



2018

Accredited Partner

EXIN

**EXIN AGILE SCRUM
FOUNDATION**



Henrique Bueno
CTO | Estabilis
henrique@estabil.is



proof of accreditation

This is to confirm that EXIN has accredited

Henrique Bueno
to act as an Accredited Trainer

The trainer is accredited to deliver courses for the following examination(s):

EXIN Agile Scrum Foundation
EXIN Agile Scrum Master
EXIN Agile Scrum Product Owner
EXIN Agile Scrum Product Owner Bridge
EXIN DevOps Master
EXIN DevOps Professional



Standard
Consulting
Partner

Channel Partner



A Certificação EXIN Agile Scrum Foundation

EXIN Agile Scrum Fundamentos é uma certificação que valida o conhecimento de um profissional na combinação das metodologias ágeis com as práticas Scrum.

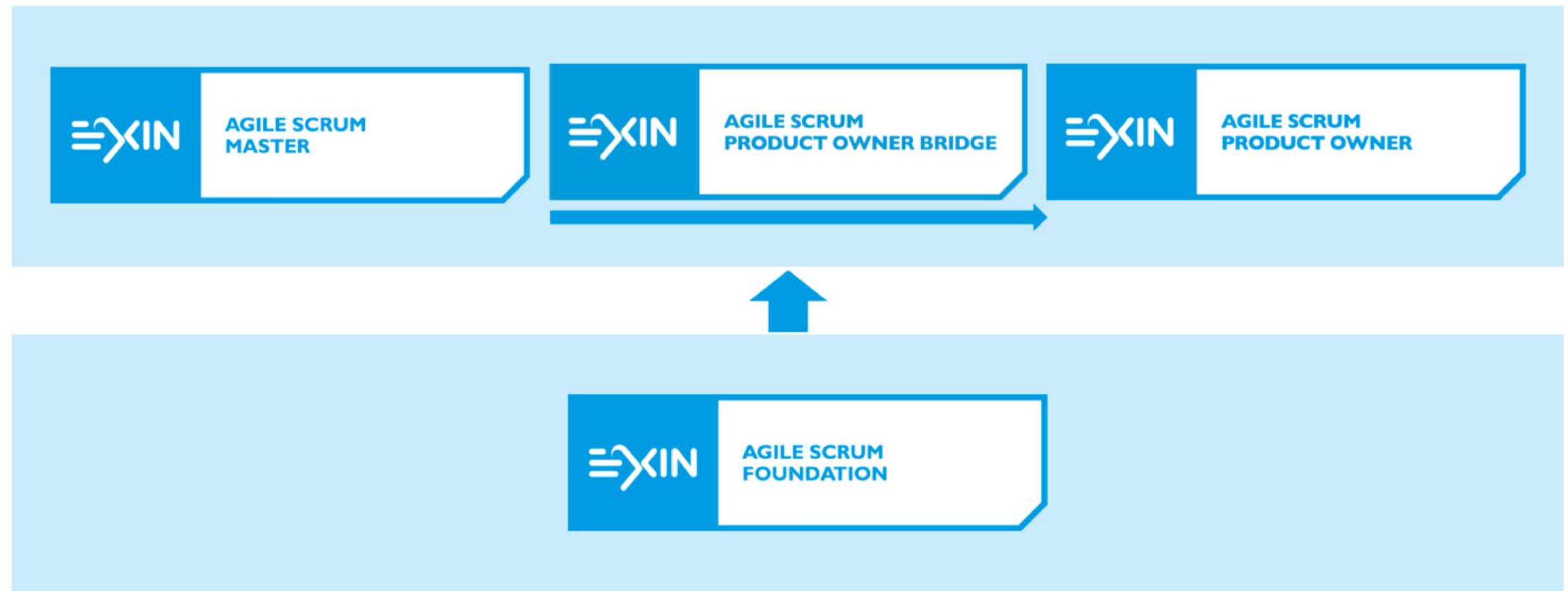


A filosofia ágil é mais conhecida na área de desenvolvimento de software, mas os seus princípios estão cada vez mais sendo aplicados em outros tipos de projetos.

O Scrum é a metodologia ágil mais utilizada e é apropriada para todos os profissionais que procuram manter seus conhecimentos atualizados com os últimos desenvolvimentos das áreas de TI e gerenciamento de projetos, particularmente aqueles que lideram e participam de projetos.

Em particular, a certificação é apropriada para os profissionais que trabalham nas áreas de:

- Gerenciamento de Projetos
- Desenvolvimento de Software
- Gerenciamento de Serviços de TI
- Gerenciamento de Negócios.



Detalhes do Exame

- Duração: 60 minutos
- Número de questões: 40
- Nota de corte: 65%
- Formas de fazer o exame:
 - Presencial
 - Online

Detalhes do treinamento

- Horas de contato: 14 horas
 - Exercícios práticos para fixação de conceitos
 - Preparação para o exame
- Carga de estudos indicada: 60 horas

1. Mindset Ágil

- Conceitos de Agile e Scrum

2. Práticas Scrum

- Papéis no Scrum
- Eventos no Scrum
- A importância do Backlog
- Definição de Pronto

3. Planejamento e Estimativas Scrum

- Planejamento
- Estimativas

4. Monitorando projetos com Scrum

- Monitoramento Scrum

5. Conceitos Avançados do Scrum

- Scrum em diferentes situações



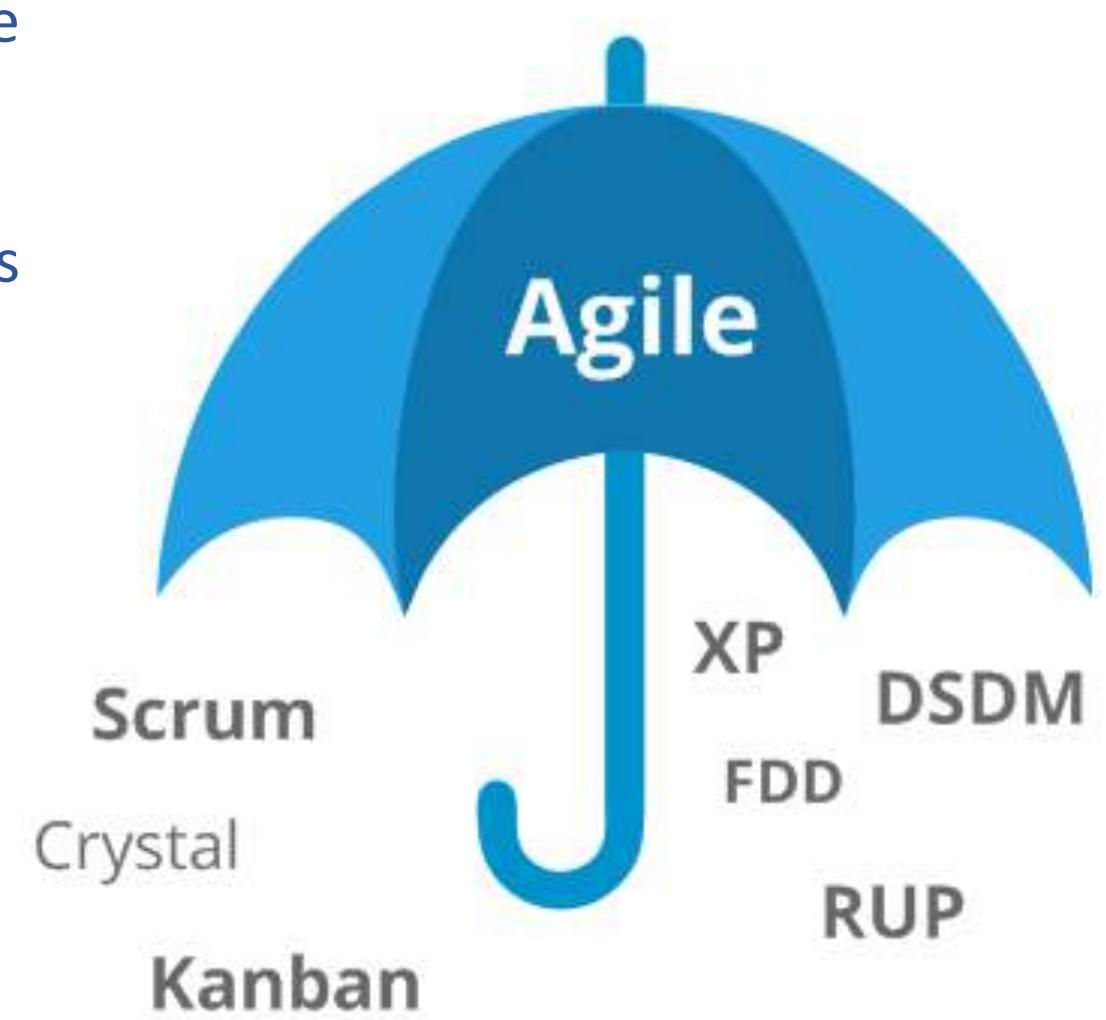
Módulo 1

Fundamentos

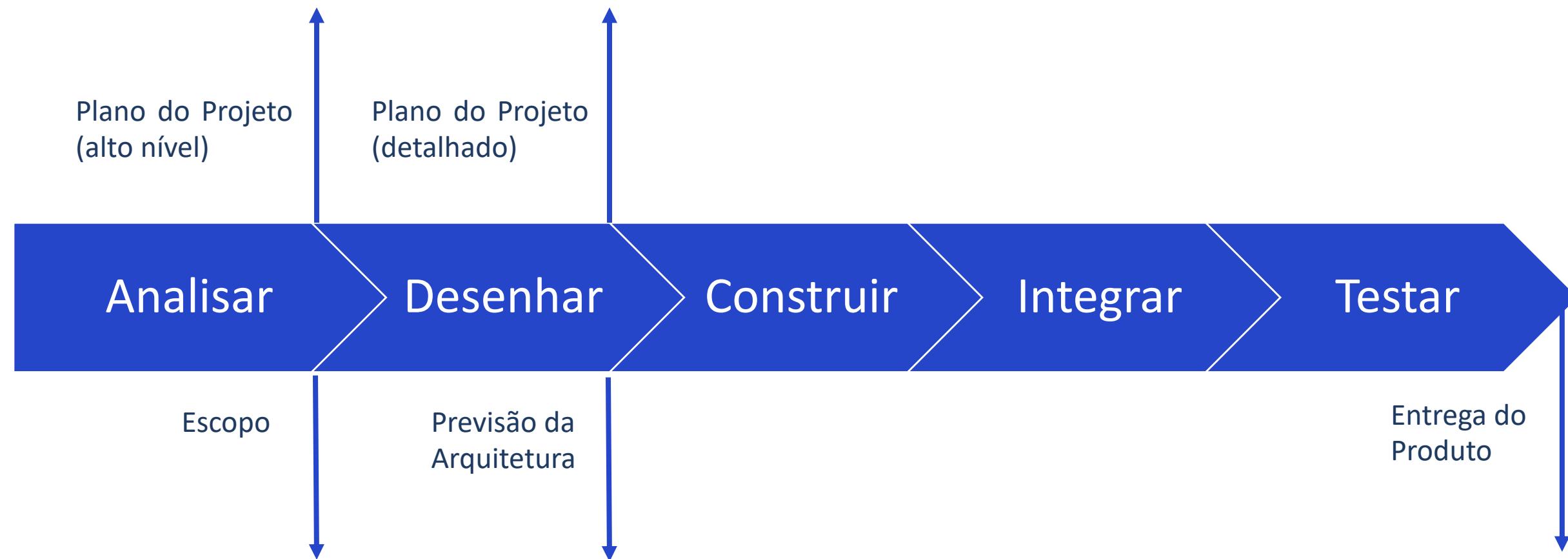
A palavra “ágil” é um adjetivo, que significa: ***Mover-se rápida e facilmente, ou pensar de forma rápida e inteligente.***

No entanto, “Ágil” (com maiúscula A) é usado para se referir a um grupo de estruturas utilizadas para o desenvolvimento e gestão de projetos.

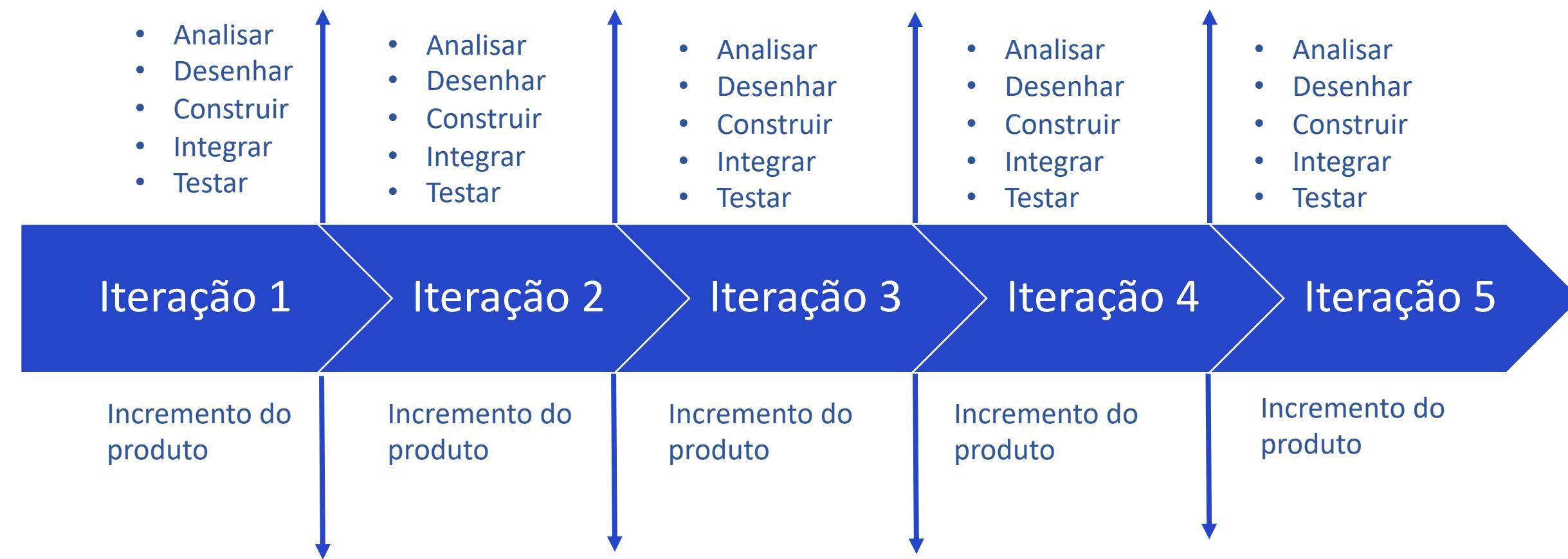
O principal objetivo dos frameworks Ágeis é tornar o desenvolvimento mais rápido e sem atritos, fornecendo entregas mais satisfatória para o cliente.



O modelo cascata



O modelo cascata



O manifesto Ágil foi criado em 2001, quando 17 dos principais pensadores em desenvolvimento de software se reuniram para encontrar uma solução melhor do que o modelo tradicional de desenvolvimento em cascata.

Contém declarações de crenças e valores em que eles acreditavam e que levaria a uma mudança radical na forma de se desenvolver software.

Suas ideias foram compiladas e convertidas em 4 valores e 12 princípios.

Indivíduos e interações,
mais que processos e ferramentas

Software em funcionamento,
mais que documentação abrangente

Valores

Responder a mudanças,
mais que seguir um plano

Colaboração com o cliente,
mais que negociação de contrato

Nossa maior prioridade é ***satisfazer o cliente***, através da entrega adiantada e contínua de software de valor

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

Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos

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

Construir projetos ao redor de ***indivíduos motivados***. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.

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

Software funcional é a medida primária de progresso

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

Contínua atenção à **excelência técnica** e bom design, aumenta a agilidade

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

As melhores arquiteturas, requisitos e designs emergem de times **auto-organizáveis**

Em intervalos regulares, o time **reflete** em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo

Timebox é um conceito muito importante no Agile. É ela que define o ritmo e dá a regularidade para o time.

Ela possui duas características principais:

1. Possui uma duração pré-definida
2. Possui um conjunto pré-definido de itens que devem ser executados

Vantagens:

- Cria regularidade e disciplina
- Ajuda a aumentar o foco no que realmente precisa ser feito
- Maximizando a produtividade



Se lembrarmos dos princípios do manifesto Ágil, precisamos ter um ritmo sustentável e constante de desenvolvimento.

Constantemente fazer muitas horas-extras pode parecer aumentar a velocidade, mas, na prática, isso realmente desacelera tudo por causa da baixa qualidade do código e da moral da equipe.

Os principais benefícios de ter um ritmo sustentável são:

- Os desenvolvedores estão mais focados em produzir do que em trabalhar;
- Eles estão focados no projeto como um todo, em vez de atividades;
- Nós teremos um ambiente mais feliz;
- No final do dia, teremos maior produtividade.

Métodos Ágeis como o Scrum possuem uma abordagem incremental e iterativa para otimizar a previsibilidade e controlar o risco.

Sprints permitem previsibilidade ao progredirem para uma meta de Sprint em uma Time-box pré-definida e imutável.

Trabalhar em Time-boxes específicos nos permite alcançar maior previsibilidade: sabemos as datas a duração das nossas próximas sprints, de modo que basta apenas definirmos em qual sprint trabalharemos em determinada funcionalidade.

Uma certeza que podemos ter é que as circunstâncias mudam.

Com métodos ágeis podemos planejar o suficiente para que saibamos a direção geral, sem todos os detalhes. Isso permite que as equipes tenham flexibilidade para alterar o curso à medida que aprende com o projeto.

Isso não significa que devemos mudar nossa direção estratégica a todo momento. É muito importante ter um norte, uma **visão** de onde se pretende chegar, e mudar o curso sempre que necessário, para atingir o objetivo da visão.

O Ágil é para qualquer tipo de projeto?



Organizações devem estar em constante mudança

Adotar uma abordagem Ágil como o Scrum é um processo de melhoria contínua.

Não há estado final pré-definido. Por causa disso, não é correto falar sobre uma “Transição completa” ou um processo de mudança que demora muito.

Mudança não é mais algo que as organizações “passam”. A mudança é agora um contínuo e perpétuo.

A melhoria contínua é parte do Ágil, não aprender e desperdiçar o conhecimento adquirido são sérias deficiências.

Vale a pena mudar para o Ágil?

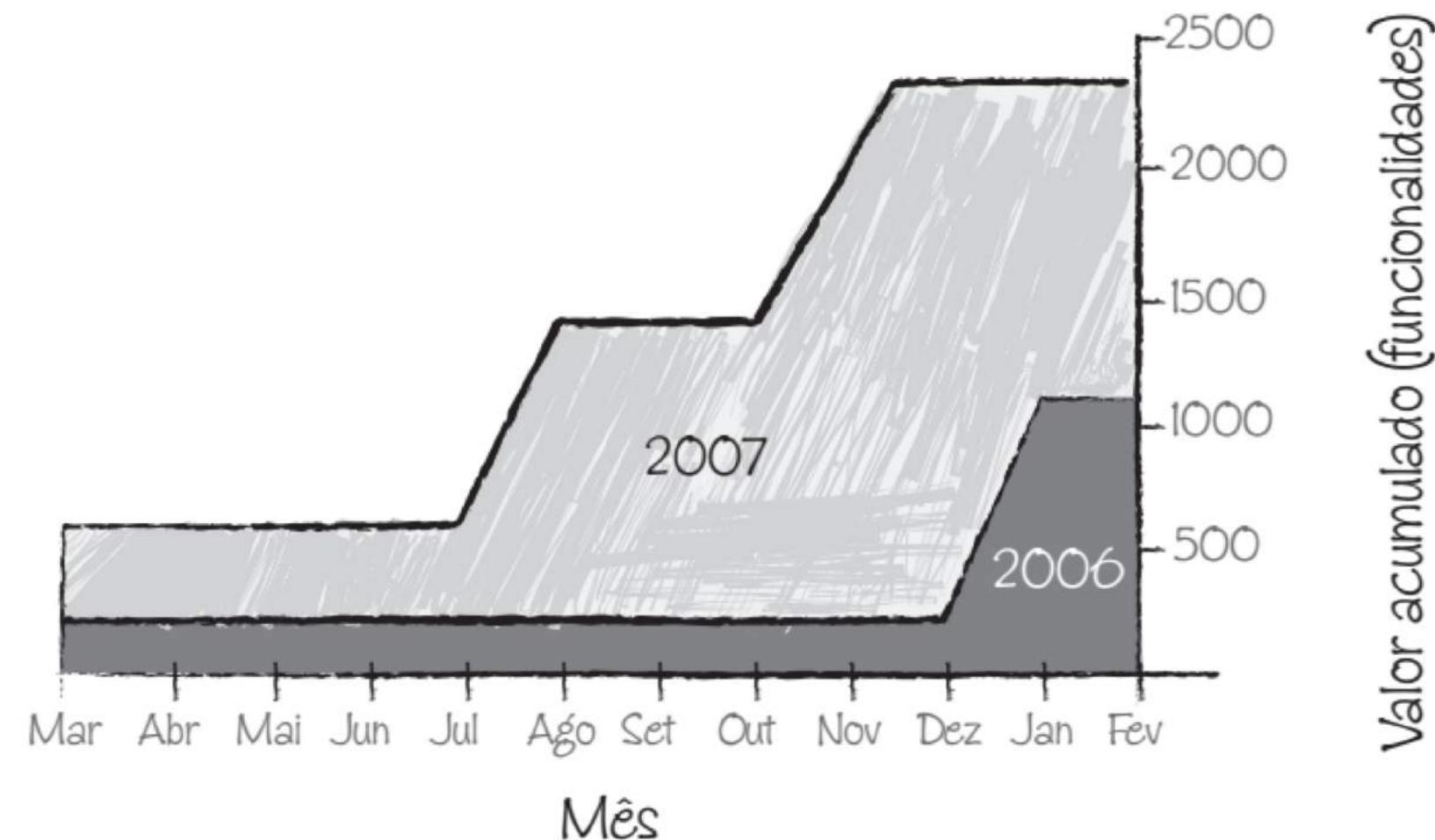
Com receita de mais de 450 milhões de dólares e 2 mil funcionários em 2006, a Salesforce.com percebeu que a frequência do lançamento de suas versões tinha diminuído de quatro por ano para uma por ano. Os clientes estavam recebendo menos e esperando mais; algo precisava ser feito. A empresa decidiu realizar a transição para o Scrum.

Durante o primeiro ano da mudança, a Salesforce.com lançou 94% mais recursos nas versões de seu sistema, distribuiu 38% mais recursos por desenvolvedor e distribuiu 500% a mais de valor a seus clientes em comparação com o ano anterior.

