

EBOOK - AULA EXTRA

GERENCIAMENTO DE PROJETOS

COM SCRUM



GERENCIAMENTO DE PROJETOS COM

SCRUM

APLICANDO AGILIDADE EM IDEIAS E DEMANDAS

LUIZ FERNANDO NUNES



Sumário

INTRODUÇÃO	4
ORIGENS.....	6
AMBIENTES.....	14
FRAMEWORK SCRUM.....	17
IMERSÃO EM SCRUM	27
USER STORIES.....	36
VISÃO DO NEGÓCIO	39
PRODUCT BACKLOG OU ESCOPO?	43
DEFINIÇÃO DE PRONTO.....	47
SPRINT PLANNING	48
ESTIMATIVAS DO PRODUCT BACKLOG	52
SPRINTS.....	55
SPRINT REVIEW, INSPEÇÃO E ADAPTAÇÃO DO PRODUTO	57
LIDERANÇA ÁGIL.....	59
REFERÊNCIAS BIBLIOGRÁFICAS	65

INTRODUÇÃO

Quando trabalhamos com mentalidade Lean dizemos que as atividades que não geram valor para o negócio ou projeto são eliminadas. O Lean foi originado na sua maior proporção na Toyota, “Manufatura Enxuta” e é composta pelos seguintes princípios: Identificar o valor, mapear o fluxo de valor, criar o fluxo de valor contínuo, produção puxada e melhoria contínua.

O Lean é centrado no cliente e podemos entender que essa relação é baseada em consistência e durabilidade da sua força. Quanto maior o laço com o cliente mais sustentável será o negócio ou projeto.

Ao se falar de ágil ou agilidade entendemos que ele é composto por uma série de valores que tiveram sua origem no manifesto ágil. Tal manifesto teve origem no desenvolvimento de software e é uma contraposição ao desenvolvimento tradicional ou como chamamos de cascata. A agilidade enquanto mentalidade é baseada em pessoas e sustenta seus relacionamentos e possui ciclos curtos de entrega, priorizando a qualidade e a transparência do que está sendo desenvolvido.

Lean e ágil são totalmente conectados, pois promovem uma mentalidade diferenciada na organização, fundamentando uma cultura necessária e possibilitam entregas que tem relação direta com valor. Juntas possibilitam flexibilidade e adaptabilidade a qualquer segmento ou empresa.

Quando exploramos mais afundo a definição de agilidade podemos entender uma capacidade de adaptação a mudanças constantes em um mundo onde chamamos de VUCA.

Quando estamos estudando tendências, comportamento do consumidor, mudanças que ocorrem nas mudanças do que chamo de “Business Mindset”, processo esse que é a capacidade do negócio em mudar e adaptar seus posicionamentos e fatores ambientais e culturais da organização em busca de fazer parte desse novo mundo baseado num pensamento complexo.

É inevitável nesse modelo encontrarmos erros pelo caminho, porém vivemos um conceito muito forte de gestão empírica, que é a aquela onde aprendemos fazendo, seja no erro, no acerto, porém é produzido aprendizagem, variabilidade

e principalmente a adaptação a todo processo de inovação diária ao qual estamos situados.

Vale nesse momento lembrarmos do acrônimo V.U.C.A (volátil, incerto, complexo e ambíguo), que hoje em dia é muito falado em contexto de agilidade, mas foi um termo introduzido pelo Exército Norte-Americano, mais precisamente United States Army War College.

As empresas que implementam ou mudam a forma de pensar para um modelo de agilidade buscam entender o comportamento dos seus clientes, prever novas tendências, buscar uma simplificação do trabalho que é realidade no presente, buscam a capacidade de se adaptar a mudanças e se posicionam de maneira mais assertiva no mundo vuca, com a geração de dados e compreensão e utilização para tomadas de decisão.

Uma empresa ágil possibilita as pessoas uma capacidade de aceleração e possuir papéis como Agile Coach é importante para possibilitar manter firme o propósito fazendo com que a empresa busque sua sobrevivência, crescimento e potencial inovador.

Estudando Scrum, Kanban e diversos outros métodos é trazer para os times e organizações novas formas e direcionamentos em busca de maximizar valor do negócio.

Sejam bem vindos a nossa viagem!

ORIGENS

Diferentes tipos de projetos possuem necessidades e características de diferentes métodos. Quando iniciamos um projeto existe uma necessidade implícita de descoberta de que formas e ferramentas iremos necessitar para atingir o seu objetivo e atender a necessidade de valor necessária pelo cliente.

Por um lado, nos dias atuais temos defensores desde a simplificação máxima na gestão de tais demandas, como também existem profissionais que inserem documentações cansativas, processos extremamente burocráticos que não agregam valor para o produto, mas apenas servem para inserir barreiras entre o desejo e a entrega final.

Estamos hoje cercados por práticas, métodos, frameworks e novas tendências que surgem a cada dia, isso traz consequências graves para o negócio, pois gera confusão e acaba proporcionando aos times falta de foco e definição de um método claro e específico de trabalho.

ESTAMOS INCONFORMADOS, VAMOS DISCUTIR?

Todos possuímos nossos momentos de grande revolta, de questionar o mundo e todos os que nos rodeiam, porém quero contar uma história bem legal de um grupo de pessoas que estavam inconformados com a maneira de desenvolver softwares. Mas leia a história pensando além do “software”, pense em qualquer situação, produto, serviço ou área que esteja inserido.

Quando ficamos inconformados com o “status quo”, é normal que vamos buscar alguma maneira de transformar e mudar as coisas de lugar. Mas também é normal que sozinhos somos presa fácil para a procrastinação e o desânimo. Então quero que leia essa história verdadeira e entenda que se ficarmos sentados vendo o mundo passar, ele vai passar.

Em fevereiro de 2001 um grupo de profissionais da área de desenvolvimento de software se reuniram nas montanhas geladas do estado de UTAH (Estados Unidos) para discutir melhores formas de desenvolvimento de software.

Segue a lista de presença:

- Kent Beck
- Mike Beedle
- Arie Van Bennekum
- Alistair Cockburn
- Ward Cunningham
- Martin Fowler
- James Grenning
- Jim Highsmith
- Andrew Hunt
- Ron Jeffries
- John Kern
- Brian Marick
- Bob Martin
- Stephen Mellor
- Jeff Sutherland
- Ken Schwaber
- Dave Thomas

Não seria necessário talvez mencionar todos esses nomes, porém é fundamental que em nossa vida, seja profissional ou pessoal busquemos referências, então aí estão dezessete nomes que fizeram a diferença.

Dessa reunião através de discussões onde cada um trazia sua realidade, experiências e proposições de melhorias surgiu um documento onde eles descreveram suas crenças e valores para o desenvolvimento de software, que mais tarde fora chamado de MANIFESTO ÁGIL.

O manifesto ágil é composto de:

- Quatro valores ágeis;
- Doze princípios ágeis.

Quando li e estudei o manifesto ágil pela primeira vez uma frase foi muito marcante, que é a seguinte:

“Estamos descobrindo maneiras melhores de desenvolver softwares fazendo-o nós mesmos e ajudando outros a fazê-lo.” (Manifesto Ágil, 2001).

Ajudar, que palavra magnífica para inserir no início do manifesto. Quando eles se reuniram naquele chalé com certeza aconteceram momentos marcantes de empatia, onde o resultado seria sempre coletivo, jamais individual. Outro ponto fundamental é que as grandes ideias e mudanças partem do localmente para o globalmente, quando analisaram as dores que possuíam ao utilizar métodos cascata no desenvolvimento de software eles trouxeram a dor de diversas empresas e desenvolvedores. E quando propuseram melhorias, elas foram muito bem aceitas, pois ajudar os outros a fazer é parte fundamental de qualquer mudança de estado.

Continuando nossa história, o manifesto ágil possui quatro valores que são:

Indivíduos e Interações Mais que Processos e Ferramentas

Certa vez uma pessoa veio me questionar qual era o item mais importante para se aprofundar e trabalhar com gerenciamento de projetos. Pensei alguns minutos e sem pestanejar respondi: “são as pessoas, elas são o ativo mais importante de qualquer negócio”. Após dar essa resposta puder ver uma certa decepção no rosto da pessoa, ela esperava técnicas, processos, ferramentas que agregassem em sua caminhada em projetos, porém quando disse pessoas ela ficou sem entender e para que você também não fique a ver navios vamos entender o porquê das pessoas.

Processos nas organizações são extremamente importantes, em nossa vida executamos processos desde que quando acordamos, se pensarmos bem. Porém temos que entender que as pessoas são o centro de tudo, dos processos, dos produtos, pois são elas que os desenvolvem e são peça fundamental na engrenagem de sucesso de um projeto. Um ponto importante é entender que os processos bem modelados e desenhados auxiliam as pessoas em como entregar resultados que satisfaçam os clientes, sem as pessoas esses processos não entregam valor.

Software em Funcionamento mais que Documentação Abrangente

Quando compramos um produto, seja um eletrodoméstico ou computador, antes de utilizar sempre lemos o manual, que é importante para o bom uso do produto. Desta mesma forma as documentações são importantes desde que representem o que foi gerado de valor e entregue ao cliente. Não faça documentações exageradas e sem necessidade, isso é desperdício de recursos e tempo que poderia ser gasto em outros incrementos do projeto.

Colaboração com o Cliente mais que Negociação de Contratos

Uma forma importante de tratar o cliente é mostrar que ele faz parte do time, ele colabora para produzir o software/produto com o máximo de valor possível. Quando se desenvolve uma necessidade do cliente é fundamental entender que a necessidade é dele, aí vem novamente aquele processo básico chamada “empatia”, se colocar no lugar do cliente é fazer entregas que satisfaçam as suas expectativas.

Isso não significa que ele sempre tem razão, significa que juntos chegaremos ao melhor resultado.

Responder a Mudanças mais que Seguir um Plano

Todo negócio que ficar preso em fundamentações ou planos com certeza estará colocando algemas no seu próprio crescimento. No desenvolvimento de software vivemos um ambiente de incerteza, onde é necessário rapidez e adaptação a mudanças, e neste caso quando possuímos um planejamento que engesse o projeto isso gera insatisfação do cliente por não agregar valor. É necessário entender que este valor do manifesto ágil não elimina o planejamento, apenas faz que este seja flexível ao cenário atual em que o projeto esteja inserido.

Os *PRINCÍPIOS ÁGEIS* foram criados tendo como base os valores ágeis, eles são:

- 1) Nossa prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.
- 2) Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
- 3) Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
- 4) Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.
- 5) Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
- 6) O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
- 7) Software funcional é a medida primária de progresso.
- 8) Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
- 9) Contínua atenção à excelência técnica e bom design, aumenta a agilidade.
- 10) Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
- 11) As melhores arquiteturas, requisitos e designs emergem de times auto organizáveis.
- 12) Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

DESCOBRINDO A AGILIDADE

Quando utilizo o termo descobrindo a agilidade coloco em ordem três pontos fundamentais que devemos estudar:

- O trabalho de Takeuchi e Nonaka;
- O manifesto ágil;
- Jeff Sutherland e Ken Schawber e o Guia Scrum.

São os conhecimentos básicos necessários para se buscar um entendimento profundo dos princípios e valores necessários para poder em seguida trabalharmos com gestão ágil de projetos.

Takeuchi e Nonaka



Hirotaka e Ikujiro Nonaka

Hirotaka Takeuchi e Ikujiro Nonaka são conhecidos mundialmente por seus trabalhos sobre gestão do conhecimento, mais diretamente ao que diz respeito ao conhecimento tácito, ou seja, aquele que é adquirido ao longo da vida através das experiências que vivemos. Um momento curioso na carreira desses grandes pesquisadores é o fato que Ikujiro Nonaka foi contratado pelo governo japonês logo após a segunda guerra mundial para realizar análises dos motivos que levaram o Japão a derrota.

