# Agile Avançado: Crie modelos e descubra o Nexus

# Cultura e Campo

## Por que aprender sobre Lean?

Seja bem-vindo a mais um vídeo do nosso treinamento, segundo vídeo do nosso curso, e eu quero começar falando nesse vídeo um pouco mais sobre filosofia, cultura e campo. “Mas Fred, por que isso é tão importante?” Porque se você não sabe onde você está, você não sabe onde você pode chegar e eu quero te apresentar agora visão holística: é você conseguir entender exatamente onde você está em relação ao gerenciamento de projetos, quem é você.

Só que para fazer isso, eu vou ter que te apresentar conceitos que talvez você ainda não conhece e, se conhece, vamos consolidar o seu conhecimento e vai mergulhar a partir desse vídeo em uma nova viagem aqui dentro da Alura quando o assunto é o gerenciamento de projetos.

Publicações recentes, em especial atualizações de guias e também de padrões, quebraram alguns paradigmas. Eu estive em algumas palestras e pude participar de alguns eventos relacionados à atualização do guia PMBOK, que é o guia patrocinado pelo PMI, o Project Management Institute.

E esse guia, ele traz basicamente o resumo das práticas melhor consolidadas no mercado gerencial. E essas práticas agora, e reconhece por meio da atualização mais recente desse guia, que deve-se, dentro de cada projeto e cada plano de gerenciamento de projeto, ter ali um caminho viável, e deve-se considerar ou levar-se em conta práticas ágeis.

Práticas ágeis e ainda, se necessário, outros modelos e metodologias. Então o que vimos, com a atualização que foi lançada em 2017, é que hoje o gerenciamento de projetos, ele não está mais representado por uma única instituição.

O gerenciamento de projetos hoje está muito maior do que qualquer grupo ou entidade possa conclamar ou entender-se como representante do campo gerencial. Isso é muito importante, esse entendimento de que o gerenciamento de projetos hoje não tem um representante exclusivo no mundo, mas sim ele é um campo, e ele é absolutamente enorme, e ele compreende diversas práticas e modelos, vai ajudar você a entender o que faz um gerente de projetos, e mais do que isso, como se tornar um gerente ágil de projetos.

E começamos essa jornada pelo Lean. Mas antes de eu falar do Lean, eu quero te trazer um conceito japonês, que é o Nemawashi. O que é Nemawashi? Por que eu estou aprendendo isso agora? Porque o Nemawashi faz parte de uma cultura que é responsável, em parte, pela grande difusão do mindset da filosofia ágil. Então o Nemawashi é a prática de plantar em coletividade uma árvore, preparar uma árvore e o terreno onde ela vai ser plantada, para que ela possa se desenvolver bem. E esse conceito nos leva à colaboração.

O Japão, onde surgiu essa prática, ele é um lugar onde é difícil a agricultura, onde ocorrem desastres naturais, onde a vida precisa ser colaborativa. Se olharmos no mapa, o Japão é uma ilha, pequena, o Brasil é muito maior, e ainda assim lá vivem mais de 120 milhões de pessoas. E o Japão, ele é um país onde a cultura, o que é a cultura? É uma série de ações, de práticas repetidas, que viram tradição, que logo se transforma em cultura.

Então esse conjunto de tradições, práticas, a cultura japonesa é muito colaborativa, e o Japão nos brindou com Kaizen, que é uma filosofia de melhoria contínua. Mas eu tenho o Nemawashi, que é um conceito que remete muito ao japonês e à sua cultura, à forma de trabalhar e colaborar. Eu tenho também uma filosofia, e da qual decorrem práticas, que é o Kaizen, de melhoria contínua, ou seja, eu vou buscar sempre melhorar aquilo que eu estou fazendo, em colaboração com o meu grupo.

Então eu tenho o Japão e a sua vida cotidiana e as suas tradições. Eu tenho uma prática já absorvida na indústria japonesa que é o Kaizen, e que também representa uma filosofia industrial, uma forma de abordar não só a indústria, mas o trabalho. E o que acontece então quando a segunda guerra eclode? Os Estados Unidos começam a desenvolver um padrão de capacitação industrial, e essa capacitação voltada para práticas dentro da indústria para a produção bélica, que é o Training Within Industry ou o TWI.

Isso surge nos Estados Unidos em 1940 e, com o fim da guerra, essas práticas consolidadas são levadas ao Japão, no pós-guerra, para a reestruturação do Japão. Então o que acontece nesse momento? O ocidente e o oriente se encontram nas práticas industriais consolidadas por meio do TWI, se encontram com a cultura japonesa e a partir daí nós temos esse desenvolvimento maravilhoso do Lean.

Que inicialmente foi publicado por um autor ainda antes do livro A Máquina Que Mudou O Mundo, que é um estudo sobre a Toyota e como que ela empregou não só o Lean, mas, acima de tudo, como que ela empregou essas práticas colaborativas em consonância com ainda boas práticas industriais e conseguiu conquistar o mundo com os seus automóveis. Então A Máquina Que Mudou O Mundo é onde encontramos pela segunda vez a expressão Lean sendo apresentada, e o Lean significa ser enxuto.

Então eu falei para você sobre o Nemawashi, que na verdade é uma prática exclusiva, a tradução do Nemawashi em si não é específica, mas ela representa uma prática e a partir da representatividade do Nemawashi temos uma visão da cultura japonesa.

Nós também temos o Kaizen, que é a melhoria contínua, que são práticas japonesas para o desenvolvimento de qualquer serviço ou linha de produção. E nós temos então o choque entre as práticas japonesas e as práticas ocidentais industriais, principalmente aquelas consolidadas nos Estados Unidos. A partir desse encontro surge então o Lean, só que ele foi nomeado assim só muitos anos depois, já com a grande ascensão da Toyota como montadora de veículos.

Então em 1986, fazendo um estudo, John Krafcik nos brinda com essa visão. Ele olha para o presente momento, ele entende que algo ocorreu na indústria de produção de veículos e cunha o termo Lean Production System, depois Lean Manufacturing, que é essa união de tradição japonesa, práticas industriais e ainda algumas outras filosofias.

## Lean e Agile

Seja muito bem-vindo ao nosso terceiro vídeo da primeira aula. Então essa primeira aula está sendo dedicada a dar à você a capacidade de entender onde você se encontra e qual é a origem de tudo que nós estamos vendo hoje no mercado. Agora, pode ser polêmico também esse vídeo, o último vídeo, porque querendo ou não, eu estou te trazendo uma visão de campo, ou seja, eu estou agora começando uma trajetória, em especial nesse vídeo, que vai culminar em um novo paradigma. Talvez até possamos discutir uma filosofia.

Mas, afinal de contas, por que você deve aprender a diferença entre filosofia, cultura, tradição, práticas, campo? Porque só por meio dessa diferenciação você vai conseguir entender o seu próprio papel como integrador. O gerente de projeto é um integrador, independente do framework com o qual ele opere. Ele deve ser capaz de combinar não só práticas, mas também de entender o mundo por diferentes filosofias.

Mas afinal de contas, o que é uma filosofia? Eu acabei de tirar o meu óculos. Por que eu fiz isso? Imagina que os meus óculos têm uma lente colorida. Essa lente aqui é transparente, ela não muda as cores que eu enxergo, mas se esses fossem óculos com uma lente cor-de-rosa e eu colocasse eles, eu ia ver o mundo cor-de-rosa. Então uma filosofia, dentro daquilo que nós estamos estudando, é mais ou menos como esses óculos que eu coloco e eu passo a ver o mundo com uma outra cor.

O que eu quero é que você deixe de adotar uma única filosofia ou de usar um único óculos ou um único chapéu. Você precisa saber utilizar múltiplos óculos, você precisa saber entender o mundo sobre múltiplas perspectivas, isso é ter visão holística, isso é ser capaz também de entender diferencie organizações. E o que eu vou te apresentar aqui é um desses óculos, é uma dessas perspectivas do mundo, a partir do Lean como filosofia, porque também já fizemos a diferenciação de tradição e cultura.

A cultura não é nada mais do que um conjunto de práticas que foram sendo repetidas por um grupo de pessoas, um povo, uma tribo, enfim, e que se constituíram em um pilar daquelas sociedades, essas práticas em conjuntos, repetidas ao longo do tempo, se tornaram então tradição e posteriormente cultura. O Lean, agora a partir do topo da pirâmide, pode ser entendido como uma filosofia, mas ele também pode ser entendido como um conjunto de práticas.

No último vídeo, eu falei sobre Kaizen. Kaizen segue a mesma linha do Lean, eu posso olhar para o Kaizen como filosofia, ou seja, eu posso pegar só as orientações gerais, a visão geral do Kaizen, e posso olhar para o mundo a partir desta perspectiva. Mas eu também posso ir mais abaixo nessa pirâmide, e entender as práticas relacionadas a essa filosofia. Então o que estamos fazendo aqui? Estamos adotando o Lean como filosofia.

O que é Lean? Lean é o nome dado inicialmente ao conjunto de práticas identificadas por autores, no fim dos anos 80 e começo dos anos 90, a aquilo que eles entenderam como uma combinação e como a evolução da Toyota, como indústria de construção de veículos, em detrimento da combinação de práticas ocidentais e filosofia oriental, e ainda práticas também orientais de trabalho. Então o Lean surge no mercado como conceito lá no início dos anos 90, fim dos anos 80, a partir das publicações que eu te apresentei no vídeo anterior.

Mas agora, ele se consolida e, principalmente dentro do campo do desenvolvimento de software, que é um campo distinto do gerenciamento de projetos, a partir do meio dos anos 90, para o começo 2000. Mas como isso acontece? Por meio da proliferação de um conjunto de práticas no mercado de desenvolvimento de software que nós conhecemos hoje como Agile. Então o Agile bebeu na fonte do Lean, porque a indústria, tanto da produção de veículos, serviços, operações, como a própria engenharia, elas tinham os seus próprios arcabouços.

Mas como o campo do desenvolvimento de software era um campo relativamente novo, ele acabou absorvendo práticas oriundas do gerenciamento de projetos da engenharia e isso causou um grande conflito, porque os problemas enfrentados pelos engenheiros de software eram diferentes daqueles enfrentados por engenheiros que trabalhavam com uma construção de pontes, rodovias, ainda construções civis, assim por diante.

Então o campo da engenharia de software puxa um monte de práticas iniciais de outros mercados, de outras corporações, e isso cria um problema, muito dinheiro se perde porque você aplicando um paradigma ou utilizando ainda uma forma de entender o mundo que não é adequada ao seu próprio universo, esse novo universo, você perde alguma coisa.

Então Agile surge como uma tentativa, bebendo na fonte do Lean e dessa revolução que foi o Lean, da cultura japonesa, da colaboratividade e ainda de estudos que demonstraram que na área de serviços, de desenvolvimento software, essas práticas colaborativas eram importantes e davam certo, o Agile então surge por meio do framework Scrum, por meio do Extreme Programming e ainda N práticas e metodologias, ainda também frameworks, que se tornaram populares ao longo dos últimos 20, 30 anos.

Então o Agile aqui eu estou representando como cultura. Por que como cultura e não como um mindset, como filosofia? Porque o Agile se alimentou em outras fontes e eu, aqui você pode até discordar de mim, se você discordar, entra lá no fórum e publica a sua discordância. Mas aqui não podemos entender o Agile como uma força filosófica, única. Porque ele se alimenta de conceitos já existentes, para que então se consolide essa cultura, sedimentada em frameworks e metodologias.

Então temos em 1991 a publicação do livro Lean: A Máquina Que Mudou O Mundo, analisando o sistema Toyota de produção e também essas questões culturais integradas. Tivemos em 1986, casualmente não só a publicação do artigo inicial que nos apresenta o conceito de Lean, mas também a publicação de um artigo científico que nos fala sobre a nova forma de se desenvolver produtos, que era a base do Scrum.

Olha só que interessante: no mesmo ano, temos o surgimento do Lean e das práticas e dessa análise contextual, e tem também o surgimento do artigo científico que depois vai ser a base sobre a qual vai ser desenvolvido o Scrum, hoje tão popular.

Ainda tivemos, nos anos 90, uma série de eventos, não estão conectados, mas que ocorreram dentro dessa mudança de práticas e de abordagem gerencial no desenvolvimento de software, que eclodiram posteriormente na elaboração de um manifesto, que é o manifesto ágil de desenvolvimento de software.

Mas novamente, eu aqui estou puxando esse fio da meada, começando no Lean, como essas práticas, essa colaboração em conjunto, essas filosofias orientais, e colocando então o Agile como cultura sedimentada em frameworks e metodologias, e desses vão derivar a prática. Então, olha só que interessante: eu coloquei aqui uma pirâmide, porque eu poderia assim criar esses diferentes níveis, dentro da nossa pirâmide, até chegar ao topo, mas eu poderia muito bem inverter a pirâmide e começar falando de práticas.

Então tem as práticas como a base da minha pirâmide, elas são resultado de uma série de metodologias, que vão culminar em uma cultura, que é a cultura ágil, mas que vão se inspirar em filosofias que vão além de si mesma, e por isso o Agile aqui é colocado em segundo lugar na nossa pirâmide, como cultura e não como filosofia.

Agora, isso é só o começo, porque se nós entrarmos, por exemplo, nos frameworks, eu posso falar sobre esse Extreme Programming, eu posso falar sobre Feature Driven Design, eu posso falar sobre Scrum, eu posso falar ainda sobre N modelos e metodologias que seguem essa cultura.

E eu posso também quebrar isso em práticas. O que eu tenho no Scrum? Eu tenho uma reunião de revisão, eu tenho uma reunião de retrospectiva. O que eu tenho no XP? Eu tenho desenvolvimento em par. O que eu tenho no Feature Driven Design? Desenvolver por funcionalidade.

Então quando eu chego nesse ponto e nesse entendimento, é que eu me transformo em um gerente ágil de projetos e, mais do que isso, combinando esses conceitos que eu acabo de apresentar, desde a filosofia até as práticas, o entendimento de que Agile é uma cultura, só que é uma cultura dentro de um campo.