Nos dois anos seguintes, a receita mais do que duplicou para mais de 1 bilhão de dólares.

Com resultados como esses, não é de surpreender que tantas empresas estão tentando fazer a transição para o Ágil.

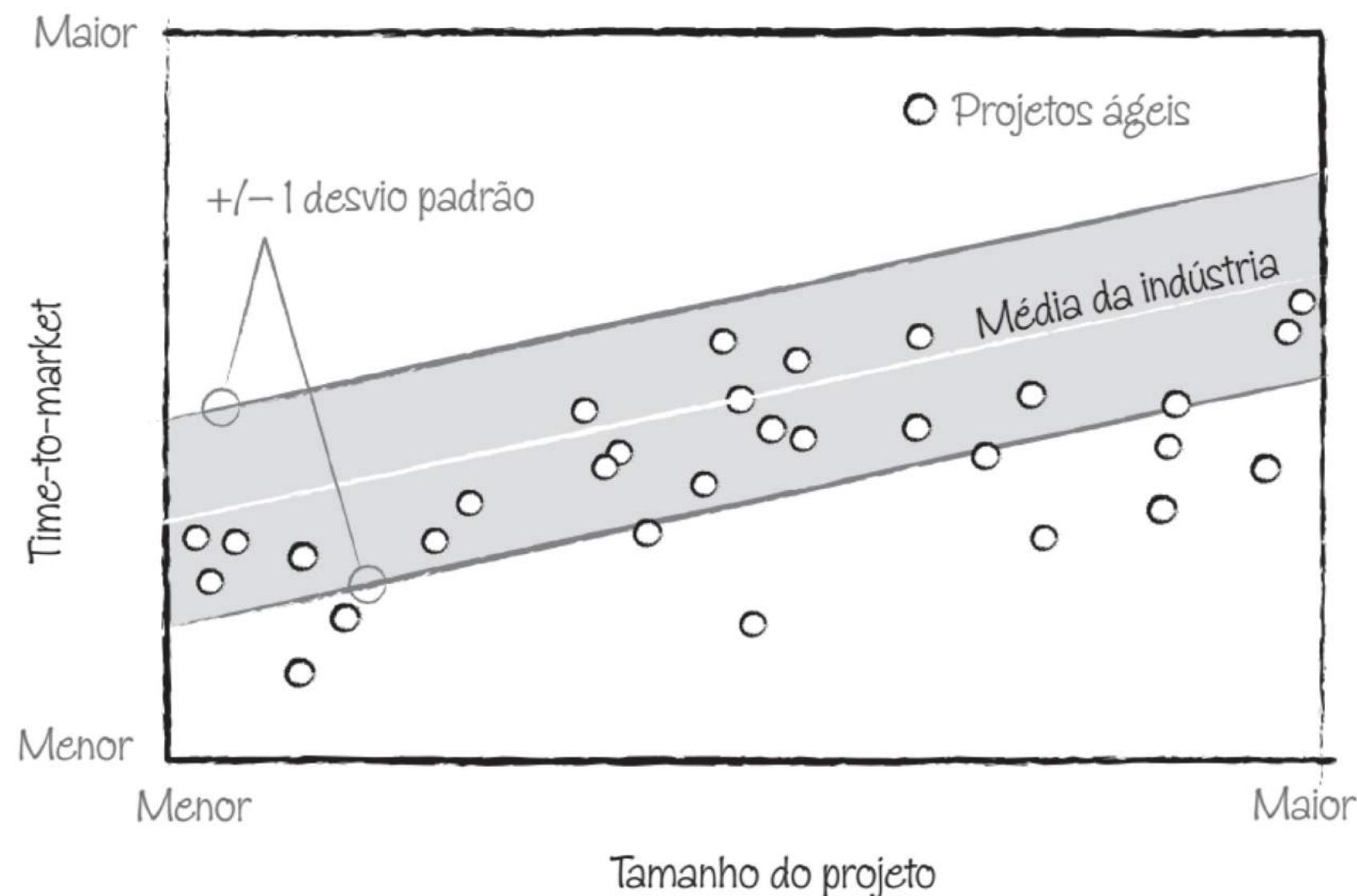
Valor acumulado (funcionalidades) entregue em versões importantes

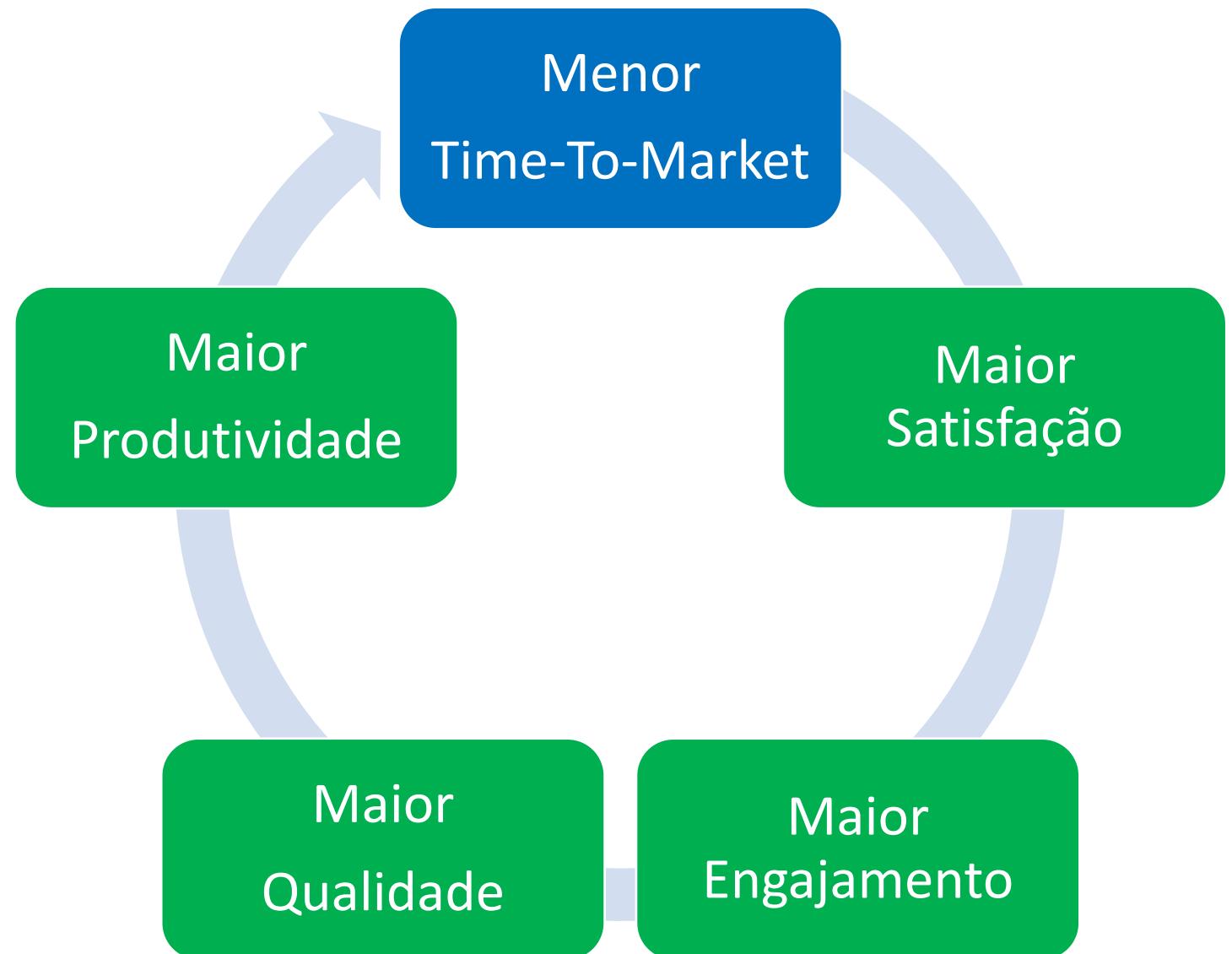


Quem conseguiu fazer a transição para o Ágil está muito satisfeito. Os principais motivos são:

- Redução do Time-To-Market
- Maior qualidade
- Maior produtividade e redução de custos
- Maior envolvimento dos funcionários e satisfação no trabalho
- Melhoria da satisfação das partes interessadas

O estudo da QSMA comparando 26 projetos Ágeis a um banco de dados de 7,5 mil projetos em sua maioria tradicionais mostrou que os projetos ágeis têm um time-to-market 37% mais rápido.





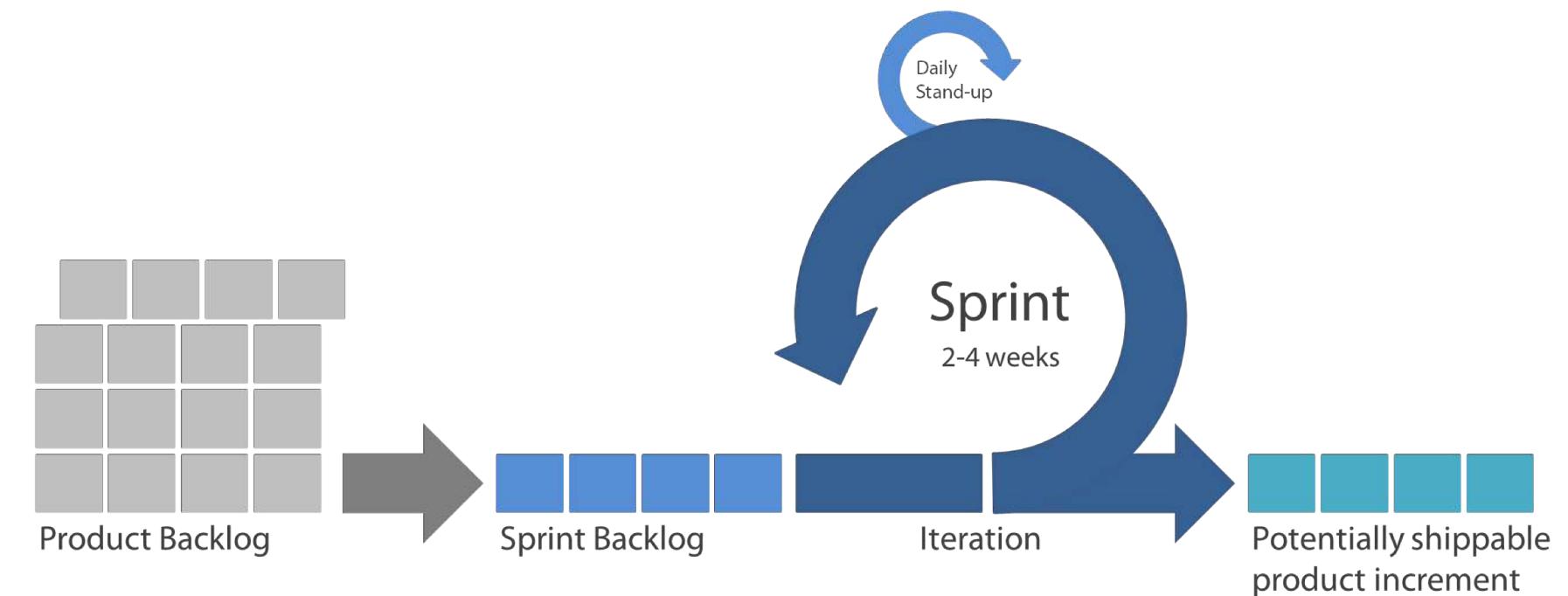
Além da maior produtividade de times Ágeis, outro fator contribui para acelerar o time-to-market:
Entregar versões incrementais do produto.

Quando os stakeholders percebem que uma equipe pode desenvolver funcionalidades importantes a cada sprint, geralmente decidem que não precisam agrupar todas as funcionalidades em uma grande versão final.

Scrum: Um framework no qual as pessoas podem abordar problemas adaptativos complexos, entregando, de forma produtiva e criativa, produtos do mais alto valor possível.

Scrum é:

- Leve
- Simples de entender
- Difícil de dominar



Método Ágil focado nas melhores práticas de desenvolvimento de software.
Promove lançamentos frequentes.

5 práticas comuns do XP:

- Desenvolvimento Orientado a Testes (TDD)
- Propriedade Coletiva
- Refatoração
- Integração Contínua
- Programação em par



O TDD se baseia em pequenos ciclos de repetições, onde para cada funcionalidade do sistema um teste é criado antes. Este novo teste criado inicialmente falha, já que ainda não temos a implementação da funcionalidade em questão e, em seguida, implementamos a funcionalidade para fazer o teste passar.

Ciclo de desenvolvimento TDD:

- Escrevemos um Teste que inicialmente **não passa** (Red)
- Adicionamos uma nova funcionalidade do sistema
- Fazemos o teste passar (Green)
- Refatoramos o código da nova funcionalidade (Refactoring)
- Escrevemos o próximo teste

Extreme Programming (XP)

Desenvolvimento Orientado a Testes (TDD)

Alguns benefícios:

- Os programadores sabem exatamente o que os outros esperam deles e podem trabalhar de maneira mais produtiva;
- Os programadores serão encorajados a codificar da forma mais simples possível, apenas para passar no teste.

Em um time Ágil, não existe um dono de um pedaço de código, um especialista em uma determinada funcionalidade do sistema. Todos são responsáveis por tudo, e todos podem trabalhar em qualquer parte do código.

Os principais benefícios da propriedade coletiva de código são:

- Incentivar a colaboração
- Fazer com que todos se concentrem no projeto e no seu valor, em vez de atividades especializadas.

Essa prática é obrigatória na maioria dos frameworks Ágeis, incluindo o Scrum.

A refatoração é a melhoria do código sem alterar suas funções externas. Quando o time gasta uma parte do tempo refatorando continuamente o código, ele está fazendo um investimento de tempo que lá na frente vai facilitar o desenvolvimento de novas funcionalidades e agilizar a manutenção do sistema.

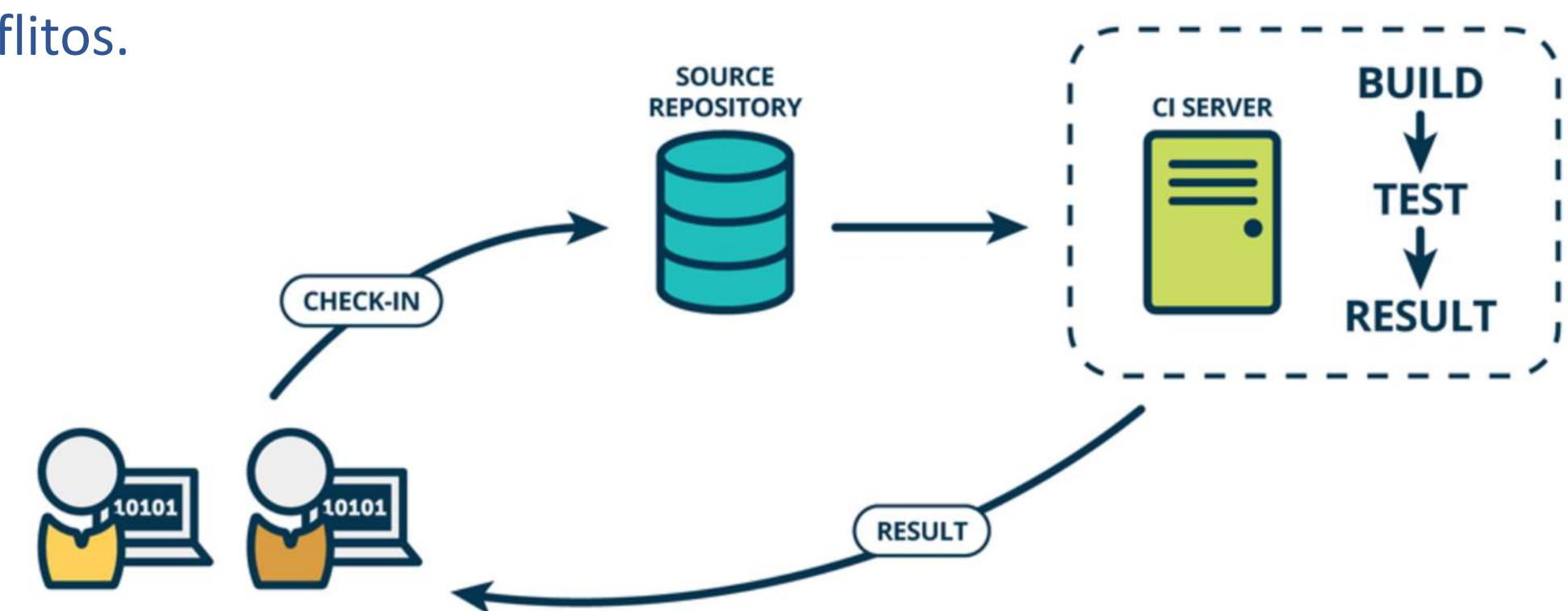
Os principais benefícios da refatoração contínua são:

- O desenvolvimento do restante do projeto é mais rápido e sem atritos;
- Podemos aplicar muitas lições aprendidas para incorporar nas partes restantes do projeto e em projetos futuros.

A integração contínua encoraja os desenvolvedores a integrar seu trabalho o mais frequentemente possível para que problemas sejam detectados e corrigidos rapidamente.

Alguns dos principais benefícios da prática de integração contínua são:

- É mais fácil ter incrementos reais no final das iterações;
- Lançar novas funcionalidades deve ser mais fácil do que nos ambientes tradicionais;
- Sabemos rapidamente se existem conflitos no nosso código;
- Temos menos trabalho resolvendo conflitos.



É um processo de desenvolvimento que incorpora a revisão contínua de código. É composta por dois desenvolvedores que escrevem código em uma única estação de trabalho, com apenas um desenvolvedor digitando de cada vez e discussões e revisões contínuas de forma livre.

Esses são os principais benefícios da programação em pares:

- O trabalho é mais agradável para as pessoas, quando estão trabalhando em pares;
- As pessoas aprendem mais quando veem umas às outras trabalhando;
- É uma atividade de formação de equipe e ajuda a criar um ambiente mais colaborativo, que é necessário para todas as equipes ágeis;
- A qualidade do produto é muito maior quando estamos usando essa prática, o que também resulta em menos defeitos, retrabalho e custo.

É uma filosofia de gestão inspirada em práticas do sistema Toyota de gestão.
O objetivo do Lean é eliminar desperdícios no processo de geração de valor.



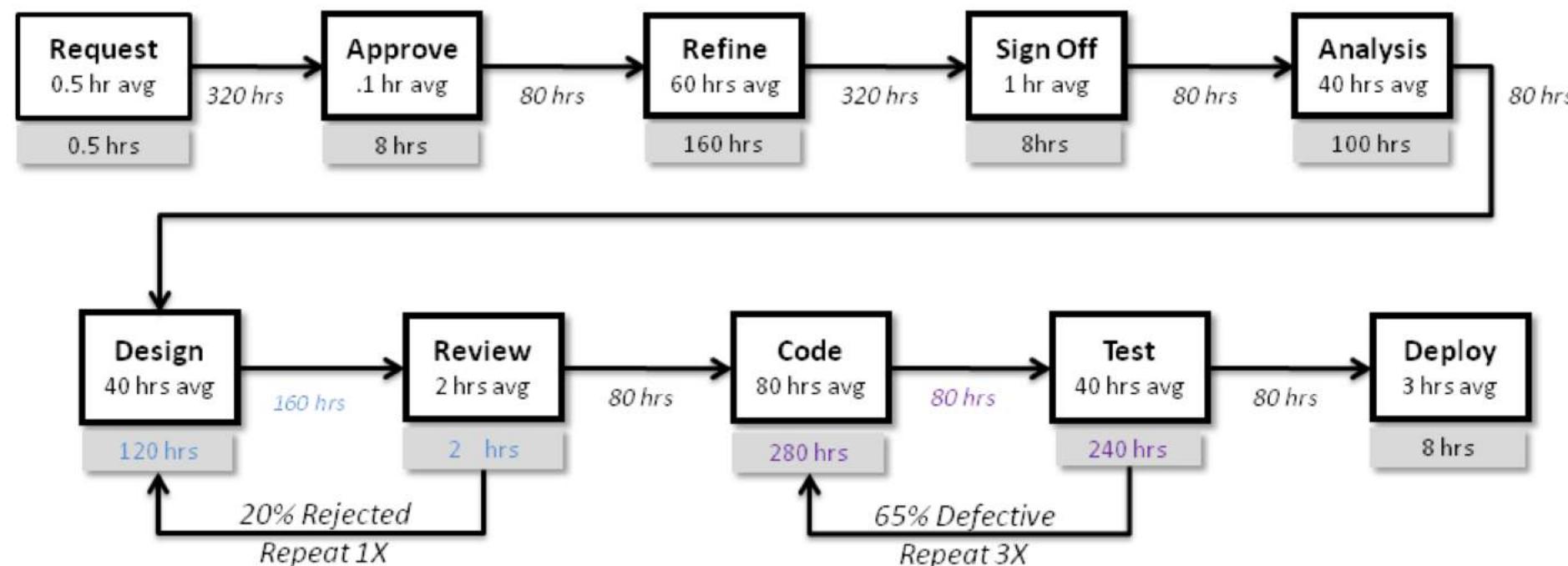
No contexto Lean, Valor é a percepção do cliente acerca do resultado do nosso produto ou serviço.

- O que é importante para meu cliente?
- Qual problema do cliente eu resolvo?
- Quais expectativas devem ser atendidas?
- Qual preço meu cliente está disposto a pagar?



