

MARY PROVINCIAUITTO e PAULO CAROLI

PREFÁCIO DE ANNELISE GRIPP

# SPRINT a SPRINT

- Erros e acertos
- na transformação cultural de um time ágil



Editora  
Caroli  
[caroli.org](http://caroli.org)

MARY PROVINCIAUTO E PAULO CAROLI

# SPRINT a SPRINT



→ Coordenação editorial e  
preparação

*Juliana Cury Rodrigues*

Revisão  
*Nestor Turano Jr*

Projeto gráfico, diagramação  
e capa

*Vanessa Lima*

Desenvolvimento de eBook  
*Loope Editora*

[www.loope.com.br](http://www.loope.com.br)

Copyright © 2020  
Mary Provinciatto e Paulo  
Caroli

Todos os direitos desta  
edição  
são reservados à Editora  
Caroli.

Avenida Itajaí, 310 – Bairro  
Petrópolis  
Porto Alegre, RS 90470-140,  
Brasil

[www.caroli.org/editora](http://www.caroli.org/editora)   
[contato@caroli.org](mailto: contato@caroli.org)

#### Dados Internacionais de Catálogo na Publicação (CIP)

Angélica Ilacqua CRB-8/7057

---

Provinciatto, Mary

Sprint a Sprint: erros e acertos na transformação cultural de um time ágil / Mary Provinciatto, Paulo Caroli. – São Paulo: Editora Caroli, 2020. 200 p. : il.

ISBN 978-65-86660-02-9

1. Administração da projetos 2. Desenvolvimento de produtos 3.  
Trabalho em equipe I. Título II. Caroli, Paulo

20-1601

CDD 658.404

---

Índices para catálogo sistemático:

1. Administração de projetos

# AGRADECIMENTOS <

**A**gradecemos a comunidade ágil brasileira por participar deste livro. Os comentários e feedback foram essenciais para a melhoria deste conteúdo. Agradecemos a todos que participaram das enquetes para escolha do título e da capa do livro. Muito obrigado, Ricardo Maciel, pela proposta do subtítulo “Erros e acertos na transformação cultural de um time ágil”.

Muito obrigado às pessoas das muitas equipes com quem trabalhamos. O nosso convívio e aprendizado em conjunto nos deu motivação para escrever e compartilhar o conhecimento descrito aqui.

Obrigado também à ThoughtWorks pela liderança e exemplo positivo nas práticas e atitudes sobre o respeito e desenvolvimento de pessoas. Muitas pessoas conhecem a ThoughtWorks como empresa líder e exemplo em agilidade e boas práticas para desenvolver produtos digitais, mas, para nós, o maior diferencial e impacto está no respeito e no desenvolvimento das pessoas.

Agradecemos aos nossos familiares e amigos, pessoas queridas que nos apoiam no dia a dia do nosso trabalho. O apoio de vocês foi fundamental para a criação e evolução deste livro!

# SUMÁRIO <

Prefácio

O que você precisa saber antes de ler este livro

Contexto e objetivo da obra

## CONTEÚDOS TÉCNICOS E TEÓRICOS

Manifesto ágil

Produto Mínimo Viável (MVP)

Lean Inception

Um overview do Kanban

Um overview do Scrum

Funcionalidades e histórias de usuário

Modelo spotify

Objetivos e resultados-chave - OKR

Sobre algumas práticas específicas

Assessment de DevOps

## O DIA A DIA DE UM TIME EM AÇÃO

Planejamento do trabalho de consultoria ágil

Entenda as sprints na linha do tempo

[Semana 1](#)

[Semana 2](#)

[Semana 3](#)

[Semana 4](#)

[Semana 5](#)

[Semana 6](#)

## PRINCIPAIS PROBLEMAS E RECOMENDAÇÕES

O resultado das seis semanas

Indefinição/dualidade dos papéis

Reuniões canceladas e cadênciа interrompida

Sprints semanais com retrospectivas quinzenais

Excesso de spikes e falta de direcionamento do negócio

Falta de visibilidade sobre a saúde do time

Pessoas compartilhadas e baixa prioridade do projeto

## OS PRÓXIMOS QUATRO MESES

Estou te vendo, estou te ouvindo

Quem facilita as reuniões de sprint?

A primeira entrega em produção

Acordos do time

Indivíduos e interações mais que processos e ferramentas

## ANEXOS

Templates

Atividades de retrospectiva

Onde encontrar mais

[Participe do treinamento Sprint a Sprint](#)

[Sobre a Editora Caroli](#)

[Deixe sua avaliação](#)

PREFÁCIO



**E**ste livro vai mudar a sua visão sobre como trabalhar dentro de um ágil!

*Sprint a Sprint: Erros e acertos na transformação cultural de um time ágil* traz um olhar mais apurado e detalhado para o leitor sobre como trabalhar com um time sem desperdícios!

Por mais que isso pareça ser bem simples, no dia a dia é bem mais complicado do que se imagina. Trabalhar com uma equipe que tenha confiança, esteja alinhada e, além de tudo, consiga cumprir prazos e metas exige dedicação.

Por envolver sistemas integrados – áreas que cuidam dos sistemas, profissionais que querem continuar na zona de conforto, profissionais que querem o novo, gestores que pensam apenas nos seus bônus, muitos terceiros dentro dos times e dezenas de outros fatores –, é preciso seguir um modelo de trabalho organizado e que integre todas as pessoas que compõem o time.

Acredito que você já está percebendo o emaranhamento que é. Mas não precisa se desesperar: sempre há um jeito de melhorarmos. A busca pela melhoria deve ser sempre contínua. Os erros nos ajudam a entender o que poderia estar de outro jeito e nos direcionam para um novo caminho que não havia sido considerado antes.

E como fazer isso? Investindo nas pessoas e dando oportunidade de todos participarem dessa transformação. Que é exatamente o que o método Sprint a Sprint proporciona: um crescimento em conjunto com todos que fazem parte do projeto, sempre ouvindo opiniões e buscando o melhor e mais eficaz meio de trabalhar.

A obra que você lerá a seguir traz a explicação de muitos conceitos importantes para equipe ágeis, apresenta muitas técnicas e ferramentas e também conta passo a passo de um exemplo real. Através de uma contação de histórias, você verá como um time utilizou do método Sprint

a Sprint para construir um produto, o que deu errado e precisou ser alterado e o que deu certo.

