutura auto-organizada sabe tomar decisões do que fazer primeiro e decidir pelo que é mais importante.

O ciclo se repete muitas vezes, dando a oportunidade da equipe crescer e ficar cada vez mais madura, assegurando que o processo repetível se transforme em um hábito. No Scrum, vemos os processos de forma empírica, ou seja, são processos vividos, experimentados e, principalmente, não são utilizados processos definidos e rígidos.

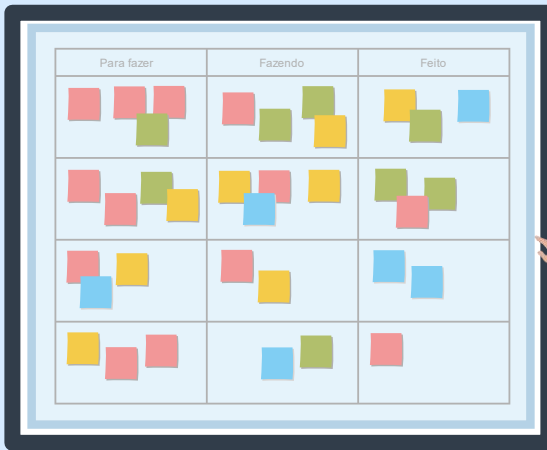
Quando falamos de processos empíricos, estamos mencionando três fatores-chave: a transparência, inspeção e adaptação. A transparência trata do aspecto da visibilidade, ou seja, a primeira etapa de um processo empírico é ser capaz de demonstrar e compartilhar um mesmo entendimento do que está sendo visto por todos os observadores.

A inspeção em um processo empírico vai minimizar a chance de algo passar despercebido e aumentar as chances de descobrir algo que esteja errado.

Se algo é descoberto, então o terceiro aspecto aparece, a adaptação. Em outras palavras, isso exige que observemos dois itens: um direciona para os resultados dos produtos ou serviços, e o outro para a forma como o trabalho é feito. A ideia aqui é verificar se os resultados são o que queremos e se precisamos mudar ou adaptar algo.



Kanban



Os quadros Kanban são super estabelecidos no ramo da manufatura, e tem sido muito usados em projetos de TI, seja de forma autônoma, ou como técnica adicional ao Scrum.

Quando ele atua sozinho, isso quer dizer que as iterações não possuem um prazo limitado como no Ágil. Na realidade, ele utiliza um fluxo contínuo onde os itens de um Backlog são sempre selecionados sempre que houver alguma capacidade livre.

Porém, mesmo com a ausência de iterações, o Kanban ainda tem desenvolvimento iterativo e entrega incremental, e quando bem aplicado, consegue ser um sistema adaptativo bastante eficiente. O Kanban é composto por três atributos: Visualização do trabalho; Limitação do trabalho em andamento (WiP); Empurrar o trabalho, em vez de puxar.

O Kanban é feito para visualizar os estágios de um trabalho e seus itens. Esses estágios variam com as preferências da equipe e o tipo de trabalho que é realizado. O quadro é composto por uma coluna dos itens “Para Fazer”, de uma ou mais colunas das atividades envolvidas no processo e uma última coluna, à direita, nomeada como “Feito”.

Reuniões Diárias

O segredo das reuniões diárias pode parecer estranho, mas elas devem ser conduzidas com todos de pé na sala. Ela não pode se estender mais do que 15 minutos, preferencialmente em frente do quadro de atividades. Caso seja detectado algum problema que precisa ser resolvido, deve ser posto fora desta reunião, pois o foco aqui é demonstrar o progresso do projeto em relação ao dia anterior, o que será feito hoje e o que está impedindo das atividades serem realizadas.



Não se escreve nada nesta reunião; o máximo que acontece é o movimento do post-it no quadro de atividades, pois a ideia aqui é garantir a integração, independentemente se todos conseguiram chegar no horário da reunião, que deve sempre começar no mesmo horário, todos os dias, preferencialmente nas primeiras horas do expediente.

Ainda para garantir a integração, os membros devem se reportar e conversar com os outros membros do time, deixando de lado o Scrum Master, pois a equipe deve ser auto-organizada e capaz de tratar destes assuntos entre si. Jamais usem celulares, sentem-se ou façam conversas paralelas. Isso é muito importante!

Todos os direitos reservados. Proibido reproduzir esse material, sem prévia autorização da
PMG Academy

www.pmgacademy.com