No Lean, definimos nosso fluxo de valor como o *processo necessário para converter uma hipótese de negócios em um produto ou serviço que ofereça valor ao cliente.*

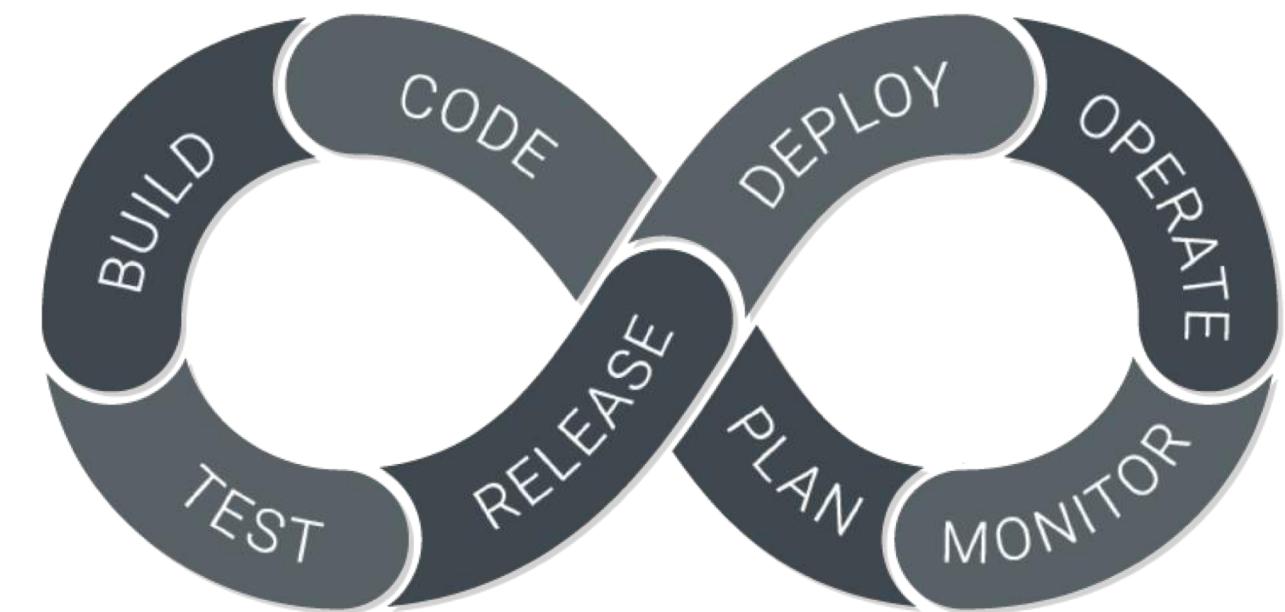
Para desenhar o fluxo de valor nós usamos a ferramenta Mapa de Fluxo de Valor.



A palavra DevOps é uma contração de 'Desenvolvimento' e 'Operações'.

O DevOps combina uma estrutura de **Entrega Contínua** altamente automatizada, repetível e testável com uma equipe unificada que se estende por todo o ciclo de vida do fornecimento de recursos à produção.

Com uma abordagem de DevOps, podemos levar pequenos lotes de requisitos rapidamente do conceito à implantação, monitorar os resultados e usar o que aprendemos para alterar o produto e o processo.

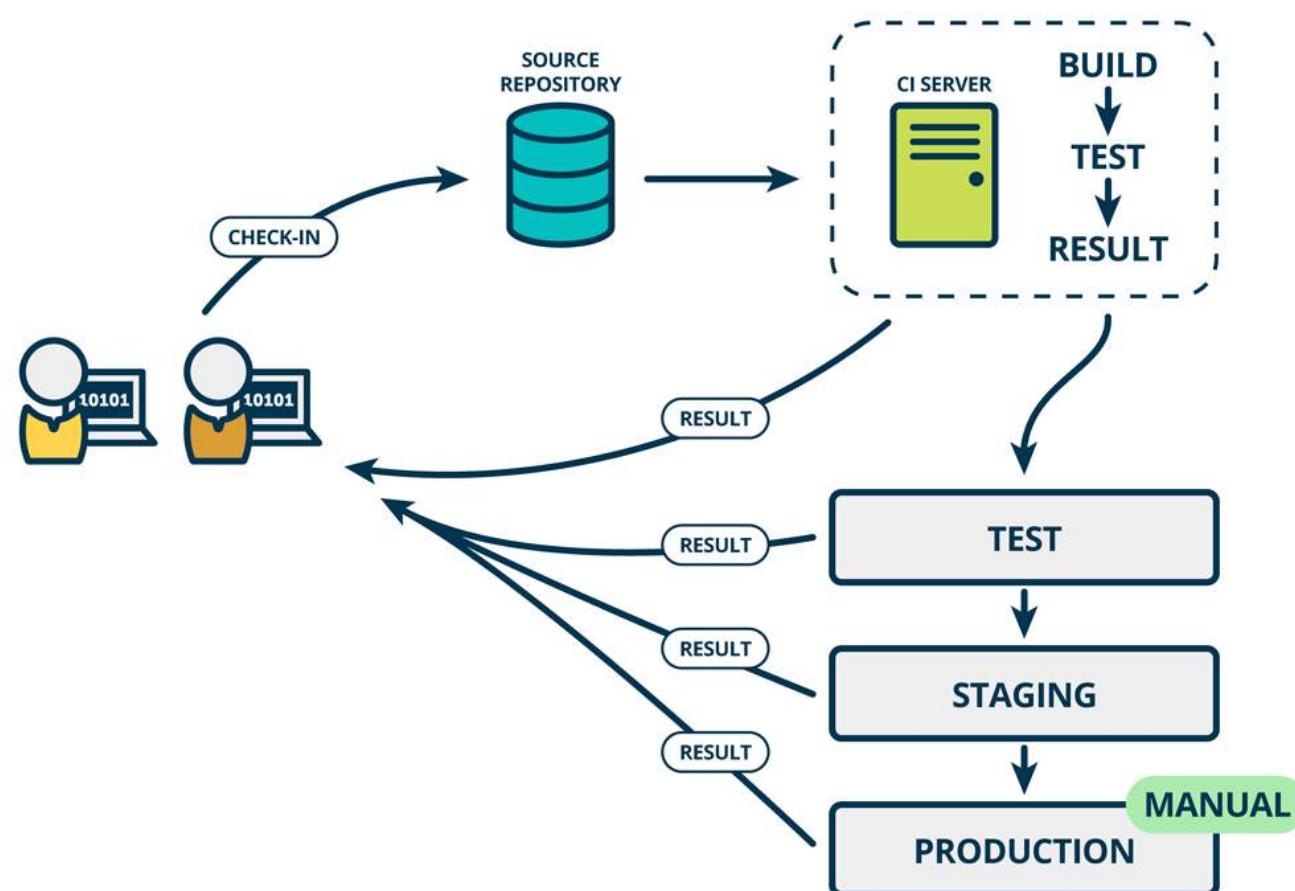


Nossa maior prioridade é ***satisfazer o cliente***, através da entrega adiantada e contínua de software de valor

Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos

Software funcional é a medida primária de progresso

A entrega contínua é uma extensão da integração contínua para garantir que você possa liberar novas alterações para seus clientes rapidamente e de maneira sustentável. Isso significa que, além de automatizar seus testes, você também automatizou seu processo de liberação e pode implantar sua aplicação a qualquer momento, com o clique de um botão.





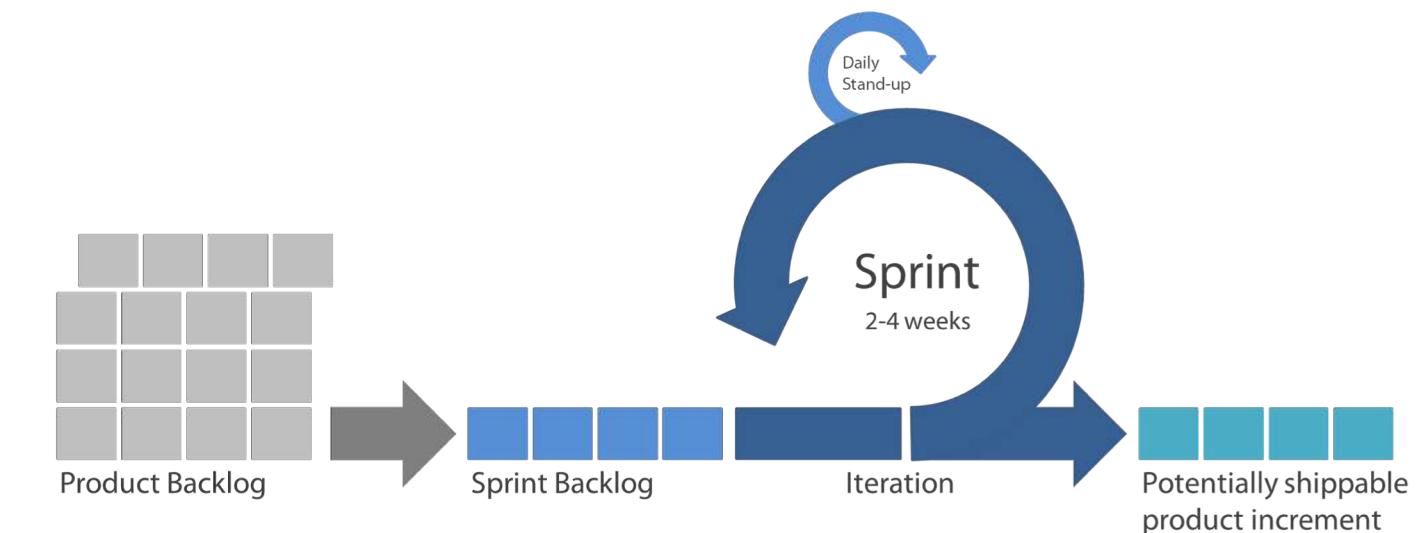
Módulo 2

O framework Scrum

Scrum (subs): Um framework dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível.

Scrum é:

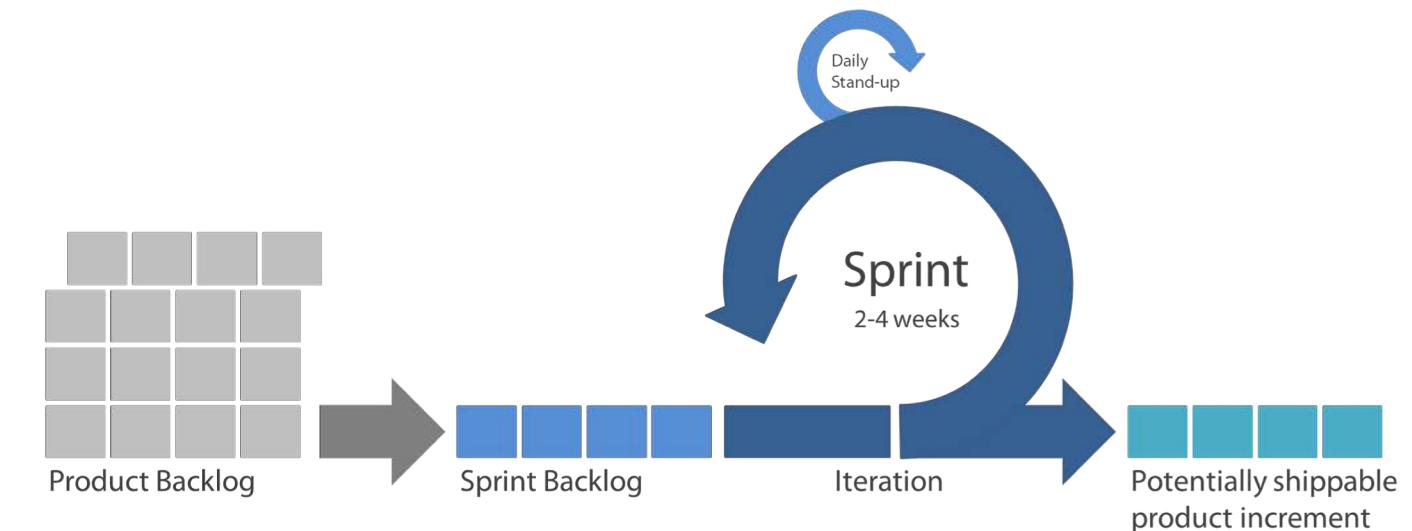
- Leve
- Simples de entender
- Difícil de dominar



Scrum é um framework estrutural que está sendo usado para gerenciar o trabalho em produtos complexos desde o início de 1990.

Scrum não é um processo, técnica ou um método definitivo. Em vez disso, é um framework dentro do qual você pode empregar vários processos ou técnicas.

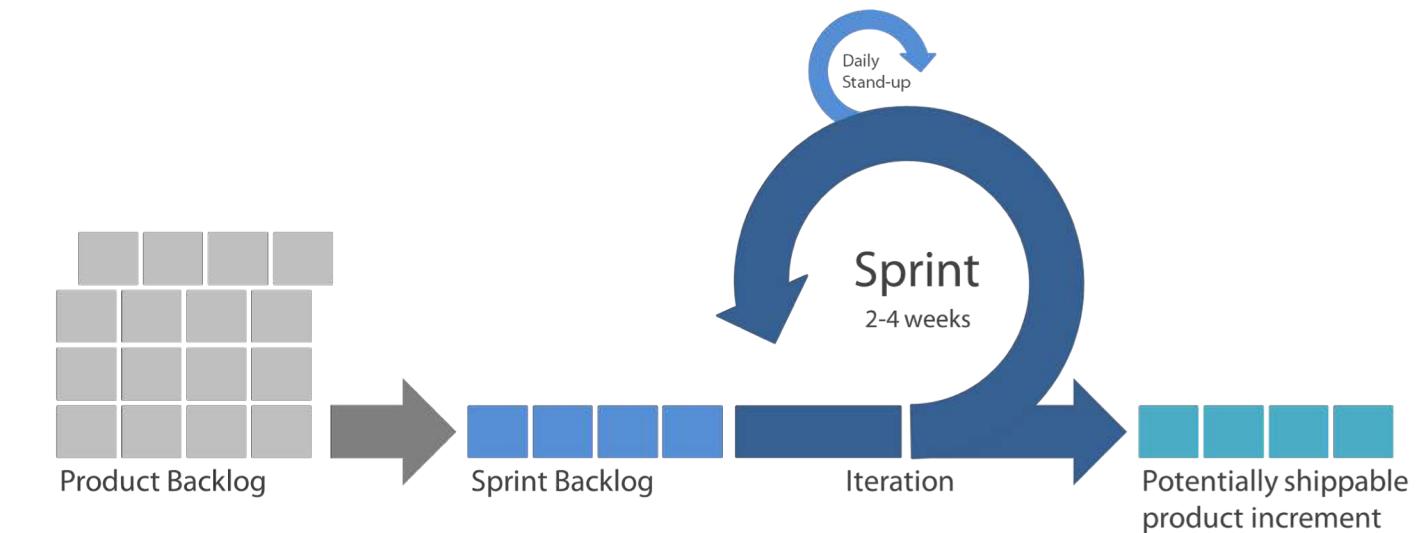
As regras do Scrum integram os papéis, eventos e artefatos, administrando as relações e interações entre eles.



Scrum é fundamentado nas teorias empíricas de controle de processo, ou empirismo.

O empirismo afirma que o conhecimento vem da experiência e de tomada de decisões baseadas no que é conhecido.

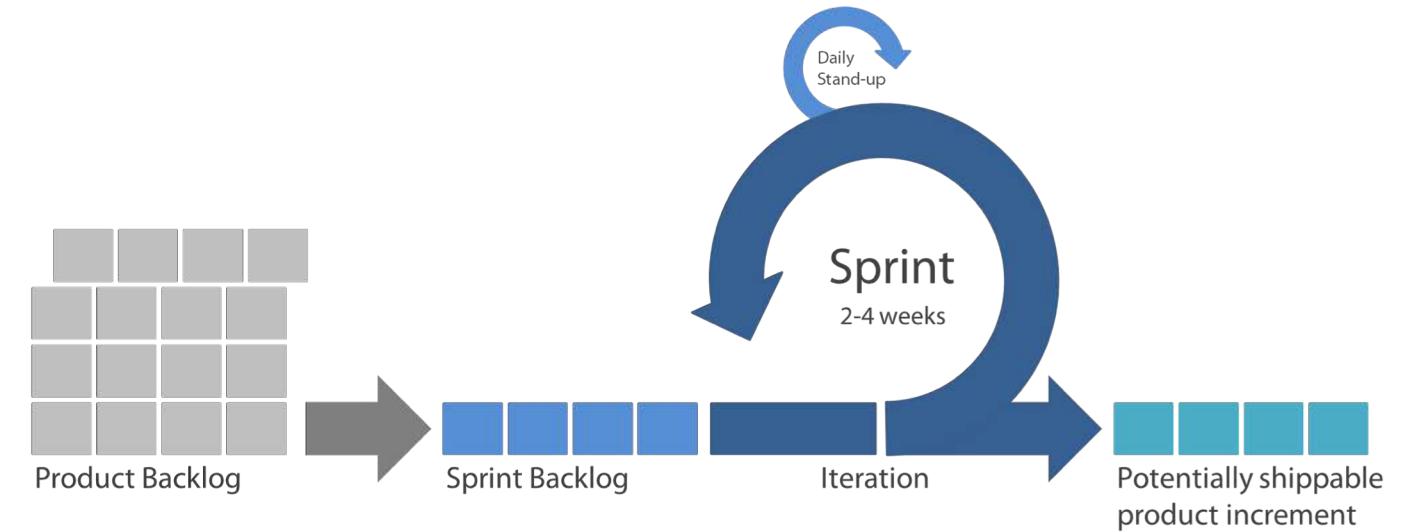
O Scrum emprega uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e o controle de riscos.



O framework Scrum

Vantagens do Scrum

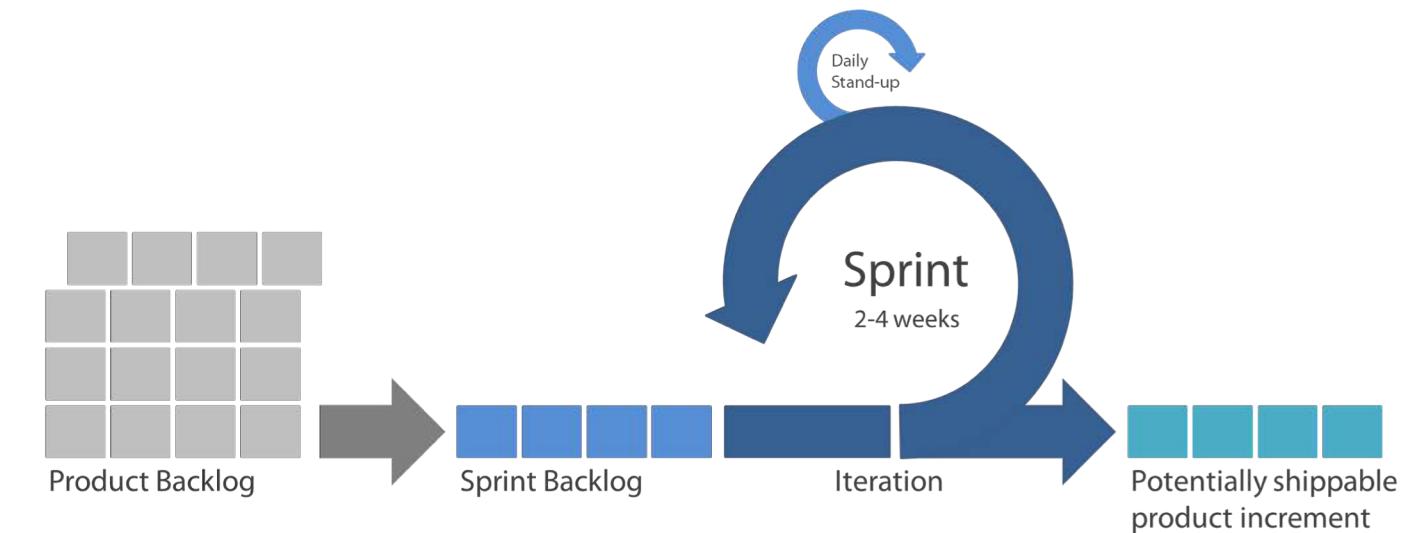
- Maior produtividade e menores custos
- Aumento do engajamento da equipe
- Time-to-Market mais rápido
- Maior Qualidade
- Aumento de satisfação das partes interessadas



O framework Scrum

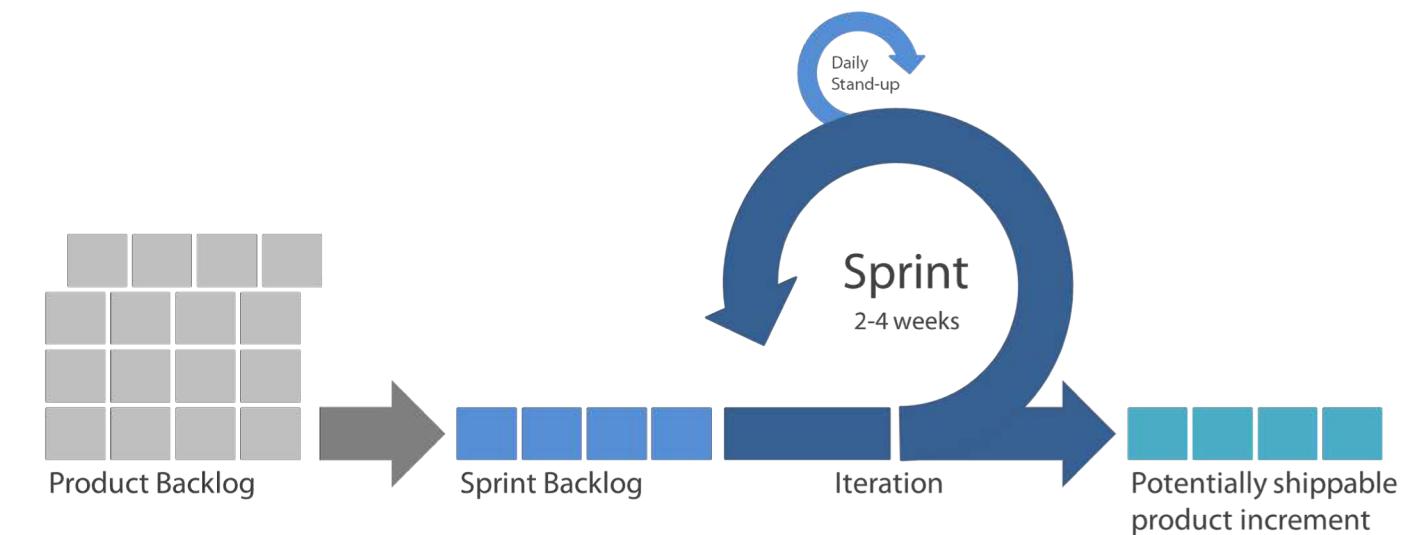
Quando não usar o Scrum

- Quando não há receptividade por parte das equipes de execução
- Quando a alta gerência ou a diretoria executiva não apoiam a mudança
- Quando a empresa não é receptiva à mudanças
- Quando existe um grande apego às boas-práticas



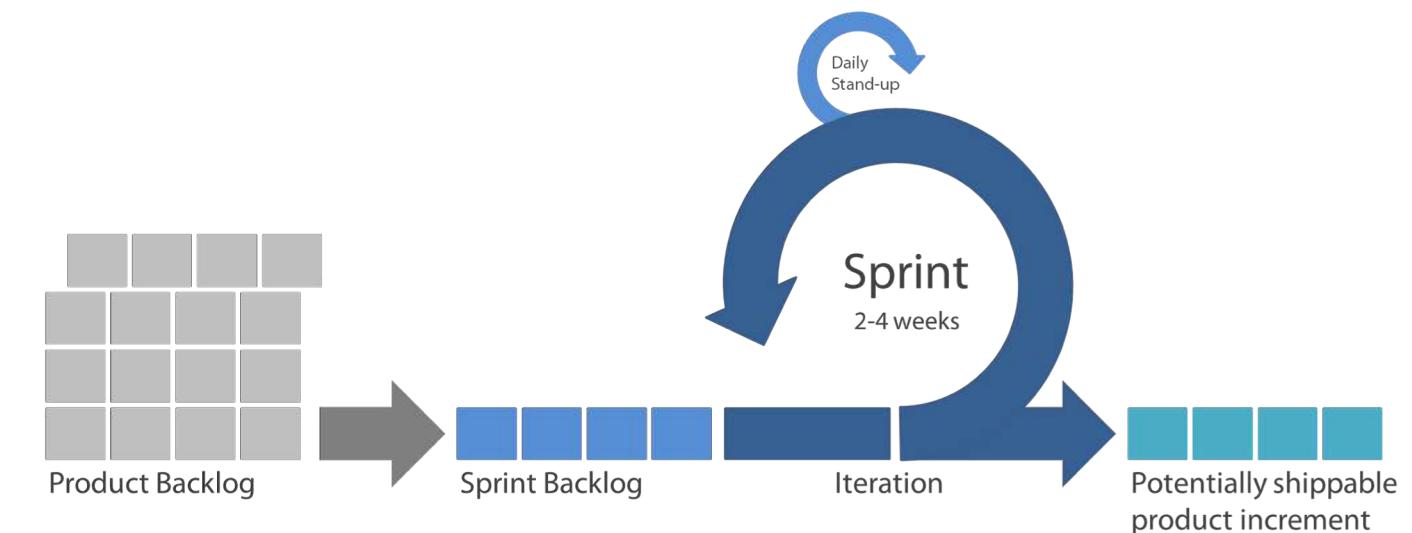
Três pilares apoiam a implementação de controle de processo empírico:

- Transparência
- Inspeção
- Adaptação



Estes cinco valores do Scrum são fundamentais para se atingir a alta performance:

- Coragem
- Foco
- Comprometimento
- Respeito
- Abertura



Existem três papéis em um projeto Scrum - nem mais nem menos.