Infelizmente nem pessoas e nem empresas nascem com um manual de instruções. O sucesso da empresa depende da boa performance de um time e o grande trabalho é justamente fazer pessoas diferentes, com conhecimentos distintos, trabalharem em direção ao mesmo objetivo: a entrega de um produto!

Desejo que você “deguste” este livro aos poucos, sentindo, refletindo e pensando em como aplicar os conhecimentos que leu aqui na rotina do seu time. Esta obra precisa ganhar espaço na sua cabeceira para que você a consulte sempre que estiver pensando “como será que posso melhorar ainda mais o trabalho com meu time?”.

Boa leitura e boa sorte!

→ [Annelise Gripp](#), consultora de transformação digital e agile.



O QUE  
VOCÊ  
PRECISA  
SABER  
ANTES DE  
LER ESTE  
LIVRO

**E**ste livro está dividido em cinco seções principais. A primeira delas – conteúdos técnicos e teóricos – apresenta materiais extras para que você tire maior proveito do livro. Caso não esteja familiarizada com o mundo ágil da criação de produtos digitais, recomendamos que a leia primeiro. Porém, se já estiver por dentro desse mundo, pode usar toda essa seção para tirar eventuais dúvidas ou aprender ainda mais!

Já a segunda — o dia a dia de um time em ação — foi escrita a partir das comunicações do time e, por isso, está organizada por acontecimentos cronológicos. A informação do dia em que ocorreu a comunicação será o título do capítulo, assim você poderá acompanhar a rotina diária da equipe e quando e como emergiu cada conceito, cada conversa. Esta seção está dividida em partes que representam cada semana do projeto.

Depois das seis semanas de consultoria ágil, há uma pequena lista com os maiores problemas e as recomendações. Ela está apresentada na terceira seção.

A quarta seção — os próximos quatro meses — resume os principais acontecimentos que ocorreram nos últimos meses do projeto.

Para complementar este livro, a quinta e última seção traz alguns anexos que deixarão a sua experiência de leitura e aprendizado ainda mais completa!

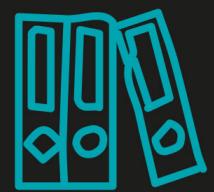


# CONTEXTO E OBJETIVO DA OBRA

**A** segunda seção do livro descreve seis meses de trabalho de uma equipe de desenvolvimento de um produto digital, um Software Development Kit (SDK) para Android e iOS. Nosso objetivo com esta obra é compartilhar os desafios e aprendizados que tivemos desde a realização da Lean Inception, passando pela primeira entrega em produção e até o fim do projeto.

O time tinha pessoas de empresas e localidades diferentes: era composto uma parte pela ThoughtWorks (consultoria), em Porto Alegre, e a outra por pessoas do cliente que ficavam alocadas em outra cidade. A composição do time foi a seguinte:

- » Um consultor ágil para as primeiras seis semanas do projeto (Paulo Caroli);
- » Uma gerente de projetos / Scrum Master (Mary Provinciatto);
- » Sete pessoas desenvolvedoras;
- » Um UX Designer;
- » Um *Product Owner*.



# CONTEÚDOS TÉCNICOS E TEÓRICOS

## » MANIFESTO ÁGIL

Em fevereiro de 2001, nos EUA, 17 desenvolvedores de software iniciaram um movimento que posteriormente serviria como base para criação dos frameworks e métodos ágeis. Insatisfeitos com os resultados obtidos na adoção das metodologias existentes, o grupo criou uma declaração de valores e princípios que representam uma maneira melhor de desenvolver software.

O Manifesto<sup>1</sup> é composto de um conjunto de quatro valores e doze princípios e representa a essência do Ágil, ou seja, eles devem guiar as opções e alternativas relativas a práticas, papéis, cerimônias, métodos ou frameworks.

### Valores:

- » Indivíduos e interações mais que processos e ferramentas;
- » Software em funcionamento mais que documentação abrangente;
- » Colaboração com o cliente mais que negociação de contratos;
- » Responder a mudanças mais que seguir um plano.

### Princípios:

1. Nossa maior prioridade é satisfazer o cliente por meio da entrega contínua e adiantada de software com valor agregado.
2. Mudanças nos requisitos são bem-vindas, mesmo tardivamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.
3. Entregar frequentemente um software funcionando, num prazo de poucas semanas a poucos meses, com preferência à menor escala de tempo.
4. Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.
5. Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.
6. O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é por meio de conversa face a face.
7. Software funcionando é a medida primária de progresso.
8. Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
9. Contínua atenção à excelência técnica e bom design aumenta a agilidade.
10. Simplicidade – a arte de maximizar a quantidade de trabalho não realizado – é essencial.
11. As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis.

12. Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

## » PRODUTO MÍNIMO VIÁVEL (MVP)

Com origem no *Lean Startup*, o MVP é a versão mais simples de um produto que pode ser disponibilizada para a validação de um pequeno conjunto de hipóteses sobre o negócio. Basicamente, isso significa que você não vai desperdiçar tempo, dinheiro e esforço construindo um produto que não vai atender as expectativas. Assim, para mitigar esses riscos, é razoável que você possa querer entender e validar as hipóteses sobre o negócio antes de tentar encontrar uma solução definitiva.

O MVP ajuda nessa validação e aprendizado da forma mais rápida possível. E, diferentemente, dos produtos criados da forma tradicional – normalmente com um período longo de criação de protótipo, análise e elaboração –, o objetivo do MVP é somente a validação de um primeiro passo ou visão – um produto mínimo, portanto, notadamente bem menos elaborado do que a sua versão final.

Um MVP foca o produto mínimo de modo que ele se mostre viável para verificar se um direcionamento ou conjunto de proposições está correto. Tratando-se de desenvolvimento de software, isso remete a um conjunto inicial de funcionalidades consideradas necessárias no curto prazo e que, depois, serão submetidas à validação, com fins de aprendizagem sobre o negócio.