Mas direcionado aos métodos ágeis o ponto fundamental foi o artigo “The New Product Development Game”, publicado na revista Harvard Business Review em 1986, onde após pesquisarem inúmeras abordagens de diferentes equipes de desenvolvimento analisaram as causas que levavam a obter resultados positivos. Podemos citar que algumas empresas estudadas foram a 3M, Honda, Hewlett-Packard, e diante do estudo levantaram algumas características semelhantes:

É repassado ao time por suas lideranças, objetivos que iria propor um desafio, mas como consequência da ao time liberdade para atingir os objetivos;

As equipes possuíam autonomia e tinham pouca interferência das lideranças, proporcionando um nível de multidisciplinaridade e auto-organização nas equipes;

Era possível identificar os momentos críticos de cada demanda, podemos entender momentos críticos como gargalos e dessa forma o time poderia estabelecer um ritmo e velocidade própria, característica de auto-gestão;

Time preparado para responder de formas eficientes a mudanças que pudessem ocorrer;

Não ocorria uma instabilidade organizacional, pois existia um controle de forma quase que despercebida pela liderança do negócio;

Todo conhecimento era compartilhado com os demais setores do negócio, propagando a cada colaborador lições aprendidas nos projetos.

Uma abordagem que propõem desenvolver novos produtos com rapidez e flexibilidade.

Takeuchi e Nonaka realizam uma comparação dessas equipes de auto desempenho com jogada o rúgbi, onde temos que avaliar a formação da equipe.

Características da Formação:

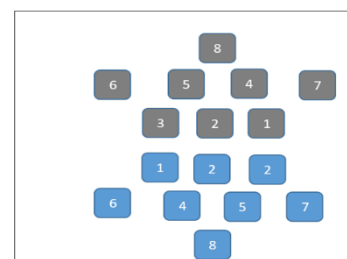
É a forma de iniciar o jogo, quando acontece uma inflação;

A finalidade do Scrum é que a equipe com a posse bola a mantenha;

Participam da estrutura oito pessoas de cada time;

Eles são divididos em três linhas, onde a primeira linha é composta pelos pilares e pelo Hooker, a segunda linha é composta por dois jogadores, a terceira linha é composta pelos asas e o oitavo;

A bola só pode ser pega pelo scrum-half;





Os times após a bola ser inserida no centro pelo scrum-half devem empurrar a equipe adversária para trás até que a bola esteja visível e disponível para que o scrum-half possa pega-la e a bola estar em jogo. Após o scrum-half pegar a bola o time deve se organizar para que consigam definir a estratégia para jogada e poder obter o ponto, chamado de try, quando a bola é colocada no chão do campo do time adversário.

Pontos que foram analisados por Takeuchi e Nonaka:

- Auto-organização do time;
- Todo time é importante para atingir o objetivo;
- Os membros trabalham juntos do início ao fim;

O trabalho de Takeuchi e Nonaka foram muito importantes para a criação do framework Scrum e todo desenvolvimento das metodologias ágeis.

FRAMEWORK SCRUM

Jeff Sutherland criou para utilizar na Easel Corporation em 1993 o Scrum. Sua utilização seria no desenvolvimento de software e para tal foram utilizadas as bases do estudo de Takeuchi e Nonaka (1986). Tanto Jeff Sutherland como Ken Schwaber desenvolveram estudos e publicações que foram fundamentais para popularizar o tema e difundir o framework no desenvolvimento de software. Podemos destacar como principal publicação “The Scrum Guide” (2011), onde constam todos pilares, eventos, papéis e artefatos do Framework Scrum.



Jeff Sutherland e Ken Schwaber (Fonte: scrum.org)

AMBIENTES

Para entender o contexto de produtos e ambientes complexos iremos analisar o framework Cynefin.

O framework Cynefin foi desenvolvido para auxiliar as organizações a determinar qual contexto operacional predominando para tomar as decisões corretas. Ele foi desenvolvido por Dave Snowden, que faz parte do Instituto de Gestão do Conhecimento da IBM no ano de 1999.

Partindo para o entendimento primeiro é importante compreender o ambiente ao qual estamos alocados, ou seja, onde vivemos. Assim como natureza o ambiente tem forte impacto na vida de cada ser vivo nos ambientes corporativos isso também é muito perceptível e segundo. O framework é dividido em cinco contextos que são definidos pela natureza da relação da causa efeito.



- Domínio Complexo** - existe um domínio da emergência, neste caso as relações entre causa e efeito. É necessária uma exploração para conseguir descobrir o problema. Como abordagem podemos inspecionar, entender e adaptar. Aqui estamos num ambiente onde o desconhecido é predominante. Neste ambiente trabalhar com Scrum é interessante, pois

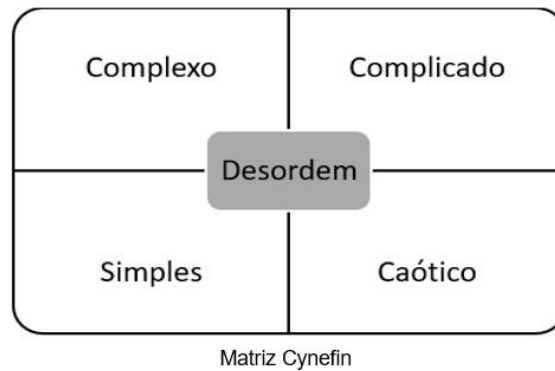
a habilidade de inspecionar e adaptar proporciona condições para operar nesse contexto.

- **Domínio Complicado** - quem domina neste contexto são os especialistas, onde a busca por resposta exige um estudo mais apurado e aprofundado, e nestes aspectos pode utilizar opinião especializada sobre determinados assuntos. Neste contexto o Scrum pode ser uma alternativa, mas o ideal é trabalhar com mais aspectos quantitativos e qualitativos podem ter maior eficiência. Nestes casos podem ser trabalhados com Six Sigma e boas práticas do Guia PMBOK®.
- **Domínio Caótico** – neste contexto podemos simplificar o entendimento dizendo que aqui é necessária uma habilidade de agir imediatamente e de forma decisiva. Neste domínio a abordagem é agir, sentir e responder. Neste contexto o Scrum não poderá ser utilizado, pois não existe tempo para definir e priorizar backlog e cumprir todos os eventos.
- **Domínio Simples** – neste contexto as causas e efeito são facilmente visíveis, a resposta certa é óbvia e sem nenhuma barreira para resolução. Neste domínio o Scrum pode ser utilizado, mas podemos dizer que poderá não ser tão eficiente, pois a capacidade de adaptar-se do Scrum pode dificultar sua utilização em contextos simples.
- **Desordem** – neste contexto imagine que está em desordem quando não tem como identificar em qual dos domínios anteriores se está localizado. Imagine um cenário totalmente inóspito, ninguém e nenhuma organização gosta ou quer estar neste contexto, é perigoso. Neste domínio não se aplica Scrum ou qualquer framework ou método, você tem que fugir ou sair da desordem o mais rápido possível.

Podemos dizer que vivemos num mundo repleto de complexidade. Nas empresas vivemos cada um dos domínios e contextos do framework Cynefin, por esta razão que sabemos a dificuldade de implementar o Scrum, precisamos ter noção e estabelecer mecanismos para identificar qual domínio se apresenta e tomar as decisões necessárias para sanar os problemas apresentados.

É importante entender o framework para aplicar metodologias ágeis em outros segmentos que não sejam desenvolvimento de software, pois exige uma série

de análises. Uma dica que pode ajudar a mapear o domínio existente na sua empresa é utilizar uma matriz de Cynefin (utilizando uma folha A1 com a imagem e post-its):



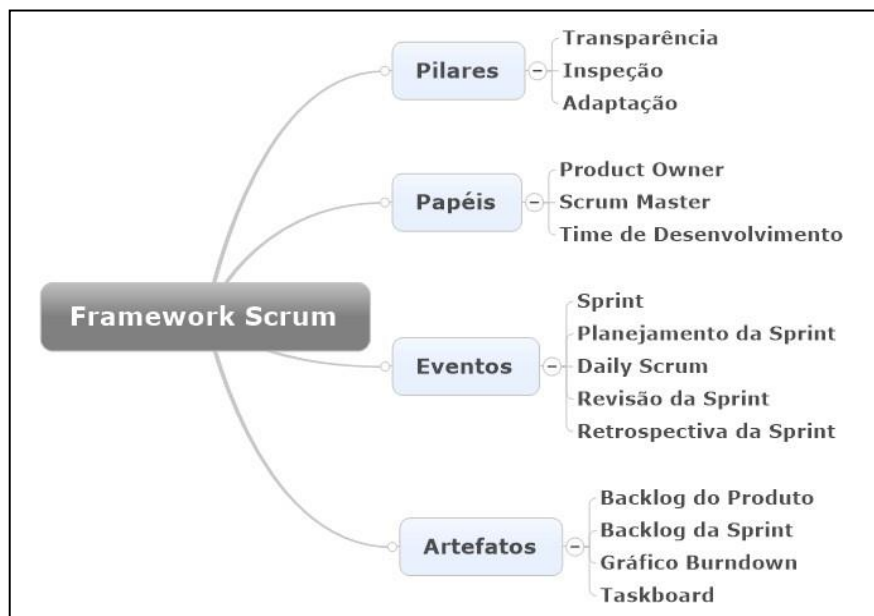
Agora que entendemos todas as possibilidades de domínio podemos continuar a contextualizar o Scrum, e Sutherland e Schawber (2017) salientam que o Scrum não se trata de um processo, técnica ou método definitivo. Ele é um framework ao qual podemos empregar diferentes processos ou técnicas, podemos dizer que é livre para se adaptar ao contexto que houver necessidade.

FRAMEWORK SCRUM

Podemos definir o Scrum como um framework para gerenciamento de projetos ágeis. Este mesmo framework poderá tratar e resolver problemas em níveis de complexidade alta e adaptativos. Não podemos confundir o Scrum como um método mais fácil ou menos burocrático de gerenciar projetos, ele não tem nenhum ponto que direcione a esse pensamento.

Um ponto fundamental do conceito ágil é centralizar o cliente no processo, ou seja, gerar valor ao cliente é fundamental para o sucesso de cada entrega e assim de todo backlog. Como framework é possível empregarmos inúmeros processos e técnicas, sempre buscando uma melhoria contínua tanto do produto e toda a equipe de trabalho.

Para entender melhor o Scrum é preciso compreender que esse framework possui Times associados a papéis, eventos, artefatos e regras, iremos estudar cada um destes itens, porém agora é importante entender o conceito do Scrum e sua importância.



Como falamos de processos o Scrum não é um processo padronizado onde se aplica toda uma sequência metódica de passos para garantir que seja entregue um produto ou serviço dentro do cronograma, custo e padrão de qualidade de

atenda os clientes. Podemos então enxergar o Scrum como uma forma de organizar e gerenciar trabalho de forma a respeitar valores, princípios e práticas.

O Scrum tem foco direto nas pessoas, baseando-se nos seguintes valores:

- Honestidade;
- Abertura;
- Coragem;
- Respeito;
- Foco;
- Confiança;
- Empoderamento;
- Colaboração.

Para continuarmos nosso estudo aprofundado sobre Scrum vamos analisar uma palavra muito importante neste contexto que é o termo “empírico”, ou seja, toda conhecimento é baseado em processos de decisão no que é conhecido. Podemos assim contextualizar como que empírico é um processo de absorver conhecimento através de experiências vividas. Esse conhecimento empírico também chamado de senso comum é baseado em experiência imediata, não metódica e que não está interpretada com organização racional.

No mapa mental é apresentado os papéis, eventos e artefatos, que são integrados pelas regras do Scrum.

Pilares do Scrum

No Scrum três pilares apoiam esse processo empírico:

Transparência

Garante que os aspectos do processo que afetam o resultado precisam ser visíveis e conhecidos a todos que controlam o resultado. Estes aspectos devem ser de comum compreensão a todos observadores para que todos possuam um mesmo entendimento do que está sendo apresentado.