Não podemos definir nenhum outro papel, porque é prejudicial à unidade da equipe e não é compatível com a filosofia do Scrum.

Product Owner: 1 pessoa, em tempo integral ou parcial, orientada para negócios

Scrum Master: 1 pessoa, em tempo integral ou parcial, especialista em Scrum

Time de Desenvolvimento: 3 a 9 pessoas, de preferência em tempo integral, especialistas em desenvolvimento

O termo **Time Scrum** se refere a todas as pessoas que participam de um projeto Scrum, ou seja: Product Owner, Scrum Master e Time de Desenvolvimento.

O Product Owner, ou dono do produto, é o responsável por maximizar o valor do produto resultado do trabalho do Time de Desenvolvimento.

É a única pessoa responsável por gerenciar o Backlog do Produto, que envolve:

- Expressar claramente os itens do Backlog do Produto
- Ordenar os itens do Backlog do Produto para alcançar melhor as metas e missões
- Otimizar o valor do trabalho que o Time de Desenvolvimento realiza
- Garantir que o Backlog do Produto seja visível, transparente, claro para todos, e mostrar o que o Time Scrum vai trabalhar a seguir
- Garantir que o Time de Desenvolvimento entenda os itens do Backlog do Produto no nível necessário

É responsável por medir o desempenho do projeto, prever a data de conclusão e tornam essas informações transparentes para todas as partes interessadas.

O Scrum Master é responsável por promover e suportar o Scrum como definido no Guia Scrum. Ele faz isso ajudando todos a entenderem teorias, práticas, regras e os valores do Scrum com o objetivo de maximizar o valor criado pelo Time Scrum.

Seu papel é de um líder-servidor que depende de uma estreita colaboração com os outros membros da equipe Scrum, e ainda assim não tem autoridade formal sobre eles.

Ele busca remover impedimentos, facilitar eventos, treinar e atuar com um coach para o time de desenvolvimento.

Auxilia o Product Owner por ensinar técnicas relacionadas ao backlog do produto, propagando informações e facilitando eventos.

Pode também atuar liderando esforços de adoção do Scrum em toda a empresa.

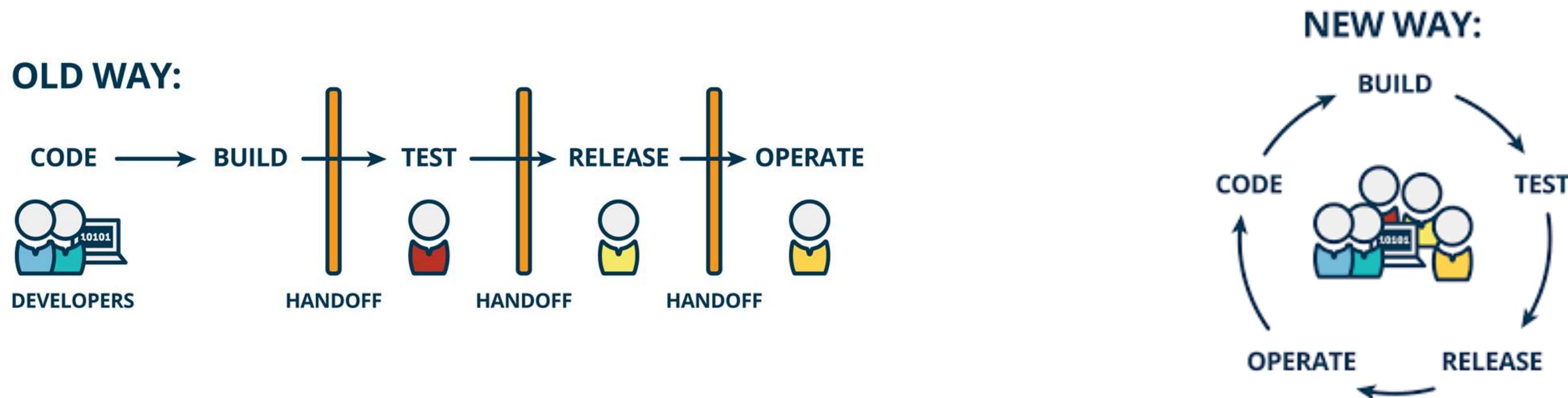
São os profissionais responsáveis por desenvolver e entregar os itens do backlog, gerenciando seus próprios esforços para isso.

São multi-funcionais, sendo capazes de executar todas as atividades necessárias para a entrega de um item do backlog do produto.

São auto-organizados, encontrando suas próprias formas ao invés de receberem ordens.

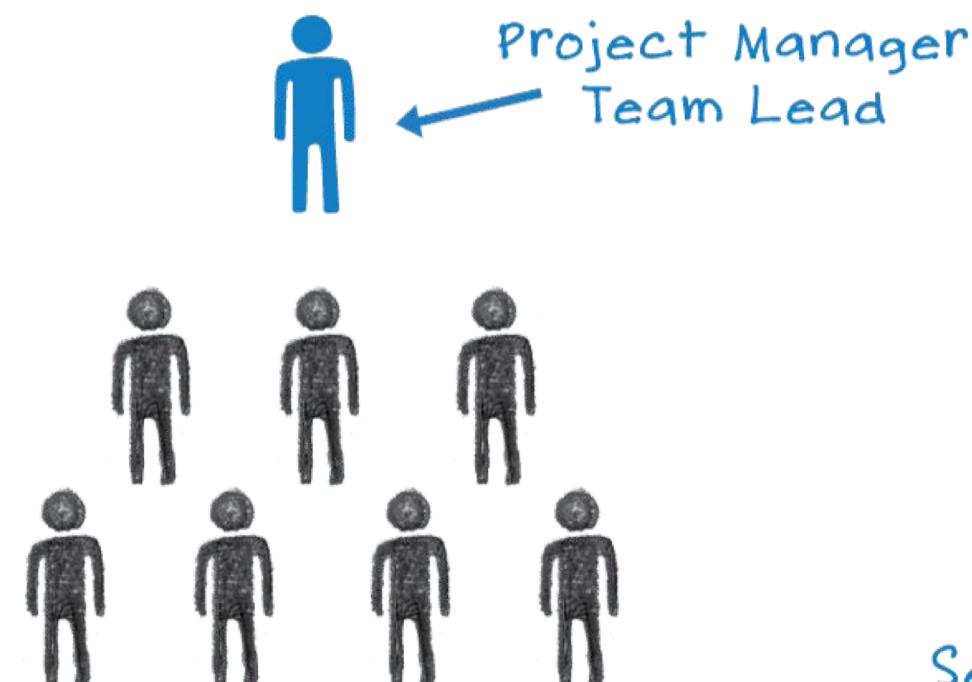
Mesmo quando uma atividade fica sob responsabilidade de um determinado membro do time durante uma sprint, todo o Time de Desenvolvimento permanece sendo o responsável pela entrega da atividade. Não existe responsabilidade individual.

Multifuncional: O Time Scrum tem todo o conhecimento e as competências necessárias para realizar o trabalho sem qualquer ajuda de fora da equipe.

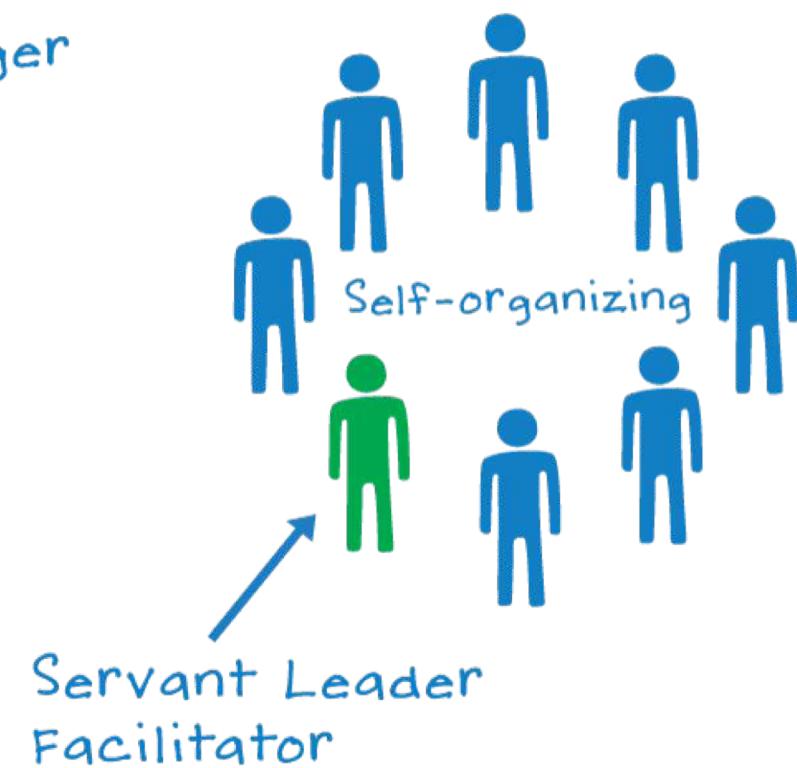


Auto-organizado: O Time Scrum gerencia seus próprios esforços, em vez de ser gerenciado ou dirigido por outros.

Traditional Teams



Agile Teams



Não existem outros papeis no Scrum.

Não existem títulos ou especialidades dentro de um Time de Desenvolvimento, todos devem ser reconhecidos como Membros do Time de Desenvolvimento.

Não existe a figura do Project Manager no Scrum, as responsabilidades de um Project Manager são distribuídas entre os três papéis do Scrum.

Eventos bem definidos são usados no Scrum para criar uma regularidade e minimizar a necessidade de reuniões não definidas no Scrum.

O Scrum reforça a ideia de que todos os eventos do time precisam ter um timebox e que é muito importante que esses tempos sejam respeitados para se manter um ritmo sustentável.

Os eventos do Scrum são:

- Sprint
 - Sprint Planning
 - Daily Scrum
 - Sprint Review
 - Sprint Retrospective

Estes eventos são definidos para termos transparência, inspeção, regularidade e adaptação.

No Scrum os eventos são ferramentas de comunicação e é importante preparar um ambiente adequado para isso.

Um requisito é ter o time fisicamente na mesma sala, ao invés de estarem distribuídos em vários locais. Isso aumenta o relacionamento entre os membros e facilita a colaboração.

Ter todo o time em uma mesma sala contribui para a **comunicação osmótica**, onde as pessoas podem obter informações úteis ouvindo, envolvendo-se e ajudando-se mutuamente, conforme necessário.

Quando trabalhamos com times distribuídos precisamos nos adaptar para conseguir a comunicação osmótica. Um exemplo é sempre copiar todo o time nos e-mails trocados.

Também precisamos tirar o máximo proveito de ferramentas de comunicação para facilitar a interação do time.

O coração do Scrum é a Sprint, um time-boxed de um mês ou menos, durante o qual um incremento de produto é criado. Um projeto é composto por várias Sprints.

Sprints têm durações consistentes ao longo de todo o esforço de desenvolvimento.

As Sprints contém os seguintes eventos:

- Sprint Planning
- Daily Scrum
- Trabalho de Desenvolvimento
- Sprint Review
- Sprint Retrospective

Durante a Sprint:

- Não são feitas mudanças que possam por em perigo o objetivo da Sprint
- As metas de qualidade não diminuem
- O escopo pode ser clarificado e renegociado entre o Product Owner e o Time de Desenvolvimento conforme o time for aprendendo mais do escopo.

Geralmente são adotadas sprints de 2 a 4 semanas.

Um tempo maior que 4 semanas poderá causar problemas por não podermos fazer mudanças de rumo durante todo este período. Isso aumenta a complexidade e os riscos do projeto.

Por outro lado, se tivermos sprints muito curtas (menores que 2 semanas), pode ser que não consigamos entregar incrementos potencialmente utilizáveis neste período.

Devemos definir o tamanho da sprint no início do projeto, de acordo com suas características. Podemos mudar sua duração no decorrer do projeto, mas isso não pode ser feito com frequência pois impactará na previsibilidade das entregas.

Nós não mudamos a duração de uma sprint durante sua execução.

Não existe Sprint 0! Não há diferença entre a primeira Sprint e as demais.

Apesar disso, obviamente no começo de cada projeto precisaremos de alguns dias para criar o estado inicial do nosso Product Backlog.

Cancelamento de uma Sprint

Uma Sprint pode ser cancelada antes do time-boxed da Sprint terminar. Somente o Product Owner tem a autoridade para cancelar a Sprint, embora ele (ou ela) possa fazer isso sob influência das partes interessadas, do Time de Desenvolvimento ou do Scrum Master.

A Sprint poderá ser cancelada se o objetivo da Sprint se tornar obsoleto. Isto pode ocorrer se a organização mudar sua direção ou se as condições do mercado ou das tecnologias mudarem.

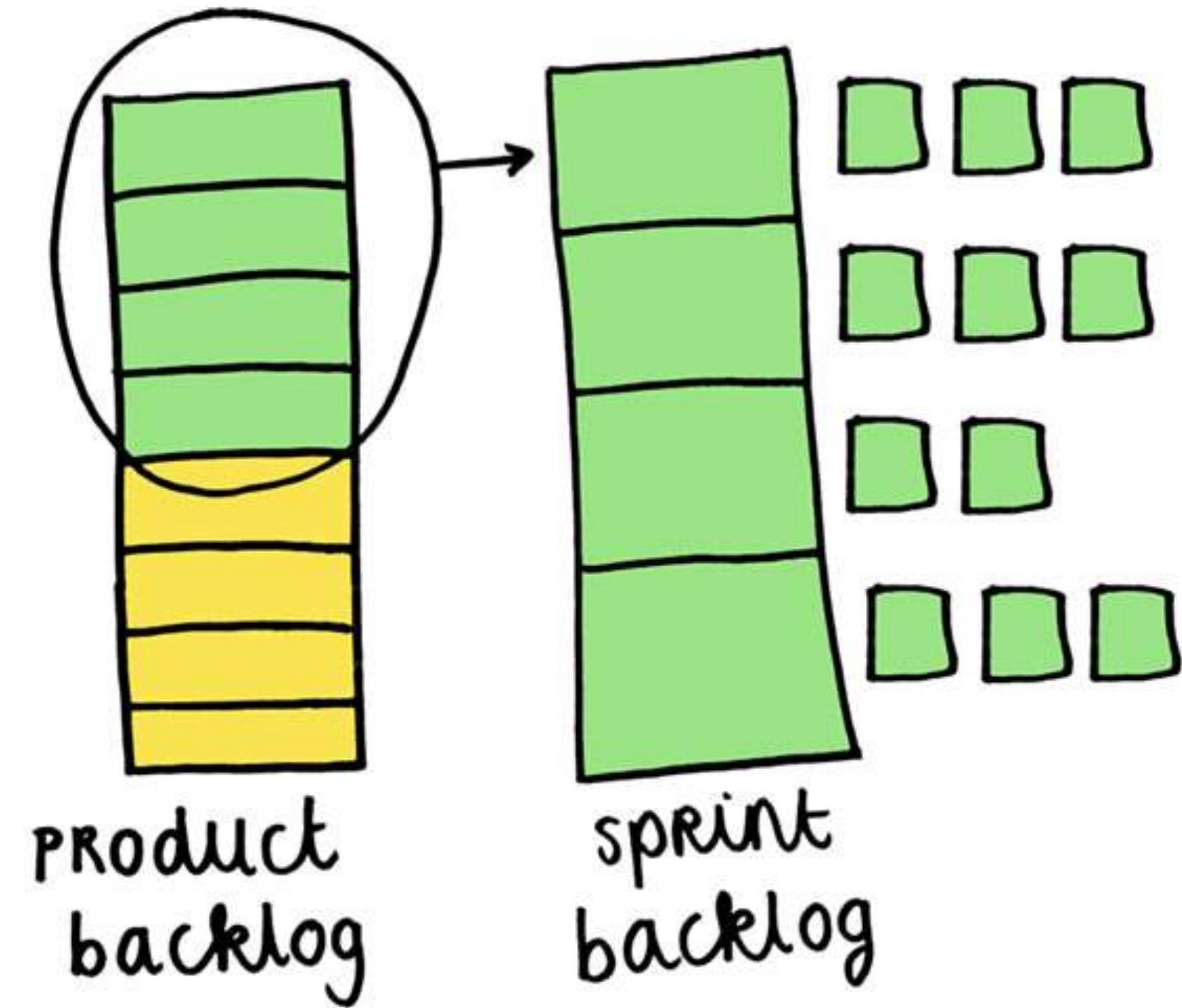
Geralmente a Sprint deve ser cancelada se ela não faz mais sentido às dadas circunstâncias. No entanto, devido a curta duração da Sprint, raramente cancelamentos fazem sentido.

O trabalho a ser realizado na Sprint é planejado durante a Sprint Planning e é criado com o trabalho colaborativo de todo o Time Scrum (Product Owner, Scrum Master e Time de Desenvolvimento).

A Equipe de desenvolvimento estima quanto trabalho eles podem fazer em uma única Sprint, e as histórias do usuário são escolhidas. Os itens são movidos do backlog de Produto para o Backlog da Sprint.

O Planejamento da Sprint é um time-boxed com no máximo oito horas para uma Sprint de um mês de duração. Para Sprints menores, este evento é proporcionalmente menor.

Após a definição de quais itens entrarão naquela sprint, o próximo passo é definir a meta da Sprint.



A Reunião Diária do Scrum é um evento time-boxed de 15 minutos para o Time de Desenvolvimento. Nela o Time de Desenvolvimento planeja o trabalho para as próximas 24 horas.

Nesta reunião, cada membro do time de desenvolvimento deverá responder às perguntas:

- O que eu fiz ontem que ajudou o Time de Desenvolvimento a atingir a meta da Sprint?
- O que eu farei hoje para ajudar o Time de Desenvolvimento atingir a meta da Sprint?
- Eu vejo algum obstáculo que impeça a mim ou o Time de Desenvolvimento no atingimento da meta da Sprint?

Horário e local constantes são os mais recomendados para estabelecer um padrão de consistência da equipe, eliminando complexidade.

A Reunião Diária é uma reunião interna do Time de Desenvolvimento. Se outros estiverem presentes, o Scrum Master deve garantir que eles não perturbem a reunião.

A Sprint Review é realizada no final da Sprint para inspecionar o incremento e adaptar o Backlog do Produto, se necessário. Nesta reunião o Time Scrum e as partes interessadas discutem sobre o que foi feito na Sprint e como melhor priorizar o backlog do produto para a próxima sprint.

Esta é uma reunião informal, não uma reunião de status, e a apresentação do incremento destina-se a motivar e obter feedback e promover a colaboração.

Antes da Sprint Review, o Product Owner deverá validar se os itens estão realmente prontos, pois somente serão apresentados itens que estiverem 100% prontos. 99% pronto é igual a 0% pronto.

A duração é de no máximo 4 horas para uma Sprint de um mês e proporcionalmente menor para sprints menores.

Sprint Retrospective

A Retrospectiva da Sprint é uma oportunidade para o Time Scrum inspecionar a si próprio e criar um plano para melhorias a serem aplicadas na próxima Sprint.

A Retrospectiva da Sprint ocorre depois da Sprint Review e antes da próxima Sprint Planning.

Esta é uma reunião de no máximo três horas para uma Sprint de um mês. Para Sprint menores, este evento é usualmente menor.

O Scrum Master garante que o evento ocorra e que os participantes entendam seu propósito.

O propósito da Retrospectiva da Sprint é:

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

Timebox é um conceito muito importante no Agile. É ela que define o ritmo e dá a regularidade para o time.

Daily Scrum: 15 minutos

Sprint: entre 2 e 4 semanas. Nunca mais que 1 mês.

Sprint Planning: máximo de 8 horas para sprint de 1 mês. *

Sprint Review: máximo de 4 horas para sprint de 1 mês. *

Sprint Retrospective: máximo de 3 horas para sprint de 1 mês. *

* É proporcionalmente menor para sprints menores.



Os artefatos do Scrum representam o trabalho ou o valor para o fornecimento de transparência e oportunidades para inspeção e adaptação.

Os artefatos definidos para o Scrum são especificamente projetados para maximizar a transparência das informações chave de modo que todos tenham o mesmo entendimento dos artefatos.