Um exemplo de MVP no mundo real é o da startup EasyTaxi. Os seus sócios decidiram que, ao invés de passarem meses desenvolvendo uma solução para intermediação digital de corridas de táxi, criariam apenas uma página web simplificada cuja função seria enviar o endereço de origem informado pelo usuário aos seus próprios e-mails. Depois, eles mesmos ligavam para as cooperativas e completavam o processo. Com esse primeiro MVP, puderam constatar que havia interesse no serviço, validando uma hipótese fundamental do negócio, com um uso mínimo de tempo e recursos, reduzindo também os riscos associados.

## Origem

É importante ressaltar que a ideia de MVP está originalmente vinculada àquelas que foram popularizadas pelo estilo Toyota de manufatura enxuta.<sup>2</sup> Steve Blank,<sup>3</sup> um empreendedor do Vale do Silício, criou uma metodologia definida com base no desenvolvimento do cliente. Este foi o início do movimento *Lean Startup*, o qual teve seu ápice com Eric Ries e o lançamento do seu livro,<sup>4</sup> com o mesmo nome do movimento.

## Evolução

O MVP não significa que o produto não irá evoluir e incrementar suas funcionalidades. Muito pelo contrário, a ideia por trás de um MVP é ter uma evolução validada e guiada pelos resultados iniciais, com a correção ou a confirmação necessária para orientar os incrementos seguintes. Incrementos são adicionados ao produto

mínimo já validado. Eles, por sua vez, são submetidos a novas verificações sobre o direcionamento do produto, agora já mais elaborado e talvez com um número maior de usuários (permitindo validar novas hipóteses ainda mais estruturadas).

Observa-se que é muito importante compreender que o MVP promove uma criação evolutiva. Logo, a arquitetura, bem como o ferramental de construção do produto, deve idealmente também possuir essa característica de evolução gradual e contínua.

Em 2010, Jez Humble e David Farley publicaram o livro *Entrega contínua*.<sup>5</sup> Nele, elaboraram um processo de entrega rápido e de baixo custo, permitindo a criação incremental de produtos de software. Isso foi batizado como “entrega contínua” e abrange a disciplina de desenvolvimento de software que promove entregas mais rápidas e com maior frequência. A essência da ideia de “entrega contínua” é a mesma que Eric Ries recomenda para a *Lean StartUp*: ciclos rápidos para validação das hipóteses.

Ciclos rápidos e frequentes que permitem tempos de liberação muito curtos e com baixo custo de experimentação. Isso traduz-se em fatores críticos de sucesso principalmente para projetos de software de inovação ou diferenciação.

## O Canvas MVP

O Canvas MVP é uma ferramenta para validar ideias de produtos. É um quadro visual que auxilia empreendedores a alinhar e definir a estratégia do MVP.

Não desperdice tempo, dinheiro e recursos criando o produto errado. Muitas vezes exageramos ao construir um novo produto.

Planejamos e adicionamos várias funcionalidades, pensando no resultado final que, supomos, irá atender a necessidade dos nossos clientes.

A verdade sobre startups e novos projetos é que ninguém sabe se vai dar certo. Se soubéssemos o que estávamos criando, então não seria um novo produto, mas algo que já existe.

Então, faça como o Facebook, o Groupon, o AirBnb, o Spotify, a EasyTaxy e vários outros exemplos de empreendimentos de sucesso. Adote uma nova forma de criar e evoluir produtos: desenvolva o produto mínimo viável, crie um Canvas MVP.

### Valide a ideia

Não crie funcionalidades para um MVP se você não souber descrever o que espera como resultado e como medi-lo.

Valide as hipóteses do negócio. Tente compreender melhor os seus usuários. Para tanto, planeje coletar dados de uso do MVP que lhe ajudarão a verificar o resultado/aprendizado desejado.

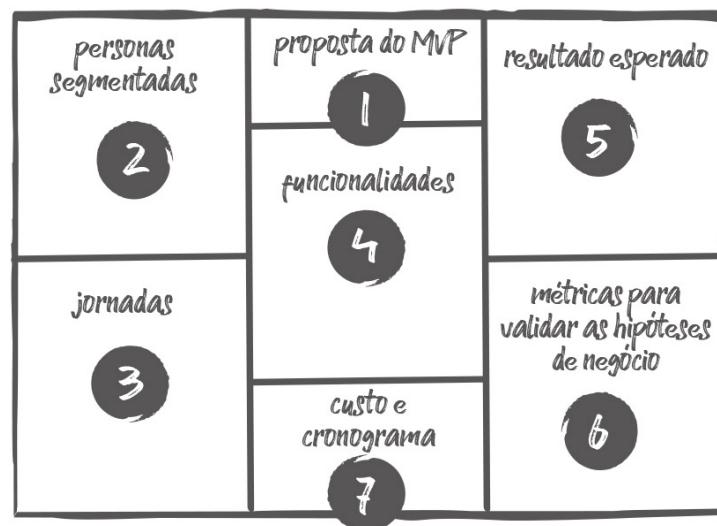
Depois de definir as funcionalidades do MVP, tente conectá-lo aos resultados esperados e às hipóteses do negócio. O modelo<sup>6</sup> a seguir auxilia com tal declaração:

Nós acreditamos que (esse MVP)  
vai conseguir (resultado esperado).  
Saberemos que isso aconteceu com base em (métricas  
para validar as hipóteses do negócio)

Você precisa preencher este modelo, pois, se não conseguir completá-lo, não saberá o que esperar do MVP ou não saberá como medi-lo. E, em qualquer um desses dois cenários, o produto estará à deriva, sem direcionamento.

## Os blocos do Canvas MVP

O Canvas<sup>7</sup> possui sete blocos que descrevem os pontos que você precisa discutir e definir sobre o seu projeto: a visão do MVP, as hipóteses de negócios, as métricas, as personas e suas jornadas, as funcionalidades e o custo e cronograma para sua criação.



Segue a ordem recomendada para preencher (e pensar) os blocos:

- 1. Proposta do MVP :** Qual é a proposta deste MVP?
- 2. Personas segmentadas:** Para quem é este MVP? Podemos segmentar e testá-lo em um grupo menor?
- 3. Jornadas:** Quais jornadas são atendidas ou melhoradas por este MVP?

- 4. Funcionalidades:** O que vamos construir neste MVP? Que ações serão simplificadas ou melhoradas nele?
- 5. Resultado esperado:** Que aprendizado ou resultado estamos buscando neste MVP?
- 6. Métricas para validar as hipóteses do negócio:** Como podemos medir os resultados deste MVP?
- 7. Custo e Cronograma:** Qual é o custo e a data prevista para a entrega deste MVP? Depois de entregue, durante quanto tempo precisaremos coletar os dados para decidir se mudaremos ou prosseguiremos?