Então, Agile é cultura. Mas o que isso significa? Significa que o gerenciamento ágil de projetos é um conceito, mas é um conceito mercadológico, e por isso que esse curso vai te ajudar a se adequar ao mercado, porque você vai aprender a falar a língua do mercado.

Mas diferente de outras pessoas que simplesmente repetem uma ideia e não refletem sobre aquela ideia, comigo você está tendo a oportunidade de refletir. Você é obrigado a concordar com tudo que eu estou apresentando nesse vídeo? Não, você não é obrigado, mas eu fico feliz que você já esteja refletindo sobre o assunto.

Então a cultura ágil surge no campo do desenvolvimento de software, mas com uma interface com o campo do gerenciamento de projetos. Você se lembra do Manifesto Ágil? O Manifesto Ágil nos fala sobre o que nós devemos buscar, valorizar, e o que nós devemos colocar em segundo plano, ou ainda manter nos nossos projetos, mas não com a relevância daquilo que nós damos prioridade.

Por exemplo, valorizar a colaboração entre clientes e fornecedores, colocar aquilo que é valor em primeiro lugar, entregar antes versus trabalhar em uma documentação abrangente, enfim, você pode ver o nosso curso de Introdução aos Métodos Ágeis se quer lembrar do Agile Manifesto e também pode fazer a minha carreira aqui, na Alura, sobre o tema, a carreira gerenciamento ágil de projetos.

Mas o Manifesto Ágil surge lá no começo dos anos 2000, em 2001, e ele consolida essa nova cultura dentro do desenvolvimento de software, em relação ao gerenciamento de projetos, ou seja, ele não vem suplantar o gerenciamento de projetos ou colocar uma nova roupagem ao campo da gestão, apesar de que sim, ele dilatou o campo, mas ele não substitui o campo, ele não substitui outras práticas. A demonização do tradicional ocorre em detrimento do desconhecimento de outros paradigmas.

Porque o gerenciamento de projetos se aplica a qualquer campo do saber, não é só o desenvolvimento de software, pode ser aplicado por exemplo na engenharia, à farmácia, à química, enfim, à múltiplas áreas. Então você limitar o campo do gerenciamento de projetos ao campo da engenharia de software é não entender o que é gerenciamento de projetos. O que eu quero fazer aqui é abrir a sua mente e trazer você para um andar acima, eu quero que você consiga olhar para esse campo e é o que vamos fazer nos próximos slides.

Então o Scrum é um framework e ele possui as suas próprias práticas e regras. Sobre isso já falamos e temos cursos aqui na Alura só sobre o Scrumming, inclusive cursos práticos, que estão sendo lançados em concomitância com esse treinamento avançado. E eu gostaria de encerrar já o nosso vídeo, caminhando para o fim dessa apresentação, com um slide que representa o campo da gestão.

Eu tenho o Agile como cultura e eu coloquei até Waterfall, como se ele fosse um equivalente da cultura ágil. Mas a questão aqui é que a visão tradicional, se você quer assim chamá-la, ou as práticas de desenvolvimento faseado, onde eu vou ter cada uma das etapas do meu projeto ocorrendo não de forma incremental, mas isolada, faseada mesmo.

Eu posso também considerar isso como uma cultura gerencial que vai decorrer uma série de práticas e também metodologias específicas. Poderia chamar isso de tradicional e, se eu fizer isso, eu tenho que reconhecer que também faz parte do campo da gestão.

Nos próximos vídeos você vai entender porque que o Waterfall não pode ser demonizado, ele é um conjunto de práticas que pode atender muito bem uma série de projetos e necessidades do mercado, ainda hoje. E olha só o Lean, como filosofia, fazendo uma interface com a gestão. Poderia também representar aqui a interface do Agile com outro campo, que é o campo do desenvolvimento de software, aqui do Waterfall com a engenharia, por exemplo, civil.

Enfim, esse slide é muito representativo daquilo que se busca com esse treinamento, que é oferecer a você essa visão holística, do andar de cima, conseguir olhar para baixo e entender o campo. E você, como gerente ágil de projetos, vai a partir de treinamento, ter também os instrumentos necessários para começar a criar as suas próprias metodologias.

Você vai criar o seu modelo, essa é a proposta do curso: crie modelos. Você vai combinar práticas, você vai combinar frameworks, você vai criar a sua própria abordagem de acordo com a identificação de necessidades na organização onde você está atuando. Se você for um consultor, você vai ajudar empresas a criarem metodologias, a determinarem uma forma de entregar e realizar o trabalho as quais se propõe, essas organizações, da melhor forma possível.

# Scrum Combinado

## Scrum e Waterfall

Considerando o Waterfall como uma metodologia e o Scrum como framework, eu vou te apresentar então uma forma de integrar essas duas abordagens distintas e aparentemente conflitivas de gerenciamento de projetos e desenvolvimento de produtos. E mais do que isso, por que eu vou combinar Scrum com Waterfall? A grande verdade, a resposta para essa pergunta, é simples: nem tudo é Agile.

Talvez na organização onde você trabalhe, você tenha uma equipe que trabalhe com Cascata e outra equipe que trabalhe com Scrum. Você precisa entender que existem formas de combinar ambas as práticas e também abordagens. Você pode sim fazer essa combinação, agora, como você vai fazer é que vai determinar o sucesso dessa integração.

Será que estamos voltando para o passado? Vamos falar sobre práticas ágeis e metodologias ágeis novamente, mas vale a pena você assistir esse vídeo para ter uma pegada diferente e entender um pouco mais sobre como conciliar ao invés de separar ou criar uma rixa entre abordagens que podem ser complementares.

Então, vamos falar sobre como combinar as práticas? Pode acontecer, numa organização, de haverem várias equipes - aliás, isso pode acontecer não, esse é o padrão. Existem múltiplos times trabalhando em objetivos distintos e às vezes trabalhando no mesmo objetivo. Agora, se você está em uma equipe que trabalha com Scrum, dentro da mesma faixa de tempo, você, nessa equipe específica, trabalhar com Cascata é impossível, ou você trabalha com Scrum ou trabalha com Cascata.

Agora pode acontecer de, enquanto você, dentro de um projeto, está construindo uma porção daqueles entregáveis, você pode utilizar Scrum, enquanto um outro time, que está trabalhando em uma outra porção do produto, pode trabalhar com Cascata. “Nossa, Fred, como?” Talvez você esteja trabalhando em uma solução que vá além do software, talvez você esteja trabalhando, por exemplo, na construção de um software as a service.

E então o seu time está desenvolvendo o produto, resolvendo um problema que é complexo, a partir do Scrum. Mas o time de marketing pode estar fazendo um planejamento para o lançamento daquele produto trabalhando com cascata, seguindo uma metodologia própria, faseada. Não há mal nenhum nisso.

Pode acontecer de você receber toda a arquitetura do seu software, todo o planejamento inicial frente ao desenvolvimento, o caso de negócios, por exemplo, pode ter sido construído assim, por meio do Cascata, uma construção faseada, “tradicional”.

Então isso pode ocorrer antes da execução do projeto, ou seja, aquela equipe de análise de negócios, aquela equipe de arquitetura, o time que faz a estruturação dos softwares a serem desenvolvidos na empresa, apesar disso não ser tradicional, no sentido de que se existe no time Scrum a capacidade de construir algo, inclusive a própria arquitetura do software, aquele time vai fazer isso. Mas pode acontecer de você receber um projeto pronto para executar e utilizar o Scrum na execução daquele projeto.

Você reconhece a necessidade de empregar o framework para a execução, que não foi reconhecida por um time anterior ao seu, não tem problema. Ou você recebe, por exemplo, um caso de negócio, o nome do produto, que justifica a existência daquele projeto, então o projeto - inclusive esse esforço pode ser considerado como parte do projeto - já vem acontecendo e entregas foram feitas utilizando o Cascata, antes da execução.

Você pode também utilizar o Scrum como abordagem de execução, ou seja, eu vou trabalhar com ciclos de desenvolvimento, um backlog, mas o planejamento daquele projeto, anterior a sua execução, como eu disse, ele pode ter sido feita de maneira distinta e você pode até absorver o quadro, por exemplo, o Scrum Board, pode absorver práticas, e aí estamos falando já não no Scrum puro, mas de práticas retiradas do Scrum, para organizar a gestão do projeto.

Porque ter um quadro cheio de post-its não significa que você é Agile, não significa que o seu projeto está sendo gerido a partir de um framework específico, ou a partir do mindset da cultura ágil. Ou ainda no final: agora fizemos aqui os nossos incrementos combinados, são releases que foram entregues, mas não testamos nada ainda. Construímos a solução, mas ainda não homologou, está tudo em teste. E agora? Vamos botar um Cascata e vai funcionar.

Sinceramente, isso aqui já não é o ideal, você tem que saber trabalhar com uma entrega e com uma dívida técnica integrada ao seu backlog, ao seu esforço de desenvolvimento, às suas Sprints, ao seu release planning. Antes e depois, aí não existe um Cascata para planejar o desenvolvimento, um Cascata na entrega, obviamente o que fica no meio também é Cascata, não tem nem como tem como discutir, não acredite nisso.

Então você já sabe que é possível combinar times que trabalham com Waterfall, Cascata traduzido do inglês, e times que trabalham com Scrum. O mesmo time, no mesmo momento, não dá, a não ser que seja uma metodologia própria, por exemplo, aí não é Scrum e nem Cascata, e talvez não seja nem Agile. Você pode tirar alguns elementos do Waterfall, ou Cascata, e colocar junto de práticas do Scrum, e criar algo novo, não existe uma lei dizendo que você não pode fazer isso.

“Frederico, mas o Manifesto Ágil diz isso, isso e aquilo.” Mas o Manifesto Ágil não é a lei. Você, como gerente de projetos ágil ou ainda como executivo de projetos, que é o que eu quero que você comece a pensar sobre, não você como um Scrum Master, P.O., gerente de projetos, nem gerente de projetos ágil, para além do gerente ágil de projetos; você como gerente executivo de projetos. Você pode ser autor das suas ideias.

Então vimos aqui nesse vídeo uma apresentação sobre sugestões de combinação e até mesmo de aceitação do Waterfall, ou método Cascata, em relação à abordagem com Cascata, ou colocamos ela então em contraponto com o Scrum.

## Lean e Scrum

Seja muito bem-vindo à nossa segunda aula e ao nosso segundo vídeo dessa segunda aula, onde vamos falar sobre o Scrum combinado com o Lean. “Mas espera aí, Frederico, como assim? Como é que eu vou combinar o Scrum com Lean? Lean não é filosofia?” Eu utilizei o Lean como filosofia e baseado nas filosofias que são inspiradas ou na filosofia Lean, que tem como forte faz inspiração a cultura japonesa e o Kaizen.

Então eu trouxe para você o Lean como filosofia, só que não se esqueça que o Lean também representa um conjunto de práticas, assim como o próprio Kaizen, que são relacionadas e que vão ser apresentadas aqui de acordo com esse mindset de colaboração. Só que o Lean originalmente introduzido a nós é aquele Lean do sistema Toyota de produção.

Então muito cuidado quando você for falar em Lean, sobre que Lean você está falando? Sobre que Lean eu estou falando? Eu estou apresentando para você o Lean como filosofia, mas também te lembrando de que existe o Lean do sistema Toyota de produção, assim como existe o Lean para startups - que nós já apresentamos aqui no nosso curso - assim como existem ainda outras abordagens.

Agora, o Lean como filosofia foi o fio condutor que nos trouxe até aqui e agora você vai conhecer práticas do Lean e oriundas do Lean, daquele que nós conhecemos lá nas publicações que deram vida a esse conceito, a partir de uma análise de contexto.

E estamos voltando para o presente momento, para o estado da arte, que é uma combinação do Scrum, que nos entrega e nos oferece uma série de regras, artefatos e também eventos para que nós possamos gerir um projeto de desenvolvimento de software de forma ágil em combinação com o Lean. Então o que o Lean nos fala? Agora esse Lean, que é o Lean identificado inicialmente nas publicações originais. Que devemos eliminar o desperdício.

Isso aqui, de novo, super relacionado com a cultura oriental: poucos recursos, você precisa eliminar o desperdício - se a luz está acesa no escritório e não tem ninguém na sala, apague a luz. Esse tipo de conceito que é super simples e que não está arraigado na sociedade brasileira da abundância, mas está arraigado na cultura japonesa.

Ampliar o conhecimento, sempre buscar conhecer e aprender com colegas, aprender de forma geral. E o que fazemos quando estamos trabalhando com Scrum? Estamos eliminando o desperdício.

Mas como que eu faço para eliminar o desperdício utilizando Scrum? Quando você trabalha com uma abordagem ágil, principalmente com ciclo de desenvolvimento de 30 dias, você está, ao invés de fazer um planejamento preditivo de algo que é complexo e que requer que você aprenda sobre aquele desafio antes de poder planejar integralmente, você está ampliando o seu conhecimento ao passo de estar eliminando toda a alocação de recursos desnecessária.

Porque você ainda não sabe, afinal de contas, efetivamente quantos recursos precisam ser empregados para resolver aquele problema específico.

As entregas rápidas ou construir rapidamente algo, isso nós também fazemos com Scrum. A cada iteração, ao fim dela, lembrando os conceitos de iteração e iterativo incremental, que se eu trabalho com ciclo de 30 dias, no começo do meu ciclo eu vou ter uma iteração e no fim do meu ciclo eu vou ter outra, então é um ciclo de desenvolvimento iterativo e incremental porque eu estou entregando uma porção a cada iteração de software funcionando.

E o Lean também nos fala sobre isso. As práticas de Lean, mas aqui também a filosofia, como vemos no Lean quando caímos aqui para o Lean aplicado na prática, o Lean mais do que como filosofia, mas como abordagem completa, gerencial, como que ele nos apresenta um conceito ou conceitos que são super afins com o que vemos no ágil.