O Backlog do Produto é uma lista ordenada de tudo que é conhecido ser necessário no produto. É a única origem dos requisitos para qualquer mudança a ser feita no produto.

O Product Owner é responsável pelo Backlog do Produto, incluindo seu conteúdo, disponibilidade e ordenação.

Um Backlog do Produto nunca está completo. Os primeiros desenvolvimentos estabelecem os requisitos inicialmente conhecidos e melhor entendidos.

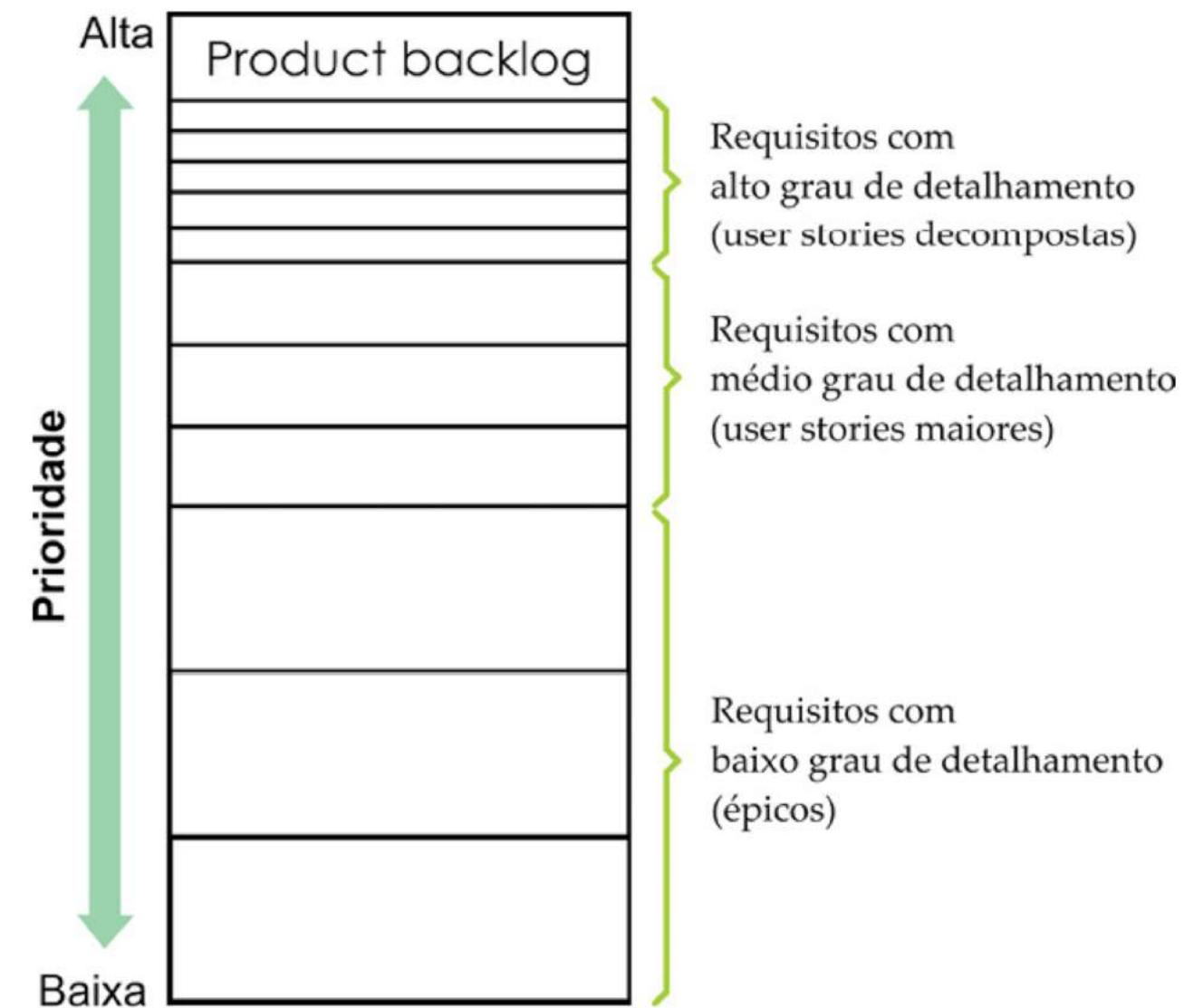
Geralmente, os critérios para priorização do backlog seguem o seguinte conceito:

- Benefícios
- Custo
- Riscos

Os itens do Backlog do Produto de ordem mais alta (topo da lista) devem ser mais claros e mais detalhados que os itens de ordem mais baixa. Estimativas mais precisas são feitas baseadas em maior clareza e maior detalhamento.

Quanto menor a ordem na lista, menos detalhes.

Os itens do Backlog do Produto que irão ocupar o Time de Desenvolvimento na próxima Sprint são mais refinados, de modo que todos os itens possam ser “Prontos” dentro do time-boxed da Sprint.

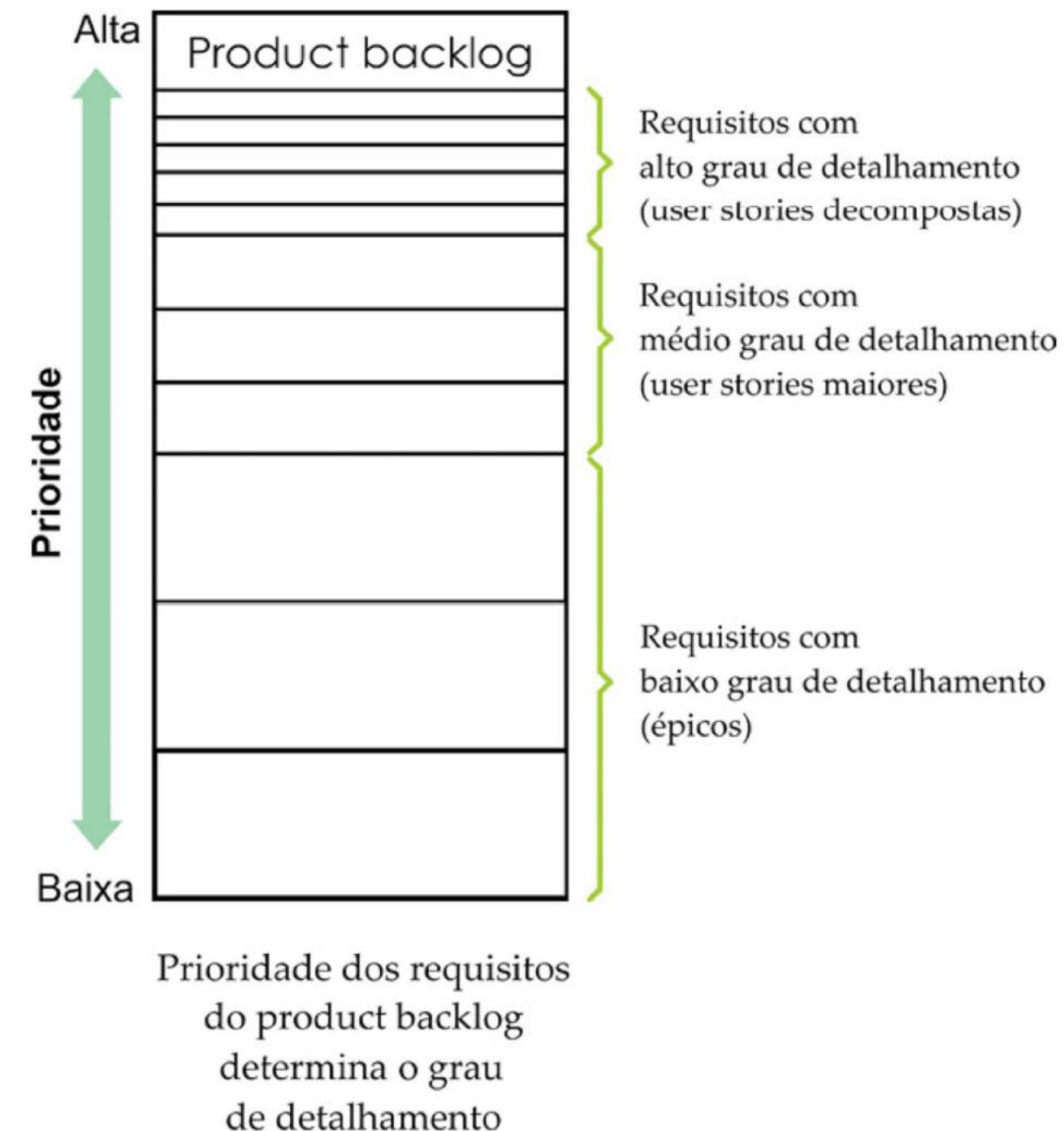


Prioridade dos requisitos
do product backlog
determina o grau
de detalhamento

O refinamento do Backlog do Produto é a ação de adicionar detalhes, estimativas e ordem aos itens no Backlog do Produto.

Este é um processo contínuo no qual o Product Owner e o Time de Desenvolvimento colaboram nos detalhes dos itens do Backlog do Produto. Durante o refinamento do Backlog do Produto, os itens são inspecionados e revisados.

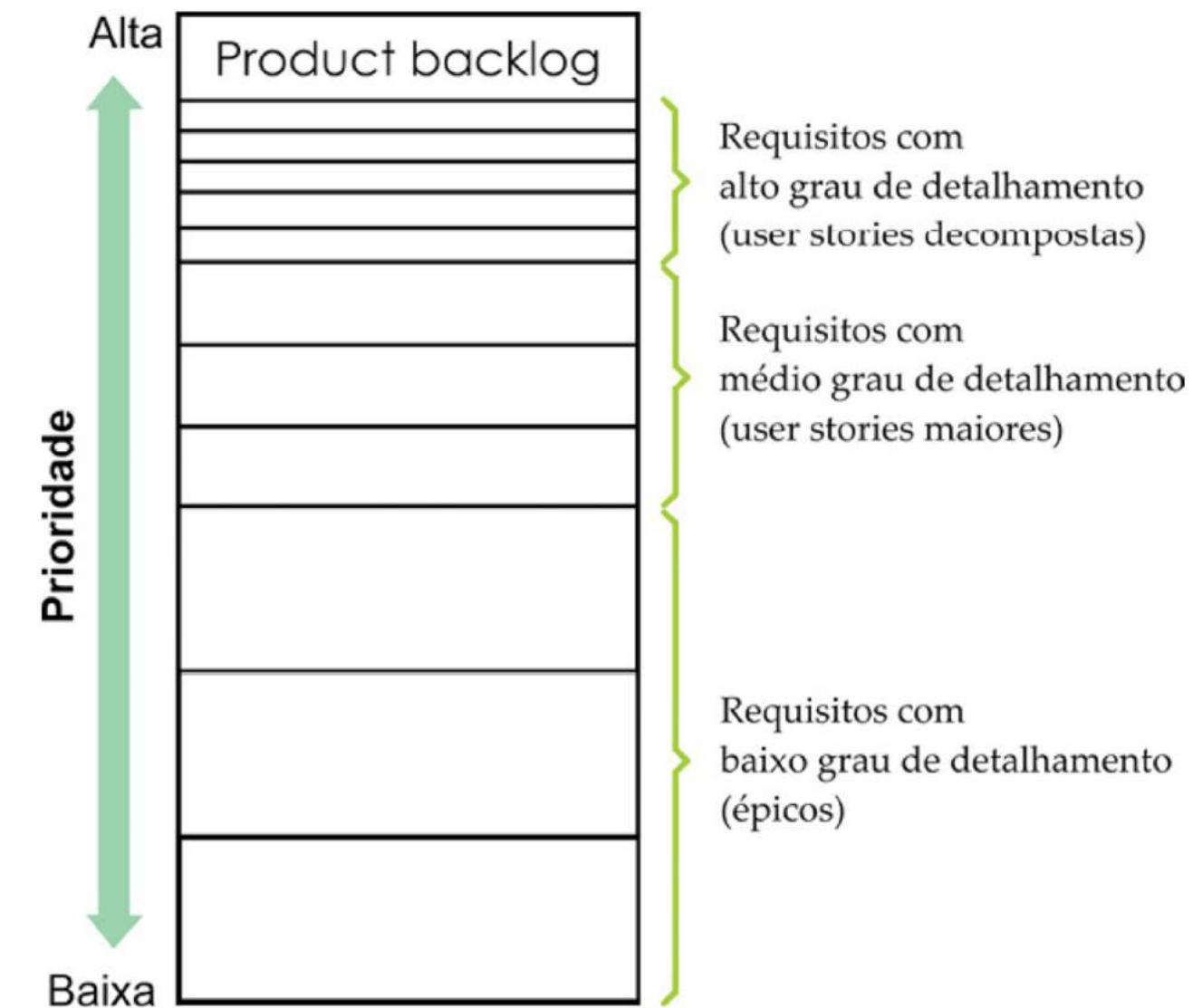
Os itens do Backlog do Produto podem ser atualizados a qualquer momento pelo Product Owner ou a critério do Product Owner.



Os itens do Backlog do Produto que podem ser “Prontos” pelo Time de Desenvolvimento dentro de uma Sprint são considerados “Preparados” para seleção no Planejamento da Sprint.

Geralmente adquirem este grau de transparência através das atividades de refinamento.

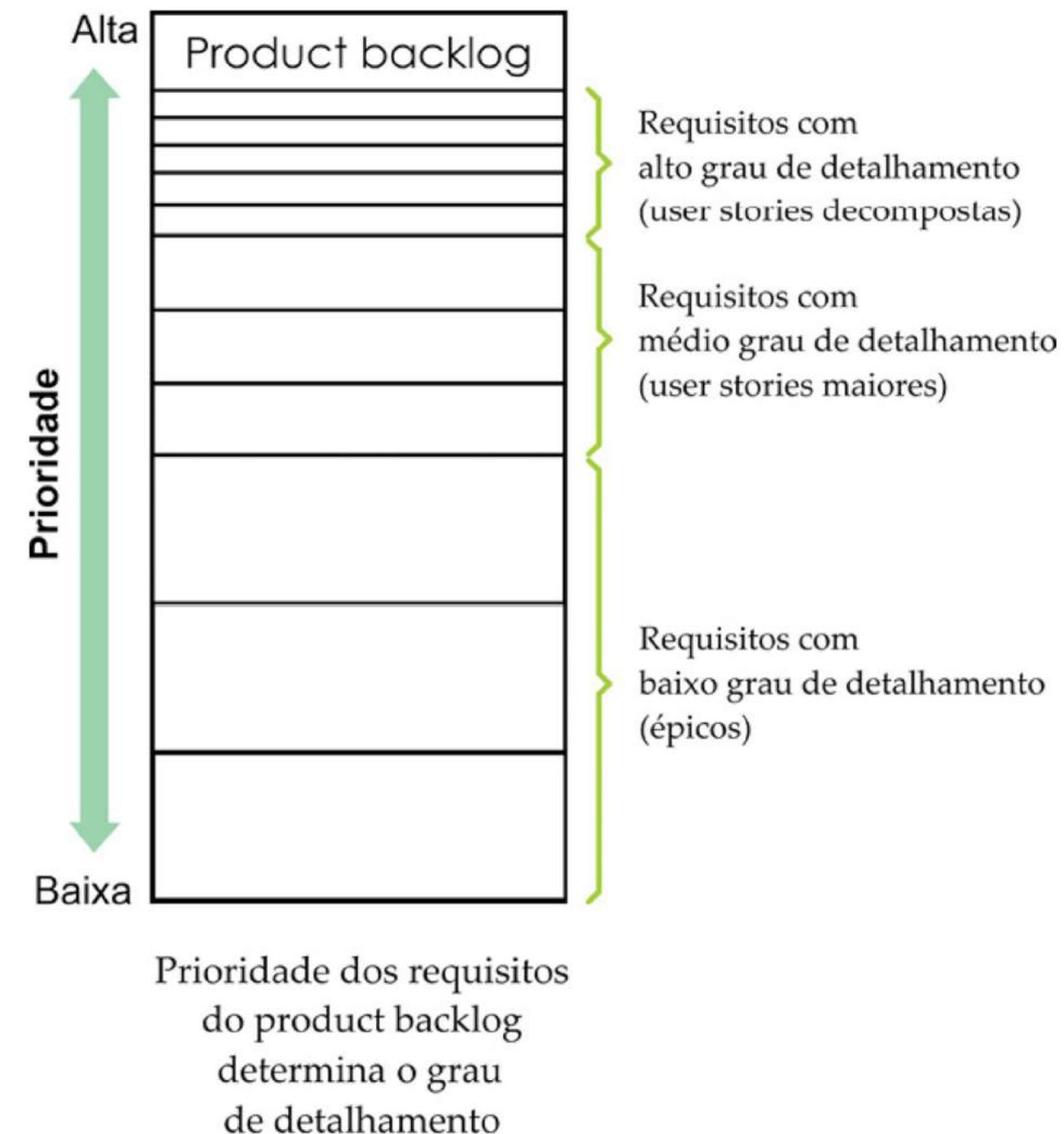
Os Itens de backlog de Produto devem ser não técnicos e independentes porque deve ser possível organizar os itens de acordo com o valor.



Prioridade dos requisitos
do product backlog
determina o grau
de detalhamento

O Time de Desenvolvimento é responsável por todas as estimativas.

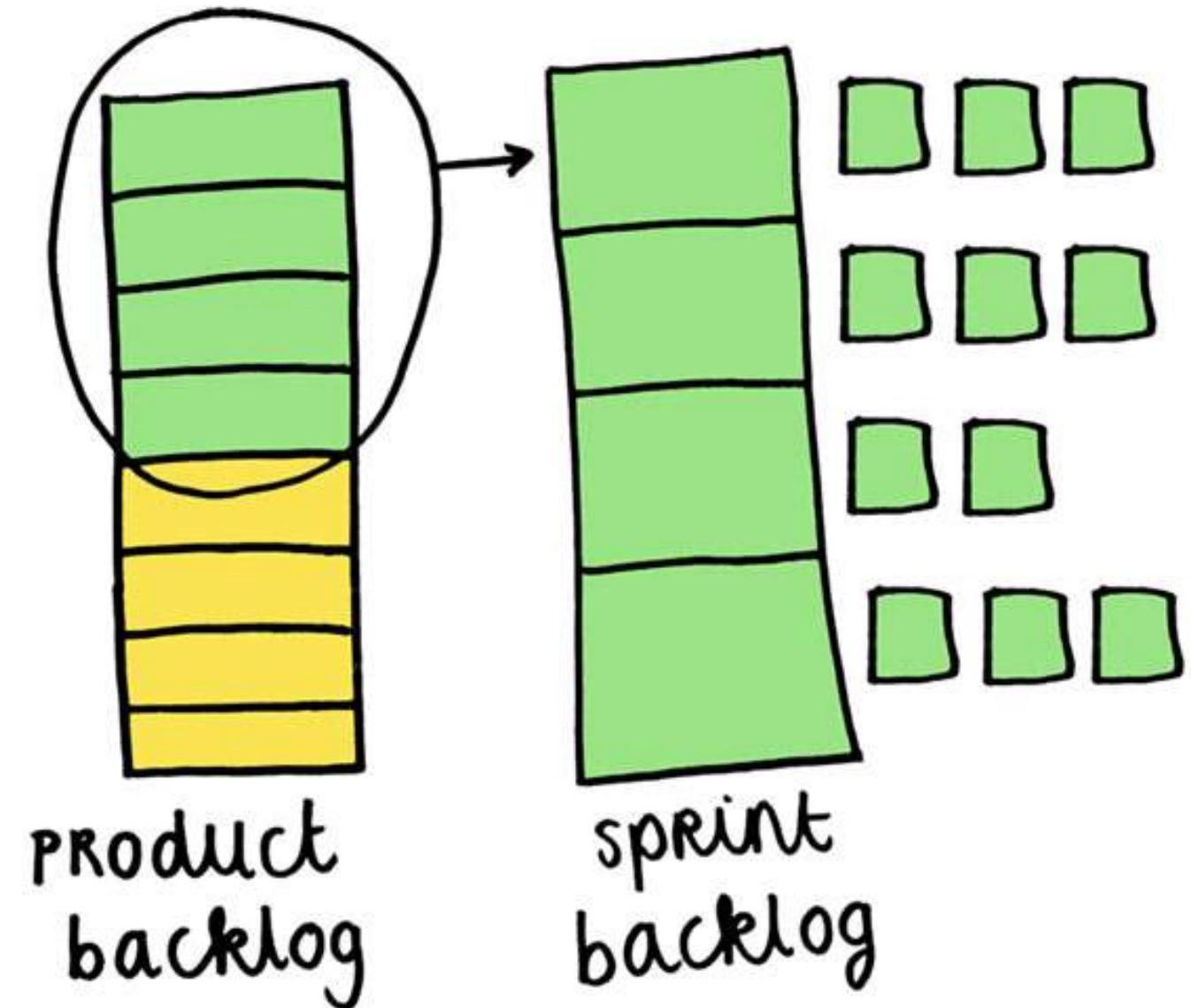
O Product Owner deve influenciar o Time de Desenvolvimento, ajudando no entendimento e nas decisões conflituosas de troca, mas as pessoas que irão realizar o trabalho fazem a estimativa final.



Sprint Backlog

O Backlog da Sprint é um conjunto de itens do Backlog do Produto selecionados para a Sprint, juntamente com o plano para entregar o incremento do produto e atingir o objetivo da Sprint.