O Canvas MVP é explicado em detalhes na obra *Lean Inception*.<sup>8</sup> O livro explora o conceito de MVP e descreve como executar efetivamente um workshop Lean Inception para alinhar um grupo de pessoas sobre o MVP. O Canvas MVP é a última atividade em um workshop Lean Inception!

## » LEAN INCEPTION

A obra *Lean Inception* traz o passo a passo completo de um workshop que busca alinhar pessoas e construir o produto certo. A seguir, vamos explicar resumidamente como ela funciona.

### Qual o objetivo da Lean Inception?

Projetos ágeis colocam ênfase nas entregas antecipadas e contínuas de software com alto valor de acordo com os objetivos do negócio e as necessidades dos principais usuários. A criação do produto enxuto promove a liberação incremental do produto, produto mínimo viável, ou *minimum viable product* em inglês (MVP), a versão mais simples de um produto que pode ser disponibilizado para o negócio.

Mas como entender o MVP e começar um projeto ágil o mais rápido possível? Como garantir que a equipe comece a criação do produto com um bom alinhamento inicial e um plano eficaz?

Em uma única semana de trabalho colaborativo, a equipe vai entender os objetivos do produto, os principais usuários e o escopo funcional de alto nível, de tal forma que a duração do projeto possa

ser estimada e uma estratégia de lançamento incremental para o produto possa ser identificada.

Durante a Lean Inception, atividades dinâmicas acontecem para definir objetivos, estratégias e definição do produto, bem como mapear e priorizar as funcionalidades desejáveis para serem entregues gradualmente, construindo cada incremento do produto. O principal objetivo do workshop é fazer com que a equipe descubra e compreenda coletivamente o que vai ser desenvolvido. No final, o time deve estar mais entrosado e com uma visão mais clara do caminho a seguir!

## Quando devo realizar a Lean Inception?

Você deve agendar uma Lean Inception se estiver buscando alinhar uma equipe para construir o produto certo. Os motivos mais comuns para marcar uma inception são:

- » Alguém investiu na sua startup e você vai colocar a sua ideia em execução;
- » A sua empresa está modernizando algo e vai refazer um produto já existente;
- » Um grupo anterior fracassou na criação do produto, agora é segunda tentativa, com mais pressão e menos recurso;
- » Pesquisas indicam um bom direcionamento de negócio, agora o grupo busca o *product-market fit*;<sup>9</sup>
- » Entre muitas outras situações.

Antes de sair fazendo, o grupo participa de um workshop colaborativo com uma sequência de atividades para alinhar e definir objetivos, estratégias e escopo do produto. Um workshop que usa técnicas de Design Thinking<sup>10</sup> com uma abordagem de Lean Startup!

## E como ela funciona?

Como um livro de receitas, com uma sequência de atividades rápidas e eficazes, a Lean Inception vai permitir que a equipe:

- » Descreva a visão do produto;
- » Priorize os objetivos do produto;
- » Descreva os principais usuários, seus perfis e as suas necessidades;
- » Entenda as principais funcionalidades;
- » Compreenda os níveis de incerteza, esforço e valor de negócio de cada funcionalidade;
- » Descreva as jornadas mais importantes dos usuários;
- » Crie um plano de entrega incremental do produto, impulsionado pelo conceito de MVP;
- » Conheça várias atividades quebra-gelo para tornar o ambiente da inception mais descontraído e divertido;
- » Calcule os custos e especifique datas e cronograma de entrega.

Colaboração é o ato de trabalhar junto para fazer uma tarefa e alcançar objetivos comuns. O sucesso de uma inception está diretamente atrelado à capacidade do grupo envolvido de colaborar

efetivamente em cada atividade relatada acima e descrita no livro *Lean Inception*.

A inception propõe um processo colaborativo de descoberta e esclarecimento no qual as pessoas envolvidas trabalham juntas em uma sequência de atividades para compreender opções e elaborar o MVP. As atividades apresentadas no livro *Lean Inception* representam métodos estruturados de colaboração, buscando um ambiente criativo com partilha de conhecimento, aprendizagem e construção de consenso. As atividades visam aumentar o sucesso de equipes, como elas se envolvem no esclarecimento e resolução de cada passo na direção ao MVP!

## Qual é o papel da facilitadora?

Workshops bem orquestrados têm algo em comum: 1. alguém pensou sobre a sua estrutura e 2. alguém facilitou o mesmo. A pessoa facilitadora da inception é um guia que fomenta as discussões dos participantes durante o workshop. Para isso, a facilitadora possui muita familiaridade e experiência com o formato da inception, sua natureza colaborativa e a sequência de atividades que serão realizadas.

Entretanto, o papel de fomentar a discussão não implica que a facilitadora seja a participante principal. Muito pelo contrário: a pessoa facilitadora deve ser uma mediadora, aquela que propicia o fluxo de ideias e conversas ativas entre todos os participantes, os principais interlocutores do workshop. Consequentemente, o trabalho da facilitadora é garantir que os participantes tenham

responsabilidade, liderança e colaboração ao longo de todas as atividades planejadas.

Em outras palavras, o objetivo da facilitadora é dar suporte aos participantes para que possam participar efetivamente de cada atividade e interação planejada para o workshop, dedicando-se ao processo e ao conteúdo, e assegurando que este último seja gerado de acordo com as expectativas e metas.

## » UM OVERVIEW DO KANBAN

Kanban é um método formulado por David J. Anderson<sup>11</sup> para gestão do fluxo de trabalho de um processo incremental e evolutivo. Influenciado pelo modelo Just-in-time da Toyota, o método se baseia em visualizar o fluxo de trabalho e, a partir disso, atuar no processo para não sobrecarregar os membros da equipe.