Inspeção

Os artefatos do Scrum devem ser frequentemente inspecionados pelos usuários Scrum e através do progresso deve ser direcionado ao objetivo da sprint para dessa forma identificar variações indesejadas.

Adaptação

Se durante a inspeção um inspetor informa que um ou mais aspectos de um processo possui desvio fora dos limites aceitáveis, impactando no resultado do produto, o processo do produto/serviço deverá ser ajustado o mais breve possível.

O Scrum apresenta quatro eventos formais para inspeção e adaptação que são:

- ✓ *Planejamento da Sprint;*
- ✓ *Reunião diária;*
- ✓ *Revisão da Sprint;*
- ✓ *Retrospectiva da Sprint.*

Papéis

Vivemos em tempos onde cada vez mais é importante buscar conhecimento e solução de problemas diários que são apresentados em todos os ambientes de negócios. Sempre é importante mencionar que cada projeto nasce com um grau de incerteza que realmente pode assustar, porém neste aspecto é importante mencionar o Time Scrum, que se enquadram perfeitamente no cenário atual pois são auto organizáveis e multifuncionais. Sendo assim sempre escolhem o melhor caminho a seguir para completar seu trabalho, dessa forma não são dirigidos por nenhuma pessoa fora do time. Quanto a times multifuncionais, possuem todas as competências necessárias para concluir o trabalho sem que existam dependências externas.

Todo esforço utilizado no desenvolvimento de um produto ou serviço consistem de uma ou várias equipes Scrum, que são compostas por:

Product Owner

O Product Owner (dono do produto), é o papel responsável maximizar o valor produto, porém esta determinação está um pouco complexa, então vamos entender de forma mais prática.

John é o Product Owner de uma empresa de desenvolvimento de sistemas, ele sabe tudo sobre a visão do produto, ou seja, para que serve, quais problemas ajuda a resolver. Porém para chegar a essa definição John teve que estudar todas as ideias apresentadas pelos stakeholders (usuários dos sistemas) e construí-las em formato de histórias dos usuários (requisitos) e estas histórias irão servir de base para que John construa o Backlog do Produto. Uma das características marcantes em John é a forma de engajamento que ele se dirige aos stakeholders, pois ele recebe muitas histórias dos usuários e se não conseguir estabelecer uma forma de controlar o backlog, ele será gigantesco e sem fim, imagine um software que nunca mais acaba (versão 150.5).

A solução tomada pelo John é gerenciar o backlog do produto pensando na capacidade de seu time de desenvolvimento e no produto a ser entregue.

Através do exemplo do John podemos determinar que o papel do Product Owner é:

- ✓ Maximizar o valor do produto e do trabalho do Time de Desenvolvimento, gerenciando de forma eficiente o backlog do produto;
- ✓ Deixar os itens do backlog do produto claros e fácil entendimento;
- ✓ Priorizar os itens do backlog do produto relacionando com as metas e expectativas do cliente;
- ✓ Garantir o valor da entrega do trabalho do Time de Desenvolvimento;
- ✓ Garantir visibilidade que o backlog do produto seja visível e transparente a todo Time de Desenvolvimento;

Importante!!!

- ✓ O Product Owner é uma pessoa, nunca em hipótese alguma um departamento, comitê, grupo entre tantas outras definições.
- ✓ Somente o Product Owner pode determinar alterações e mudanças no backlog do produto.

Neste momento você pode estar se perguntando, o Product Owner é um Analista de Negócios?

Segundo o guia Babok® 2.0 (2009) a Análise de Negócios (função do Analista de Negócios) é o conjunto de atividades e técnicas utilizadas para servir como ligação entre partes interessadas no intuito de compreender a estrutura, políticas e operações de uma organização e para recomendar soluções que permitam que a organização alcance suas metas."

Dessa forma podemos entender que sim, o Product Owner pode ser uma Analista de Negócios, como também pode ser um Gerente/Supervisor do produto ou até mesmo um membro do Time Scrum. Uma restrição que se deve atentar é que em momento algum o Product Owner poderá ser o ScrumMaster, pois isso gerará conflitos e poderá confundir os papéis e prejudicar tanto o engajamento do Time quanto a qualidade do produto.

ScrumMaster

Para entender este importante papel vou fazer uma analogia do Scrummaster com um missionário de qualquer religião existente. Vamos lá, um missionário em toda sua vida faz estudo diário da sua crença, costumes, bíblia, livros e todas as práticas envolvidas. Até chegar a sua igreja esse missionário passou por um processo de conversão, ou seja, saiu do mundo para encontra o Senhor. Diante de toda sua história de vida esse missionário agora quer catequizar e evangelizar as pessoas, apresentar toda essa crença e costume que mudou sua vida, ele se torna além de servo, um líder.

Pois bem, lhes apresento o Scrummaster, o evangelizador do Scrum. Esse papel promove e suporta o Scrum nas organizações, fazendo com que as pessoas entendam:

- ✓ A teoria;
- ✓ As práticas;
- ✓ As regras;
- ✓ Valores do Scrum.

O Scrummaster tem funções como:

- ✓ Ajuda no entendimento e utilização da auto-organização e interdisciplinaridade;
- ✓ O Scrummaster não deve gerenciar o Time de Desenvolvimento;

- ✓ Ele treina o Time de Desenvolvimento em ambientes de difícil aceitação do Scrum, garantindo que todos entendam o fluxo de processos do Scrum;
- ✓ Facilita os eventos do Scrum;
- ✓ Remove os impedimentos que possam interferir no objetivo do Time de Desenvolvimento;
- ✓ Tem o papel de Coach (técnico do time), ensinando e liderando o Time de Desenvolvimento na criação de produtos de alto valor;
- ✓ Tem como principal papel orientar o time na realização dos trabalhos. Neste ponto importante entender o Scrummaster não é gerente é apenas um guia que aponta o melhor caminho.

O Scrummaster é o papel mais importante do Scrum, porém vamos entender o porquê de tal afirmação:

Cenário 1

Temos um Time Scrum (Product Owner + Equipe de Desenvolvimento) totalmente inexperiente e também um Scrummaster inexperiente, podemos afirmar que será um total fracasso.

Cenário 2

Temos um Time Scrum (Product Owner + Equipe de Desenvolvimento) totalmente inexperiente, porém agora temos um Scrummaster experiente, com boa vivência em projetos ágeis, teremos uma boa chance de sucesso, pois o Scrummaster irá alimentar a engrenagem do processo empírico do Time Scrum.

Equipe de Desenvolvimento

Quem nunca quis fazer parte de uma grande equipe? Quando você assiste filmes de equipes como “Vingadores”, “Liga da Justiça” fica fascinado pela união do Super-heróis e como seria legal se na equipe de trabalho fosse tão legal quanto no cinema.

Pois bem vou te apresentar a Liga do Scrum, mais conhecida como Equipe de Desenvolvimento, fiz está analogia com heróis porque é assim que os vejo nas organizações. Eles trabalham com recurso reduzido, com um cenário de incerteza nos projetos e principalmente fugindo do famoso “Go-horse”, neste caso sabemos que o cliente quer algo de 3 meses em 3 dias e com a mesma qualidade e nessas horas se a equipe não for unida com um bom Product Owner,

um experiente Scrummaster e uma Equipe de Desenvolvimento auto-organizada e multifuncional não conseguirá organizar todo cenário desfavorável para algo real e entregável.

É responsabilidade do Time de Desenvolvimento executar o desenvolvimento e transformar o backlog do produto em incremento e funcionalidades, criando um sistema pronto que possa ser entregue ao cliente.

As características do Time de Desenvolvimento são:

- ✓ Auto-organizados;
- ✓ Multifuncionais;
- ✓ Não reconhece títulos para integrantes;
- ✓ Não reconhece sub-times;
- ✓ A responsabilidade pertence ao Time de Desenvolvimento, independente das habilidades individuais.

Tamanho do Time de Desenvolvimento

O Guia do Scrum descreve que o tamanho ideal do Time de Desenvolvimento deve ser pequeno o suficiente para se manter ágil e grande o bastante para completar uma parcela significativa do trabalho sem ultrapassar limites da Sprint. Porém podemos trabalhar como sugestão Time de Desenvolvimento com sete pessoas, e podemos ter variações de três a nove integrantes.

Eventos

Os eventos no Scrum são utilizados para criar uma regularidade e minimizar a necessidade de reuniões. Uma característica presente é que todos os eventos são time-boxed, ou seja, todo evento tem uma duração máxima estabelecida. Dessa forma quando inicia uma Sprint por exemplo sua duração é fixada e não pode ser reduzida ou aumentada.

Podemos enxergar o Sprint é um container que alocação os demais eventos, cada evento no Scrum é uma oportunidade de inspecionar e adaptar alguma coisa, os eventos permitem transparência e inspeção criteriosa. Quando ocorre alguma falha na inclusão destes eventos será resultado redução na transparência e perda de oportunidades para inspecionar e adaptar.

Sprint

É o trabalho realizado em iterações ou ciclos de até um mês (quatro semanas), onde o trabalho completo em cada sprint deverá criar algo que agrega valor ao cliente.

Toda Sprint tem uma data de início e data de fim bem definidas (Timebox) e devem ter sempre a mesma duração. Um novo Sprint começa imediatamente a finalização do anterior.

Restrições durante a Sprint:

- ✓ Não é permitido mudança no escopo;
- ✓ Não é permitido mudança de pessoal;

Sprint Planning

Uma característica do Product Backlog é que ele pode representar várias semanas ou meses de trabalho, muito maior que uma única Sprint (4 semanas). Devido a esse Product Backlog com várias semanas, o product owner e a equipe de desenvolvimento acordam um sprint goal que define o que o sprint que vai começar deve alcançar.

Sprint Execution

Quando a equipe do Scrum finalizar o sprint planning e concordar com o conteúdo do sprint seguinte, a Equipe de Desenvolvimento, orientada pelo Scrummaster, irá realizar o trabalho necessário no nível das tarefas para aprontar as features, onde “pronto” (done em inglês) significa que há um alto grau de confiança que todo o trabalho necessário para produzir features de boa qualidade tenha sido feito.

As tarefas que a equipe realiza irão depender da natureza do trabalho, isso mesmo, não estamos falando apenas de desenvolvimento de software, mas também da construção de uma casa, um prédio, de qualquer segmento que pertença o projeto.

O sprint backlog não poderá ser direcionado a equipe com uma ordem estabelecida, pois quem irá definir tal ordem será a equipe, por isso se chamam auto organizáveis.

Daily Scrum

Em cada dia do sprint, no mesmo horário, os membros da equipe de desenvolvimento realizam uma reunião de duração fixa de quinze minutos, que

recebe o nome de daily scrum. Esta é uma atividade de inspeção e adaptação, que possui uma abordagem comum de ter o Scrummaster como um facilitador da reunião, provocando em cada membro da equipe a responder três perguntas:

- ✓ O que eu realizei desde a última daily scrum?
- ✓ O que eu planejo trabalhar para a próxima daily scrum?
- ✓ Quais são os obstáculos ou impedimentos que estão evitando que eu avance?

A daily scrum não é:

- ✓ Uma atividade de solução de problemas;
- ✓ Uma reunião de status report;
- ✓ Uma reunião para lavar roupa suja ou discutir problemas de relacionamento.

A daily scrum é uma poderosa ferramenta de comunicação, utilizada para se comunicar os itens do backlog, veja bem ela é focada nos itens, é através dela que ocorre uma sincronização entre a equipe que proporciona que o trabalho seja executado da melhor forma.

Definição de Pronto (“Done”)

Quando um item do backlog do produto é descrito como “Pronto”, devemos entender o que significa pronto. Cada equipe de desenvolvimento possui um propósito, que na verdade é o propósito de cada sprint:

“Entregar incrementos de funcionalidades potencialmente liberáveis que aderem à definição atual de “Pronto” do Time Scrum (Guia Scrum, 2017).