E aí eu pergunto para você: quem surgiu antes, o Scrum ou o Lean? O Scrum foi publicado pela primeira vez em 1995. Lean já existe há muito tempo, desde esse choque cultural entre oriente e ocidente. Adiar decisões, o que significa adiar a decisão? É, por exemplo, se eu tenho que comprometer um recurso por um período enorme, será que eu vou fazer isso se eu estou trabalhando a partir do Lean ou com Scrum? Não, eu vou comprometer os recursos por um prazo determinado e não maior do que 30 dias.

Se eu tenho, por exemplo, um fornecedor, eu vou fechar com ele um contrato super grande, amplo, com entregas planejadas ou ainda transferindo para ele um risco, eu vou tomar a decisão de comprometer esses recursos no início de um projeto? Ou eu vou adiar essa decisão o máximo possível, focando em entregas rápidas, focando em aprender conforme eu desenvolvo aquela solução? Então você, com o Agile ou ainda com Scrum, melhor dizendo, consegue adiar decisões, consegue empregar esse conceito também no seu dia a dia.

Fortalecer o time por meio de reuniões, encontros, pelo debate, o trabalho colaborativo, tudo isso está super arraigado no Lean, está arraigado na cultura japonesa que, quando se encontrou com o Training Within Industry se transformou e quando o Training Within Industry se encontrou com o Kaizen também.

Talvez seja essa a melhor definição do nascimento do Lean, que é o TWI se encontrando com a filosofia Kaizen, as práticas de Kaizen, nos trazendo o Lean como nós conhecemos hoje, claro, empregado inicialmente e talvez com maior sucesso em escala global pela Toyota.

Construir qualidade: eu quero entregar algo que tenha valor. Quando falamos em Scrum, qualidade é entregar valor, mas o que é a qualidade? Qualidade é atender as necessidades, mais do que os requisitos do meu cliente ou do meu usuário, quando eu falo em qualidade, é atender às necessidades, é um passo além dos requisitos, não é só ver os requisitos funcionais ou não funcionais, mas é entregar valor, é saber o que é mais importante e isso vem a partir da priorização.

Quando com o Scrum eu priorizo o meu trabalho, o trabalho de desenvolvimento, o Product Owner faz a priorização e repassa essa priorização para o time de desenvolvimento, ele não está pensando só nos requisitos, ele tá fazendo a priorização com base no valor, para entregar antes aquilo que tem mais valor.

E por fim, otimizar o todo, pensar nos processos: Construir qualidade está super relacionado, claro, com o produto, até porque falamos em construir, mas quando falo em otimizar, é olhar para o processo.

E temos dois eventos no Scrum que nos permitem fazer isso, que é a revisão da Sprint, onde vamos verificar o trabalho que foi Pronto e aquilo que não foi Pronto, vai validar o esforço de construção durante a Sprint, aquilo que foi entregue, efetivamente Pronto, de acordo com a definição de Pronto criada pelo time de desenvolvimento.

E também vai poder, na retrospectiva da Sprint, olhar para forma como o time construiu aquele incremento e como ele pode melhorar o seu trabalho, desde os relacionamentos humanos, dentro do grupo, até mesmo os processos de desenvolvimento que foram empregados, metodologias, técnicas, enfim. Então eu sei que às vezes pode ser um pouco difícil de entender o Lean como guarda-chuva, porque vamos ter o Lean Manufacturing, o Lean para startups, Lean como filosofia.

## Scrum e XP

Seja muito bem-vindo a nossa segunda aula e terceiro vídeo da aula de Scrum combinado. E agora eu apresento a vocês uma combinação que foi preconizada há muito tempo, foi inclusive apresentada na forma de material que você encontra gratuitamente na internet e também que eu já te apresentei aqui no nosso treinamento, Scrum e XP por trás das trincheiras.

E essa combinação é obrigatória para quem quer trabalhar com métodos ágeis e também desenvolvimento de software, até porque eu poderia dizer que o Scrum é um filho do XP.

O XP, entre o fim dos anos 80 e começo dos anos 90, na verdade o correto seria dizer até o fim dos anos 90, ele foi a abordagem ágil predominante no mercado. Foi só na virada do século, a partir dos anos 2000, que o Scrum começou a ganhar muito mais fôlego e muito mais espaço no mercado.

Mas o Scrum, ele se alimenta muito no XP. E qual é a diferença essencial e por que que você deve ver esse vídeo? Por que combinar Scrum com XP? Porque o Scrum, ele é um framework, ou seja, um conjunto de regras, artefatos, eventos e papéis que você deve seguir se quiser aplicar Scrum como ele é, framework puro, você deve seguir todas essas regras.

E este framework, o Scrum, ele é voltado para o desenvolvimento de produtos complexos, mas não necessariamente software, e não é exclusivamente para software, então você pode trabalhar com qualquer desafio complexo utilizando o Scrum.

Enquanto o XP, é absolutamente orientado para o desenvolvimento de software. Apesar de nós encontrarmos no XP algumas práticas que remontam a filosofia Lean, e que vamos encontrar também no próprio Scrum. Então o Scrum se apropria de alguns conceitos que surgiram no início dos anos 90 e também que estão presentes do XP para propor a sua própria abordagem, mas aquilo que fica no XP é muito interessante e merece um vídeo.

O Scrum, nós já conhecemos. Temos aqui os eventos como, por exemplo, a Sprint, que é o evento central no Scrum; temos o Sprint Planning; vamos ter também o Daily Scrum, enquanto ocorrer a Sprint, todos os dias em que o desenvolvimento estiver ocorrendo, o time deve se encontrar uma vez por dia; vamos ter a Review do produto; e também a retrospectiva da própria Sprint. Você já conhece o framework, eu não vou apresentá-lo em detalhes.

Eu trago aqui esse slide para que pensemos junto na relação que esse slide e o framework tem com o XP e para você ver a proximidade entre ambos. Temos o conceito de equipe inteira, todo mundo trabalhando no mesmo projeto, colocação, colaboração, são os conceitos que nós buscamos e conseguimos ver no Lean como filosofia. Vamos ter os jogos de planejamento e olha só o Scrum, com seu Sprint Planning, e também se considerarmos o Nexus, como o planejamento do Nexus, o Nexus Planning, ou o Sprint Planning do Nexus.

Então esses jogos de planejamento, no XP, é quando a equipe se reúne para planejar o incremento que vai ser entregue na próxima semana, na próxima iteração, sempre preconizando também ciclos curtos de desenvolvimento. E olha só o que temos no Scrum: Sprint de até 30 dias, o conceito de desenvolvimento iterativo e incremental estava no XP, antes do Scrum absorver esse conceito. As entregas curtas também. Esse ciclo curto de desenvolvimento não é exclusivo do Scrum, já existia antes.

O Scrum então, ele absorve muito das práticas do XP, só que ele também deixa outras práticas de lado, como os testes de usuário: você escrever um teste unitário antes de desenvolver a funcionalidade, ou seja, Test Driven Development. Colocar o seu produto em primeiro lugar, trabalhar com entregas curtas, ter um planejamento voltado para o iterativo incremental, ou seja, trabalhar com incrementos. E mais, aqui estamos vendo um direcionamento já para o desenvolvimento e isso nós não temos no Scrum.

A padronização do código, a coletividade, o grupo, a equipe, determinar os padrões que deve seguir e também o mantenimento de um ritmo sustentável, fala assim 40 horas de trabalho por semana, Sustainable Pace. O que seria isso? Seria respeitar os limites do time e não ultrapassar esses limites para garantir que o time consiga entregar continuamente.

Então, olha só como nós temos conceitos interessantes e que vão além daquela visão, já um pouco batida, mas que parece que no Brasil nós gostamos de manter, que é a supervalorização do trabalho ou a supervalorização do herói.

Aqui não. O time responde coletivamente pelo trabalho, o time busca e a empresa deve sustentar essas práticas, o ritmo sustentável, a padronização do código. A metáfora também é algo que até poderia ter no Scrum e que nós consideramos importante, até mesmo aqui na Alura. O que seria a metáfora? A metáfora é saber e criar um glossário comum para equipe, para que todos consigam se comunicar. Isso vai desde a comunicação mais simples até a comunicação escrita em forma de código.

Apesar de que você ainda tem outras práticas que deve considerar no desenvolvimento do seu software, como a integração contínua. A metáfora relacionada mais à comunicação entre as pessoas. E ainda vamos ver alguns elementos que estão relacionados ao desenvolvimento propriamente dito. Integração contínua está relacionado, de novo, em entregar uma porção de software, um incremento, e permitir que esse incremento se conecte ainda a outros incrementos.

Eu vou te apresentar, logo em seguida, o Nexus e você vai ver que o Extreme Programming, ele já traz em si um conjunto de práticas que permite a você dar escala para o seu trabalho de desenvolvimento, e isso fica faltando no Scrum, até pela sua ambição de ser mais do que uma metodologia para desenvolvimento de software, ser um framework para o desenvolvimento de qualquer tipo de produto.

Então ele acaba ignorando certas questões e vemos agora, e como eu vou te apresentar em um framework para Scrum escalado, que é o Nexus, questões relacionadas principalmente à integração de software de forma contínua. A programação em par para reforçar não só o entendimento que cada um tem do trabalho do outro, mas também a própria metáfora, o próprio glossário. E o que seria a programação em par? É isso mesmo: duas pessoas ocupando a mesma máquina.

Então se temos que comparar o Scrum com o XP, eu poderia dizer que o Scrum eu vou usar para organizar o esforço do meu projeto, e o XP eu vou utilizar para organizar o trabalho de desenvolvimento. Então o Scrum é uma ocupação do Scrum Master, agora o XP, ele pode ser empregado pelo time de desenvolvimento livremente na execução do projeto, a integração contínua, a programação em par.

A propriedade coletiva, o que significa isso? Ninguém precisa pedir permissão para ninguém para alterar o código fonte. Isso é interessante, isso desconstrói a noção de hierarquia no gerenciamento do produto e até nos faz pensar que não temos o Product Owner nomeado oficialmente.

Até que ponto essa liberdade pode ser traduzida no nosso projeto quando estamos rodando com Scrum? Como é que fica a questão da priorização? Como é que ficam certas liberdades que o Extreme Programming oferece ao time de desenvolvimento frente a propriedade única e exclusiva do dono do produto, frente ao backlog? O desenvolvimento orientado a teste, que eu já comentei aqui: escrever o teste antes de começar a programar.

E a refatoração. O que é refatoração? É retomar o desenvolvimento feito, reescrever, sempre buscando manter em linha os padrões determinados e refinando esses padrões, incrementando e melhorando também as entregas. Até porque conforme vamos construindo um incremento, se torna mais difícil garantir que a integração contínua seja mantida, a não ser que refinemos os nossos próprios padrões e permita que o time reescreva, refatore.

E o conceito que surge no Scrum e que está relacionada à refatoração, é o da dívida técnica. Podemos considerar a refatoração e a necessidade de refatorar frente a evolução do projeto como uma dívida técnica. Você se lembra desse conceito? Quando entregamos algo muito rapidamente, acabamos deixando algo para trás. Uma porção do trabalho fica preterida em relação a outra, pela priorização ou pela necessidade de entregar o que possa ser utilizado pelo cliente. Então a refatoração e a dívida técnica conversam.

E o design simples é manter a simplicidade, objetividade, também está relacionado com valor. A simplicidade de design nos leva a pensar nas necessidades reais do nosso usuário e isso também nós vemos no Scrum por meio da priorização. E eu deixei aqui o design simples para mostrar aqui o que importa é comunicar esse conceito, mais do que uma imagem ou a beleza do meu slide. Eu poderia ter colocado aqui uma imagem bacana.

Se estivéssemos trabalhando com software, esse poderia ser um slide inicial e eu poderia refatorá-lo, eu poderia ampliar os requisitos desse slide, conforme as necessidades do aluno, por exemplo, adicionar uma imagem, mudar a apresentação, e assim por diante. Mas, ficamos por aqui e eu te vejo no próximo vídeo, quando vamos falar de Scrum combinado com Feature Driven Development.

## Scrume FDD

Seja muito bem-vindo ao nosso vídeo de número quatro da segunda aula, e nesse vídeo eu vou te apresentar uma combinação muito próxima do dia a dia do desenvolvedor de software, que é o Scrum combinado com Feature Driven Development. E a minha visão, com essa aula foi, do começo ao fim, te dar exemplos de combinação para que você comece a pensar não só como um desenvolvedor, não só como um Scrum Master, mas como alguém que tem essa visão de campo e consegue integrar diferentes práticas.

Isso aqui é como um motor de arranque, essa aula. Eu quero te preparar para integrar práticas gerenciais, práticas também de desenvolvimento de produtos, para combinar isso tudo e montar aquela abordagem que vai ser utilizada na sua empresa. Verdade é que pouquíssimas exceções, trabalham puramente com Scrum. As empresas costumam adequar, e isso ocorre naturalmente, uma prática, um framework, ao seu dia a dia, ao seu ambiente interno, à sua equipe, aos desafios que encontra.

Então é natural que você, em um curso de Agile avançado, seja instruído e entenda que o seu trabalho como um líder ágil, como um gerente ágil de projetos, como um executivo de projetos, seja primeiro entender as necessidades, depois oferecer uma abordagem e então executar essa abordagem.

E agora, encerrando então essa sequência de aulas e tendo apresentado a nossa visão, eu faço a seguinte pergunta: por que eu devo combinar Scrum com FDD? E espero já ter resolvido essa questão anteriormente, inclusive no nosso vídeo de Extreme Programming.

Scrum é um framework para o desenvolvimento de resolução de problemas complexos, por meio do desenvolvimento de produtos complexos. Enquanto FDD, Extreme Programming, TDD, são metodologias para o desenvolvimento de software que estão relacionados à engenharia de software.