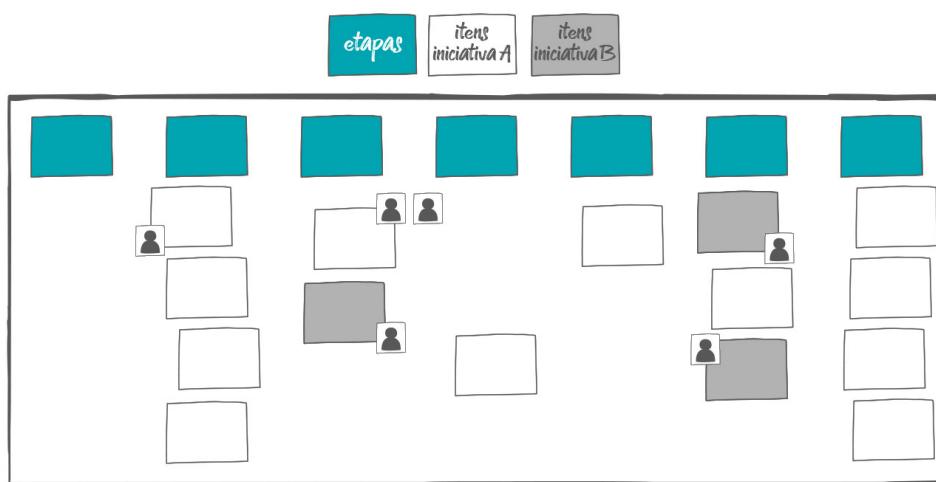
Graças a uma abordagem de gestão visual perante a cadeia de valor, o processo, desde sua etapa inicial até a entrega do trabalho, é exposto aos membros da equipe. Tipicamente a cadeia de valor é representada em quadros brancos com post-its ou ferramentas online. Os itens de trabalho estão visualmente representados nesses cartões, comumente chamados de kanban, o mesmo nome do método (kanban com k minúsculo é o cartão, enquanto Kanban com K maiúsculo é o método de David Anderson). A partir do quadro, fica mais fácil para a equipe decidir o quê, por quê, quando e quanto produzir.

No desenvolvimento de software, normalmente uma pequena tarefa leva horas e até dias para ser concluída. Além disso, você não consegue visualizar quantos requisitos estão atualmente em análise ou quantos requisitos estão atualmente sendo codificados ou testados. O fato é que não conseguimos “ver” o item de trabalho relacionado ao software e nem como ele se move ao longo das

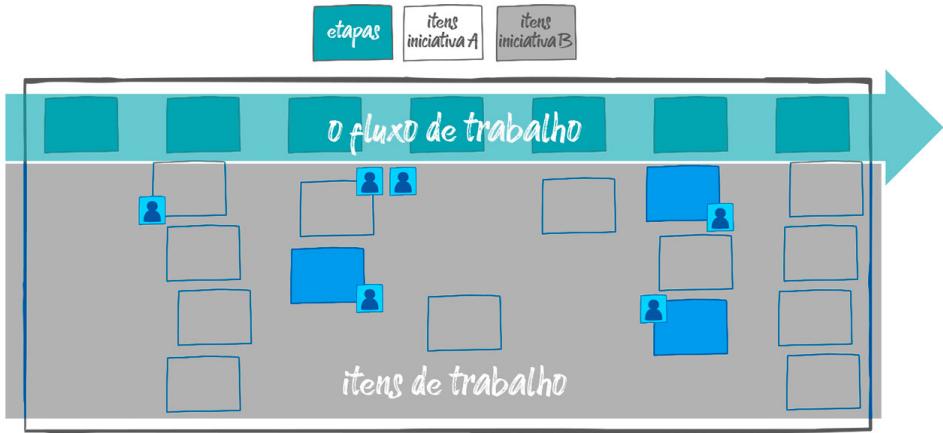
etapas do processo, até que esteja pronto. Aqui é onde tudo começa: o kanban torna tais itens em construção visíveis!

## Visualize o workflow

A ideia principal do kanban é colocar o fluxo de trabalho na frente de todos. Por exemplo em um quadro branco ou na própria parede. Abaixo está uma representação de um kanban de uma equipe de desenvolvimento de software.

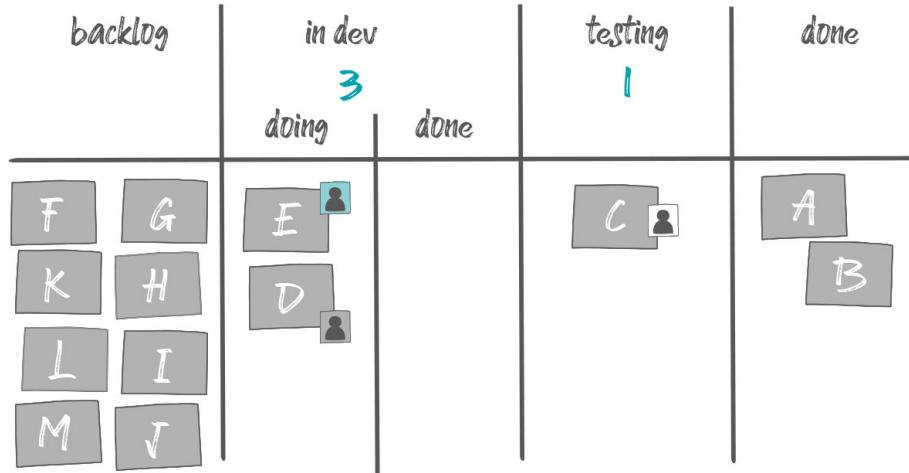


Como a parede é uma superfície bidimensional, o kanban é apresentado em um formato tabular, em que as etapas de trabalho são títulos de colunas e os itens de trabalho, as fotos das pessoas e outras marcas relacionadas ao projeto preenchem o espaço na parede. Esses cartões podem ser organizados em uma linha horizontal ou não, tudo depende da equipe e de como ela representa e organiza o seu trabalho na parede.



## Limite o Work In Progress (WIP)

Limitar o trabalho em andamento, ou WIP, implica que o kanban seguirá um sistema puxado. O trabalho em cada etapa do processo é limitado de forma que uma nova tarefa somente seja adicionada/“puxada” para a próxima etapa quando houver capacidade disponível dentro do limite WIP de tal etapa. As restrições de WIP identificam gargalos e áreas problemáticas no processo, auxiliando o time a tomar decisões para resolvê-las. Limitar o WIP é o grande diferencial do método Kanban: tal artimanha é o divisor de águas entre *task boards* – ou quadros visuais, como eram conhecidos antes da influência de David Anderson com a divulgação do método Kanban – e os quadros kanban.