A definição de pronto faz parte:

- ✓ Das convenções;
- ✓ Padrões ou diretrizes de desenvolvimento da organização;

Revisão da Sprint (Sprint Review)

É a apresentação da Sprint. Seu objetivo é a revisão do Product Owner, ou o próprio cliente, em todos os itens concluídos pelo Time. É um evento time-boxed de quatro horas. É realizada uma comparação do que foi entregue e o que deveria ter sido entregue.

Retrospectiva da Sprint (Sprint Retrospective)

É uma oportunidade para o Time Scrum inspecionar tudo que foi realizado e criar um plano de melhorias a serem aplicadas na próxima Sprint. Ela ocorre após

a Sprint Review e antes do planejamento da próxima Sprint. A reunião é um evento Time-boxed de no máximo três horas para a Sprint de um mês. O Scrummaster garante que o evento ocorra e que os participantes entendam seu propósito.

A retrospectiva da sprint tem os seguintes propósitos:

- ✓ Inspeccionar como a última sprint foi em relação às pessoas, aos relacionamentos, aos processos e às ferramentas;
- ✓ Identificar e ordenar os principais itens que foram bem e as potenciais melhorias;
- ✓ Criar um plano para implementar melhorias no modo que o Time Scrum faz seu trabalho.

Artefatos

Os artefatos são utilizados como apoio ao Time Scrum para aplicar regras que criam ligação entre os eventos, os papéis e os artefatos.

Backlog

- ✓ É o principal artefato do Scrum;
- ✓ Trata-se de uma lista ordenada de requisitos (estórias de usuários) necessário do produto;
- ✓ O Backlog do Produto lista todas as características, funções, requisitos, melhorias e correções que formam as mudanças que devem ser feitos no produto nas futuras versões;
- ✓ O Backlog do Produto possui os atributos de descrição, ordem, estimativa e valor;

O Backlog é dividido em conjuntos menores que contribuem para objetivos específicos:

- ✓ **Backlog do produto** - todo backlog que será trabalhado no projeto;
- ✓ **Backlog da versão de entrega** – parte do backlog que será trabalhado na versão de entrega definida entre o Time Scrum e o Cliente;
- ✓ **Backlog da Sprint** – parte do backlog considerada “preparada” e selecionada para ser trabalhada na versão da Sprint.

Existem artefatos não oficiais:

- ✓ Histórias;
- ✓ Taskboard e gráfico Burndown.

IMERSÃO EM SCRUM

Scrum entre Contextos e Segmentos

Quando mencionamos o fato da aplicação de métodos ágeis em outros segmentos que não a tecnologia da informação, podemos encontrar cenários onde esse trabalho pode ser árduo para implementação. Existem cenários onde o negócio não utiliza nenhum método padronizado em seus projetos e uma grande mudança nesse aspecto pode representar uma restrição à liberdade até encontrada. Outro cenário é aquele onde já existem metodologias ou práticas waterfall implementadas e podemos esbarrar num enfrentamento de conceitos que de maneira alguma é produtivo para a organização,

Mas para seguirmos numa implementação deveremos nos aprofundar em alguns conceitos ágeis direcionados ao Scrum que possibilitarão maior entendimento e visibilidade das vantagens do pensamento e métodos ágeis.

Definição oficial de cada papel

Sempre quando estou em minhas aulas ou até mesmo em conversas com profissionais sobre gestão de projetos, um dos assuntos que mais questionamos é a dificuldade que as organizações possuem para definir e assegurar os papéis de cada colaborador. No Scrum existem três papéis oficializados que são o Scrum Master, o Product Owner e o Time de Desenvolvimento, que constituem o Time Scrum. Cada um dos papéis possui função e razão muito bem definida de existir e seguindo esse parâmetro podemos garantir tanto o sucesso como maior desempenho do time nas entregas e geração de valor ao cliente.

Não estamos falando num aspecto hierárquico, não existe um “chefe” ou “gerente”, mas estamos dizendo que o importante é que cada papel cumpra as atribuições que são fundamentais para não desorganizar todo processo ágil sustentado pelo Framework Scrum.

Product Owner, o profissional da análise de negócios

Vamos abordar o Product Owner de uma forma muito relevante, ele é o fator crítico de sucesso do projeto. Podemos dizer sem medo se caso ele falhe nas suas atribuições todo projeto irá desmoronar, pois seria semelhante a realizar uma viagem seguindo um mapa errado do caminho a percorrer.

O P.O tem como objetivo desenhar esse mapa e demonstrar a equipe como atingir os objetivos propostos para realizar entregas de qualidade e construam valor. Pode parecer um pouco drástico essa responsabilidade ao P.O, porém ele é o analista, a pessoa que irá entender a “dor” dos Stakeholders e transcrevê-la em formato do backlog do produto. Neste aspecto cada funcionalidade ou item do backlog deve ficar muito bem entendido tanto pelo P.O quanto para os Stakeholders, quando isso acontece é minimizada a possibilidades de mudanças mais drásticas na execução do projeto.

Como encontrar um Product Owner?

Quando um ramo de negócio decide optar por um método, framework, metodologia ou simplesmente um pensamento voltado para uma melhor gestão de projetos se torna obvio que em inúmeros casos a empresa terá que investir em capital humano e formação para todos os níveis da organização.

Porém quando procuramos um profissional para assumir o posto de Product Owner no Scrum quais os fatores que devem ser levados em consideração?

Um Product Owner é um ponto de liderança do produto, em muitos aspectos pode até parecer com um gerente de produto, deve saber comunicar o que deve ser realizado e ter a capacidade de estabelecer uma ordem para a construção do produto. Outro fator importante é entender de prioridades e senso de urgência, porém podemos destacar algumas responsabilidades fundamentais como:



Figura 2.1 – Habilidades Product Owner

Comprometimento – o PO deve ser comprometido com a empresa e principalmente com cada projeto e time ao qual está alocado, como elo

fundamental para a qualidade final do produto ou serviço que deverá ser entregue sua dedicação é fator crítico de sucesso;

Relacionamento com o Cliente – um dos grandes desafios encontrados em qualquer equipe de projeto é entender qual a real necessidade do cliente e transformar aquilo em um conjunto de features que o satisfaçam. O sucesso de um projeto é definido pelo atingimento das necessidades dos stakeholders, e para isso é preciso saber se relacionar com todos buscando sempre os dados necessários para a construção do produto;

Negociação – deve possuir atitude para conseguir melhores resultados, através de hábitos e ferramentas que façam as partes envolvidas sair satisfeitas e com suas necessidades atendidas.

Comunicação – ter habilidade de se comunicar de forma clara e conseguir dimensionar tanto métodos e planos básicos de comunicação são pontos fundamentais.

Trabalho em Equipe – uma capacidade do P.O é auxiliar o Time no entendimento do Backlog e dessa forma interagir com a equipe de forma produtiva, sanando duvidas e deixando claro que os objetivos só serão atendidos se houver um esforço colaborativo de todos os talentos, pois sabemos que um único talento pode até ganhar um jogo, mas quando o trabalho é em equipe se ganha campeonatos.

Visão de Negócio – toda empresa possui sua estratégia que determina tanto objetivos a curto, médio e longo prazo, um PO deve assim como um analista de negócios entender a visão e objetivos estratégicos do negócio para compreender melhor cada item a ser construído no Product Backlog.

Entender de Priorização – quando falamos em priorização é de forma eficaz determinar que item é mais importante e agrega mais valor, um PO deve possuir essa capacidade de analisar cada item e sequenciá-los pensando no valor que será entregue ao cliente;

Conhecimento do Core Business – esse é um item de extrema necessidade, hoje em dia podemos perceber a existência de gestores que não sabem direito a área de negócios que estão inseridos, ou seja, qual o ramo que atuamos. Por esta razão podemos dizer que o PO pode possuir habilidades e conhecimento de um gerente de produto.

Mas como encontrar o PO em minha organização?

Para encontrar um Product Owner que atenda às necessidades da organização podemos buscar por gerentes de produto, analistas de marketing ou profissionais que tenham conhecimento do negócio e do segmento que atuam e tenham uma capacidade de entender as necessidades dos stakeholders. Neste aspecto é importante que o próprio departamento de Recursos Humanos defina um perfil de profissional para esta posição e que o mesmo tenha uma descrição de cargos e o mesmo pode ser inserido no plano de carreira da organização. Dessa forma o profissional se torna um recurso de valor na organização e assim como gerentes funcionais ou de projetos tradicionais pode participar de um programa de crescimento, com metas e objetivos estabelecidos pela organização.

Scrum Master, o formador ágil

Esse papel podemos considerar o fator chave para o Scrum funcionar em uma empresa ou área de negócios. O Scrum Master podemos fazer uma analogia com um consultor empresarial, peguemos o seguinte exemplo:

“Uma empresa de um segmento de infraestrutura necessita remodelar os seus processos operacionais, porém os seus gerentes funcionais não possuem a expertise para realizar essa mudança sozinhos. A empresa então contrata um consultor em gestão de processos que irá orientar toda a organização nessa atividade, ele não irá executar a mudança, mas será o facilitador para que ela ocorra.”

O Scrum Master é um facilitador das metodologias ágeis na organização. Pode perceber que não falei de somente de Scrum, mas de metodologias ágeis num contexto mais geral, justamente pensando em eventuais necessidades e projetos de diferentes situações que possam ocorrer.

O Scrum Master deve ser um profissional que possui bom conhecimento de metodologias ágeis e principalmente de Scrum, então podemos destacar duas situações para a empresa encontrar esse profissional:

- ✓ Promovendo um gerente de projetos ou gerente funcional e fazer que o mesmo passe por treinamentos e formações;
- ✓ Contratando um Agile Coach com experiência para difundir o ágil na organização e treinar possíveis candidatos a Scrum Master.

Nos aprofundando mais nas responsabilidades do Scrum Master podemos elencar as seguintes segundo Rubin (2017):

- ✓ Escudo;
- ✓ Coach;
- ✓ Líder Servil;
- ✓ Agente de Mudança;
- ✓ Autoridade do Processo;
- ✓ Removedor de Impedimentos.

E como habilidades podemos destacar:



Figura 2.2 – Habilidades Scrum Master

Quantos Scrum Master posso ter em minha empresa?

Essa resposta irá depender da quantidade de projetos, tamanho dos projetos e capacidade de organização de cada Time Scrum. Alguns frameworks que trabalham com Scrum escalado, ou sejam com vários times Scrum estabelecem limites tanto para Scrum Master e Product Owner por quantidade de Times, mas isso depende muito da realidade e maturidade da organização.

O que seria interessante num aspecto estratégico é que o departamento de Recursos Humanos em conjunto com Scrum Masters mais experientes promove programas de formação de novos profissionais para estarem disponíveis num crescimento futuro.

Time de Desenvolvimento, auto organizado e sustentável

O Time de Desenvolvimento é responsável pela produção das funcionalidades ou partes do projeto, num âmbito de software ela é composta por:

- ✓ Desenvolvedores;
- ✓ Arquitetos de Softwares;

- ✓ Analistas de Testes;
- ✓ Designer UI
- ✓ Analista de banco de dados entre outros

Porém quando falamos de outros segmentos esse time poderá ser composto por funções próprias de cada negócio, por exemplo na construção civil:

- ✓ Engenheiro Civil;
- ✓ Projetistas;
- ✓ Arquitetos;
- ✓ Desenhista entre outros.

O time tem como principais responsabilidades:

- ✓ Realizar todo trabalho da Sprint Execution;
- ✓ Realizar inspeção e adaptação;
- ✓ Auxiliar o Product Owner no refinamento do Backlog;
- ✓ Realizar o Planejamento da Sprint;
- ✓ Determinar técnicas para realização das estimativas da Sprint.

O Time de Desenvolvimento deve conhecer da metodologia ágil utilizada, saber que fazem parte de um Time Scrum, indiferente do setor ou ramo de atividade é importante se situarem dentro do time.