Scrum, eu colocaria dentro do campo da gestão. FDD, TDD e Extreme Programming, enfim, como engenharia de software. Sabemos que Scrum é muito centrado, é muito utilizado hoje no desenvolvimento de software, apesar de poder ser utilizado em outras áreas. Agora, se você quer ser um Scrum Master, um Product Owner proficiente, um gerente ágil de projetos proficiente, precisa entender também de como colocar na prática e como gerir o desenvolvimento de produtos, principalmente se for software.

Se não for software, aí a conversa é diferente, mas se você busca e ambiciona trabalhar com software, quer entrar nesse mercado, já está nesse mercado e quer se aprofundar, tem que ter pelo menos noções relacionadas ao desenvolvimento de software.

E aqui está o bom e velho Scrum que você já conhece e o Scrum, ele tem os seus eventos, papéis, regras. E o que eu vou fazer agora é te apresentar o Feature Driven Development e fazer uma ponte com o Scrum, sempre em tela, indo e vindo, para refletirmos juntos sobre como podemos integrar práticas de FDD e integrá-las com o Scrum.

Então vamos analisar o FDD e depois vai pensar no FDD incluindo ele aqui no fluxo do Scrum. Então vamos considerar agora o Feature Driven Development. O que é Feature Driven Development? O nome é bem claro: é desenvolvimento orientado às funcionalidades. Então o FDD, ele considera cinco etapas: desenvolver o protótipo, listar funcionalidades, planejar por funcionalidade, desenvolver a funcionalidade, e entregar a funcionalidade.

Mas onde que podemos encaixar essas etapas dentro do Scrum? Vamos voltar agora para o Scrum, então. Vamos pensar no desenvolvimento de um protótipo, antes de começar a planejar essa funcionalidade e seu desenvolvimento, temos que ter um protótipo. Quem é que pode prototipar? O time de desenvolvimento pode prototipar, mas também o Product Owner.

E como é que ele pode fazer isso? De muitas formas. Você já trabalhou como Product Owner, dono do produto? Sabe que você tem o contato direto com o cliente, você é a voz do cliente, então é importante que você conheça as necessidades do cliente. Você pode fazer o seguinte: pode usar a técnica de Vison Box, que apresentamos no curso de Scrum e Agile na Prática. A Vision Box é uma forma de apresentar um protótipo e a listagem das funcionalidades.

Desenvolver o protótipo e listar as funcionalidades. Eu tenho aqui duas etapas que eu posso reunir numa única técnica de Agile e trazer ela para o Scrum a partir do Product Owner. Então o Product Owner, ele pode desenvolver prototipagem de tela por exemplo, ele pode até mesmo fazer uma Vision Box, um desenho e compartilhar esse protótipo não funcional com o cliente. Ele fez isso, faz as histórias de usuários, que são as funcionalidades. Listou as histórias de usuário? Você tem aí já um bom backlog.

Agora chegou a hora de tirar esse backlog do papel e transformar ele em um Sprint Backlog, para desenvolver de fato o produto. Então vamos planejar por funcionalidade. O que nos diz o Scrum Guide? Que temos que dirigir 10% do nosso tempo durante todo o Sprint para o Grooming. O que é Grooming? É o aperfeiçoamento das nossas estimativas, é realizar estimativas, se não foram feitas, e refinar o backlog com estimativas a partir da colaboração do time de desenvolvimento.

Então o Grooming vai ser liderado pelo Product Owner e vai contar com o apoio dos time de desenvolvimento. Eu desenvolvi o protótipo e listei funcionalidades - aqui o Product Owner fez o backlog dele, ele mesmo pode ter criado um protótipo. Agora ele tem que planejar por funcionalidade. Podemos aqui, no Sprint Planning, fazer esse planejamento ou no Grooming. No Grooming é o lugar adequado para isso, no Sprint Planning vamos determinar qual é a meta da Sprint e qual é o backlog da Sprint.

E no backlog da Sprint vamos ter só o quê? As funcionalidades priorizadas e já estimadas pelo time de desenvolvimento para cada Sprint. Então eu planejei por funcionalidade e agora, durante a Sprint, eu vou desenvolver por funcionalidade, não tem nada de misterioso na combinação dessas duas práticas.

E olha que temos ao fim da Sprint: um incremento e o que é esse incremento? Se estivermos combinando FDD com Scrum, é uma funcionalidade entregue. E uma funcionalidade que funciona, que pode ser integrada à outras funcionalidades, que pode ser testada, que cumpre quaisquer que tenham sido os critérios de Pronto do nosso time de desenvolvimento.

Enfim, você viu como não é difícil, então vamos recapitular? Temos o desenvolvimento do protótipo, que pode ser feito em conjunto com o time de desenvolvimento ou iniciado pelo P.O. e depois refinado pelo time de desenvolvimento.

Temos a listagem das funcionalidades, que pode ser esse trabalho de P.O., que não é especificado como deve ser feito no Scrum Guide, logo você pode trabalhar com Scrum, e ainda cada papel pode empregar práticas específicas para o melhor exercimento da sua função no time. O P.O. pode utilizar BPMN, que é uma anotação para processos. Podemos utilizar como P.O. ainda práticas como a Vision Box, que eu te apresentei e que você já deve ter visto no nosso curso de Agile na Prática.

Então o P.O., ele desenvolveu o protótipo e listou as funcionalidades. Planejar por funcionalidade requer estimativas, então o time de desenvolvimento colaborou com essas estimativas. Desenvolver por funcionalidade vai ter exigido que tenha sido feito uma priorização pelo Product Owner e que o time tenha comprometido as funcionalidades a partir das histórias de usuário já priorizados por meio de um Sprint Planning.

Fez isso, tem o backlog da Sprint, vai para a Sprint, desenvolve por funcionalidade, ao fim da Sprint, entrega um incremento. Te vejo na próxima aula então quando vamos finalmente conhecer o Nexus, falei um pouco dele, ainda muito por cima, mas nas próximas duas aulas vamos estudar o Nexus, que é Scrum escalado e é também o framework criado pelo Ken Schwaber e pela Scrum.org, e é o framework no qual eu estou investindo a maior parte do meu tempo como profissional ao estudá-lo.

Porque é um framework simples, é um framework ao mesmo tempo complexo, como o Scrum. Mas que não requer licenciamento, você pode adotá-lo livremente, não precisa contratar uma consultoria certificada. Então eu prefiro te ensinar, assim como trabalhamos com o freeware aqui para os nossos cursos de desenvolvimento, ferramentas às quais você não precisa se credenciar ou pagar nada a alguém.

# Introdução ao Nexus

## Por que Nexus?

Seja muito bem-vindo ao nosso treinamento de Agile Avançado. E cobrimos a primeira parte do nosso treinamento com a criação de modelos e abordagens gerenciais, e agora eu vou te introduzir, e te apresentar ao Nexus, que é a cereja do nosso bolo.

Eu sei que muitos de vocês que estão assistindo esse treinamento já trabalham com Scrum, já trabalham com Agile, e já também criaram soluções para abordar projetos que necessitem de múltiplas equipes trabalhando no desenvolvimento de um produto ou de um software.

Então por que vamos aprender o Nexus? Justamente para apresentar, mais do que resolver, uma sugestão de um framework gratuito, proposto pela Scrum.org para o Scrum escalado. Ou seja, você vai aprender agora práticas, regras, artefatos, papéis, relacionados ao gerenciamento do desenvolvimento de um software, a partir de três ou mais equipes.

Então com este modelo - até porque modelo eu utilizo como tradução de framework - este framework vai te ajudar a resolver problemas relacionados à escala no desenvolvimento de software ou de produtos complexos, a partir da utilização do Scrum por meio do Nexus, que é esse framework de Scrum escalado.

Então o principal foco do Nexus é integração. Você já conhece o Scrum, temos apresentado nos nossos vídeos esse slide introdutório, para sempre lembrarmos o que está nos conduzindo aqui, qual é a visão que eu tenho buscado te oferecer.

Existem múltiplas abordagens ágeis no mercado, a mais consolidada é o Scrum. Inclusive tivemos a oportunidade de conversar sobre a ascensão do Scrum versus o XP. Até o fim dos anos 90 tínhamos o Extreme Programming como a forma de desenvolvimento de software ágil mais utilizada no mundo e a partir dos anos 2000 em diante, o Scrum ganha essa predominância.

Você já conhece o Scrum e é muito importante relembramos algumas das questões e algumas das práticas, principalmente as regras, papéis e artefatos do Scrum, além obviamente dos eventos. Nós vamos ter um backlog por produto. E um conceito importante é que enquanto existir um produto, existe um backlog, e mais do que isso, apenas um backlog por produto. Então independente da quantidade de pessoas com as quais você esteja trabalhando, um backlog por produto, não mais.

Vamos ter também a cada Sprint por equipe, um Sprint backlog, que vai refletir aquilo que foi priorizado no backlog do produto. Então o time vai fazer um planejamento, vai determinar uma meta para a Sprint, vai definir o seu backlog da Sprint, e vai executar aquela Sprint até um mês corrido. Tivemos uma atualização recente no Scrum Guide que retirou o limite mínimo da Sprint, então agora temos até um mês no calendário como Timebox da nossa Sprint.

O time vai então se encontrar diariamente, fazer a Daily Scrum, antes de entregar, obviamente, o incremento, fazer uma revisão do produto que foi entregue, junto com o Product Owner, cliente, partes interessadas. O incremento foi desenvolvido, foi entregue, então vai haver uma retrospectiva. E aqui você já vê o Nexus. E por que eu trouxe então o Nexus em conjunto, já encadeado com a nossa visão geral do fluxo do Scrum? Porque o Nexus se baseia fortemente no Scrum, inclusive o Nexus é uma sobreposição ao Scrum.

O Nexus é um framework para que você consiga integrar de 3 a 9 equipes, então com o Scrum você sabe como trabalhar em um único time, até nove integrantes do time de desenvolvimento, considerando o Scrum Master e P.O., podemos chegar até 11 pessoas no total. E limitação de pessoas se dá a partir da necessidade de promover a transparência, a saturação comunicacional, permitir que todos consigam conversar e ser ágeis, isso é muito importante.

O número de indivíduos em um time facilita a agilidade, muitas pessoas trabalhando no mesmo time seria menos ágil porque seria mais complexa a comunicação. Então aqui nós vamos ver o reforço, vamos ver também no Nexus, assim como vemos no Scrum, esse conceito de único Product Backlog. E além do planejamento da Sprint, como vemos no Scrum, e eu vou falar sobre isso ao longo das próximas aulas, vamos ter um planejamento da Sprint do Nexus.

O que é um Nexus? O Nexus é essa relação entre pessoas. Então vai haver um backlog de Sprint para esse conjunto de equipes, vai haver também um planejamento para esse conjunto de equipes, vai haver um desenvolvimento integrado por meio de um novo papel que emerge no Nexus, que é o time de integração, que vai facilitar o trabalho dessas equipes que estarão trabalhando no Scrum, então não vai ser apenas um time, mas múltiplos times, uma revisão de Sprint do Nexus e também uma retrospectiva do Nexus.

E a grande sacada e o grande pano de fundo do Nexus é a necessidade de integração. Só que antes de eu prosseguir nessa apresentação, e antes de nos aprofundarmos no que é o Nexus, nós temos de fazer aqui o reconhecimento, em primeiro lugar, de que o Nexus é um framework, é o Scrum profissional escalado, e ele foi desenvolvido pelo Ken Schwaber e pelo time da Scrum.org.

E é um trabalho primoroso e inclusive hoje existem certificações que você pode tirar na Scrum.org - eu vou deixar mais informações para você nas aulas complementares - e essas certificações relacionadas ao Scrum escalado.

Além disso, é importante reconhecer aqui os tradutores, o pessoal que, em parceria com a Scrum.org, traduziu o framework para o português, que foi o Eduardo Rodrigues e o Fábio Cruz. Então esses dois profissionais, você pode nos contatos que eu apresento aqui. Nosso agradecimento a esses membros da comunidade ágil que fizeram este grande esforço, inclusive Fábio Cruz é um grande nome da comunidade brasileira, já há muitos anos tem contribuído muito com material de qualidade.

E eu falo do Fábio, não conheço o trabalho do Eduardo, mas posso dizer que o que ele fez no Nexus framework, que é a tradução que ambos fizeram, está de altíssima qualidade. Agora sim podemos prosseguir com a justificativa do Nexus e porque ele existe.

Afinal de contas, por que temos que ter um framework para Scrum escalado? Por que eu não posso usar o Scrum simplesmente? Porque o Scrum não vai te permitir uma visão de cima, que é o que nós temos tentado fazer ao longo desse treinamento.

Então para você ter essa visão, você precisa de novos instrumentos, você precisa de novos papéis e, mais do que isso, é muito mais complexo organizar o trabalho de múltiplas equipes do que uma única equipe: novos conflitos, novas necessidades vão surgir a partir da participação de múltiplos times Scrum.

E uma coisa importantíssima para você que está estudando para certificações da Scrum.org é: a prova PSM, Professional Scrum Master, a qual abordamos conceitos lá no nosso curso na carreira Agile, carreira de gerente ágil de projetos, também na carreira agilista, ela sofreu um aumento de complexidade agora. Nós vamos encontrar nas provas PSM questões relacionadas à múltiplas equipes de Scrum, antes não víamos isso nos exames, questões situacionais também.

Então a prova PSM ganhou em muito em complexidade justamente pela ascensão do Nexus, então fica ligado nisso porque isso pode cair nos seus exames de certificação. Agora o backlog do produto deve ter um esforço maior de identificação de dependências. Por que isso? Porque agora não é apenas somente um time trabalhando naquele backlog, nós vamos ter múltiplos times trabalhando nesse backlog.

E como é que eu posso, por meio do backlog, facilitar o trabalho de múltiplas equipes, dependências? Eu tenho que saber quais são as dependências, porque se eu não sei quais são as dependências, eu não sei qual o trabalho de cada equipe vai afetar o trabalho de ainda outra equipe e isso vai me prejudicar muito, eu vou ter uma dívida técnica enorme se eu não tiver identificado essas dependências e distribuído adequadamente o trabalho.