Na imagem acima, os números 3 e 1 representam, respectivamente, o limite do WIP para a etapa in dev e o limite do WIP para a etapa testing. Ou seja, a etapa in dev deverá ter no máximo três tarefas ao mesmo tempo, enquanto a etapa testing deverá ter somente uma.

## Foco no fluxo de trabalho

Segundo David Anderson, o ponto principal de implantar um kanban é criar uma mudança positiva. Antes de provocar essa alteração, o time tem que saber o que mudar. Para tanto, deve verificar como os itens de trabalho fluem ao longo do processo e analisar possíveis áreas problemáticas que restringem o sistema. Dessa forma, o time deve implementar as mudanças no processo de trabalho para resolver tais problemas.

E assim sucessivamente: identificando problemas e agindo para resolvê-los. Tudo isso baseado na visualização e limites WIP do Kanban. Melhorando o trabalho e o processo, na busca contínua por maior eficiência e eficácia.

## Puxando a melhoria contínua

O kanban pode ser visto como uma ferramenta de comunicação para rapidamente adquirir e divulgar conhecimento entre os membros de uma equipe. O objetivo principal é dar a todos os membros do time uma visão compartilhada do processo e do estado atual do trabalho. Para esse fim, o kanban tem representações visuais para as fases de trabalho, para as pessoas e para o trabalho em si.

Mas o método Kanban vai além disso. A partir de uma simples abordagem – visualização do workflow e limites de WIP –, a equipe trabalha em um sistema puxado, que auxilia na identificação de gargalos e na atuação sobre áreas problemáticas no processo. Sendo assim, o Kanban fornece à equipe uma ferramenta, uma prática e um processo de melhoria contínua!

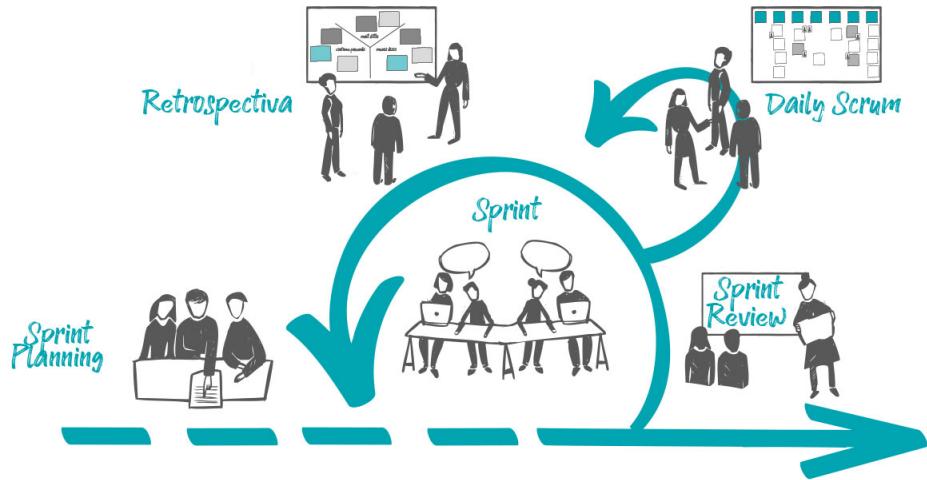
## » UM OVERVIEW DO SCRUM

Scrum é um framework ágil<sup>12</sup> para desenvolvimento, entrega e sustentação de produtos complexos. Scrum foi inicialmente formalizado para projetos de desenvolvimento de software, mas tem sido aplicado para qualquer âmbito de projetos complexos e trabalhos inovadores.

O Scrum é especialmente adequado para projetos com requisitos que mudam rapidamente ou são altamente emergentes. O desenvolvimento de software com Scrum progride através de uma série de iterações chamadas de Sprints, que duram, tipicamente, de uma a quatro semanas.

### As reuniões do Scrum

O framework Scrum sugere que cada Sprint comece com uma reunião de planejamento, para definir o trabalho que deverá ser feito durante a Sprint, e termine com uma reunião de retrospectiva, na qual a equipe busca a melhoria contínua quanto ao processo, à entrega e à interação entre as pessoas.



Além das reuniões de planejamento e retrospectiva, o Scrum sugere mais dois eventos que ocorrem a cada Sprint: reunião de revisão do trabalho, em que o time inspeciona o resultado da Sprint e decide futuras adaptações; e a reunião diária, na qual o time inspeciona o progresso em direção à meta da Sprint e adapta o backlog da Sprint conforme necessário. Nessa reunião, geralmente todos os membros do time ficam de pé (para que a reunião não demore demais) e respondem três perguntas, as quais auxiliam o time a se auto-organizar, buscando o alinhamento diário em relação ao trabalho da Sprint. As três perguntas são: O que fiz ontem? O que vou fazer hoje? O que está impedindo o progresso do meu trabalho? Além das reuniões sugeridas pelo Scrum, muitos times também fazem reuniões de refinamento durante as Sprints. Nessas reuniões, o backlog do produto é revisitado para buscar o entendimento dos próximos requisitos, candidatos à próxima Sprint.

## A Sprint

Sprints promovem um cadênciia, tipicamente de uma a quatro semanas, dependendo da preferênciia do time. Essa cadênciia permite previsibilidade, garantindo a inspeção e adaptação do progresso.

No mundo ágil Scrum, evitamos descrições completas e detalhadas sobre como tudo deverá ser feito na Sprint. Isso porque o time vai saber a melhor forma de resolver o problema em questão.

por isso que a reunião de planejamento da Sprint é descrita em termos das metas e do resultado desejado. O resultado desejado é um compromisso com um conjunto de funcionalidades ou histórias a serem desenvolvidas na próxima Sprint, buscando, assim, o equilíbrio entre autonomia, flexibilidade e comprometimento do time. Tal compromisso é revisitado na reunião de revisão.

Todo o trabalho necessário para atingir o objetivo do produto é realizado pelo time durante as Sprints, incluindo os demais eventos: reunião de planejamento, reuniões diárias, reunião de revisão e retrospectiva. No decorrer de uma Sprint, não deverá ocorrer nenhuma mudança que possa colocar em risco a meta da Sprint.