Figura 2.3 Habilidades do Time de Desenvolvimento

Focada no Trabalho não em Cargos – o time de desenvolvimento não tem foco em cargos, todos tem o mesmo objetivo e não existe distinção de júnior, pleno ou sênior dentro do time, essas classificações ficam de responsabilidade exclusiva do Departamento de Recursos Humanos para fins de remuneração e promoções. As tarefas são organizadas conforme habilidades e capacidades de entrega de cada membro.

Trabalho em Equipe – o time forma uma unidade, não existe talento individual, todos estão na mesma direção e o trabalho em harmonia e união é ponto crucial para sucesso das entregas em cada Sprint.

Cross Discipline Competence – a equipe é composta por profissionais com background diferentes, é ideal possuir uma mistura de profissionais juniores e seniores para promover um ambiente de aprendizado e extremamente colaborativo.

Comunicação – a equipe deve possuir uma habilidade de comunicação entre seus membros e com o Product Owner para sanar dúvidas do Backlog e também com o Scrum Master quando esse for acionado. Em qualquer ambiente empresarial um grupo de pessoas que forma um “silo” de trabalho promove “n” quantidade de canais de comunicação e somente quando calculamos esses canais conseguimos medir o impacto quando um indivíduo não cumpre seu papel de emissor-receptor, veja por exemplo:

Fórmula Canais de Comunicação:

N = Quantidade de Pessoas

$$CN = N (N-1) / 2$$

$$CN = 9 (9 - 1) / 2$$

$$CN = 36 \text{ canais}$$

Análise agora com uma pessoa a menos:

$$CN = N (N-1) / 2$$

$$CN = 8 (8 - 1) / 2$$

$$CN = 28 \text{ canais}$$

Neste exemplo perde-se 8 canais de comunicação, que podem impactar na agilidade e velocidade do Time de Desenvolvimento.

Tamanho Certo – o Scrum funciona muito bem com equipes pequenas, é sempre recomendado de 3 a 9 pessoas por Time Scrum. Caso a demanda necessite de mais pessoas é aconselhável criar uma nova equipe e trabalhar com Scrum Escalado. Cohn (2009), menciona algumas razões para manter a equipe pequena, sendo:

- ✓ Menos pessoas ociosas ou fazendo pouco esforço porque acreditam que outras darão conta do trabalho;
- ✓ A interação entre equipes menores é mais construtiva;

- ✓ Menos tempo gasto com coordenação dos esforços;
- ✓ É fácil perceber quando alguém não está atuando;
- ✓ Evita-se a superespecialização.

Sustentável – trabalhar de forma sustentável significa que o time deve trabalhar num ritmo dentro da realidade e velocidade aceitável. Quando se trabalha num ambiente altamente estressante e com prazos irreais e pressão total ao ritmo de trabalho o impacto é mensurado na saúde dos indivíduos do time e em um ambiente totalmente desmotivador.

Auto-organização – o Time de Desenvolvimento possui a cada Sprint que é entregar o incremento planejado na Sprint Planning. A equipe deve possuir a capacidade de se organizar para encontrar a melhor maneira de atingir esse objetivo. Não existe um chefe ou líder que irá determinar como a equipe irá trabalhar, mas sim todo time define como cada tarefa da Sprint será trabalhada e entregue.

Habilidades T-Shaped – desde que iniciei meu trabalho e estudo dos métodos ágeis e particularmente do Framework Scrum algo que chamou muita minha atenção foi a possibilidade que cada membro do time pode se desenvolver tecnicamente. Quando falamos de habilidade T-Shaped, referimos a sair da zona de conforto e participar de novas atribuições.

A habilidade T-Shaped tem como características prever ao membro do time a capacidade de compreender várias disciplinas, além é claro de possuir o conhecimento em uma disciplina específica.

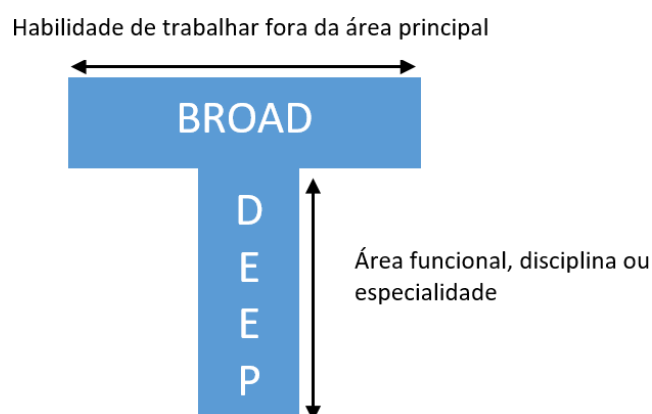


Figura 2.4 Habilidades T-shaped

Transparência – assim como pilar do Scrum, a transparência em cada Sprint auxilia o time a compreender todo trabalho realizado e ainda o que tem para ser realizado. A transparência no Scrum na questão do Time pode ser interpretada

como a facilidade de enxergar cada item do Sprint backlog através do taskboard e do gráfico burndown.

USER STORIES

Coletar requisitos é sempre um desafio árduo, precisamos definir um método que facilite o entendimento para:

- ✓ Comunicar as características, prioridades e valor;
- ✓ Permitir padrões de comunicação sobre as mudanças;
- ✓ Permitir que maiores detalhes surjam conforme as informações são aprendidas;
- ✓ Realizar o refinamento contínuo do Product Backlog, para que cada recurso e o Time aprenda mais.

Podemos entender como User stories a forma com que o usuário (cliente) descreve uma funcionalidade ou atributo, a partir destes dados é realizado o planejamento. Cada história pode ser mais bem avaliada e refinada sucessivamente, dessa forma podemos ir versionando cada item representado pelo cliente.

As User Stories foram desenvolvidas como forma do cliente expressar as características que agregam valor ao longo do tempo ao produto. Nelas os clientes podem descrever detalhes sobre design. E todos estes detalhes são descritos através conversas e anotações. Segundo Mike Cohn (2004) o modelo a seguir é extremamente recomendado:

Como um <usuário>, eu quero <objetivo> [para que <razão>]

No modelo proposto ele inclui:

- ✓ Função do Usuário – um cliente que se beneficia do projeto nesta história;
- ✓ Objetivo – recurso ou função que será necessário no projeto;
- ✓ Razão – relata o benefício para o cliente quando esta função ou recurso for implementada.

Exemplo:

Como <morador>, eu quero <que a casa possua três janelas> para que <aumente a ventilação e menor sensação de calor>.

Podemos também adicionar mais detalhes a uma pequena história, a isto chamamos de condição de satisfação (conditions of satisfaction, CoS), uma ferramenta muito poderosa que auxilia a equipe a entender melhor o objetivo de

cada história e demonstrar se os recursos foram utilizados de forma correta na Sprint Review, por exemplo:

Como **<morador>**, eu quero **<que a casa possua três janelas>** para que **<aumente a ventilação e menor sensação de calor>**, **<elimine a necessidade de instalação de ar condicionado>**.

Quando ocorre situações com nível de incerteza como a história mencionada é interessante decompor em histórias menores para adicionar detalhes, por exemplo:

User Story #1

Como **<morador>**, eu quero **<que a casa possua três janelas>** para que **<aumente a ventilação e menor sensação de calor>**.

User Story #2

Como **<morador>**, eu quero **<que a casa possua três janelas>** para que **<elimine a necessidade do uso de ar condicionado>**.

Perceba que em uma história o cliente manifesta o desejo de três janelas para diminuir a sensação de calor, porém fica uma pergunta, e se mesmo com as três janelas a casa ficar quente?

Por esta razão na segunda história é inserido para eliminar a necessidade de ar condicionado, neste aspecto será necessário um estudo que extra na história para avaliar a necessidade de ar condicionado. Este estudo será um projeto realizado por um engenheiro, caso o estudo comprove que mesmo com três janelas irá ficar quente em determinadas épocas do ano, ocorrerá uma atualização nas histórias, podendo ficar assim:

Story #3

Como **<morador>**, eu quero **<que a casa possua três janelas>** para que **<aumente a ventilação e menor sensação de calor>**.

Story #4

Como **<morador>**, eu quero **<instalação de ar condicionado>** para que **<para diminuir o calor no verão>**.

No exemplo mencionado já utilizamos uma situação simples da área de construção civil. Sei que pode parecer complicado imaginar representar requisitos de uma área tão técnica, porém não estamos deixando de lado os aspectos técnicos, eles serão discutidos com o Time de Desenvolvimento no

momento da Sprint Planning e ali cada história receberá sua pontuação (estimativa).

Então sempre quando ocorrer várias situações nas histórias de condições de satisfação, será necessário ir desagregando uma história para aumentar seus níveis de detalhes. Porém caso a história mesmo possuindo várias condições e cabe em uma Sprint não é necessário decompor em outras histórias.

As User Story podem ser definidas também pelos chamados de “os três C’s”, que são:

- ✓ Cartão
- ✓ Conversas
- ✓ Confirmação

VISÃO DO NEGÓCIO

Considero uma das maiores falhas que ocorrem em equipes de projetos independente da metodologia utilizada, o isolamento que estas equipes protagonizam no interior das empresas. As equipes não entendem e as vezes nem sabe que existem um planejamento estratégico na organização. Este perigoso equívoco ocorre por uma série de fatores que podem ser distintos dependendo de cada área de negócio e no âmbito que os projetos estão nascendo.

A estratégia é fundamental em qualquer negócio, e jamais deve ser fragmentada em pedaços que não se comunicam e interagem entre si. Um planejamento estratégico é um organismo que responde a um grande sistema de comunicação em uma empresa ou unidade de negócio, e quando setores ou grupos não caminham em conjunto existe grande possibilidade da estratégia ser toda perdida e isso para uma organização pode significar palavras tenebrosas como “morte”, “falência” e isso resultada danos em todos os aspectos hierárquicos. Então somos como parte desse organismo que representa os projetos da organização também responsáveis por não permitir que a estratégia seja perdida e suas metas não atingidas.

Para entender um ramo de negócio é fundamental saber qual é o seu conceito, ou seja, em que ramo de atividade está inserido e a partir desta contextualização podemos enxergar qual a missão, a razão de existir daquele negócio, quais dores os produtos ou serviços são curadas nos clientes. Podemos utilizar para visualizar isso de forma mais clara o modelo proposto por Osterwalder (2014), o Value Proposition Design:

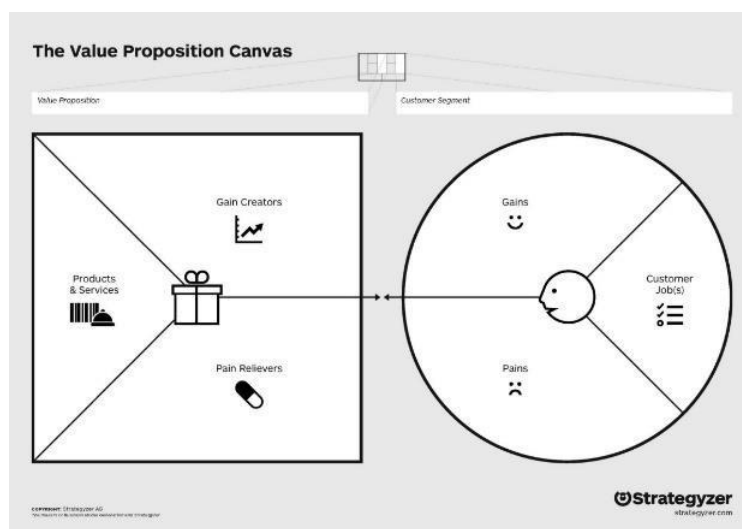


Figura 2.6 – Value Proposition Design (Osterwalder, 2014)

Neste modelo apresentado fica muito simples para encontrar as dores e ganhos que o cliente obtém adquirindo nossos produtos. De forma simples do lado esquerdo da figura 2.6 temos os produtos e serviços que constroem uma proposta de valor, os analgésicos que demonstram como os produtos e serviços aliviam as dores do cliente e por fim os criadores de ganho, que descrevem como geramos ganho ao cliente. Do lado direito temos o cliente com as tarefas que ele tenta realizar tanto pessoal como profissionalmente e nesse aspecto isto pode gerar resultados ruins e obstáculos que geram a dor no cliente e também evidência os ganhos e resultados que os clientes querem alcanças em suas tarefas.