A comunicação entre equipes: sem o Nexus, não vamos ter também, ou eu diria sem o Nexus, mas você pode absorver práticas do Nexus no seu próprio modelo, na sua própria metodologia. Nós não temos somente utilizando o Scrum Guide, um time de integração. Então surgem novos papéis, surgem novas designações, e surge também uma nova visão em relação ao trabalho a partir do Nexus.

E a comunicação entre os times é facilitada por meio desses papéis, a saber o time de integração. Então o time de integração surge no Nexus para fazer com que os times conversem mais e você vai entender melhor esse papel a partir dos próximos vídeos, eu vou explorar em detalhe cada papel, todo o framework.

Testes integrados, e aqui vamos ver um Grooming que vai ocorrer em múltiplas equipes e reuniões que vão ser orientadas pelo time de integração, vamos ver também uma arquitetura sendo patrocinada pelo time de integração, a capacitação dos times sendo feitas por esse time de integração, vamos ver também uma coisa muito interessante que ocorre no Nexus: imagina só, estamos falando de nove equipes, estamos falando de potencialmente até 99 pessoas.

Vamos desconsiderar, na verdade, um papel, que é o do Product Owner. Se eu tenho um backlog, eu tenho somente um Product Owner. Então nós podemos ter 91 pessoas, considerando nove equipes com até dez membros e um Product Owner, então 91 pessoas aí que eu posso ter nesse Nexus.

Então imagina como integrar o trabalho de nove equipes, já parou para pensar que complexo seria isso? Então você tem nove equipes trabalhando e eu tenho que integrar isso tudo.

Novamente, o time de integração vem para nos ajudar, mas temos ainda outros eventos, como a revisão do Nexus, que vai suplantar a revisão de produto do Scrum. Então nós vamos poder, ao longo do nosso trabalho, contar com o time de integração e, na hora de fazer a revisão, também vamos poder olhar para o nosso produto como um todo, como um Nexus.

É realmente muito difícil trabalhar quando temos múltiplas equipes, o projeto se torna ainda mais complexo, ele ganha em complexidade. Se antes tínhamos a complexidade do desafio em si, agora temos uma complexidade gerencial, e o Nexus surge para colaborar e para diminuir a complexidade do gerenciamento do trabalho, enquanto conseguimos nos concentrar na resolução do problema, ou seja, no desenvolvimento do produto.

E conflitos de escopo e requerimentos vai acontecer sempre. Imagina se eu tivesse 99 pessoas de fato, cada time Scrum com o seu Product Owner? Eu ia ter nove backlogs, já pensou o quão conflituoso isso seria? Nove Product Backlogs, nove pessoas podendo dizer o que elas pensam e decidir quanto à priorização e assim por diante? Seria impossível. Então o Nexus também nos ajuda a eliminar esse conflito ao propor apenas um Product Owner.

A organização das equipes também se torna mais fácil a partir do Nexus, esse framework que permite então, e você vai entender melhor isso, de novo, nos próximos vídeos, mas já te dando um gostinho do que está por vir: vamos ver no Nexus uma certa descaracterização dentro do time de integração. E que descaracterização é essa? É a dos papéis.

Você vai ter no time do Nexus um Scrum Master, vai ter um P.O., mas vai ter também membros do time de desenvolvimento que não necessariamente precisam focar no time de desenvolvimento ao qual pertence, eles tem que primeiro trabalhar no time de integração e depois eles podem trabalhar no time de desenvolvimento.

E também não há um limite para o número de pessoas escaladas no time de integração, então isso vai ajudar a organizar melhor as equipes, você não precisa de preocupar com uma regra de, por exemplo, seis, mais ou menos três membros no time de integração. Não, não existe isso. O que acontece é você, como equipe, como Nexus, tem a capacidade de auto organizar esse time de integração sem um limite de recursos.

Então vamos ver que muitas pessoas podem sair dos seus times de Scrum, esses times Scrum que vão estar utilizando framework Scrum também, e nós vamos ter esse time de integração, que é um time que não tem muitas regras e que vai colaborar com essa integração. Então aí temos uma facilitação muito grande por meio do time de integração, de novo, que é o principal papel do Nexus.

## Visão Geral do Nexus

Seja bem-vindo a mais um vídeo, segundo e o último da nossa terceira aula, onde eu vou te dar agora uma visão bem mais específica do Nexus, mas ainda ampla, por que? Porque agora eu vou te apresentar todo o Nexus, todos os seus papéis, eventos, regras, e isso já vai te sanar muitas dúvidas.

Só que ainda não vou entrar no detalhe. Você vai entender agora o Nexus, vai entender melhor os papéis, mas ainda não em profundidade. Depois, na próxima aula, começamos a aprofundar cada um dos papéis para entender melhor qual é, de fato, o valor de cada um desses.

Então por que você quer entendeu fluxo Nexus? Porque você quer aprender a escalar Scrum. Vamos já rever então o fluxo. Eu apresentei o Nexus, no último vídeo, de forma bem passada, para você entender mais do que o Nexus, a justificativa por trás dele. Agora vamos começar a mergulhar no Nexus propriamente dito.

Então nós já sabemos que o Nexus aponta apenas um backlog para o produto. Isso é norma: não existe produto com mais de um backlog e enquanto houver produto, vai haver backlog, independente se ele é administrado por um P.O. ou não.

Então esse conceito é universal, não esqueça disso: se é um produto, se é Agile, independente se for Scrum ou não, é um backlog de produto, e um backlog só tem um P.O. E no Nexus isso aqui é reforçado. E agora já está aparecendo nas provas de certificação da Scrum.org, também em questões de prova de PSPO, Scrum Product Owner, e também Scrum Master, PSM. Então Product Backlog é um planejamento da Sprint do Nexus.

O que vai acontecer aqui? Você, como Product Owner, está trabalhando no backlog, identificando as dependências, tornando elas claras, e também a necessidade ou as necessidades do cliente e dos usuários. Chegou o momento de planejar o incremento que vai ser desenvolvido por todos os times e aqui já não é mais incremento, é incremento integrado.

Então como você vai fazer isso? Vai planejar a Sprint do Nexus, ou seja, vai fazer a priorização e as pessoas necessárias, adequadas a essa organização, não só do time de integração, mas de todos os times, vão fazer parte desse planejamento e depois vai correr o que? O planejamento individual de cada time.

Cada time vai se auto organizar depois dessa primeira parte da reunião do planejamento do Nexus e vai distribuir entre si, um time para o outro, vai se organizar e vai assumir propriedade sobre parte daquele backlog do Sprint do Nexus. Então isso vai ser quebrado em backlogs de Sprint que vão ser executados por meio do Scrum. Lembre-se, o Nexus ele sobrepõe-se ao Scrum. Então ok, foi feito o backlog da Sprint de cada time, agora isso é negociado e encerrado na segunda parte do planejamento da Sprint do Nexus.

Temos agora não só o backlog da Sprint do Nexus, mas cada times Scrum vai ter o seu backlog, uma porção daquele trabalho que representa, obviamente, uma porção do backlog da Sprint do Nexus. Vai ocorrer então a execução do ciclo de desenvolvimento, que vai durar o mesmo mês de calendário e vamos ter o time de integração colaborando ao longo desse processo, do emprego de diversos processos de desenvolvimento de software.

O Nexus surge para sanar problemas majoritariamente do desenvolvimento de software, apesar de que você também pode empregar o Nexus em outros campos, assim como fazíamos com o Scrum inicialmente: ele era focado em software, mas fomos utilizando a abordagem em projetos de natureza distinta até que ele se consolidou como um framework universal.

O time de integração do Nexus vai colaborar na execução do projeto, vamos ter dentro do time um Scrum Master e esse Scrum Master vai ser o Scrum Master do Nexus. Vamos também ter no time de integração um P.O., que vai ser o P.O. de todo o Nexus, além de pessoas que sejam necessárias dentro desse time de integração e essas pessoas podem mudar ao longo do tempo do ciclo de vida daquele projeto.

Então, diariamente, vai ter uma reunião diária em cada equipe e também vai ter uma reunião diária do Nexus, que de novo vai ser promovida pelo time de integração, o Scrum Master vai facilitar essa reunião, e aí é que vai ocorrer a integração entre os times. Nós vamos ter, ao fim do esforço do segundo desenvolvimento, da Sprint, uma revisão da Sprint, um incremento integrado e uma retrospectiva do Nexus. Bem simples: a revisão da Sprint, aliás, nada simples.

A revisão da Sprint do Nexus pode ter lá uma multidão enorme de pessoas, o Nexus não te diz como fazer essa reunião de revisão, você pode até ter aquelas 91 pessoas que já calculamos no vídeo anterior, em uma única revisão. Mas você vai ter que encontrar formas de fazer isso. O framework Nexus não te diz como fazer isso, ele te dá orientações gerais.

E uma retrospectiva que vai acontecer também em três etapas: a primeira retrospectiva do Nexus, depois cada time faz a sua retrospectiva e um retorno para consolidar o conhecimento das equipes dentro do Nexus.

Te dei aqui uma visão geral, agora em mais detalhes. Vamos ter múltiplas equipes e vai ter um incremento integrado. O que isso significa? Que, ao longo do esforço do desenvolvimento do meu software, ao longo, e não no final, eu vou integrar os meus incrementos e todo esse esforço de integração, ele começa no backlog de produto. “Como, Frederico, isso vai ser integrado no backlog?” Porque é no backlog que o P.O. vai identificar as dependências.

Logo em seguida, indo adiante, você deve recordar o nosso fluxo, temos o Product Backlog e um planejamento do Sprint da Nexus, ou do Nexus, e é nessa reunião de planejamento que nós vamos reforçar a integração. Como? Quebrando essa reunião em partes: eu vou ter a primeira etapa dessa reunião, a definição do que vai ser feito e a meta da Sprint do Nexus. Depois disso eu vou pegar e vou quebrar essa meta da Sprint e vou também quebrar o backlog da Sprint do Nexus em múltiplos backlogs, e essa reunião facilita também a integração.

Depois vamos ter a execução, com as reuniões diárias de cada time, também a reunião diária do Nexus, vamos ter o trabalho de Grooming e o trabalho de integração vai prosseguir ao longo do desenvolvimento por meio das reuniões diárias e por meio, principalmente, das reuniões diárias, do estabelecimento de uma estratégia para o trabalho do dia em consonância com aquilo que for definido na reunião diária do Nexus.

Então representantes de cada time podem participar dessa reunião diária do Nexus e depois eles voltam para os seus times e montam a sua estratégia com a sua equipe, facilitada pelo Scrum Master de cada equipe, para que seja dada continuidade ao trabalho e também que seja garantida a integração contínua do que está sendo desenvolvido. Que sejam respeitadas as dependências, que novas dependências sejam não só absorvidas pelo backlog, mas entendidas pelo grupo e assim por diante.

O foco em dependências e interoperação é muito grande. Como que nós vamos fazer o nosso trabalho? Como o nosso trabalho afeta o dos demais grupos? E nós vamos ver perguntas que podem ser feitas, e são recomendadas pelo Nexus, que a partir dessas respostas das perguntas que são propostas você consegue ampliar a capacidade do time de trabalhar no ambiente interoperacional.

E ainda sobre a questão da integração, uma coisa muito importante: não é na reunião de revisão que se faz a integração. Tenho que reforçar isso: não é na reunião de revisão de produto, revisão de Nexus, que nós vamos fazer a integração, é ao longo do desenvolvimento.

O incremento, ele deixa de ser somente um incremento no Nexus, ele passa a ser um incremento integrado, e ele nasce por meio da facilitação proposta pelo Nexus, com apoio do time de integração, vamos ter não só mais um incremento como resultado do fluxo Nexus, vamos ter um incremento integrado.

Então o produto aqui, ou seja, não o produto que entregamos para o cliente, mas o fruto do Nexus, é um incremento integrado. Ou seja, não temos nove incrementos no Nexus, ou conforme for o número de times, nós vamos ter um: um Product Backlog, um incremento integrado. Muita atenção nisso.

Vamos ter papéis e quais são os papéis. Já te apresentei também, e vamos reforçar, o time de integração do Nexus, que vai ser composto pelas pessoas que forem necessárias, de todas as equipes, todos os times Scrum, mais um Scrum Master do Nexus e o P.O. do Nexus.

Vamos ter o P.O., que vai ser um para todo o Nexus, vai ter o Scrum Master do Nexus e vai ter também o Scrum Master em cada equipe. Não vai ter P.O. nas equipes, o time de integração, que surge como esse principal novo papel no Scrum. E vamos ter também artefatos, quais são os artefatos? Um único backlog. Vamos ter um backlog da Sprint do Nexus, vamos ter um backlog de Sprint individual por time, ou seja, o Nexus, ele não desconstrói o Scrum, ele é colocado sobre o Scrum.

Então nós vamos ter o backlog no Nexus, vamos ter ainda o backlog da Sprint do Nexus, e vamos ter cada time, o seu backlog. E quais são os eventos do Nexus? Nós vamos ter o planejamento da Sprint do Nexus, onde vão ser determinadas, acima de tudo, a meta e também o próprio backlog da Sprint do Nexus.

Vamos ter a reunião diária do Nexus, que vai também se sobrepor à reunião diária de cada time. A reunião de revisão da Sprint do Nexus, essa aqui ela elimina a revisão de cada time Scrum e se torna única para todos os times, isso aqui é importante.

Reunião de retrospectiva da Sprint do Nexus, também, ela é um evento que vai permitir que você trate questões relacionadas ao Nexus, faça a reunião diária de cada time, e depois feche esse grande evento, encerre a reunião de retrospectiva somente após ter feito a retrospectiva de cada time, assim se torna mais rica a retrospectiva do Nexus.

# O Nexus

## Papeis do Nexus

Seja muito bem-vindo à nossa quarta aula e nessa aula eu vou falar sobre os papéis do Nexus. Você já viu que o Nexus não tem nada de difícil quando o assunto é conteúdo. O conteúdo está lá, você já deve ter acessado, eu disponibilizei o material em PDF, você pode acessar diretamente na Scrum.org. O Nexus tem 12 páginas - ele não é volumoso.