## Os papéis do Scrum

Scrum fomenta uma equipe multifuncional e com auto-organização. A eficiênciia do time depende da capacidade dos membros de trabalhar em conjunto e de fazer o melhor uso das habilidades dos indivíduos: multifuncionais. O time Scrum é auto- organizado, pois não deve existir um líder de equipe que decide quem vai fazer qual tarefa e como. Tarefas e problemas são levantados por todos.

Os times Scrum possuem três papéis específicos. O primeiro é o papel de Scrum Master (SM), uma pessoa experiente com o framework que possui a responsabilidade de garantir que o framework Scrum seja entendido e aplicado. Seu principal objetivo é maximizar a eficiência do time. Os melhores Scrum Masters são aquelas pessoas que sentem mais satisfação em facilitar o sucesso dos outros do que os próprios. A pessoa no papel de Scrum Master deve se sentir confortável e segura com o framework a ponto de dar todo controle em relação ao produto para a Product Owner (PO), e também todo controle em relação ao desenvolvimento à equipe.

O segundo papel específico é o de Product Owner (PO). A pessoa no papel de PO representa o negócio, os clientes ou usuários, e orienta o time para a construção do produto certo. É sua principal função priorizar o backlog com base no alinhamento entre as partes interessadas, tanto internas quanto externas ao time Scrum. A PO maximiza valor através de um processo contínuo de desenvolvimento de produtos, guiado por aprendizado e experimentação. Ela deve liderar o esforço de desenvolvimento por meio de esclarecimentos e priorizações sobre o trabalho. Como tal, a PO deve estar disponível para a equipe para responder perguntas e direcionar o time em cada momento ou indagação.

Essa combinação de autoridade e disponibilidade para a equipe de desenvolvimento faz com que o papel de PO seja peça-chave do framework. Scrum valoriza a auto-organização e autonomia do time. Portanto, a PO deve respeitar o direcionamento e a capacidade da equipe para criar o próprio plano de ação.

Todas as demais pessoas do time Scrum são consideradas como desenvolvedoras, sendo este o terceiro papel do Scrum. Essas são

as pessoas de habilidades variadas comprometidas em criar todo e qualquer incremento do produto.

## Os artefatos Scrum

Scrum tem três artefatos: o backlog do produto, o backlog da Sprint e o incremento do produto. Eles compartilham o mesmo objetivo: maximizar a transparência e promover o alinhamento sobre o trabalho.

O backlog do produto e o backlog da Sprint descrevem o trabalho a ser desenvolvido, respectivamente, para o produto e para o Sprint. O incremento é um pedaço do produto a ser desenvolvido. Cada incremento é adicionado aos incrementos previamente entregues. Vários incrementos podem ser criados em uma Sprint. Para fornecer valor, todos e cada incremento deve ser utilizável.

## Trabalho em equipe, alinhamento cadenciado e clareza

A equipe Scrum – Scrum Master, PO e todas as desenvolvedoras do time (com suas variadas formações) – participa ativamente de todas reuniões com um alto nível de autonomia, transparência e comprometimento. Na reunião de planejamento da Sprint, a equipe decide o backlog da Sprint – o qual é acompanhado diariamente e reavaliado na reunião de revisão. Por intermédio da busca de melhoria contínua (principal objetivo da reunião de retrospectiva), tipicamente o time Scrum alcança níveis elevados de rendimento. Muito disso é alcançado pelo trabalho em equipe, pelo alinhamento

cadenciado via Sprints e pela clareza de cada papel e de cada reunião.

## » FUNCIONALIDADES E HISTÓRIAS DE USUÁRIO

As funcionalidades são tipicamente descritas em um nível mais elevado do que as histórias do usuário. Portanto, antes de começar a trabalhar em uma funcionalidade, a mesma deve ser analisada e detalhada em suas respectivas histórias.

Funcionalidade é a descrição de uma ação ou interação de um usuário com o produto. Por exemplo: imprimir nota fiscal, consultar extrato detalhado ou convidar amigos do Facebook. A descrição de uma funcionalidade deve ser o mais simples possível: o usuário está tentando fazer uma coisa. O produto deve ter uma funcionalidade para isso. Qual é ela?

Tipicamente, as equipes de desenvolvimento trabalham com histórias de usuários. Portanto, você deve detalhar um pouco mais e realizar o mapeamento de funcionalidades para histórias.

### História de usuário

História de usuário é um formato textual para a descrição concisa de um requisito que busca responder três indagações básicas: quem, o quê e por quê. Tipicamente, um MVP tem uma ou mais

funcionalidades que são convertidas em algumas histórias do usuário no seguinte formato:

- » Como um(a) (perfil de usuário/persona)
- » Eu quero (ação a ser realizada/funcionalidade específica do produto)
- » Para que (valor para o negócio e/ou para o usuário)

Bill Wake, em seu livro *Extreme Programming Explored*,<sup>13</sup> criou o acrônimo INVEST que define um conjunto simples de regras usadas na criação adequada de histórias de usuários. Cada letra do acrônimo representa uma das seis características importantes de uma história de usuário: independente, negociável, valiosa, estimável, small (pequena) e testável.

Alguns anos após a criação do acrônimo, Mike Cohn,<sup>14</sup> um dos grandes autores sobre histórias de usuários, renomeou a letra S, de small, para *sized appropriately* (sob medida), refletindo que algumas pessoas criavam histórias um pouco maiores, mas adequadas ao seu contexto.

- » **Independente:** Uma história não depende de outra.
- » **Negociável:** Uma história captura a essência do que é desejado. Não é um contrato fechado, conversas e negociação são bem-vindas.
- » **Valiosa:** Uma história descreve, claramente, o valor para o cliente.
- » **Estimável:** Uma história fornece informações suficientes para o time elaborar uma estimativa de alto nível.

- » **Sob medida:** Uma boa história deve ser relativamente pequena em tamanho para ser concluída no menor tempo possível e caber em uma iteração, considerando o contexto do time.
- » **Testável:** Uma história deve estar clara o suficiente para que os testes possam ser definidos para ela.

O acrônimo INVEST ajuda a escrever boas histórias: Ela é independente? Negociável? Valiosa para o negócio? Estimável? Sob medida? Testável?