Este modelo apresentado deixa muito claro o encaixe que deve existir entre o produto/serviço e a real necessidade do cliente. Se um time de projeto não conhece pelo menos de forma superficial essa relação, podemos definir que existe um déficit de empatia na equipe, pois a mesma não foca suas tarefas e resultados no principal stakeholder envolvido que é o “cliente”. Se o modelo de pensamento ágil afirma que devemos entregar valor constante num ciclo de inspeção e adaptação, pode parecer de forma muito evidente falhas na implementação e execução de projetos ágeis, quando a visão do negócio não é percebida pela equipe e dessa forma o valor que deve ser percebido pelo cliente passa a ser manipulado para atender os desejos do time do projeto e não do principal interessado, responsável pela existência da demanda.

Outro item importante na estratégia do negócio é a visão, ou seja, onde a empresa quer chegar dentro de período temporal. Pense de forma prática referente a crescimento, faturamento, Market Share e como o negócio deseja ser visto no futuro. Por parecer difícil as vezes para colaboradores enxergarem esse quesito, porém é importante entender novamente que somos parte de um grande sistema que irá proporcionar os resultados desejados em curto, médio e longo prazo. Então o time de projetos deve se perguntar como nossas atividades podem auxiliar a organização a atingir seus objetivos estratégicos? Esse deve ser um questionamento frequente, pois estabelece uma conexão entre negócio e pessoas e sabemos que o maior ativo de uma empresa é seu capital humano.

Por fim temos os valores que norteiam a organização e para entendê-los melhor, podemos fazer um pequeno questionamento: Quais valores ou comportamentos as pessoas dizem que existem em cada um de nós? Essa pergunta com certeza irá demonstrar na resolução um importante dado, como somos vistos. Agora uma empresa passa pelo mesmo processo em maior escala, pois cada cliente enxerga cada negócio do seu jeito por terem experiências particulares. A empresa então deve demonstrar a todos os seus consumidores, fornecedores e mercado em geral quem ela é, isso é realizado através dos valores apresentados na definição da sua estratégia.

Num aspecto de negócio, cada time de projeto, e repare o seguinte falo time de projeto e não “scrum”, pois cada empresa tem seus métodos para atingir resultados, e responsáveis por utilizarem esses métodos encontramos os times ágeis, tradicionais e aqueles que não possuem método algum definido, mas mesmo assim são equipes de projetos, deve entender essas informações básicas:

- ✓ Conceito do negócio (core business);
- ✓ Missão;
- ✓ Visão;
- ✓ Valores;

A partir daqui o trabalho do time fica muito mais fácil, pois compreendendo da visão do negócio (produto) facilitará a Sprint Planning e compreensão das histórias apresentadas no Sprint Backlog. O Product Owner como personagem

fundamental deve possuir uma expertise ainda maior na compreensão da visão do negócio, pois através dessa visão ele será capaz de construir um Product Backlog consistente e que atenda às necessidades e anseios dos stakeholders.

Quando a visão for compreendida iremos saber como podemos satisfazer essa visão e também a partir desse ponto ter capacidade de:

- ✓ Definir restrições para o projeto;
- ✓ Entender as premissas;
- ✓ Definir itens básicos de qualidade;
- ✓ Definir estrutura de custos;
- ✓ Realizar uma correlação entre prazo e escopo.

Mas para que tudo isso? Não estamos falando de ágil?

Então, ser ágil não significa ser imprudente, não precisamos criar burocracia exagerada, mas como já mencionamos um P.O deve possuir como habilidades básicas entender do negócio e enxergar restrições, premissas, dimensionamento de recursos entre outras habilidades. Veja bem, ele não é gerente de projetos, mas possui algumas habilidades que são similares.

Conforme o P.O vai coletando as User Stories, ele já vai analisando, pensando em possíveis restrições, premissas, custo, prazo e padrões de qualidade, para poder na conversação e confirmação das histórias construir um item para o Backlog mais certo.

PRODUCT BACKLOG OU ESCOPO?

Nesta etapa vamos entender algo que posso dizer com clareza que é o fator crítico de sucesso de qualquer projeto ágil, e além disso gostaria de fazer uma ligação a uma área do Guia PMBOK® e demonstrar como uma compreensão da relação entre escopo e Backlog do Produto pode facilitar o entendimento e resultados a serem obtidos. Estou falando da área de conhecimento de gerenciamento do escopo do projeto.

Quando comecei a ideia de escrever este livro para apresentar metodologias ágeis em qualquer cenário ou segmento de negócios procurei entender que é necessário expandir o conhecimento em projetos e gestão para trazer bons resultados e sendo assim vamos entender essa relação product backlog x escopo.

No gerenciamento do escopo passamos por todo um planejamento que tem como principal autor o papel do gerente de projetos, ele é o responsável em conjunto com a equipe do projeto em:

- ✓ Coletar os requisitos;
- ✓ Definir o escopo;
- ✓ Criar a estrutura analítica do projeto;
- ✓ E trabalhar no monitoramento e controle do escopo.

Gostaria de focar no processo direcionado a coleta dos requisitos que é a etapa onde é realizado a determinação, documentação e gerenciamento de todas as necessidades das partes interessadas do projeto. Lembrando que todo esse processo tem como objetivo listar os itens importantes para cumprir as metas do projeto.

A partir destes requisitos coletados é estabelecido o escopo do projeto que será continuado pelo cronograma, custo e demais planos de gerenciamento. Como podemos perceber após definido o escopo com base nos requisitos o mesmo somente poderá ser alterado mediante a aprovação de um comitê de gestão de mudança e para isto existe um processo específico que se chama realizar o controle integrado de mudanças, nele são revisadas todas as solicitações de

mudança, aprovações, comunicações e atualizações de mudança em todos os planos auxiliares de gerenciamento de projetos. Dessa forma entendemos que podem ocorrer mudanças e essas são controladas em um nível extremamente alto de gerenciamento, pois qualquer mudança pode alterar escopo, tempo, custo, qualidade e pode aumentar os percentuais de riscos em projetos.

Este pequeno lembrete do guia PMBOK® serve apenas para que possamos entender que o escopo também sofre alterações e segue todo um processo para ser estabelecido e a principal relação com o product backlog está nos impactos que qualquer mudança pode representar tanto em projetos ágeis como tradicionais. Falar de mudança neste momento serve para alertar que um projeto está sujeito às mesmas durante todo seu ciclo de vida, porém devemos olhar para elas com criticidade e cautela, nenhum método ou prática irá extinguir os impactos, porém eles precisam ser muito bem gerenciados.

Entendendo o Product Backlog

O Product Backlog se trata de uma lista de tudo que será desenvolvido pelo Time de Desenvolvimento no projeto. Podemos dizer que o product backlog é toda matriz de requisitos encontrada no Framework Scrum, nela consta tudo que precisa ser feito para entregar o produto nos padrões e necessidades descritos pelas partes interessadas nas suas user stories. E cabe ao Product Owner realizar a busca pelo conteúdo, disponibilidade e ordenação.

Algo que merece destaque é a afirmação que um product backlog nunca está pronto, pois os requisitos no andamento do projeto podem ser melhores entendidos e dessa forma o product backlog passa por uma evolução constante e está sempre ligado à necessidade do produto, para que este se torne mais útil.

O product backlog segundo o Guia Scrum (2017) lista todas as características, funções, requisitos, melhorias e correções que formam as mudanças que devem ser realizadas no produto em seu versionamento.

Os itens do product backlog possuem:

- ✓ Descrição;
- ✓ Ordem;
- ✓ Estimativa;

✓ Valor.

Podemos entender que o product backlog é um organismo vivo, pois ele está em constante transformação, devido a alterações que sejam solicitadas pela área de negócios, mercado em que está inserido e todas suas alterações. O Product Owner deve possuir grande habilidade de percepção e negociação com as partes interessadas para garantir apenas mudanças realmente necessárias no projeto, sabendo separar reais necessidades de “perfumaria” apresentada pelas partes interessadas.

Dependendo do tamanho do projeto podemos ter múltiplos Times Scrum trabalhando no mesmo produto, neste caso pode-se ter um product backlog que servirá como base para descrição do trabalho previsto, porém pode-se estabelecer product backlog que agrupe parte de itens para cada Time Scrum poder trabalhar no desenvolvimento do produto.

O product backlog é composto por itens que podem ser características, defeitos, correções entre outros trabalhos que passem pela avaliação do product owner e seja entendido como item prioritário e importante.

Um bom backlog segundo Roman Pichler (Pichler 2010) e Mike Cohn pode ser melhor entendido no acrônimo DEEP, que descreve as diversas características importantes para bons product backlogs, sendo eles:

- **Detalhado apropriadamente** – itens do backlog possuem tamanho e detalhamento diferentes, os que estão no topo e mais prioritários possuirão tamanhos bem definidos para se ajustar no Backlog da Sprint e contém detalhamento suficiente para seu entendimento. Devemos entender que o princípio de sustentabilidade caminha com a gestão ágil de projetos, não iremos detalhar itens muito cedo para não correr risco de retrabalho.
- **Emergente** – o product backlog irá passar constantemente por repriorização e reequilíbrio por estar no processo de inspeção e adaptação, dessa forma podemos concluir que o product backlog está emergindo por todo projeto. Trata-se de incorporar as mudanças e consequências que as mesmas irão ocasionar.

- **Estimado** – cada item possui uma estimativa de tamanho que é relacionada a todo esforço necessário e velocidade do Time. O Product Owner utiliza tais informações para priorização de cada item.
- **Priorizado** – quando falamos em priorizar devemos entender que não existe aquela frase comum no meio corporativo “tudo é prioridade”, no Scrum deve-se entender que os itens que serão priorizados são aqueles que estão próximos das Sprints. O Product Owner será responsável em analisar novos itens que vão surgindo e ordena-los conforme sua prioridade.

Outro item importante no Scrum é o que chamamos de **Grooming**, pense nele como uma forma de gerenciar, organizar e administrar o product backlog, ele é composto por três atividades:

- ✓ Criar Itens do Product Backlog;
- ✓ Estimar Itens do Product Backlog;
- ✓ Priorizar Itens do Product Backlog.

O Grooming é realizado pelo Product Owner, porém ele poderá envolver partes interessadas, Time de Desenvolvimento e também do Scrum Master e o Guia Scrum não descreve quando ele deve acontecer, mas isso pode ser definido pelo Product Owner, algumas dicas são:

- ✓ Pequenos workshops após a Daily Scrum;
- ✓ Inserir pequenos eventos durante a Sprint;
- ✓ No início de uma nova Sprint.

DEFINIÇÃO DE PRONTO

Quando um item é definido como pronto, isso deve ser visível para todos do Time Scrum, ou seja, deve existir um consenso na interpretação do que foi definido como pronto. Podemos dizer que possuem um entendimento compartilhado e isso é próprio de cada Time Scrum, então cada um pode possuir sua definição de pronto e para isso podem ser utilizadas ferramentas, processos devendo estar alinhado com o Product Owner.

Importante entender que o conceito de pronto, quem define é quem utiliza. Definir com o Time Scrum a definição de pronto podendo atender:

- ✓ Itens de qualidade;
- ✓ Funcionalidades ou características finalizadas;
- ✓ Todos os incrementos adicionados foram testados e funcionam em conjunto?