Mas como framework Scrum nos levanta muitas dúvidas: como é que eu vou fazer, por exemplo, a revisão com 90 pessoas? Isso é uma super dúvida. A própria quebra do planejamento da Sprint do Nexus é uma quebra que não está clara no guia, você tem que considerar que eu estou facilitando inclusive o seu entendimento do Nexus e te dando aqui dicas que estão ligadas à minha prática gerencial, com Agile, ainda outros projetos com outras abordagens.

Então como tudo no Agile, vamos ver principalmente na Scrum.org ou que são oriundos ou adventos da Scrum, orientações gerais, algumas regras e aquela velha máxima: é fácil de ler o material, está tudo ali, está tudo aparentemente simplificado, mas existe uma complexidade enorme no domínio e na aplicação desses frameworks.

E daí surgem consultorias, e cursos, e uma mídia de produtos e serviços ao mercado que essas empresas, parceiros, e ainda outros profissionais oferecem. Aqui inclusive vale uma reflexão: será que, e somente será que, eu não estou dizendo que é ou que não é.

Será que não vemos tanto o framework Scrum como framework Nexus, e assim como outros frameworks de outras metodologias ágeis, se embasarem no conhecimento pré-existente do mercado para permitir essa simplificação? Então será que essa simplificação não é uma máscara? Será que estamos vendo de fato algo que ágil e, mais do que isso, perguntas para sua reflexão: Sendo ágil, será que não é um ágil que se sustenta em conceitos já consolidados no mercado? Porque pode ser ágil e não necessariamente ser resumido.

Pode ter um conhecimento profundo, algo que requisite horas de estudo, um mergulho em práticas ágeis. Você vai ver isso principalmente estudando para certificações como ACP.

Então eu te pergunto isso, reflita sobre isso: E qual é a relevância dos papéis do Nexus? Obviamente toda, sem os papéis do Nexus, não existe Nexus. E você vai entender aqui cada um dos papéis e como que se organizam esses papéis em função do Nexus e do próprio Scrum. Então já que eu já consigo identificar os papéis do Nexus no fluxo do Nexus, no time de integração acima de tudo.

O time de integração vai ter o Scrum Master do Nexus, vamos ter o P.O., Product Owner, dono do produto do Nexus, e vai ter também um grupo de desenvolvedores, que vão fazer parte desse time de integração, sem limite de pessoas por time de integração. Então aí já temos três papéis claros relacionados ao Nexus, e dentro de cada time, também vamos ter o Scrum Master do time de desenvolvimento.

Só que se estamos trabalhando dentro do Nexus, somente um P.O. “Mas agora vamos trabalhar com o Nexus, mas eu já tinha um P.O., eu era P.O. em um time de Scrum”. Ok, você vai ter que trabalhar com o time de desenvolvimento, não vai poder vestir a camiseta de P.O., porque só tem um P.O. por produto e um P.O. por backlog. Então fique atento a isso.

Vamos dar uma navegada dentro do time de integração? Eu fiz um recorte para você do próprio framework, para dar zoom mesmo em como se relaciona o time de integração aos demais times Scrum. O time de integração poderíamos considerar, se você olhar no slide, está vendo essas engrenagens? O trabalho integrado. Olha o trabalho entregado, está vendo aquele triângulo, que tem nas suas pontas as engrenagens? Ótimo, isso é o time de integração.

O time de integração, eu até diria mais do que fora da Sprint, dentro do Nexus obviamente, ele está integrando, está ali no coração do Nexus. Agora óbvio, é uma forma de representação gráfica, eu entendo que essa forma ainda deve mudar ao longo dos próximos anos, o Nexus não é tão antigo quanto o Scrum, na verdade é muito mais recente, temos aí poucos anos de vida do Nexus em comparação ao Scrum.

Mas quem faz essa integração, óbvio, é o time de integração do Nexus, ele que permite a integração ao longo do ciclo de desenvolvimento de até um mês de calendário.

Então quando eu chegar no fim da minha Sprint, eu vou ter lá um incremento integrado. Essa integração tem que ter ocorrido antes. O time de integração é uma combinação de membros do time Scrum. Não há limite de pessoas. “Quantas pessoas eu tenho que ter no meu time de integração?” Quantas forem necessárias. E muito importante, dentro desse time só tem um P.O. e só tem um Scrum Master, não mais do que isso.

E quais são as competências desse time, antes de falarmos de cada papel, do P.O. e do Scrum Master e também dos demais membros? E eu acho que quando eu falo aqui das competências do time, eu já estou falando dos demais membros e do grupo como um todo, isso vai te ajudar a entender depois cada papel. Primeiro lugar, um gerenciamento de um backlog único e aqui, por meio desse backlog único, temos aquele esforço de integração ao qual eu te apresentei nos últimos vídeos.

Um backlog único vai dar uma noção de todo e de dependências dentro desse backlog e, a partir dessas dependências, eu vou consegui um melhor planejamento da Sprint do meu Nexus. Também temos uma orientação bem interessante no Nexus que é de que agora e, em função do Nexus, talvez seja mais interessante que os grupos de trabalho, os times Scrum, façam até mesmo uma reorganização por meio de conhecimento, então vamos ter times mais especializados.

Isso aqui é um ponto para debate: eu tenho times especializados, multifuncionais, trabalhando no meu Nexus. Isso quer dizer que agora eu tenho que trabalhar com times especialistas? Mas está no Nexus Guide e isso vai facilitar o trabalho, as pessoas se auto organizarem em função do seu conhecimento. É um direcionamento que o Scrum Master pode dar.

Já pode ir para o fórum da Alura, vamos discutir isso aí, porque, nós temos um guia aqui que nos diz que as equipes devem se auto organizar e ser multifuncionais; e nós temos um outro guia, que é o Nexus Guide, que eu disse que talvez seja mais interessante organizar os times por meio de conhecimentos específicos que eles possam ter para facilitar o desenvolvimento do incremento integrado do Nexus. Mas isso é passível de debate.

Mapeamento de dependências: como que você vai integrar esse incremento do seu produto, se você não tem as dependências definidas de forma clara? E aí de novo, fica muito mais fácil, obviamente, fazer depois desse mapeamento de dependências, uma distribuição durante o planejamento da Sprint do Nexus para os times que sejam mais capazes de atender cada aspecto específico daquele incremento.

O time de integração, e aqui é o foco dessa apresentação, neste momento, colabora na organização das equipes. O time de integração do Nexus vai ser composto então pelo Scrum Master, pelo P.O. e por um número indefinido de profissionais dos times de desenvolvimento. Mas o papel de P.O., de Scrum Master, é um para cada: um P.O., um Scrum Master, número ilimitado de desenvolvedores.

O que vai acontecer? Esse time não só vai organizar na colaboração da organização das equipes e impedimentos aqui, principalmente na figura do Scrum Master, que vai ser o líder de todos os demais, temos o time aqui, dentro do time integração, ele tem que ser o responsável pelo fortalecimento do conhecimento, pela centralização do conhecimento aqui, mais ainda, a distribuição do conhecimento.

Ele representa o estado da arte do conhecimento do Nexus e de todos os times, ele consolida esse conhecimento, ele transforma este conhecimento em padrões e ele multiplica esses padrões de volta para as equipes. Então ele puxa para si essa responsabilidade de ser o pilar do saber em desenvolvimento de software, padrões, arquitetura e depois ele replica isso para as equipes.

Os impedimentos do Nexus vão ficar mais fáceis de serem identificados porque vamos ter ali um Scrum Master, que vai ter comunicação com outros Scrum Masters, mas também temos membros representantes dos times de desenvolvimento, pode ser líder técnico, pode ser alguém fale pelo grupo, pode ser o próprio Scrum Master daquela equipe. E ele vai estar dentro do time de integração, como “desenvolvedor”, porque lá ele vai poder representar até mais de um time Scrum.

Imagina que temos dois ou três times trabalhando no mesmo conjunto de especificações, requisitos, e esse time pode ter um representante macro, ele pode chegar no time de integração do Nexus e ser a voz desse time, pode ser um Scrum Master dentre 13 Scrum Masters, enfim, a organização fica a critério dos grupos, não é algo prescrito.

Então esses impedimentos e tudo que gire ao redor da comunicação, da organização, pode ser melhor integrado por meio do time de integração do Nexus e a partir também do próprio Nexus.

A garantia desse incremento integrado ocorre porque a integração é facilitada pelo time de integração, então o time de integração trabalha constantemente para facilitar a integração. Pessoal, está escrito no nome: time de integração. Ele trabalha para isso, ele vive para isso.

No guia do Nexus vamos encontrar algo muito importante que é: quem trabalha no time de integração primeiro trabalha no time de integração, não trabalha no seu próprio time Scrum, ou seja, coloca como prioridade a integração. Então temos efetivamente um time integrador.

É que esse papel de mentoria também surge no próprio framework, no guia do Nexus, esse papel de coach, de orientação, esse time é um pilar do conhecimento. Então a absorção e a devolução desse conhecimento em forma de padrões, orientações, metáfora, isso tudo volta para os times de forma organizada. E também é um ponto focal para questões de problemas, conflitos internos, é um fórum para que seja lavada a roupa suja quando necessário.

Além, é claro, da reunião que esse time vai promover, que é a reunião diária do Nexus, e deve também coletar feedbacks, e vamos ver nos próximos vídeos o que compõe essa reunião diária do Nexus. Então o time de integração tem um P.O. - o papel de P.O. não muda, ou seja, o P.O. que trabalhava antes em um time Scrum, vai continuar trabalhando como P.O. no time de integração do Nexus, vai ser um P.O. por produto, só que é um P.O. que precisa ser muito mais experiente, não vai ter P.O. nos times Scrum.

E o Product Owner, eu tirei isso aqui direto do guia do Nexus porque eu achei importante: O Product Owner é responsável por gerenciar o backlog do produto para que o máximo de valor seja derivado através do Incremento Integrado criado pelo Nexus. Como isso é feito pode variar amplamente entre organizações, Nexus, Time Scrum e indivíduos.

Ou seja, faça o seu trabalho contanto que você respeite essas mínimas regras, que é só um P.O. por backlog. Que o backlog você pode gerir utilizando as melhores práticas disponíveis ao seu alcance, contanto que você não fira as regras do Nexus e do próprio Scrum, simples assim e complicado assim.

É daí de novo que surgem aquelas questões que eu te apresentei no começo do nosso vídeo: será que não estamos promovendo aqui com esses frameworks uma simplificação absurda, mas apoiada nos ombros de práticas consolidadas? E o time de integração do Nexus, de novo, um P.O. e também um Scrum Master. Ele pode ser Scrum Master de um ou mais times.

Não existe uma especificação de que cada time Scrum tenha que ter um Scrum Master exclusivo. Isso acontece também no Scrum. O que acontece no Scrum? Podemos ter em um time Scrum, um Scrum Master que trabalhe sem exclusividade para aquele time Scrum, então ele está lá, trabalha naquele time Scrum, mas ele também atende a múltiplos times como Scrum Master, como esse facilitador.

Ele é full time Scrum Master, não é Scrum Master/Dev mas ele não trabalha com exclusividade em um time. Então pode acontecer de vários times Scrum terem o mesmo Scrum Master, mas terem esse papel representado por um ótimo profissional, especialista em Scrum e em Nexus.

Só que isso é nos times Scrum, no Nexus, vamos ter um Scrum Master do Nexus. Então os times podem ter um Scrum Master, dois ou três, dependendo da organização de todo o Nexus, mas o Nexus tem só um Scrum Master e ele também pode acabar chegando ao Scrum Master do Nexus por ser o Scrum Master de outros times.

Vai ser o ponto focal, ele vai identificar impedimentos, vai identificar dificuldades, ele vai ser esse Scrum Master chefe. Ele vai facilitar a integração entre os times, ele vai ser esse grande líder servidor de todo o Nexus. Ele vai ser a representação do Scrum Master de todos os times, ele vai ser com quem os Scrum Master vão conversar se necessário. Ele também vai estar presente em todos os times por meio dos demais Scrum Masters, ele vai ser um grande facilitador.

Ele vai fortalecer o Nexus, o trabalho dele é dominar o Nexus, então esse Scrum Master aqui não pode ser um Scrum Master que tenha somente experiência com um time Scrum, ele tem que ter experiência com múltiplos times Scrum. E esses Scrum Masters nós já vemos surgindo nas provas da Professional Scrum Master, da Scrum.org.

Eu falo com alunos, tenho muitos depoimentos, pessoas que fazem os exames depois de estudar conosco, e já vemos na Scrum.org, dentro das provas, questões relacionadas à múltiplas equipes.

Então o Scrum Master está começando, ainda não é uma questão obrigatória, a ter de saber participar de múltiplas equipes, integrar múltiplos profissionais, incrementos, enfim, incrementos de diferentes equipes formando um único incremento integrado. Não se esqueça: um incremento por Nexus, um backlog por produto.

E os outros membros então são membros que podem vir de múltiplas equipes distintas, equipes na verdade sem limite, e eles estão lá porque eles têm esse papel integrador, porque eles são mais experientes, eles conhecem os desafios em pauta, eles conhecem o trabalho e eles podem ajudar a fortalecer o Nexus e também integrar o Nexus ainda com outras áreas da organização.

Eles podem ajudar na unificação, no estabelecimento dessa metáfora, do glossário, eles podem estabelecer também a arquitetura do software, ensinar as pessoas a trabalharem dentro dos padrões da empresa.

Ele é uma ponte. Apesar de não parecer muito glorioso fazer parte dos demais membros no time de integração, eles são possivelmente os mais importantes, porque eles são, dentro dos times de desenvolvimento, pessoas respeitadas pelo seu amplo domínio no conhecimento XYZ, pela sua habilidade também de colaborar e trabalhar em equipe.