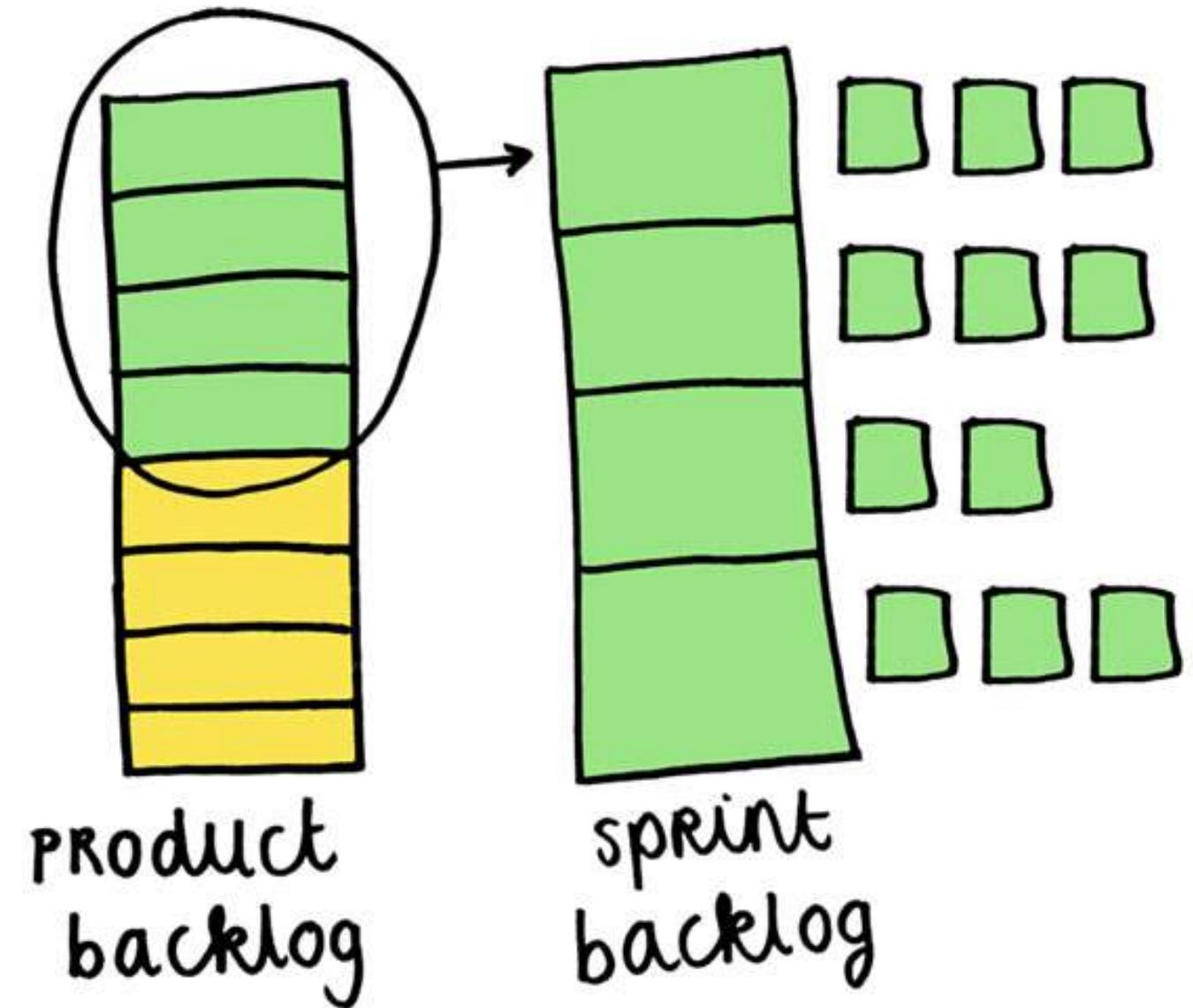
O Backlog da Sprint é a previsão do Time de Desenvolvimento sobre qual funcionalidade estará no próximo incremento e sobre o trabalho necessário para entregar essa funcionalidade em um incremento “Pronto”.



Sprint Backlog

O Backlog da Sprint torna visível todo o trabalho que o Time de Desenvolvimento identifica como necessário para atingir o objetivo da Sprint.

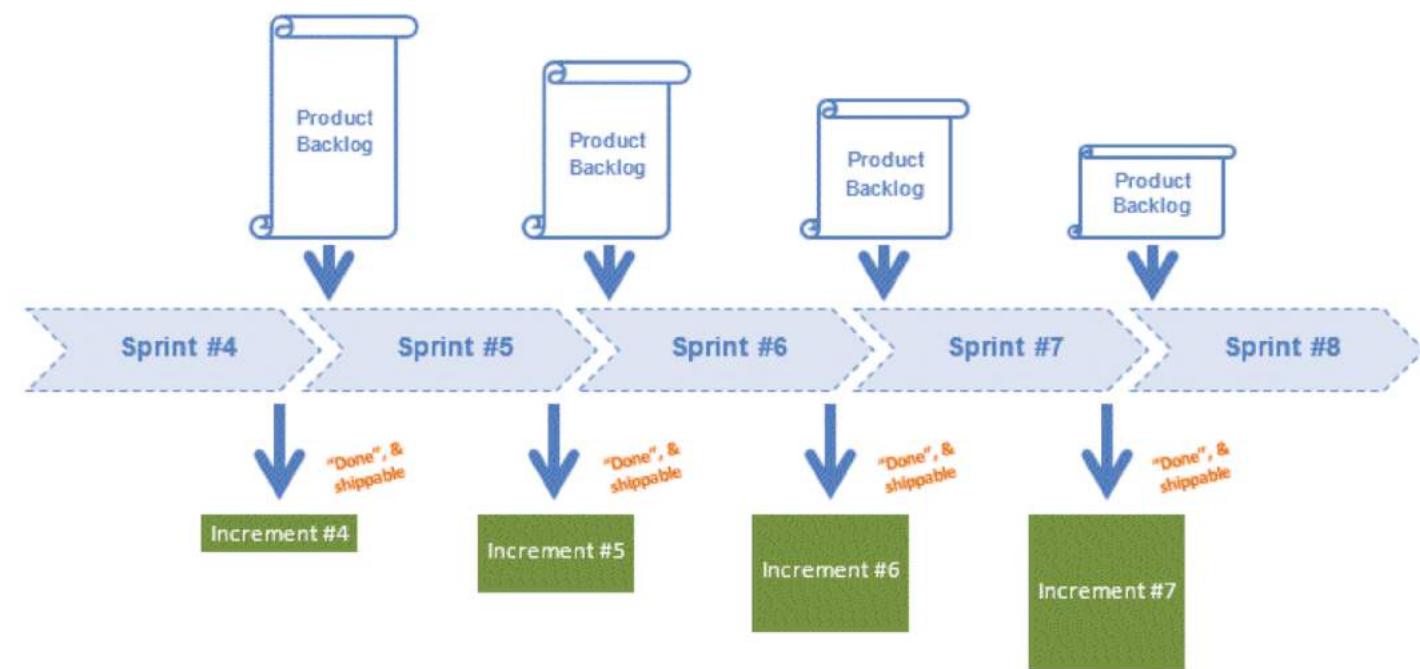
Para garantir melhoria contínua, é incluído no mínimo um item de prioridade alta sobre melhoria do processo identificado na última Reunião de Retrospectiva.



O incremento é a soma de todos os itens do Backlog do Produto completados durante a Sprint e o valor dos incrementos de todas as Sprints anteriores.

Ao final da Sprint um novo incremento deve estar “Pronto”, o que significa que deve estar na condição de ser utilizado e atender a definição de “Pronto” do Time Scrum.

O incremento é um passo na direção de uma visão ou de um objetivo. O incremento deve estar na condição de ser utilizado independente do Product Owner decidir por liberá-lo ou não.



Definição de Pronto (DoD)

É uma descrição dos critérios que todo incremento deve cumprir. Geralmente exige que os itens do Backlog do Produto sejam transformados em software funcional testado e documentado adequadamente.

Antes do primeiro Sprint, o Product Owner deve se reunir com o Scrum Master e com a equipe para criar uma Definição de Pronto que inclua as propriedades que cada incremento deve cumprir.

É importante incluir na DoD que os itens sejam testados, pois uma vez concluídos, eles serão liberados para entrar em produção.

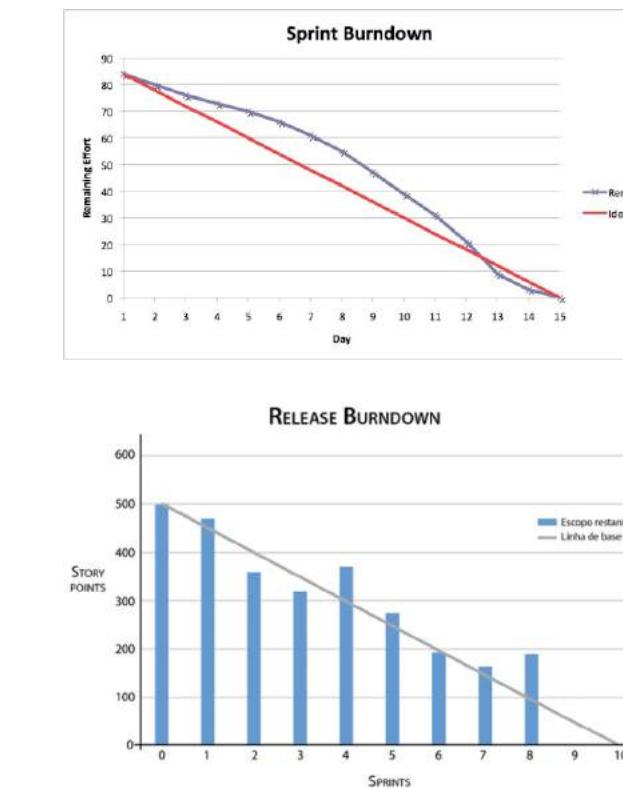
Quando um item do Backlog do Produto ou um incremento é descrito como “Pronto”, todos devem entender o que o “Pronto” significa.

A definição de pronto geralmente contém:

- Processos de desenvolvimento
- Processos organizacionais
- Requisitos não-funcionais
- Critérios de qualidade

Independentemente da quantidade de Times de Desenvolvimento, para um mesmo projeto deve-se adotar uma única definição de “Pronto”. Isso garante o mesmo nível de qualidade na entrega e ajuda aos stakeholders entenderem quando um incremento está de fato pronto.

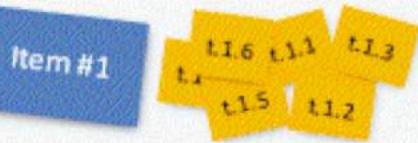
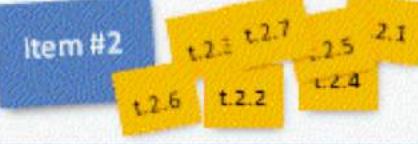
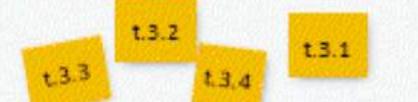
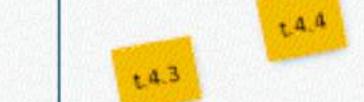
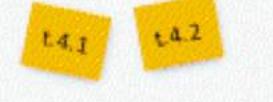
Qualquer tipo de dashboard altamente visível pode ser chamado de um radiador de informação. O objetivo é que qualquer pessoa envolvida no projeto possa facilmente ter acesso às informações de status do projeto ou qualquer outra informação relevante. Isso aumenta a transparência do projeto.



	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Edit name here	😊	😢	😢	😢	😢	😢	😢
Edit name here	😊	😢	😐	😊	😊	😊	😊
Edit name here	😊	😐	😐	😢	😢	😢	😢
Edit name here	😊	😐	😐	😊	😊	😊	😊
Edit name here						😢	😢

Características de Radiadores de Informações:

- Simples
- Atual
- Transitório
- Influente
- Altamente visível
- Mínimo em número

Sprint Goal	To Do	Doing	Done
The goal of this Sprint is to make the purchasing part of the website mature enough to be able to handle the whole process and users can experience a full purchasing process, through which other functionalities of the website will be more meaningful.			
			
			
			
			

Uma história de usuário conta uma história sobre um cliente ou usuário do produto.
Ela contém um nome, uma breve narrativa e critérios de aceitação.

	<p><i>Eu, enquanto <QUEM>, quero <O QUÊ> para <POR QUÊ></i></p>

Histórias de Usuários - Épicos

Épico descreve a essência de um recurso, mas deixa de fora os detalhes.

É um plano inicial que capta a essência do que será entregue, mas deixa os detalhes para mais tarde.

Embora não haja um tamanho mágico, geralmente um épico é uma história de usuário que levará mais de uma sprint para ser concluída.

Histórias de Usuários – Granulação grossa

Como é muito grande e vago para ser entregue em um Sprint, o épico é dividido em várias Histórias de Usuário de granulação grossa.

A História é então decomposta em Histórias de Usuário refinadas. Estas devem ser pequenas o suficiente para caber em um Sprint.

Considere a seguinte história de usuário:

Como usuário, tenho que fazer login no sistema para que minhas informações possam ser acessadas apenas por mim.

Veja um possível conjunto de histórias de usuário menores:

Como usuário registrado, posso efetuar login com meu nome de usuário e senha para que eu possa confiar no sistema.

Como novo usuário, desejo registrar-me criando um nome de usuário e senha para que o sistema possa lembrar minhas informações pessoais.

Histórias de Usuários – Refinando histórias de usuários

Muitas vezes, as histórias de usuário precisam ser lapidadas, pois podem estar complicadas de entender, grandes demais para serem desenvolvidas, dependentes demais para serem entregues, ou podem não ser verificadas.

Para esse refinamento, utiliza-se o acrônimo INVEST, o qual traz 7 características. Aplicar todas elas não é tão fácil; até dizem que INVEST é o nirvana das histórias de usuários, mas deve-se tentar ao máximo obtê-las.

Refinando com INVEST, uma história de usuário deve ser:

- **Independente** de outras histórias para facilitar a negociação, a priorização e a implementação;
- **Negociável** com o cliente, para que o time possa manter um ritmo sustentável para a entrega contínua de valor;
- **Valorosa**, assim cada entrega agregará grande valor ao negócio do cliente;
- **Estimável**, desse modo o time poderá estabelecer quais histórias serão implementadas de acordo com sua velocidade;
- **Small** (pequena) o suficiente para ser implementada dentro de uma iteração, sem correr grandes riscos de não a completar;
- **Testável**, apenas assim poderá ter qualidade na entrega e certeza de que o cliente aceitará a história como pronta.

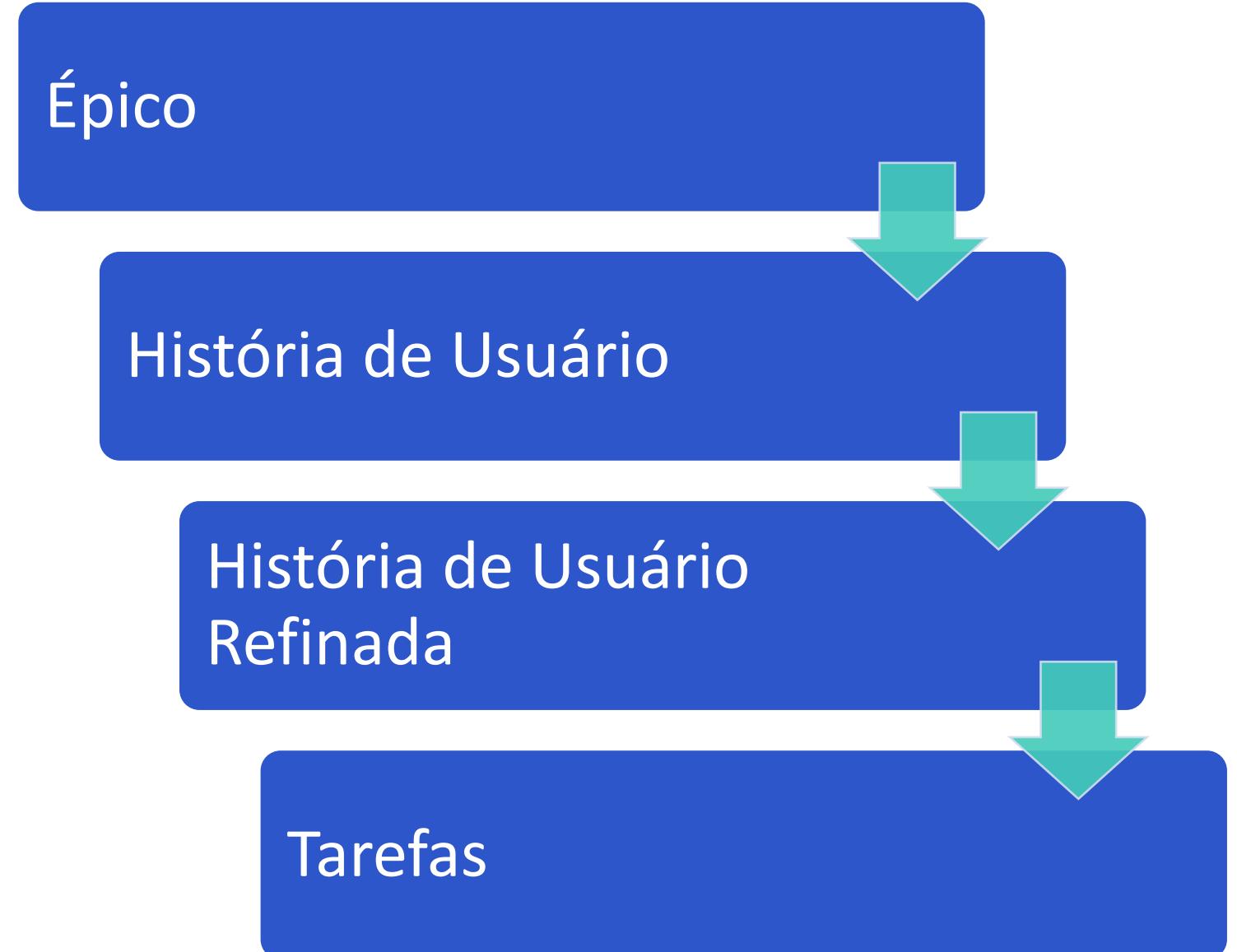
Cada história de usuário possuirá tarefas a serem desenvolvidas, como:

- Implementação
- Teste
- Integração
- Deploy

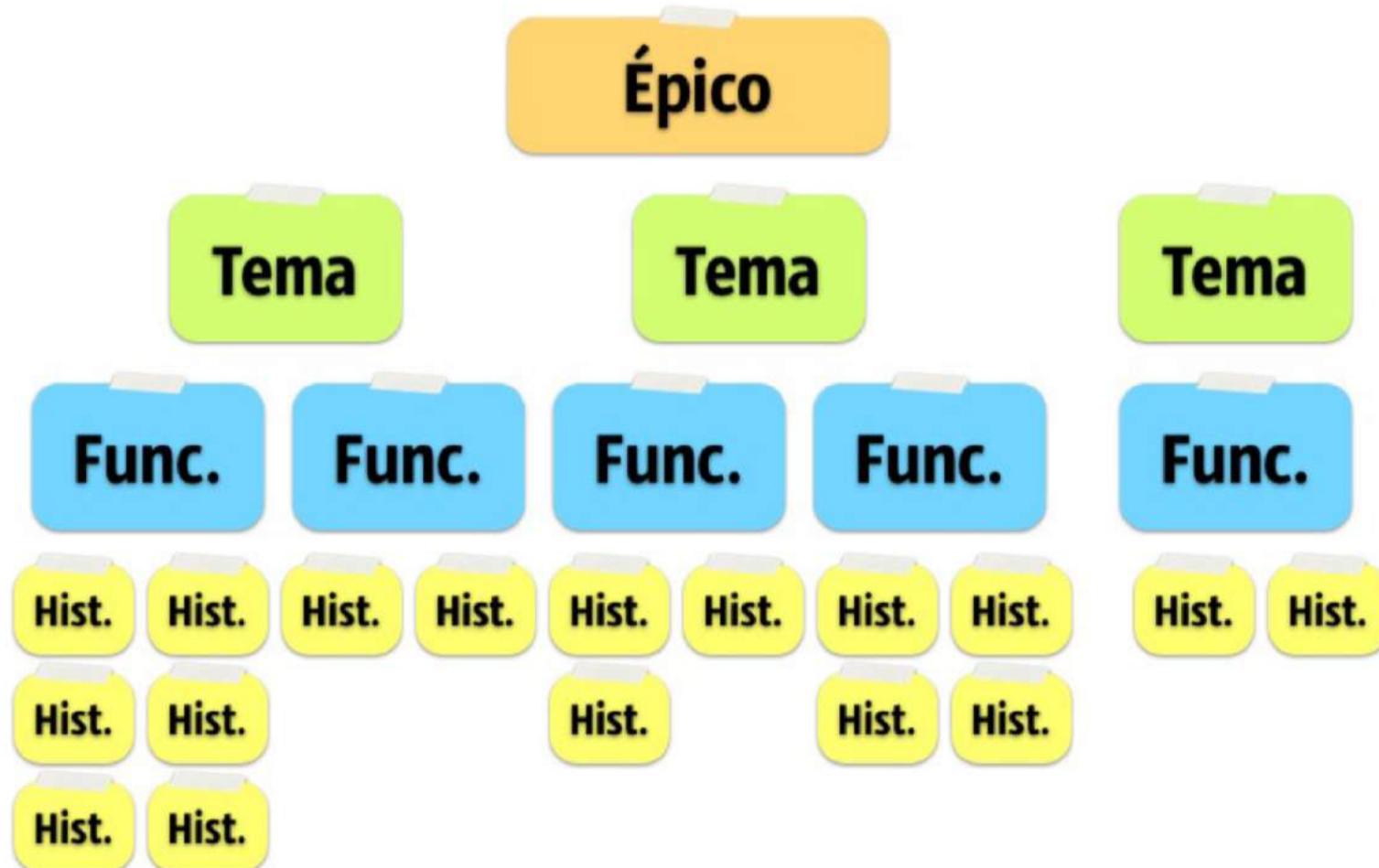
Quebrá-las em tarefas gera mais certeza do trabalho que o time precisará entregar.

Cada uma delas deverá possuir as 5 características do acrônimo SMART:

- **eSpecífica** o suficiente para ser entendida, para que se implemente exatamente o que é necessário, evitando a sobreposição de outras tarefas;
- **Mensurável**, possuindo um tamanho para garantir que a tarefa seja concluída (contabilizando testes e refatoração);
- **Alcançável** para que todos possam ter acesso às informações necessárias para sua conclusão;
- **Relevante**, contribuindo para a implementação da história;
- **Time-boxed**, limitada a uma duração específica. Tarefas com grande duração devem ser evitadas, podendo serem divididas em tarefas menores ou sendo revisadas.



Podemos agrupar pequenas histórias em temas. Desta forma, até mesmo o Product Backlog de um projeto imenso pode ser ajustado.



Requisitos funcionais são funções ou recursos específicos do produto, como poder fazer ou receber chamadas.

Os requisitos emergem e o Product Backlog evolui à medida que a equipe Scrum aprende mais sobre as necessidades do cliente e como atendê-las.

Também chamados de requisitos operacionais, qualidades do sistema e restrições - são patinhos feios do desenvolvimento de software. Eles costumam ser negligenciados, embora descrevam propriedades importantes, como desempenho, robustez, escalabilidade, usabilidade, bem como requisitos técnicos e de conformidade.

Os requisitos não funcionais devem ser detalhados desde o início - ao criar a visão ou ao montar o Product Backlog.