SPRINT PLANNING

Quando o Product Owner organiza o product backlog é estabelecido um nível de trabalho que pode compreender a semanas ou meses, dessa forma entendendo que cada Sprint possui Time Box de quatro semanas é necessário organizar cada item nas Sprints, seguindo sua prioridade definida no Product Backlog, esse processo é um evento chamado de Sprint Planning (Planejamento da Sprint). Consiste em uma reunião de quatro a oito horas para uma Sprint de trinta dias, que acontece no início de cada Sprint.

Todo o Time Scrum participa da Sprint Planning, sendo:

- **Product Owner** – apresenta o product backlog priorizado, compartilha informações e responde as dúvidas que ocorrerem do Time de Desenvolvimento;
- **Time de Desenvolvimento** – análise o que pode entregar e de forma realista estabelece o que podemos chamar de compromisso ao final da reunião;
- **Scrum Master** – participa apenas como um facilitador, desafiando a equipe e rompendo impedimentos que possam ocorrer. Não interfere nos itens do product backlog.

O processo da Sprint Planning consiste em entradas de informações (Item Product Backlog), que será analisado determinando capacidade do Time. Nesta etapa o Time pode definir técnicas e ferramentas que poderão ser utilizadas para facilitar a construção do Sprint Backlog. A saída neste caso seria a capacidade da equipe, o objetivo da Sprint (o que o Time entende como “Pronto”) e também podem levantar possíveis restrições além da velocidade do Time.

Kanban como artefato a ser criado na Sprint Planning

Podemos entender o Kanban como uma estratégia para trazer ganho aos Times Scrum, partindo de um conceito de fluxo puxado, onde existe um movimento contínuo otimizando todo processo. O Kanban traz uma melhora empírica para o processo, pois seu foco é baseado na transparência, gestão visual e fluxo

puxado, em conjunto com o Scrum se torna uma base extremamente eficaz para gerar valor real ao cliente.

Segundo o Guia Kanban for Scrum (2018), os Times Scrum otimizam o fluxo utilizando quatro práticas:

- ✓ **Visualização do Fluxo de Trabalho** – forma que o Time Scrum torna o seu fluxo de trabalho transparente, levando a constantes conversas e colaboração para melhoria;
- ✓ **Limita o WIP (Work in Progress)** – este item refere-se aos itens de trabalho que o Time Scrum iniciou, porém ainda não finalizou. Os Times que utilizam kanban devem controlar estes itens iniciados até o momento que estiverem concluídos;
- ✓ **Gerenciamento ativo dos itens de trabalho em andamento** – é responsabilidade do Time Scrum garantir o gerenciamento contínuo de forma proativa dos itens de trabalho, respondendo rapidamente itens bloqueados, garantindo que os itens de trabalho não fiquem velhos e esquecidos, desobstruir trabalho que se acumula nas colunas do kanban.
- ✓ **Inspecionar e adaptar sua definição de “fluxo de trabalho”** – o kanban deve mostrar todas as políticas ou direcionar os membros do Time Scrum onde podem encontrar. Podemos entender políticas como as regras do jogo, ou seja, o que pode ou não pode no framework Scrum.

Existem quatro métricas básicas que os Times Scrum devem acompanhar quando utilizam o kanban:

- ✓ **Trabalho em andamento** – número de itens iniciados e não concluídos;
- ✓ **Tempo do ciclo** – tempo decorrido entre início e fim de um item;
- ✓ **Idade de cada item de trabalho** – tempo decorrido entre início e a hora atual;
- ✓ **Taxa de transferência** – Números de itens de trabalho concluídos por tempo.














TAREFAS	A FAZER	FAZENDO	FEITO
	 		
	 		
	 		
			
	 		

Figura – Kanban resultante da Sprint Planning

Após finalizar a Sprint Planning todos os itens (tarefas) podem ser alocados no kanban para que se torne o mecanismo de gerenciamento da Sprint Execution.

Velocidade

A forma mais clara de definir velocidade é dizer que é a soma dos itens do product backlog que foram completados no término da Sprint. A velocidade não está relacionada a valor entregue (entenda valor como incremento no produto), mas sim o tamanho do que foi entregue.

As métricas de velocidade são utilizadas para planejamento e também para uma forma de avaliação de desempenho, porém pense nesse item como avaliação não de pessoas mais de processo, sendo analisado na Sprint Retrospective.

Para calcular tempos de finalização de release fazemos o seguinte:

1. Temos uma release de 120 pontos que precisa ser finalizada entre 10 e 12 Sprints;
2. Sabemos que o Time de Desenvolvimento tem uma velocidade de 10 pontos;
3. $120 \text{ pontos} / 10 \text{ pontos} = 12 \text{ sprints}$
4. Podemos dizer que 10 pontos será a velocidade baixa que o Time poderá trabalhar;
5. $120 \text{ pontos} / 12 \text{ pontos} = 10 \text{ sprints}$

6. Podemos dizer que 12 pontos será a velocidade máxima que o Time poderá trabalhar.

Quando trabalhamos com dados históricos tudo fica mais fácil para realizar o planejamento, porém e quando não possuímos tais dados?

Devemos então realizar o que chamamos de previsão da velocidade que consiste na análise da equipe em avaliar os itens do Product Backlog e verificar quais são os itens que ela considera capaz de ser entregue em uma Sprint. A partir destas pode-se realizar as pontuações e cálculos de intervalos de velocidade para analisar meta e objetivos da Sprint Backlog.

ESTIMATIVAS DO PRODUCT BACKLOG

Neste momento entendemos que o projeto foi aprovado e liberado pelas partes interessadas e o patrocinador. Então devemos realizar as estimativas de:

- ✓ Itens do Product Backlog;
- ✓ Tarefas da Sprint Backlog

Rubin (2017), descreve duas formas de realizar a estimativa dos itens do product backlog, sendo elas:

- ✓ **Story Points** – utilizam a combinação de fatores como a complexidade e tamanho físico dos itens do product backlog. A ideia é comparar User Stories e atribuir pontos a cada uma delas. Dessa forma é medido o tempo necessário e esforço para construir a história.
- ✓ **Ideal Days** – é aplicado para realizar estimativas na Sprint Planning, basicamente é a velocidade calculada com base em um número de horas que o Time de Desenvolvimento consome para completar um trabalho que tenha relação a um dia de trabalho (*ideal days*).

Planning Poker

Técnica de estimativa desenvolvida por Grenning (2002) e popularizada por Cohn (2006). A dinâmica apresentada através de um jogo de cartas que é realizada pelo Time de Desenvolvimento de forma colaborativa e segundo Cohn (2006) alguns conceitos são fundamentais para o sucesso da técnica que são: baseado em consenso, opinião especializada, discussão intensa, dimensionamento relativo, agrupamento e binning acurado e por fim faz uso do histórico de estimativas.

O planning poker é uma técnica altamente colaborativa e envolve todo Time Scrum, onde:

- ✓ **Product Owner** – realiza a apresentação e detalhamento dos itens do product backlog;
- ✓ **Scrum Master** – trabalha como um coach para auxiliar na aplicação da técnica da forma mais correta possível;

- ✓ **Time de Desenvolvimento** – responsáveis por gerarem as estimativas.

Como funciona? Quais são as etapas?

Etapa 1 – Cada membro do Time de Desenvolvimento recebe um conjunto composto por 14 cartas;

0	1/2	1	2	3	5	8
13	20	40	100	∞	?	π

Figura – Cartas Planning Poker

Carta	Descrição
0	Item concluído ou pequeno que não será necessário numera-lo.
1/2	Itens muito pequenos.
1	Itens pequenos.
2	
3	
5	Itens médios.
8	
13	
20	Itens Grandes.
40	
100	
∞	Item tão grande que não faz sentido numerá-lo.
?	Um integrante do Time de Desenvolvimento não entendeu o Item do Product Backlog e solicita explicações do Product Owner.
π	Quando alguém do Time de Desenvolvimento estiver cansado, ele poderá solicitar um tempo de pausa.

Tabela – Descrição Cartas Planning Poker

Etapa 2 – o Product Owner seleciona o item do Product Backlog a ser estimado e realiza a apresentação para o Time de Desenvolvimento.

Etapa 3 – Ocorre a discussão e respostas as dúvidas apresentadas.

Etapa 4 – Após a discussão cada membro do Time de Desenvolvimento seleciona uma das cartas em sigilo e apresenta.

Etapa 5 – Caso todos cheguem a um consenso o número que foi apresentado é o resultante da estimativa.

Etapa 6 – Caso não aja consenso é aberto novamente a discussão e a posteriormente as cartas são lançadas novamente.

Curiosidade

A escala mais utilizada no planning poker e sugerida por Cohn (2006) é baseada em partes na sequência de Fibonacci, nela ocorre uma sucessão de números que seguem um padrão onde cada elemento é a soma dos dois anteriores, por exemplo:

1,1,2,3,5,8,13,21,34,55,89,144,233,377...

Na escala proposta para o planning poker é utilizada os seguintes elementos:

0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100 além dos elementos? ∞ , π .

SPRINTS

Vamos imaginar agora uma situação bem particular, você é um piloto de uma dessas categorias de automobilismo, sua equipe desenvolveu um carro muito rápido e seu objetivo é vencer a corrida (momento crucial do processo). Antes da corrida seu Time irá se reunir para discutir estratégias, planejamento e possíveis alternativas de obter o resultado final que é chegar em primeiro lugar. Durante cada volta do circuito sua equipe vai inspecionando como foi a volta anterior, adaptando a estratégia conforme a corrida está acontecendo, talvez tenha que trocar pneu, ficar atento com possíveis acidentes na pista, se adaptar ao clima que poderá mudar e após um número “N” de voltas a corrida termina.

O final da corrida entrega o incremento completo, seja ele o primeiro, segundo ou apenas completar o circuito, a meta e definição de como está pronto é definida pelo Time conforme a realidade de cada equipe de automobilismo.

No exemplo apresentado fizemos uma analogia rápida do Scrum com uma corrida automobilística e entendemos que cada volta do circuito é a Sprint, ponto mais importante do Framework Scrum, se pensarmos no Scrum como corpo humano podemos dizer que o Product Backlog é o cérebro e a Sprint o coração do Scrum e cada papel são os braços e pernas responsáveis pela movimentação do projeto.

Definindo os Eventos

Algo que desde que conheci o Framework Scrum chamou muita minha atenção foram os eventos Time-box, que são eventos com tempo máximo de duração. Entendi que somente esse fator já corrigia um grande problema que ocorre não somente em gerenciamento de projetos, mas também em todo segmento empresarial.

Barbosa (2006), menciona dados de uma pesquisa que informa que 65% das pessoas acreditam que as reuniões são malconduzidas. Falando sobre essa condução podemos entender que as reuniões habituais que participamos hoje em dia não possuem nenhum tipo de planejamento ou método estabelecido, dessa forma as empresas perdem tempo e isso resulta em recursos alocados

em reuniões pouco fundamentadas e que tem impacto em custo (\$), ou seja, o impacto financeiro será consequência certa.

No Framework Scrum cada evento é muito bem detalhado tanto em processos, quanto no tempo para cada momento do projeto, dessa forma se trabalha num formato evitando desperdício de tempo, material e recursos humanos.

Daily Scrum mais produtivas

A daily scrum é um evento Time Box de quinze minutos e tem como objetivo responder três perguntas sobre as tarefas da Sprint, sendo:

- ✓ O que eu fiz ontem para atingir a meta da Sprint?
- ✓ O que farei hoje para atingir a meta da Sprint?
- ✓ Existe algum impedimento que impeça o atingimento da meta da Sprint?

É importante que essa reunião seja realizada sempre no mesmo horário, local e se possível com todos os membros do Time Scrum em pé. Quando se define uma regra clara como local e horário aumentamos a consistência e reduzimos a complexidade.

Esse modelo de reunião pode ser aplicado mesmo fora do framework Scrum, para ambientes tradicionais de qualquer empresa, imagine que todos os dias um departamento de produção faça uma reunião de quinze minutos para analisar a demanda seja ela oriunda de projetos ou processos, a implementação desse estilo de reunião é simples e traz grandes resultados.