E eles vão chegar no time de integração do Nexus e vão criar uma cultura para todo o Nexus, e essa cultura, obviamente, também tem que estar alinhada e de acordo com a cultura organizacional.

Podem ser membros de time Scrum? Obviamente, se necessário, podemos convidar também pessoas de equipes, enfim, que façam parte do Nexus. Mas aqui surge uma questão interessante: eu posso ter no meu time Nexus outras pessoas, de fora? Nós vamos abordar isso em breve. E tem como foco primário, e eu já expliquei isso, o time de integração. Qual é que é o foco primário do time de integração? É ser essa ponte, é ser esse canal, esse ponto focal, é sustentar práticas, multiplicar práticas, ser o farol do Nexus.

E quem trabalha no time de integração dá prioridade para o time de integração, se sobrar tempo, vai voltar a desenvolver, codar, fazer o que for necessário. No próximo vídeo vamos tratar sobre eventos do Nexus e continuar a explorar esse framework, enquanto no material complementar e nas aulas complementares você vai encontrar cases, aplicações de Scrum escalado, e muito mais.

## Eventos do Nexus

Bem-vindo a mais um vídeo do nosso treinamento e nesse vídeo eu vou explorar com você os eventos do Nexus. E na última aula, eu perguntei e deixei em aberto a dúvida: será que eu posso ter no time de integração do Nexus pessoas que não fazem parte dos times Scrum? A resposta é: isso não está especificado no guia do Nexus, não existe uma proibição, na verdade o que existe é uma abertura para esse tipo de membro se juntar ao time de integração.

Quando o Nexus Guide apresenta que geralmente o time de integração é composto por membros dos times Scrum, ou seja, geralmente, pode acontecer que você tenha membros que não façam parte dos times Scrum. Agora, isso também vai ficar a seu caráter. Vamos descobrir os eventos do Nexus? E por que nós vamos descobrir os eventos? Porque é por meio dos eventos que você vai conhecer as práticas de Scrum escalado.

Então de novo, olhando o nosso fluxo do Nexus, eu consigo já identificar os eventos do Nexus. Nós temos aqui iniciando com backlog, e logo em seguida um evento - planejamento da Sprint do Nexus. Isso é um evento. Backlog da Sprint do Nexus é evento? Não, backlog da Sprint do Nexus é artefato. Então o planejamento da Sprint do Nexus é um evento. A reunião diária do Scrum do Nexus? Mais um evento. A revisão da Sprint do Nexus - ainda um outro evento. E também temos a retrospectiva da Sprint no Nexus.

Todos estão eventos do Nexus. E vamos ter também, o Nexus vai nos recomendar manter o Scrum, ele vai nos dizer que o Scrum deve seguir como o motor do Nexus, então se eu tenho planejamento da Sprint do Nexus, eu também vou ter um planejamento da Sprint do meu time. Eu vou ter a Sprint, eu vou ter uma revisão da Sprint do Nexus e também da Sprint do meu time, individualmente? Não, esse é o único evento que o Nexus suplanta.

Vou ter a reunião diária do Scrum do Nexus e também vou ter a reunião diária do meu time de Scrum. Retrospectiva da Sprint do Nexus abraça a retrospectiva do time Scrum. Então quais são os eventos do Nexus? Os Timeboxes no Nexus são absorvidos dos Timeboxes do Scrum.

Então se eu tenho um Sprint do Nexus de 30 dias, o meu planejamento da Sprint do Nexus vai ser de oito horas. Se eu tive um ciclo de desenvolvimento na Sprint de um mês, que é a melhor forma de apresentar isso como está no Scrum Guide, um mês que pode ter até 31 dias ou 28 dias, dependendo do mês. Eu vou ter para esse mês, além das oito horas de planejamento, quatro horas para revisão, três horas para a minha retrospectiva e 15 minutos para a minha reunião diária.

Então os Timeboxes do Scrum eu absorvo nos eventos do Nexus. O que isso significa? Que o planejamento da Sprint do Nexus vai ter oito horas, que a minha revisão do Nexus vai ter quatro horas, que a retrospectiva do Nexus vai ter quatro a três horas e que também a minha reunião diária do Nexus, 15 minutos.

Agora temos que entender também que o Nexus não é absolutamente específico para os seus eventos. O Scrum é mais específico do que o Nexus. E as práticas de Scrum já estão mais consolidados do que as práticas do Nexus, as práticas do Scrum escalado.

Então note o seguinte: a minha retrospectiva vai ser quebrada em alguns momentos, e vamos falar sobre isso um pouco mais adiante. Mas quando há essa quebra também, eu entendo, e aí o seu entendimento deve prevalecer mais do que o meu inclusive, em consonância com a sua visão de gestão ágil de projetos, como já entendemos também acima de tudo.

É que onde há uma ruptura no evento, por exemplo, se eu tenho uma reunião de retrospectiva e eu tenho de abarcar dentro dela a retrospectiva de cada time, então eu vou ter a retrospectiva do Nexus e a retrospectiva de cada time.

Então há uma quebra, logo há uma interrupção nesse Timebox, nós vamos discutir isso mais adiante. O foco aqui é, lembre-se, os eventos têm esse Timebox absorvido pelo Scrum dentro do Nexus. E o que nós temos aqui em termos de refinamento? O refinamento também é uma atualização não tão recente, mas não é absolutamente antiga, vimos nos últimos anos o Grooming, ou o refinamento do backlog, se consolidar no Scrum Guide, os 10%.

Isso inclusive não está presente em alguns dos nossos treinamentos, mas agora está estabelecido no próprio guia Scrum que 10% do tempo de uma Sprint, não do Sprint Planning, você vai investir em Grooming. E você, enquanto faz o Grooming, que é esse refinamento do backlog, em conjunto, ou seja, o P.O. não é que faz estimativas, quem faz estimativas é o time de desenvolvimento. Nesse caso, vamos ter o time de integração colaborando no Grooming e fazendo uma interface com os times de Scrum.

Então esse esforço deve ocorrer durante a Sprint, assim como também deve ocorrer a integração do esforço do time para resultar em um incremento ao fim daquela Sprint, durante a Sprint, não alternou nos eventos posteriores ao ciclo de desenvolvimento em si. Você vai fazer o refinamento por meio de workshops, práticas diversas, como você julgar melhor, e aí vai mapear as dependências e os conflitos. Então mantenha esse entendimento, que você pode buscar no Scrum Guide, de refinamento e 10%.

O planejamento da Sprint do Nexus vai ser liderada pelo Product Owner. E a Sprint do Nexus, você já sabe, vai herdar oito horas de Timebox do planejamento da Sprint do Scrum. E o P.O. vai liderar esse evento, por quê? Porque ele entende das dependências. Ele mapeou as dependências, ele contou com o time de integração, ele fez por meio do time de integração essa interface com os demais times e ele conhece as dependências, e ele conhece os times. Com o suporte do Scrum Master ele vai então organizar essa reunião de planejamento.

Ele vai levar para a reunião tudo que está Preparado. O que é Preparado? É aquilo que atende a definição, estabelecida pelo próprio P.O., do que pode ser levado a uma reunião de planejamento, aquilo que pode ser apresentado aos times de desenvolvimento para que os times de desenvolvimento então criem a solução. E quem é que define o Preparado? É o dono do produto, como você vai ver também o Pronto, quem define é o time de integração, e esse Pronto pode ser refinado, pode ser aperfeiçoado por cada time Scrum.

Então o que vai chegar na reunião de planejamento do Sprint do Nexus, esse evento de oito horas de duração? Vai chegar o P.O. com o seu backlog priorizado, com as definições apontadas, e ele vai então apresentar ao time esses itens priorizados e o time, em conjunto com o P.O. e com o Scrum Master, vai determinar, na reunião de planejamento da Sprint do Nexus, a meta dessa iteração.

Agora, é importante também salientar aqui no planejamento da Sprint do Nexus, vamos trazer as pessoas que sejam mais apropriadas para aquela reunião: pode ser o time de integração com outros membros de times diferentes de Scrum.

Agora, é obrigatório todo time de integração ou todos os times Scrum? Não, apenas quem for relevante para essa reunião, a meta da Sprint do Nexus então vai ser definida e o que nós temos que fazer? Foi feita a meta da Sprint, as pessoas adequadas estavam presentes. Então se faz necessário agora o quê? A execução de uma reunião de planejamento de cada time.

Eu já tenho uma meta, eu já tenho aqui uma formação de backlog inicial, só que antes de encerrar o meu planejamento da Sprint do Nexus, eu tenho que promover uma quebra e essa quebra vai ocorrer para que seja executado então o planejamento de Sprint de cada time individualmente, onde eles vão comprometer porções do backlog da Sprint do Nexus. Então, depois disso, retornam esses participantes à reunião de planejamento da Sprint do Nexus e encerra-se a reunião.

Então notamos o quê? Uma quebra. Você pode ler dez vezes o guia do Nexus e você não vai encontrar uma orientação clara, de mantenimento do Timebox dessa reunião, assim como você não vai encontrar em outros eventos. Então a minha sugestão para você, já entrando na prática, é: faça a sua reunião de planejamento da Sprint, quebre ela em duas partes. Primeiro é o que nós vamos fazer, pode ser aplicado aqui quatro horas, e uma meta da Sprint.

Cada time faz a sua reunião de planejamento, individual por times Scrum, já sabendo o que deve ser feito, o que se busca fazer. Agora o como vai ser feito, depois de ser feita uma reunião de Sprint por time, de oito horas também, considerando um mês de calendário, o time volta para segunda parte da Sprint do Nexus e termina o Timebox de quatro horas da segunda parte da reunião, com o comprometimento por equipe de itens do backlog. Fica aqui a minha sugestão.

A demonstração da visão de incremento de Pronto, ou seja, o que se espera entregar ao fim da Sprint também vai acontecer idealmente na primeira parte da reunião da Sprint do Nexus. Então, como eu disse, cada time tem que fazer o seu planejamento, mas posterior à primeira parte da reunião. Vamos adotar aqui a mesma prática que já adotamos na Sprint Planning um e dois: A primeira parte do planning do seu time individual vai ser focada em determinar a meta.

A segunda parte é o como, estimativas e apropriação, determinação de atividade por membros, enfim, do grupo, ou refinamento até, se necessário. No Sprint do Nexus vai ser a mesma coisa: na primeira parte vamos determinar a meta da Sprint, na segunda parte, depois de ter sido feita a reunião de planejamento de cada time, então fazemos a apropriação e a consolidação dessa apropriação do comprometimento das histórias de usuário dentro do backlog do produto.

E também, obviamente, do Sprint do Nexus, mas já comunicando com cada time individualmente.

E cada time, quando estiver fazendo a sua reunião de planejamento individual por time, pode também ter uma meta. E como eu disse antes, sabemos que hoje também já mudou um pouco esse entendimento de definição de Pronto. A definição de Pronto é criada pelo time de desenvolvimento. O time de integração do Nexus é que vai fazer essa definição de Pronto. Só que a definição de Pronto, assim como a meta, podem ser melhor especificadas por cada time.

Então o time sabe qual é a meta da Sprint do Nexus, mas pode pegar aquela meta e detalhar ainda mais ela ou criar uma meta que seja uma fração daquela grande meta. Assim também pode acontecer com a definição de Pronto: o time de integração do Nexus vai determinar o que é pronto, e então cada time pode melhor detalhar, mas não pode mudar a definição base. E também são bem-vindos nesta reunião os Stakeholders, partes interessadas, diretoria, cliente, para ter essa noção de todo e até mesmo colaborar.

A reunião diária, também de novo, vai ter a participação de quem for mais adequado à reunião diária do Nexus. E ela ocorre antes da reunião diária de cada equipe. Então quem for necessário à reunião diária do Nexus, com aquele Timebox de quinze minutos, vai participar e depois cada time vai promover a sua própria reunião diária. Aqui o foco é na integração, eu quero com a reunião diária do Nexus trabalhar principalmente na integração. Essa é a principal dor que o Nexus vem sanar.

E depois então dessa reunião, vai ocorrer a reunião diária da equipe. E aqui eu tenho algumas perguntas que devem ser feitas dentro da reunião diária do Nexus, e essas perguntas vão ser respondidas por quem estiver participando da reunião, seguindo também a mesma ritualística da reunião diária do Scrum.

Então vamos às perguntas: o trabalho do dia anterior foi integrado com sucesso? Ou seja, conseguimos promover a integração que você buscava? Não? Por que não? Quais novas dependências surgiram do esforço de integração? Quais foram identificadas enquanto nós trabalhávamos? E que informações precisam ser compartilhadas entre as equipes? A partir dessas respostas, os partícipes dessa reunião, vão voltar para a sua reunião diária.

E aqui podemos ter quem for necessário, de novo, na diária do Nexus, vão voltar para a diária do Scrum com os seus times, e aí, com base nessas respostas, vão elaborar a estratégia do dia para o trabalho da equipe.

Vai ser discutido o que eu fiz, o que eu não fiz e o que eu vou fazer. E as respostas devem ser já embasadas, dentro da reunião de cada time, naquelas respostas dadas na reunião diária do Nexus. Então aqui eu vejo o Scrum Master como principal facilitador: ele vai fazer essa integração, mas também vemos o desenvolvedor. Apesar de não ter o chapéu de líder técnico, sempre surge aquela pessoa com notório saber que pode representar o time de desenvolvimento.

Então as perguntas já estão dadas, mas respostas devem buscar esse alinhamento com a diária do Nexus dentro da diária do Scrum.

E a revisão da Sprint? Eu trago aqui para você palavras do próprio Ken Schwaber e da Scrum.org, os mantenedores desse framework: A revisão da Sprint do Nexus substitui a reunião individual de revisão do Scrum. Está rodando o Nexus, então não vai ter reunião de revisão do Scrum - é a única que é cortada, é o único evento que é cortado. Porque todo o incremento integrado é o foco para captar o retorno das partes interessadas.