Descobrir e refiná-los tarde demais pode causar escolhas erradas e impactar negativamente o sucesso do produto.

O framework Scrum

Quiz de revisão

EXIN

AGILE SCRUM
FOUNDATION



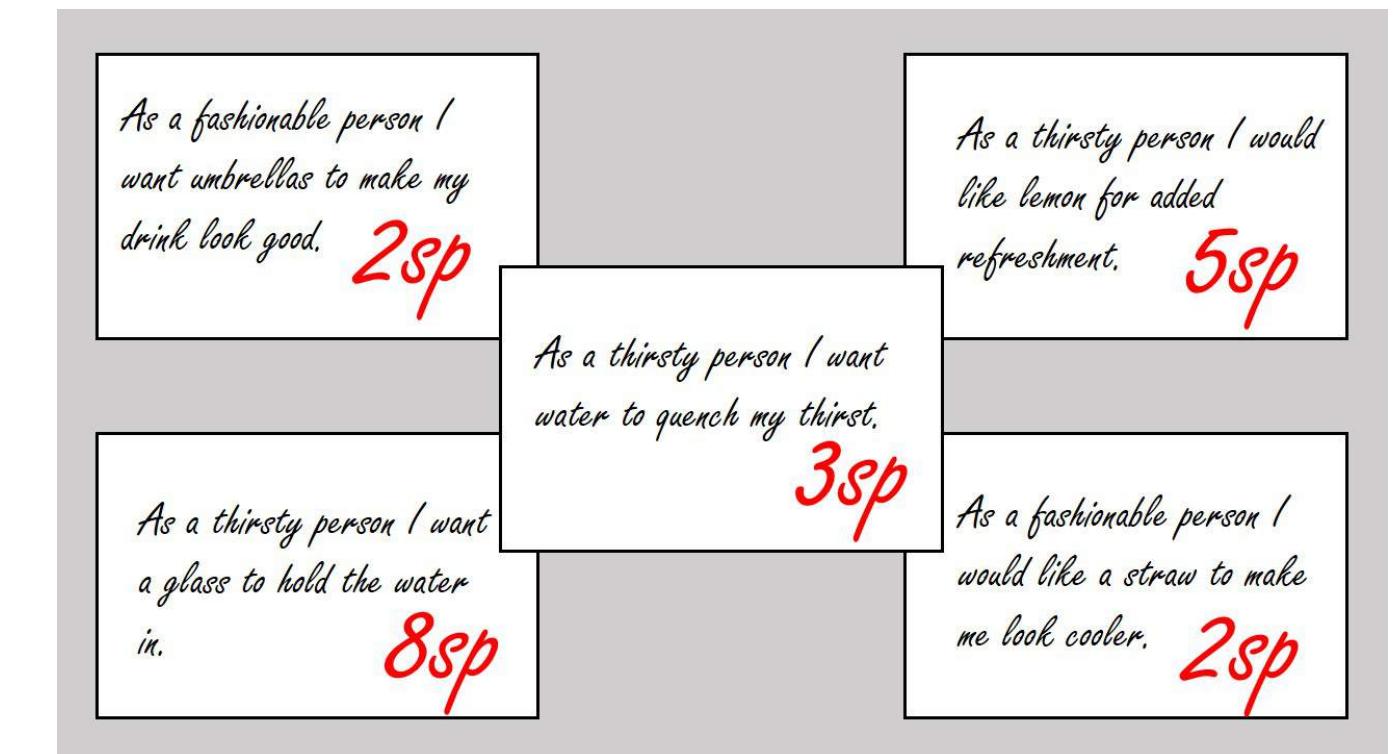
Módulo 3

Planejamento

e Controle

A velocidade pode ser usada como um indicador de quanto trabalho a equipe pode fazer em um Sprint. Isso permite acompanhar e prever o progresso do projeto.

Velocidade é o número de Pontos por história que uma Equipe pode terminar em 1 Sprint.



Throughput nada mais é do que uma medida que determina quantas unidades de trabalho foram completadas em uma unidade de tempo. Olhando pela perspectiva de vazão (ou saída), é possível afirmar que o throughput é uma medida que determina quanto o meu fluxo é capaz de processar.

Conforme ilustrado na figura seguinte, trazendo para o dia a dia de desenvolvimento de software, diríamos que, em determinada semana (unidade de tempo), o throughput de uma equipe foi de três funcionalidades (quantidade de itens).

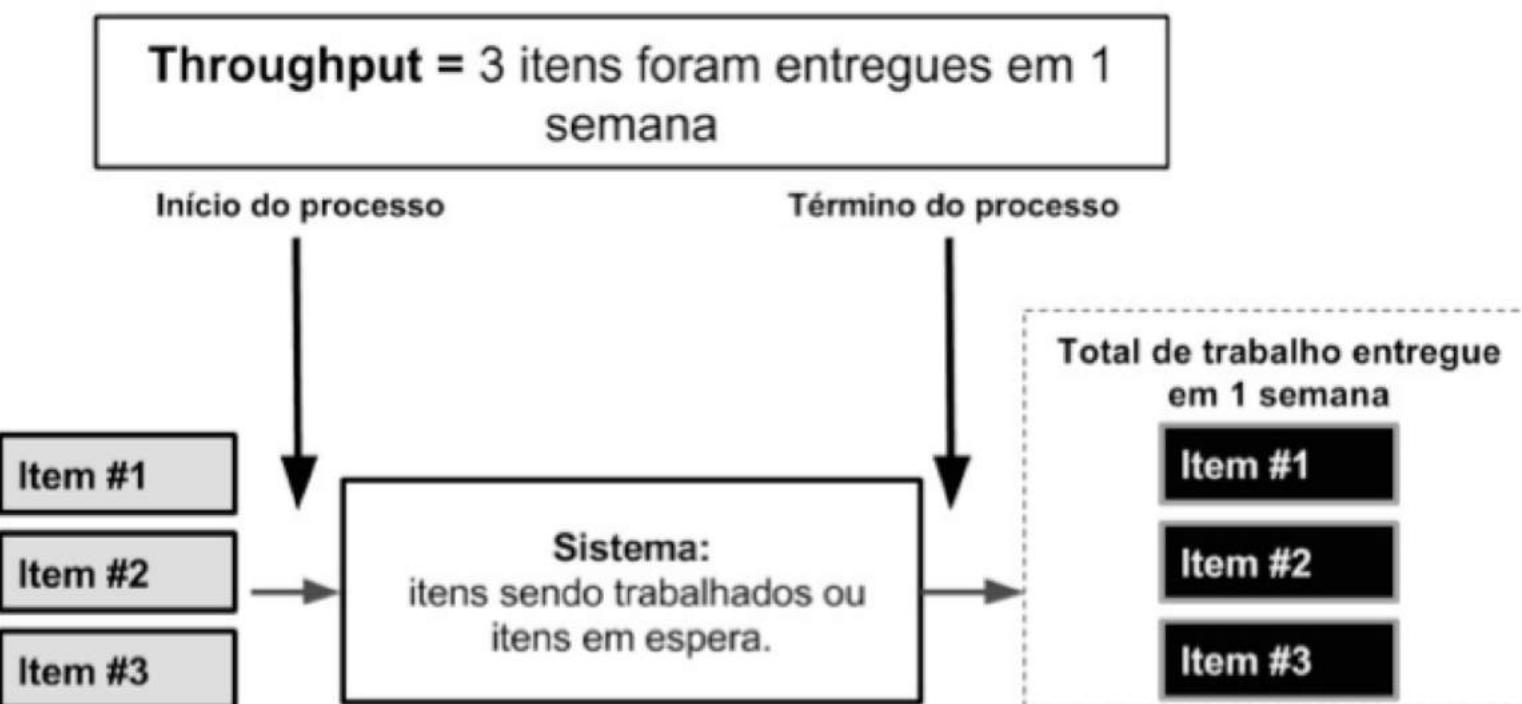
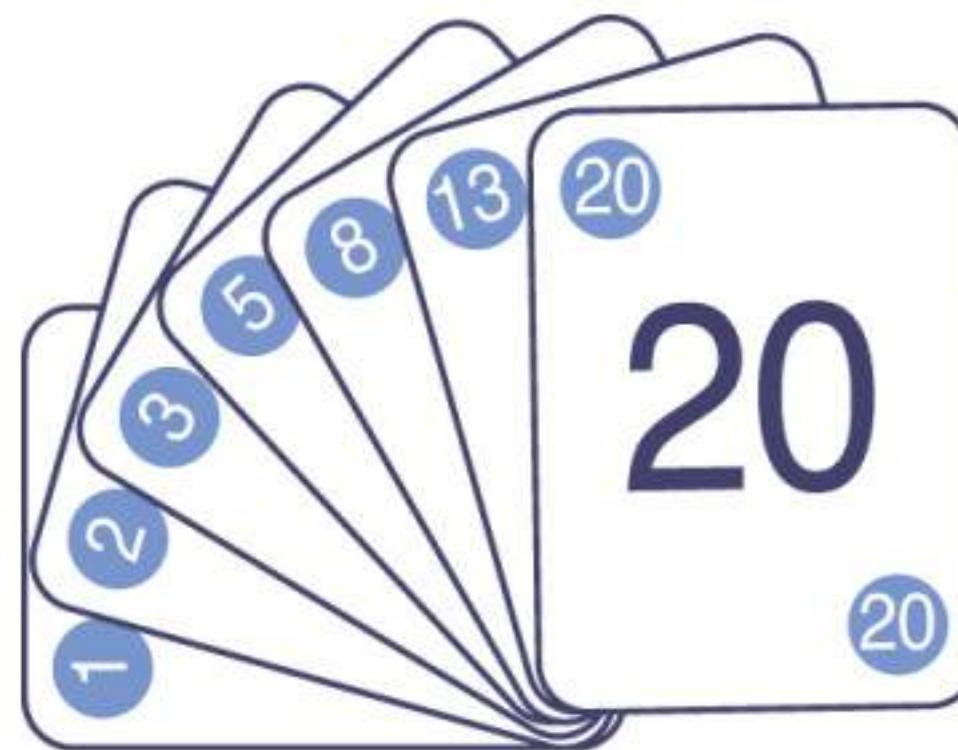


Figura 4.1: Exemplo de cálculo do throughput

Story point é uma medida arbitrária usada por times na medição do esforço necessário para desenvolver determinado requisito, funcionalidade, história do usuário, bug etc. Usualmente, os times que utilizam tal medida se baseiam na sequência de Fibonacci (1, 2, 3, 5, 8, 13 etc.) para atribuir o esforço de trabalho necessário para implementar uma demanda.



É o processo de estimativa por referência que utiliza âncoras para determinar o tamanho ou esforço de uma nova história ou item. A técnica de triangulação é aplicada naturalmente quando se utiliza Planning Poker Card para estimar, pois a triangulação se dá quando o time pega uma história e a compara com outras duas para saber exatamente onde encaixá-la.

Exemplo:

O time já estimou as seguintes histórias:

História 22 – Tem o esforço Estimado de 10

História 35 - Tem o esforço Estimado de 30

História 43 – Tem o esforço Estimado de 5

História 18 – Tem o esforço Estimado de 15

O time recebe a nova história 89 e discute brevemente entre si: a história 89 é maior que a história 22 e é menor que a história 18. Essa comparação de três pontos é chamada de triangulação e permite encaixar uma história entre outras duas com menor e maior esforço.

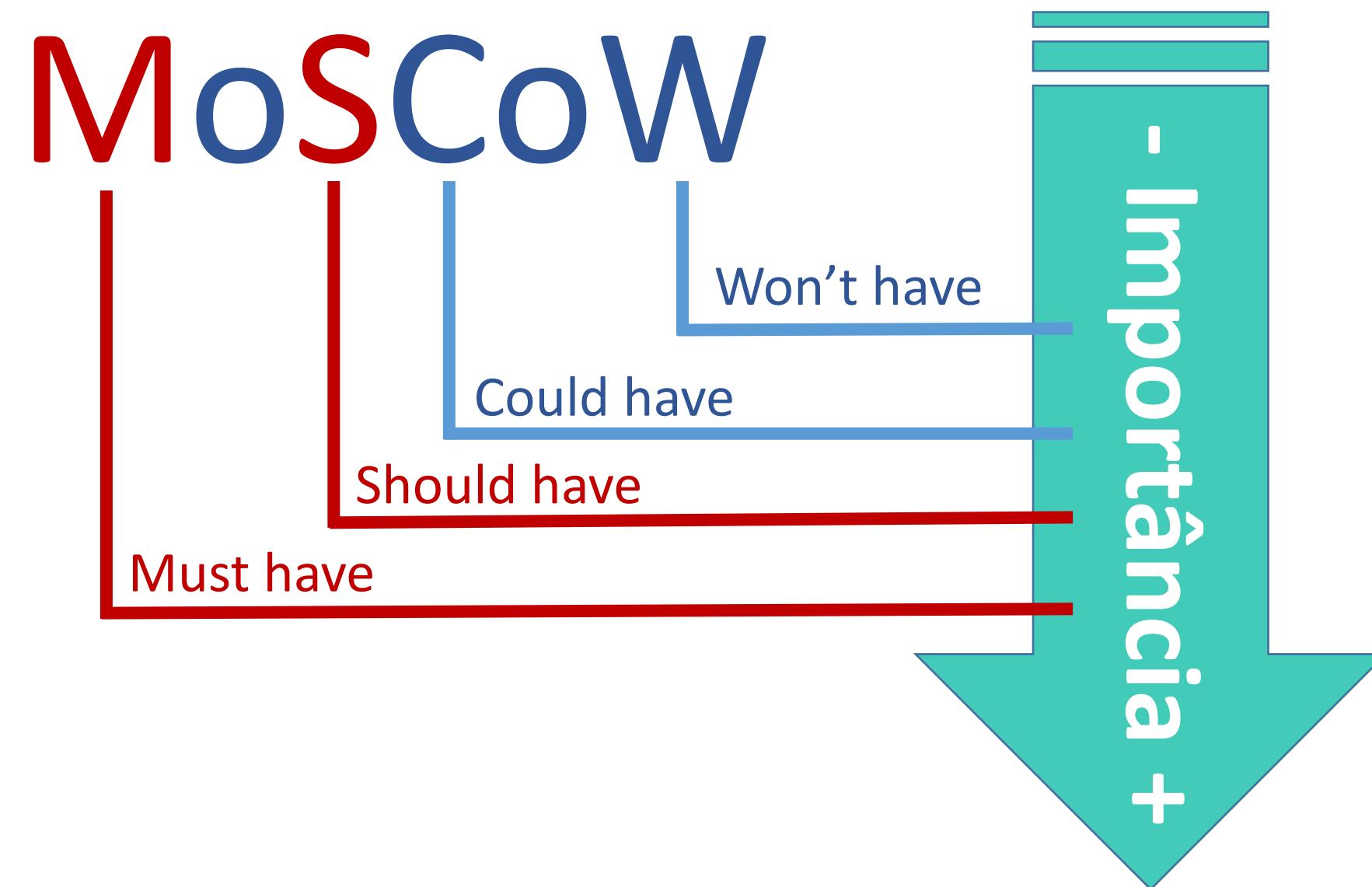
Estimativa por afinidade é a ação de determinar o tamanho e/ou esforço de grupos de histórias com classificação similares. Esta prática é geralmente aplicada quando um projeto é grande e possui uma enorme quantidade de histórias. Em vez de estimar história por história, individualmente, o Time faz um agrupamento de histórias e o estima.

Frequentemente são utilizadas as definições de tamanho da estimative T-shirt, que separa os tamanhos e/ou esforços das histórias como camisetas, tais como PP, P, M, G e GG.

Tais equivalências podem ser convertidas para pontos por história, horas ou outra unidade de medida que o Time entenda e com a qual tenha familiaridade.

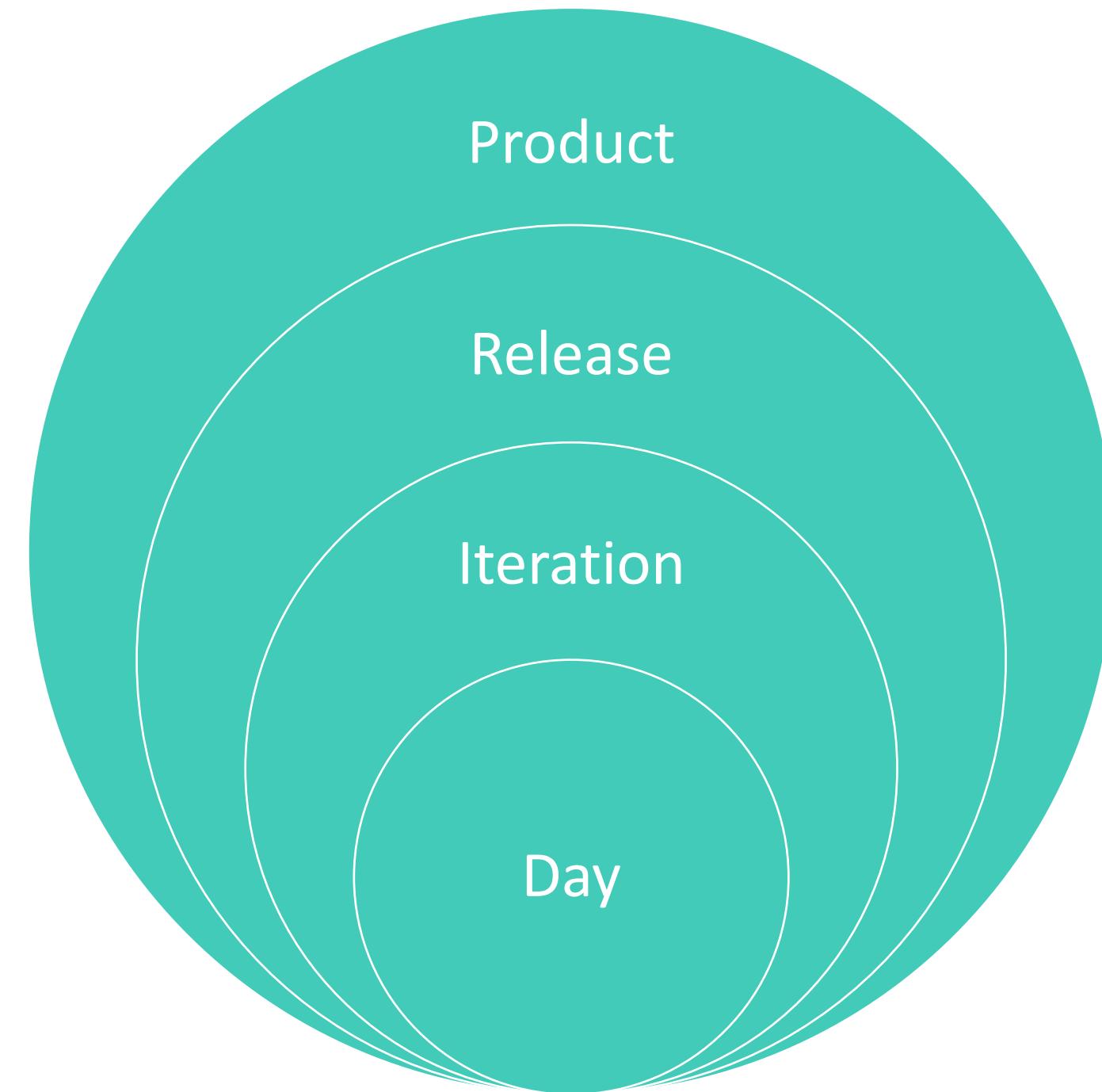
A estimative por afinidade é uma técnica para ganhar velocidade na estimative inicial de muitos itens de backlog e/ou classificação e categorização de requisites de negócio.

MoSCoW é uma ótima técnica para gerenciamento de escopo pois facilita o entendimento do que realmente é importante para o projeto.



Planejamento

Planning Onion



O número mínimo de desenvolvedores que você precisa perder antes de perder a capacidade de continuar o projeto (mesmo adicionando novos desenvolvedores) é chamado de Bus Factor.

Quanto mais alto o número, melhor.

Formas de aumentar o Bus Factor:

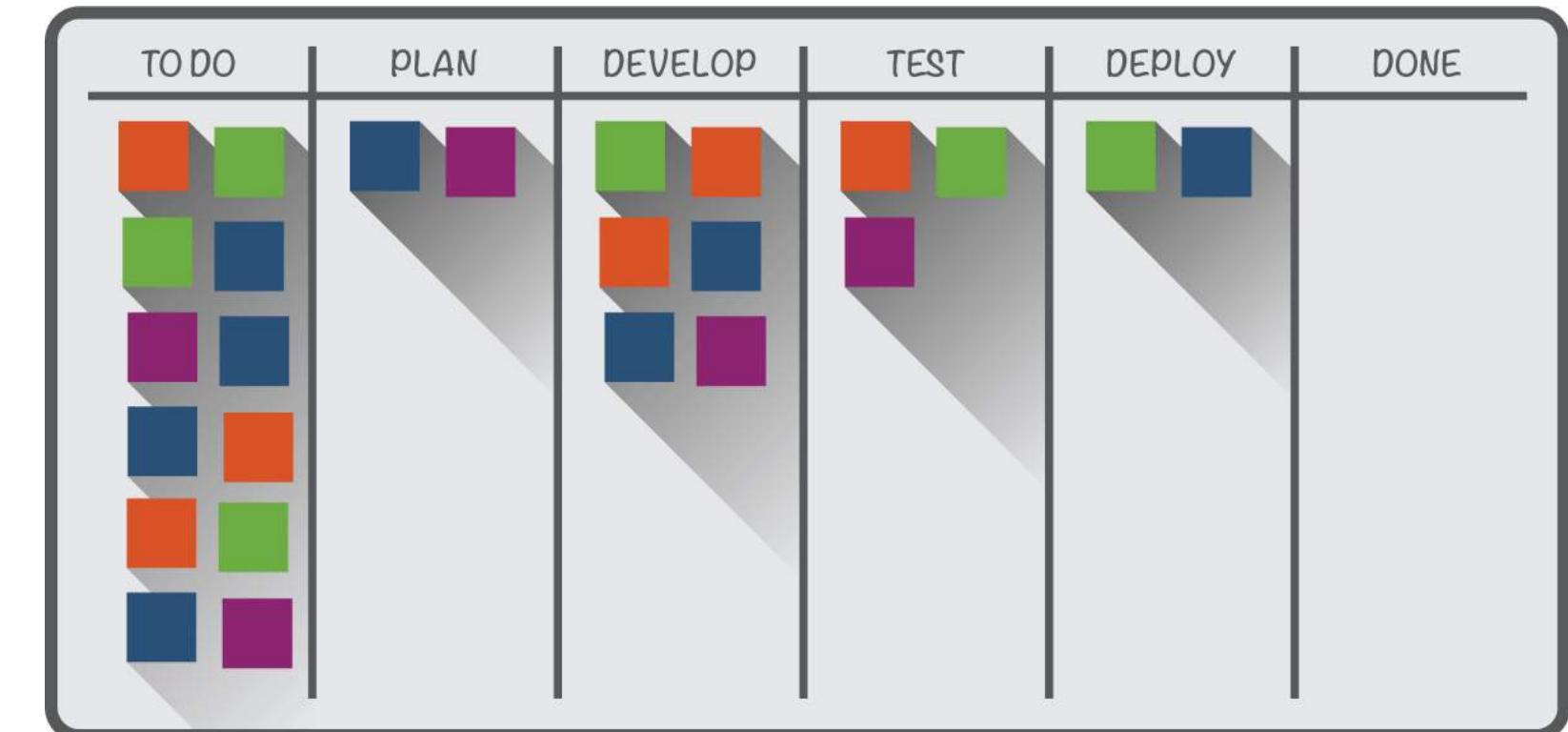
- Programação em par
- Aumentar a transparência
- Melhorar a comunicação
- Propriedade compartilhada



É um método de gestão do fluxo de trabalho de um processo incremental e evolutivo.