SPRINT REVIEW, INSPEÇÃO E ADAPTAÇÃO DO PRODUTO

Esse evento sempre considere como o ponto chave de qualidade do Framework Scrum, após a Sprint Execution o Time Scrum e partes interessadas se reúnem para analisar todo trabalho realizado. Aqui inspecionamos e adaptamos o incremento do produto, esse evento ocorre ao fim da Sprint antes da Sprint Retrospective. Possui time-box de quatro horas.

Na Sprint Review é realizado um processo simples onde o incremento potencialmente entregável é demonstrado, discutido por todos participantes (inspeção) e adaptação (vão ocorrer mudanças?).

Um aspecto muito importante nesse evento é a conversação entre os participantes, onde todos mesmo quem não faz parte do Time Scrum realize as perguntas certas que resultarão num ótimo feedback e se estamos atendendo os objetivos da Sprint.

Ao fim dessa reunião teremos o product backlog novamente passando pelo processo de grooming (criar, refinar, estimar e priorizar).

SPRINT RETROSPECTIVE, INSPEÇÃO E ADAPTAÇÃO DO PROCESSO

No Scrum podemos utilizar inúmeros processos e técnicas, na Sprint Retrospective vamos discutir a Sprint finalizada num nível de processos detectando o que foi positivo e negativo, e mudanças que podem ser realizadas para otimizar o processo. Possui time-box de três horas.

Nesta reunião será inspecionado os seguintes itens:

- ✓ Relacionamentos;
- ✓ Processos;
- ✓ Ferramentas.

Não se trata apenas de uma discussão simples para preencher informações positivas e negativas, deve ser entendida como uma oportunidade de buscar melhoria contínua e aumentar o nível de maturidade em gerenciamento de projetos ágeis do Time Scrum.

Toda empresa possui seus ativos organizacionais e culturais e quando a reunião de retrospectiva é bem fundamentada auxilia a o negócio a definir a cada Sprint qual o rumo e padrões de gestão de projetos que desejam seguir, e isto causará um impacto muito importante na organização após cada projeto, cada Sprint, pois irá produzir novas bases de conhecimento e organização dos processos ágeis.

LIDERANÇA ÁGIL

Existem diferentes tipos de liderança, cada uma delas poderá ou não produzir resultado em nós e nos outros. É importante entender primeiramente que quando abordamos um tema como liderança nos deparamos com um conceito do Scrum que diz que não existe gerente, chefe ou supervisor. Porém liderança está muito além disso, necessitamos nos adaptar a um estilo de liderança que se adapte na melhora de nossa personalidade em todas as circunstâncias em nosso ambiente de projetos.

Quando cada membro de um Time se adapta a um melhor estilo de liderança ele alcança:

- Uma redução no estresse no trabalho;
- Melhoria profissional;
- Maior satisfação pessoal com o trabalho.

Para falar de cada um dos estilos de liderança devemos entender que sua aplicação pode ser resultante de uma combinação entre eles, será difícil encontrar alguém que utilize apenas um estilo isolado. Os estilos que abordaremos são:

Liderança autocrática - é um estilo onde os líderes possuem um grande poder sobre sua equipe. O membro de equipes tem pouca ou as vezes nenhuma possibilidade de manifestar opiniões sobre seu trabalho.

Algumas características são:

- As decisões são tomadas rapidamente;
- Quando se trabalha com equipes pouco qualificadas este estilo é muito favorável;
- Pode ocorrer resistência da equipe;
- Pode acontecer grande rotatividade de pessoal (turn-over);
- Um líder autocrático tem papel importante papel quando existe a necessidade de tomar decisões difíceis, fazendo que os membros da equipe tenham foco e energia no trabalho.

Liderança democrática – Neste estilo os líderes tomam a decisão final em situações de trabalho e projetos, porém eles incluem a participação da equipe nas decisões e dessa forma incentivam fortemente a criatividade.

Algumas características são:

- Traz grande satisfação a equipe com seu trabalho;
- Desenvolve habilidades dos membros do Time;
- Aumenta motivação sem necessidade de aspectos financeiros;
- Como deficiência pode causar uma lentidão na tomada de decisões;
- Garante a participação de todos os membros do Time, mesmo os mais novos.

Liderança orientada a tarefas – é direcionada como o próprio nome menciona as tarefas, e pode facilmente caminhar para a direção da liderança autocrática. Neste estilo o líder define as atividades para cada membro do time, ou seja, planejamento e organização do trabalho.

Algumas características são:

- As metas são alcançadas em nenhum transtorno;
- Ideal para times que sabem gerenciar seu tempo;
- O bem-estar e qualidade de vida da equipe pode ser ignorado;
- Redução da motivação.

Liderança orientada a relações – este estilo é focado em organizar, desenvolver e dar todo suporte aos membros do time. Podemos entender como uma forma participativa que motiva a colaboração com base na criatividade e eficiência de todo time.

Algumas características são:

- Cada membro do time é tratado de forma igual;
- É criado um ambiente de trabalho onde as pessoas queiram estar;
- Aumento de produtividade;
- A equipe dá o apoio que o líder necessita;
- Uma dificuldade encontrada é quando o líder se envolve demais em gestão pessoal e deixa as tarefas do projeto de lado.

Liderança liberal – neste estilo a liderança confia na sua equipe e as deixa livres para trabalharem com suas iniciativas e autogerenciamento.

Algumas características:

- Muito eficiente para equipes responsáveis e proativas;
- Liderança apenas monitora os resultados;
- Não é ideal quando a equipe não possui maturidade e necessita de uma direção.

Liderança transacional – neste estilo temos um fator que traz uma crença muito comum onde as pessoas são motivadas pelas suas recompensas e penalidades.

Algumas características:

- Funciona muito bem quando existe uma hierarquia de comando;
- A autoridade é total do líder;
- O papel da equipe é fazer o que lhe é determinado;
- Se cumpre suas atividades tens recompensas;
- Se fracassa a empresa poderá lhe punir de alguma forma;
- Os membros dos times possuem pouco espaço para melhorar sua satisfação com o trabalho.

Liderança transformacional – é o estilo dos verdadeiros líderes. Um líder transformacional é um exemplo a seguir, motiva a todos que trabalham com ele, é comprometido com a geração de ideias e equipes criativas além de ter percepção do lado humano de cada colaborador.

Algumas características:

- O Líder inspira sua equipe com uma visão compartilhada de resultados;
- Possuem autopoder de comunicação e a promovem nas equipes;
- Trabalham com uma gestão visual, clara e transparente;

Qual o estilo de liderança presente nos seus Times Scrum?

Pode seguir o modelo proposto para monitorar os estilos de liderança em cada colaborador do Time Scrum e dessa forma auxiliar nas melhores estratégias para entregar valor aos clientes.

Estilos de Liderança	Product Owner	Scrum Master	Time de Desenvolvimento
Autocrático			
Democrático			
Orientado a Tarefas			
Relações			
Liberal			
Transacional			
Transformacional			

Na tabela de estilos de liderança entendemos que não é uma tarefa fácil identificar esses estilos em cada membro do Time Scrum, porém esse é um trabalho que a organização com apoio do departamento de recursos humanos e gestão de pessoas terá forte participação, não somente na identificação, mas também no desenvolvimento das lideranças.

Gostaria de deixar no que diz respeito a liderança ágil alguns fatores mencionados por Anthony Robbins no seu livro Poder sem Limites (2017), um verdadeiro best-seller da programação neurolinguística, menciona sete mecanismos de caráter que quando acionados são fundamentais para se alcançar o sucesso, sendo:

- I. **Paixão** – são os motivos que nos fazem acordar todos os dias e encarar todas as barreiras que estão a nossa frente. São os motivos pelos quais buscamos estímulos e nos fazem crescer, é o fator que torna cada líder capaz de realizar grandes transformações. Sem paixão não existe êxito, não existe recompensas.
- II. **Crença** – tudo que acreditamos de verdade que é possível de se realizar irá se realizar. Nossas crenças definiram nosso passado, guiam nosso presente e projetam nosso futuro.
- III. **Estratégia** – cada objetivo que possuímos em nossa vida pessoal ou profissional precisa de meios para que sejam alcançados, dessa forma traçamos estratégias que constroem o caminho que devemos percorrer para atingir os resultados.

- IV. **Clareza dos Valores** – devemos saber o que fazemos e o por que fazemos, valores são julgamentos que fazemos sobre nossa vida que dão a ela sentido e dignidade. Todos possuímos valores ou como podemos dizer um sistema de crenças do que é certo ou errado em nossos atos durante toda a vida.
- V. **Energia** – de nada adianta paixão, crenças, valores e estratégia se não possuímos a vitalidade suficiente para agarrar cada oportunidade que aparece e realizar uma construção para bons resultados. É necessária energia física, intelectual e espiritual que possibilite obter tudo o que queremos e precisamos.
- VI. **Poder da união** – devemos formar relações forte e duradouras com os outros, essa capacidade avassaladora de unir-se as pessoas de forma que nossas vidas tenham sentido, sozinhos nenhuma vitória será boa, será vazia e sem cor.
- VII. **Domínio da Comunicação** – Robbins traz uma definição fantástica quando fala da comunicação, ele diz que é o que faz um grande pai, um grande artista, um grande político ou grande professor. Através do domínio da comunicação podemos construir novos caminhos e compartilhar novas visões que irão fazer diferença em nossa vida.

Agora pergunto para, como está a combinação dos sete mecanismos em sua vida pessoal e nos times em que trabalha?

Sempre enxerguei a liderança como alguém com uma lanterna iluminando um caminho escuro para que sua equipe possa seguir sem cair ou trombar com os obstáculos. Porém durante a vida profissional nem sempre encontramos esse tipo de liderança e assim invés de aprendermos com os exemplos, iremos aprender nas quedas e nos obstáculos, pois através delas teremos maior compreensão do caminho que estamos percorrendo.

Num Time Scrum é fundamental que cada papel tenha suas características de liderança, isso será utilizado em diferentes momentos dos projetos. Pensar que um Time irá depender sempre de um Scrum Master ou de um Product Owner é errado, pois o que torna as metodologias ágeis e nesse caso particularmente o

Scrum um framework altamente poderoso é a centralização nas pessoas, elas são a razão e condição para produzir produtos que agreguem valor ao cliente.

REFERÊNCIAS BIBLIOGRÁFICAS

COHN, M. **Agile Estimating and Planning**. Upper Saddle River, NJ: Person Education, 2005;

COHN, M. **User Stories Applied: for agile software development**. Upper Saddle River, NJ: Pearson Education, 2004;

SCHWABER, K. **Agile Project Management with Scrum**. Remont, WA: Microsoft Press, 2004;

MANIFESTO ÁGIL. Disponível em: <<http://www.agilemanifesto.org/iso/ptbr/>>;

MASSARI, V. **Gerenciamento ágil de projetos**. Rio de Janeiro: Brasport, 2014;

SCHWABER, K.; SUTHERLAND, J. **The Scrum Guide - The definitive guide to Scrum: the rules of the game**. Disponível em: <<http://www.scrumguides.org>>;

RUBIN, KENNETH.S, **Scrum Essencial: um guia prático para o mais popular processo ágil**. Rio de Janeiro: Alta books, 2017;

OSTERWALDER, Alexander; PIGNEU, Yves. **Business model generation: a handbook for visionaries, game changers, and challengers**. New Jersey: John Wiley and Sons, 2010;



Luiz Fernando Nunes é especialista em gestão de projetos, business agility, atua na área de TI, inovação e negócios há 15 anos e possui experiência em gestão de projetos e negócios, tendo atuado em grandes empresas de vários portes e segmentos. Possui várias certificações na área de gestão de projetos, comportamento e negócios. É Professor dos cursos de Pós Graduação de diversas Faculdades.