Pode não ser possível mostrar todo o trabalho completado em detalhe. Aí de novo: técnicas que você quiser podem vir a ser empregadas nessa reunião. Imagina 90 pessoas em uma única reunião. Como é que você vai reunir esse pessoal? Esse é um desafio que você vai ter que resolver utilizando N práticas e a sua experiência gerencial. De novo, o Nexus não nos dá essa resposta.

E aí que surge a necessidade de ter um profissional que seja mais do que um Scrum Master, que seja mais do que um P.O., uma pessoa que tenha essa visão holística, justamente a proposta do nosso treinamento. E a retrospectiva dentro do Nexus, como é que vai ocorrer? Em três etapas. Na primeira parte as pessoas adequadas, de novo, sem limite, vão apontar questões e eu já vou te apresentar questões.

Na segunda parte é feito um intervalo, então, onde cada time Scrum faz a sua própria retrospectiva. E na terceira parte temos um reencontro para fazer um acordo sobre o que de fato deve ser monitorado e acompanhado na Sprint seguinte. E você vê que temos partes aqui definidas na retrospectiva da Sprint do Nexus, mas não temos essas partes definidas lá no Nexus Planning, que é quando fazemos o planejamento da Sprint do Nexus. Por isso que eu já te dei essa sugestão de cara.

E então o que que nós vamos fazer em termos de questionamentos, assim como fizemos na reunião diária do Nexus, dentro da retrospectiva da Sprint do Nexus? Será que foi deixado de ser realizado um trabalho? Alguma coisa ficou pendente? Tem alguma dívida técnica? O Nexus gerou débito técnico? Todos os artefatos, o código foi integrado com sucesso? Aqui você pode buscar aquelas práticas do XP que já vimos também e empregar elas no seu projeto.

Você, quando estiver executando a Sprint, pode entregar e integrar, com o seu esforço de escalar o Scrum, múltiplas práticas de desenvolvimento. Então tudo foi integrado? Essa pergunta aqui está relacionada diretamente ao código. Tudo o que construímos aqui está integrado? Tudo o que foi feito ontem, quantas vezes por dia ao longo da minha Sprint eu fiz isso. Ontem, claro, nós vamos nos perguntar no máximo na reunião diária, quando estivermos fazendo a retrospectiva do trabalho como um todo.

Mas isso só mostra para você como o meu mindset está ajustado para trabalhar com o Nexus, porque eu vou me preocupar diariamente com a integração, eu não vou deixar isso para uma reunião de retrospectiva. Na retrospectiva eu vou olhar como que eu integrei e se eu integrei bem ou mal, e como eu posso melhorar a integração. Não é nem na retro e nem na revisão que eu vou fazer a integração.

Nós construímos um software com sucesso? Ele foi testado, implantado, com a frequência necessária para medir o acúmulo de dependências não resolvidas? Será que eu estou conseguindo fazer o meu trabalho? E aí podemos nos lembrar da história que a Carol Azevedo conta no curso de Agile na Prática, inclusive que eu ajudei ela a construir, sobre o que é PEC, sobre o surgimento do XP.

E você, escutando a história que a Carol conta, você vai conseguir entender como é importante medir o sucesso, medir o Pronto, ter entregas realizadas, porque se você não entrega, você não agrega. Tem que ter um incremento integrado no fim de cada Sprint. Então, essas perguntas foram feitas na primeira parte da reunião, houve o intervalo, e os times foram fazer as suas próprias retrospectivas.

Depois de ter feito a sua própria retrospectiva, voltam para a terceira parte da reunião de retrospectiva do Nexus com apontamento de ações e soluções. Por que aconteceu o que aconteceu? O que eu discuti com o meu time e que eu posso trazer agora para a minha retrospectiva? Como é que podemos resolver o débito técnico? Como que podemos melhorar questões relacionadas à integração? Como que eu posso evitar problemas que se repetem, problemas que estão ocorrendo e estão nos prejudicando?

Eu fico por aqui e, no próximo vídeo, eu vou entrar em detalhe nos artefatos do Nexus. Espero que você tenha gostado, estou te trazendo uma perspectiva nova sobre o Scrum, que é o Scrum escalado, e o nosso curso de Agile Avançado está quebrando alguns paradigmas.

Eu quero que você reflita sobre Scrum, sobre simplicidade e complexidade, conceitos consolidados e, mais do que isso, esteja pronto para montar a sua própria abordagem e utilizar tudo que você está vendo nesse treinamento e resolver os problemas que você encontra na sua empresa.

## Artefatos do Nexus

Seja bem-vindo ao nosso último vídeo sobre o Nexus, vai ser um vídeo rápido e nesse vídeo eu vou te falar sobre os artefatos que encontramos no Nexus, utilizando a própria tela com todo o fluxo do Nexus. E qual é a relevância dos artefatos? É a organização.

Esses artefatos vão nos ajudar a organizar o trabalho. Eles vão representar a visão do produto, vão representar o esforço a ser executado, o produto ou parte desse produto, o incremento a ser construído, a Sprint, enfim, são simples. É um vídeo rápido e vamos encerrar essa jornada pelo Nexus.

E qual é o primeiro artefato que nós temos no Nexus? É o backlog do produto. O backlog do produto tem que ser absolutamente transparente, tem que ter as dependências muito bem mapeadas, tem que ser visível, acessível, todo mundo tem que conseguir ler e entender o que está escrito e apresentado. O Nexus, ele depende muito dessa transparência. Se não há transparência no Nexus, todo trabalho vai por água. Então talvez o pilar central do Nexus seja a transparência.

O backlog do produto, você sabe, é uma lista com todos os requisitos necessários referentes ao produto, ordenados, ele pode conter inclusive riscos - como já vimos no gerenciamento ágil de projetos, na nossa carreira. Ele pode, enfim, ter tudo que é necessário para que aquele produto siga existindo, seja construído, atenda a necessidade de mercado. Enquanto houver um produto, vai haver um backlog e um Product Owner.

Nós temos também o backlog do Sprint da Nexus, que vai ser uma reunião dos itens do backlog do produto, priorizados e levados para a próxima iteração. Lembre-se: para chegar no backlog do Sprint da Nexus, vamos quebrar o nosso planejamento da Sprint da Nexus em três partes, assim como quebramos a retrospectiva.

A primeira parte eu vou fazer a definição da minha meta da Sprint, vou criar um intervalo, vou fazer a minha reunião de planejamento com cada time e, na terceira parte, eu vou fazer a compilação desse planejamento individual por equipe em um backlog da Sprint do Nexus.

E o último artefato, pasme, realmente esse é o último artefato, é um incremento integrado, mas não menos importante, obviamente, que vai ser fruto desse esforço de organização do Nexus, do time de integração trabalhando para facilitar os times Scrum, dos eventos que já exploramos, o planejamento da Sprint do Nexus, vamos ter o backlog da Sprint do Nexus.

Até um incremento é resultado de uma retrospectiva, se você considerar o fluxo contínuo de Sprints. Então, a cada Sprint, o time entrega melhor, mais integrado, claro, se todos estiverem fazendo a sua parte.

Então vou falar agora sobre alguns conceitos gerais para encerrarmos esse vídeo e essa apresentação sobre o Nexus. E vou buscar te trazer, ainda, conteúdo prático, cases, para você estudar e evoluir. Deixe para mim aqui, na Alura, recados, mensagens, referentes aos próximos cursos de Agile Avançado que você deseja ver.

Então, o time de integração do Nexus, que é responsável pelo desenho de toda a forma como o time vai trabalhar, em colaboração com o time, obviamente, representa esse desejo, ele representa esse conhecimento e esse time de integração vai definir o que é Pronto. Então o Pronto vai ser definido pelo time de integração. Agora, o Pronto pode ser refinado para cada time por cada time individual, só que ele não pode ser modificado. Posso aumentar o nível de detalhamento, eu não posso diminuir.

E temos também a necessidade de transparência. Sem transparência o Nexus não ocorre. E, por fim, quero te dar um toque: Preparado. É uma definição criada pelo Product Owner, ele vai dizer o que está preparado ou não, mas sempre em conjunto com o time de integração, que vai representar a voz do time ou dos times, de desenvolvimento, os times Scrum.

E o Preparado é tudo aquilo que pode ser apresentado, por exemplo, em uma reunião de planejamento do Nexus. Por quê? Porque está pronto para ser desenvolvido, ou melhor, está preparado. Eu te vejo no próximo vídeo e vamos encerrar esse treinamento e quero muito conversar com você, quero saber sua opinião sobre o nosso curso, esse é o primeiro curso de Agile Avançado que lançamos aqui na Alura e eu preciso do seu feedback.

# Mostre que Sabe

Seja muito bem-vindo ao nosso vídeo de encerramento do nosso treinamento de Agile Avançado - Crie Modelos e Descubra o Nexus. Foi um grande desafio, porque, saiba, que os conceitos que eu te apresentei aqui, eles foram construídos e eu apliquei eles na prática por meio de consultorias ao longo dos anos, desde o meu primeiro contato com a Agile até a primeira implementação.

Então eu construí esse curso realmente com muito carinho e aproveitei também um momento, antes da gravação desses vídeos, que eu participei de treinamentos presenciais, que eu ministrei treinamentos, que eu participei de palestras como palestrante e colhi muito feedback das pessoas que estavam nessas palestras, nesses treinamentos, e pude gravar esse vídeo e trazer para você uma visão contemporânea da gestão de projetos para além do Agile, mas com foco ainda no Agile, que foi o que nós fizemos aqui, dando ênfase ao Nexus.

Então eu te ensinei a pensar como um consultor, a se posicionar como um gerente de projetos de Agile, mas também mantive a sua mente aberta para outros modelos, para que você consiga adaptar e adaptar-se. E quanto ao Nexus, que é o que dá a conotação de Agile Avançado a esse treinamento, acima de qualquer outro vídeo que nós tenhamos apresentado, o Nexus é de fato o Agile Avançado, que dá escala para o esforço de um time Scrum.

E, a partir das práticas do Nexus, você pode também construir o seu modelo, dar escala para o trabalho das equipes na sua empresa e isso é avançado, é um tópico que merece a sua atenção, esse estudo pode determinar o seu futuro profissional.

Então tendo terminado essa sequência de vídeos e de exercícios que foram, nesse curso, muito ricos e muito completos, que nos deram todo o embasamento necessário, tanto teórico como complementação prática, e tendo chegado até aqui então, de fato, concluído esses exercícios, você pode se considerar um gerente ágil, um gerente de projetos que vai além do Agile, mas que também retém esse conhecimento e detém esse conhecimento avançado em Scrum por meio do Nexus.

Conquiste o mercado. Agora está nas suas mãos colocar isso de fato em prática, para além de um treinamento, vai determinar o seu futuro no mercado, porque cada vez mais buscam-se profissionais de gestão que consigam orbitar entre diferentes climas e organizações, equipes, padrões e métodos. Isso está muito alinhado com a nova forma de se entender as empresas.

E o que esperamos desse gerente ágil? Lembrando que ser um gerente ágil ainda é ser um gerente de projetos, você pode vestir essa roupa, mas sempre lembrando que o campo da gestão é maior do que o Agile. Então você tem que entender o campo de jogo, justamente tem que entender que campo é esse, do gerenciamento de projetos, quais são as interfaces desse campo com ainda outros campos, e dentro do campo, quais são as culturas vigentes, filosofias que também se integram à gestão de projetos.

Tem que entender de governança, tem que entender de gestão, práticas, não só de Agile, para além do Agile. Você tem que ter a capacidade de criar modelos de gestão específicos para desafios específicos, depois de identificar as necessidades da organização, da equipe, entender o desafio, conhecendo práticas e a sua combinação, como nós vimos na aula em que estudamos alguns exemplos de combinação entre diferentes práticas ágeis. Você agora tem que ter essa capacidade também de saber fazer essa avaliação e combinar essas práticas.

E essa combinação é o que se buscou, principalmente na nossa segunda aula, onde eu te apresentei como combinar Scrum com Waterfall, Scrum com Feature Driven Development, e também Scrum com Lean e Extreme Programming.

Dar escala a projetos: isso foi objeto das nossas aulas três e quatro, onde eu te apresentei o framework Nexus. O framework Nexus, ele não é tão recente no mercado, mas a relevância dele ainda está por ser demonstrada e eu acredito que houve uma absorção no passado a modelos mais complexos e menos ágeis, como o Scale da Agile e também ainda outros.

Eu não quero fazer muitas menções, porque isso pode ser polêmico, mas o Nexus, no meu ponto de vista, é onde você deve direcionar a sua atenção a partir de agora, e, claro, não havendo necessidade de conhecer outros modelos. Conheça a todos, se possível, mas estude o Nexus porque o Nexus representa aquilo que há de mais ágil dentro dos frameworks de Scrum escalado ou de Agile em escala, na minha concepção.

Liderar pessoas, organizações: para ter chego aqui você tem que ter feito outros cursos. Esse curso, assim como eu falei em governança, eu falei em liderança, ele possui vários pré-requisitos, então você tendo chego até aqui demonstrou não só que tem muito afinco pelos estudos, mas que está pronto para liderar pessoas e organizações. E nós esperamos, obviamente, de todo o gerente de projetos ágil, tradicional, como você quiser denominar, que ele se capacite continuamente.

Aliás, que tal começarmos a pensar em você como profissional, mais do que um gerente ágil, e como executivo de projetos? Essa é uma boa sacada: você tem que se colocar diante do mercado da forma que o mercado espera que você se coloque, mas eu e você sabemos que você é muito mais do que somente um Scrum Master, um Product Owner, um gerente de projetos. Você é um executivo de projetos – essa é uma boa denominação.

Então encerramos aqui e eu realmente espero que você tenha aproveitado do nosso momento juntos, estudando juntos e que você tenha realizado os exercícios. Se ficou alguma dúvida, entre em contato comigo pelo Fórum da Alura, eu vou ficar super feliz em conversar com você, até porque você já não é mais um iniciante, você é um profissional que detém conhecimento profundo e pode inclusive fazer as suas proposições, pode ser autor do seu próprio pensamento. Vejo você no fórum.