O método se baseia em visualizar o fluxo de trabalho e, a partir disso, atuar no processo para não sobrecarregar os membros da equipe.

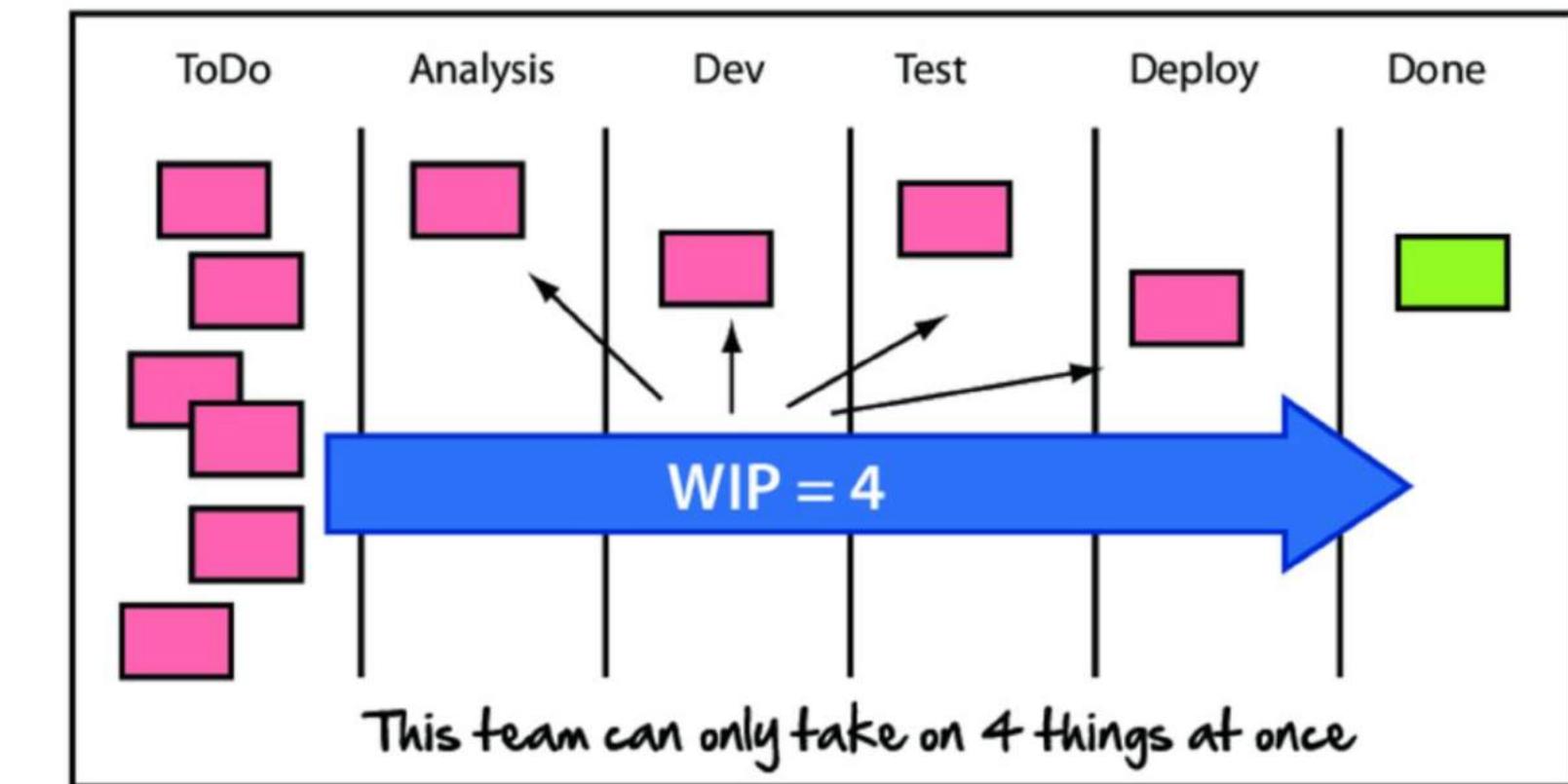
Ao tornar o trabalho visível podemos ver onde o trabalho está fluindo bem e onde o trabalho está na fila ou interrompido.



É a tarefa que está em execução em um determinado ponto do processo.

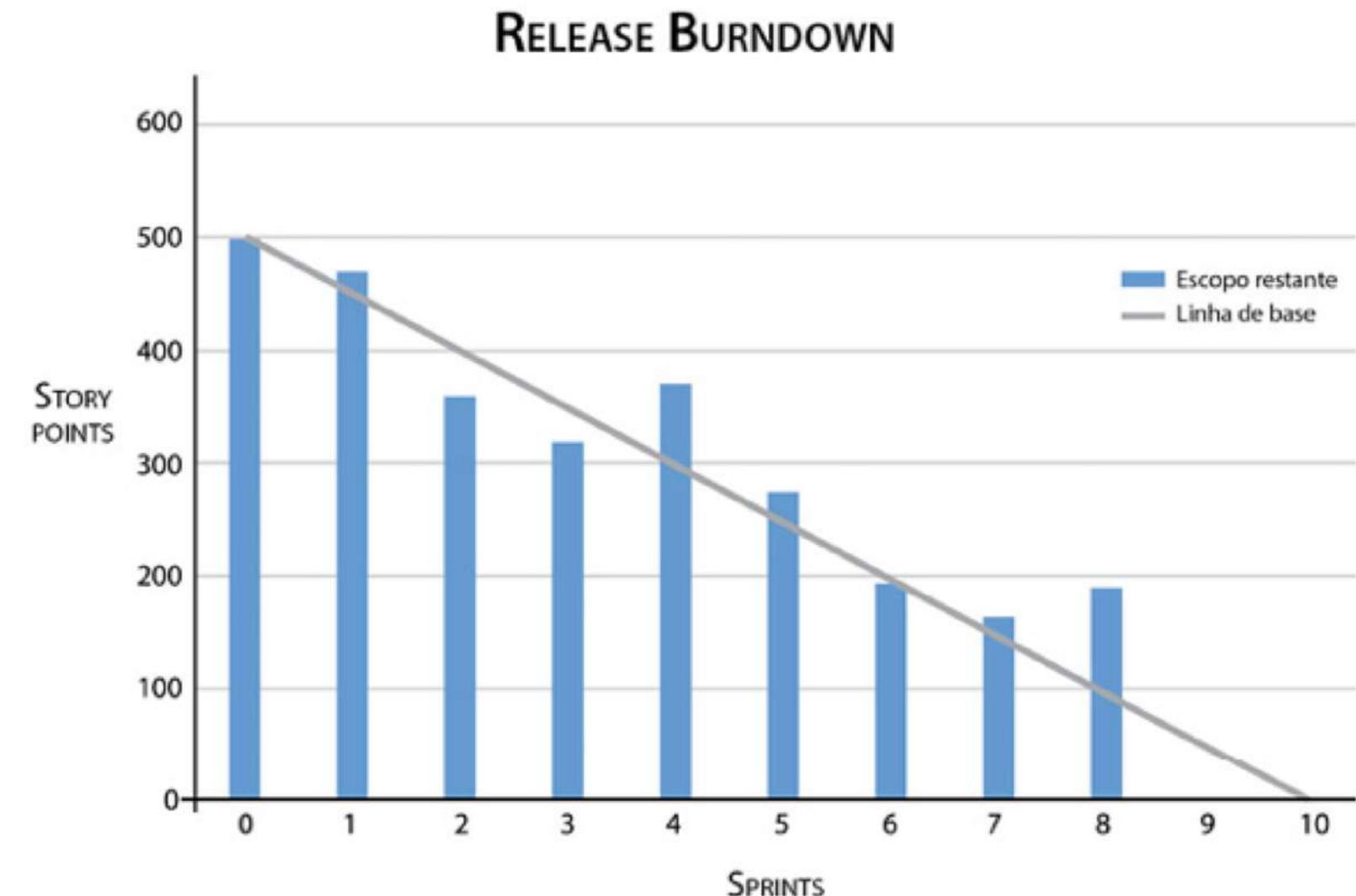
Um método de controlar e limitar o trabalho em progresso (WIP) é utilizando quadros KANBAN e limitando a quantidade de cartões que podem ser adicionados em cada coluna.

Limitar o trabalho em progresso também ajuda a identificar problemas que impedem determinada atividade de ser concluída.



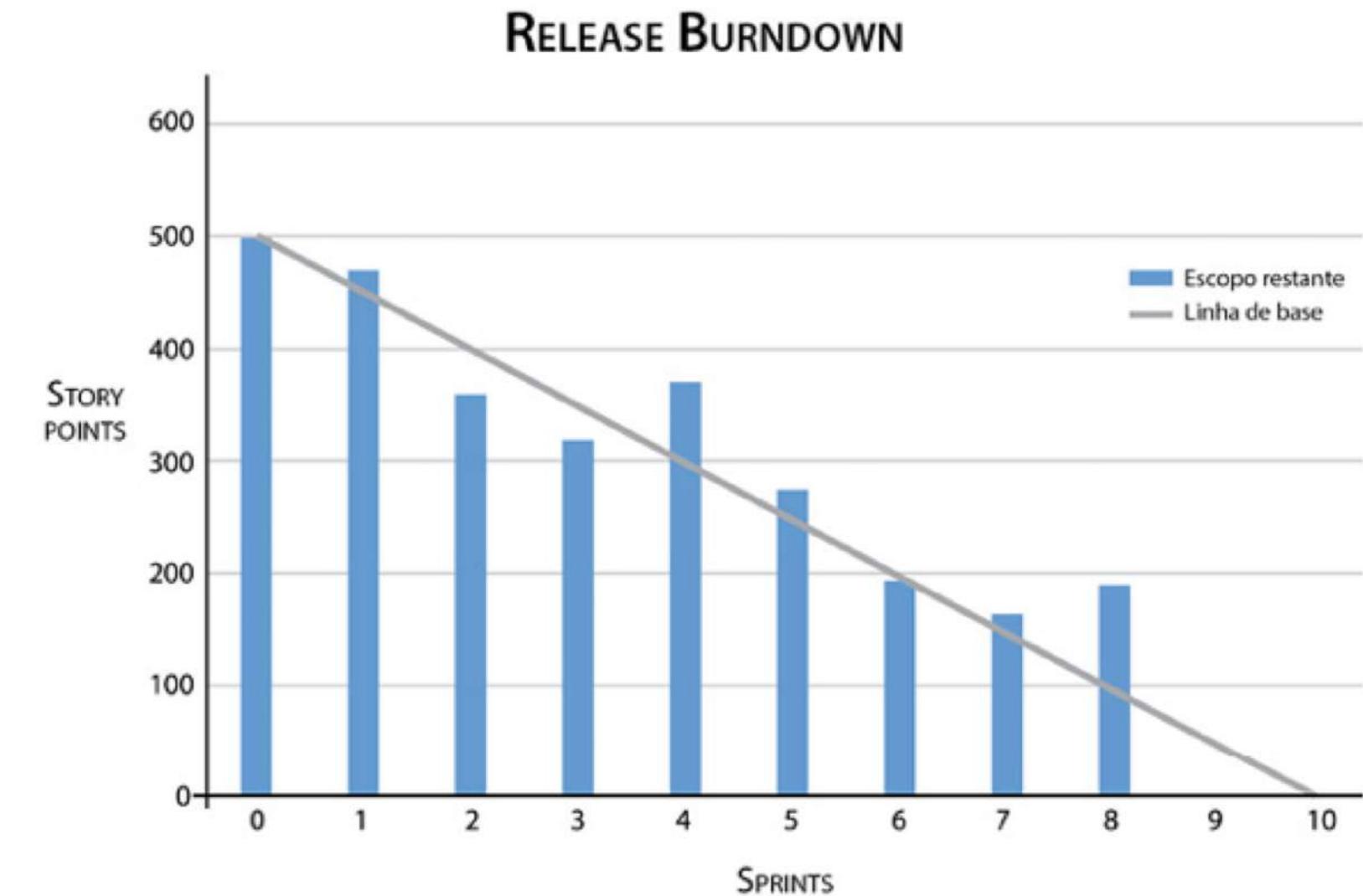
O gráfico Burndown fornece subsídios para o Product Owner e o restante da Equipe Scrum avaliarem o andamento da versão de entrega/release do produto e projetarem tendências.

Neste gráfico listamos a quantidade de trabalho restante na linha do tempo, representado pela quantidade de Sprints no eixo horizontal e pelas unidades de esforço (story points, horas, etc.) no eixo vertical.



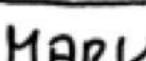
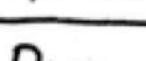
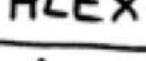
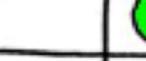
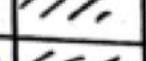
O gráfico pode revelar três tendências:

- Barra vertical abaixo da linha de base significa que a versão de entrega/ release do produto está **potencialmente adiantada**.
- Barra vertical acima da linha de base significa que a versão de entrega/release do produto está **potencialmente atrasada**.
- Barra vertical exatamente na mesma posição que a linha de base significa que a versão de entrega/release do produto está **seguindo conforme o planejado**.



Quando os integrantes de um Time deixam o escritório no final do dia, vão até uma parede com um calendário pendurado e desenham ali uma carinha alegre, uma carinha neutra ou uma carinha triste, isso é chamado de calendário NIKO-NIKO (NIKO-NIKO calendar).

O objetivo é armazenar as emoções de todos os integrantes do Time nos dias de trabalho de forma que todos possam ver.

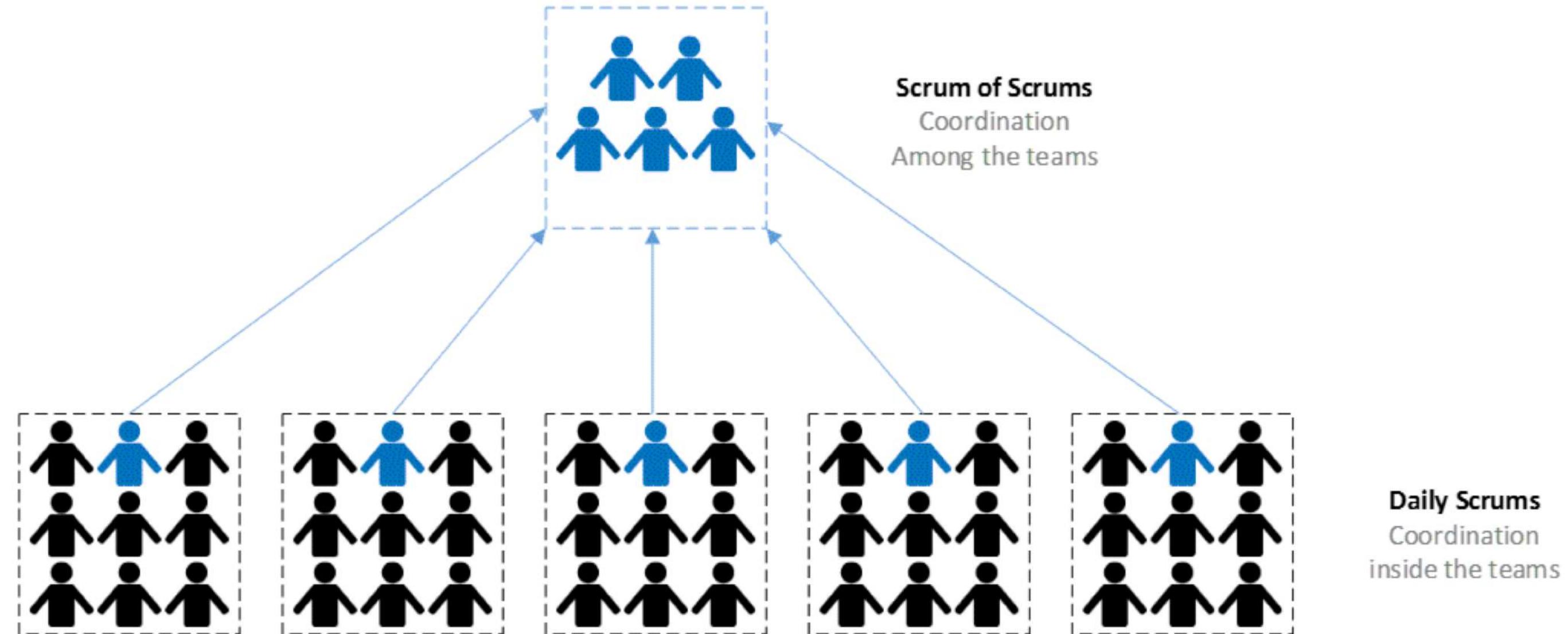
SEPTEMBER 2011		1	2	3	4	5	6	7
		BERND	MARIKO	ALEX	MEIKE	STEFAN		
								
								
								
								
								

Escalando o Scrum

Scrum of Scrums

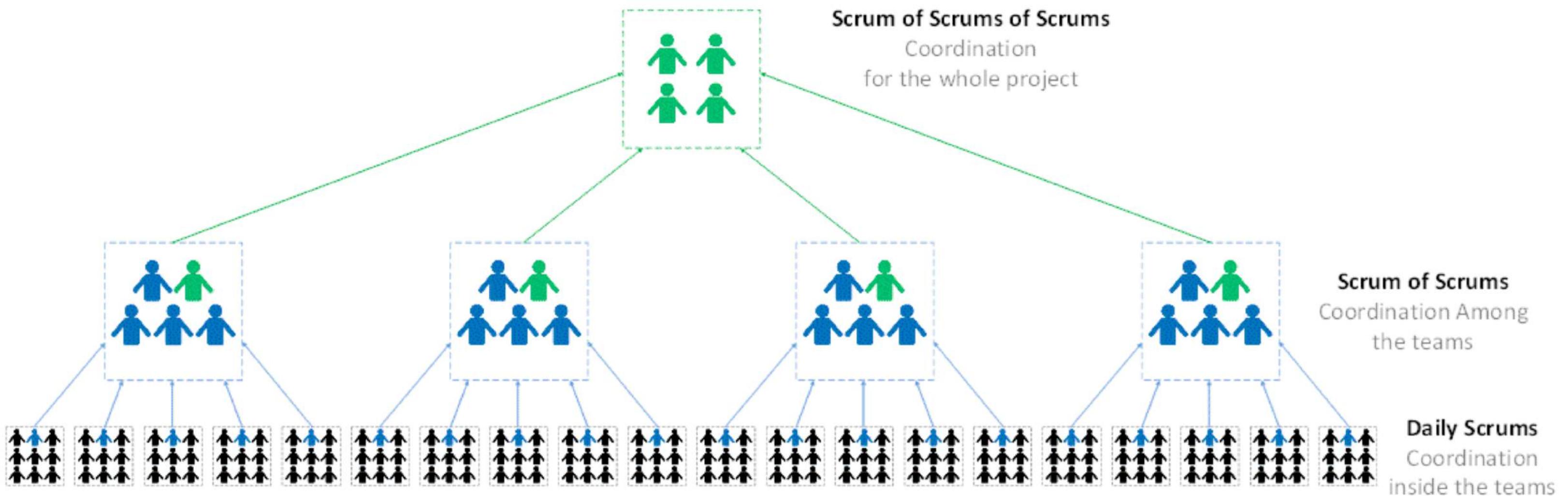


AGILE SCRUM
FOUNDATION



Escalando o Scrum

Scrum of Scrums of Scrums



A reunião do Scrum-of-Scrums permite que várias equipes se coordenem diariamente durante o Sprint. Os representantes da equipe se encontram seguindo o Daily Scrums de suas equipes para discutir o status quo, o trabalho planejado e quaisquer dependências entre as equipes.

Esta reunião é de natureza tática. Não pode compensar a falta de trabalho de preparação da Sprint, como o planejamento antecipado.

O Scrum-of-Scrums é uma forma de manter todas as equipes Scrum atualizadas com as informações de todas as outras equipes.

Um representante de cada equipe vai à reunião do Scrum-of-Scrums e ouve o que as outras equipes estão fazendo, conta o que sua equipe está fazendo e depois da reunião atualiza seus colegas.

O Scrum of Scrums permite que as equipes compartilhem informações com representantes de todas as outras equipes Scrum.

Além desses dispositivos de comunicação, as equipes Scrum de alto desempenho encontram maneiras de conversar diretamente com as pessoas sobre outras equipes.

É comum, por exemplo, para banco de dados os desenvolvedores conversam com outros desenvolvedores de banco de dados e os designers de interface de usuário para conversar com outros designers de interface de usuário.

Você pode encontrar dois contratos em um projeto de desenvolvimento de TI:

Tempo e meios ou unidade fixa

Este é o tipo de contrato preferido, que é compatível com a natureza adaptativa de projetos Scrum. No caso, podemos cobrar o cliente com base em horas-homem ou por Sprint.

Preço fixo

Embora este seja o tipo preferido pela maioria das empresas e até mesmo algumas delas sejam obrigados por lei a ter apenas contratos de preço fixo, este tipo de contrato não é aderente ao Ágil. Isso ocorre porque todos os contratos de preço fixo são baseados em uma definição inicial de escopo, que não é adaptativa. Esse tipo de contrato é considerado de baixo risco e um alto nível de determinismo para os clientes.

Planejamento e Controle

Quiz de revisão

EXIN

AGILE SCRUM
FOUNDATION