Além do INVEST, uma boa história de usuário consiste em três elementos, comumente chamados de 3C's:

- » **Cartão:** A descrição da história do usuário deve caber em um cartão índice, contendo o suficiente para identificá-la. O formato mais comum é:  
Como (usuário específico)  
quero (executar esta ação)  
para que eu (alcance determinado objetivo)

- » **Conversa:** A principal intenção de colocar as histórias de usuário em um formato de cartão de índice é que não há espaço para escrever muito. Logo, muita conversa é necessária para esclarecer as dúvidas e detalhar o trabalho necessário para implementar a mesma. Trabalhar com histórias de usuário significa aceitar que as conversas sobre o trabalho serão contínuas, não somente no início quando o requisito é inicialmente definido.

» **Confirmação:** É onde determinamos se o objetivo da história do usuário é alcançado. Para tanto, os critérios de aceitação confirmam que a história do usuário foi implementada corretamente e entregue com êxito. Os critérios de aceitação devem ser definidos para cada história antes que a equipe comece a implementá-la, dessa forma, não há surpresas no momento de verificar a entrega da mesma.

Histórias de usuários têm três aspectos críticos: o **cartão** da história deve seguir um modelo simples (como um(a)... Eu quero... para que...). A partir disso, teremos **conversas** colaborativas para melhor entender e detalhar a história, até que, quando pronta, receberemos a **confirmação**, geralmente pelo dono do produto (PO).

## Critério de aceite

Critério de aceite é um formato textual que descreve como testar uma funcionalidade. Tipicamente uma história do usuário terá alguns critérios de aceite (ou ACs, do inglês *acceptance criteria*).

- » **Dado que** (cenário inicial)
- » **Quando** (ação realizada)
- » **Então** (estado esperado)

Os ACs definem os limites para uma história de usuário. A partir deles, todos os envolvidos irão saber se a história está completa, uma vez que haja confirmação do PO sobre a mesma.

## Tarefas

É muito comum quebrar uma história em pedaços ainda menores sobre o trabalho que deve ser feito: essas são as tarefas. Ao listar as tarefas necessárias para construir uma história, a equipe de desenvolvimento entra em detalhes técnicos de como os pedaços menores serão implementados. Diferente das histórias, as tarefas não seguem um formato textual definido. Elas são mais diretas, com uma linguagem bem técnica, da equipe de desenvolvimento para a equipe de desenvolvimento.

Uma tarefa identifica algo que precisa ser feito, algo necessário para uma história. Como tal, a tarefa não será necessariamente independente e não irá demonstrar o valor de negócio. A maioria delas tende a ser para pessoas programadoras, descritas com termos utilizados pelas mesmas. Alguns exemplos de tarefas são: alterar campos da tabela de partida, criar contas de teste para usuários, automatizar os scripts de geração de dados e assim por diante.

## Um exemplo

Segue abaixo um exemplo de funcionalidade que foi dividida em três histórias, com alguns critérios de aceite e tarefas.

Funcionalidade: consultar partidas de futebol disponíveis sem geolocalização

### História 1

- › Como um *jogador não cadastrado*,

- › eu quero consultar partidas próximas a um endereço informado
- › para que eu encontre um jogo próximo ao meu local atual.

## História 2

- › Como um jogador cadastrado,
- › eu quero consultar partidas próximas à minha residência
- › para que eu encontre um jogo próximo à minha residência.

## História 3

- › Como um jogador cadastrado,
- › eu quero consultar partidas próximas ao meu trabalho
- › para que eu encontre um jogo próximo ao meu trabalho.

### Critérios de aceite da história 3 (exemplo 1)

- › Dado que existe uma partida a menos de 10 quilômetros do meu trabalho,
- › quando procuro por uma partida próxima ao meu trabalho
- › então encontro tal partida.

### Critérios de aceite da história 3 (exemplo 2)

- › Dado que não existe nenhuma partida a menos de 10 quilômetros do meu trabalho;
- › Quando procuro por uma partida próxima ao meu trabalho;
- › Então não encontro nenhuma partida.

## Tarefas da história 3

- › Criar UI para mostrar partidas próximas ao local de trabalho (com dados *hardcoded*);
- › Criar lógica no *backend* para busca por proximidade (*hardcoded* 10 km);
- › Alterar parâmetro *hardcoded* para campo de configuração;
- › Criar dados de teste para buscar partida nas proximidades do trabalho;
- › Alterar DB para incluir endereço de trabalho.

## » MODELO SPOTIFY

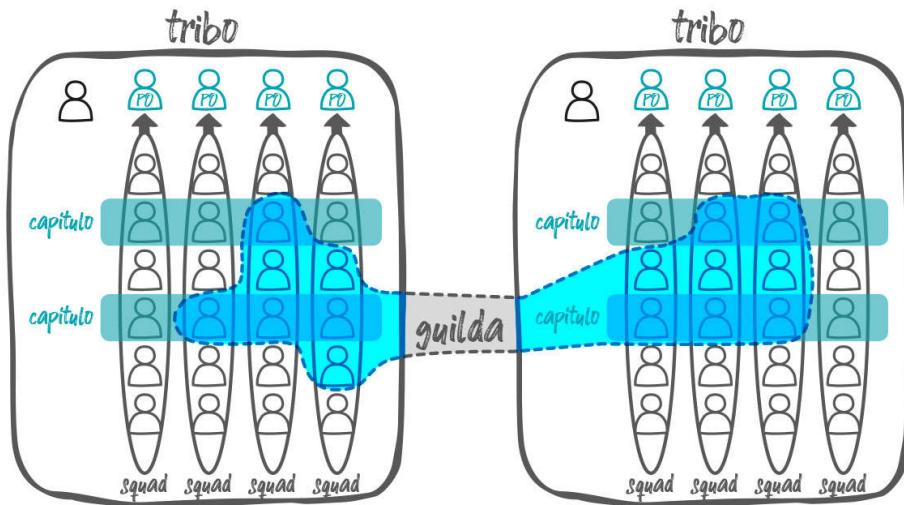
O modelo Spotify é o nome dado a um tipo de organização de times de produto. A Spotify é uma empresa de desenvolvimento de produtos, famosa por seu aplicativo de música. Em 2012, a empresa estava em um crescimento acelerado, com trinta times de produto, distribuídos entre três cidades.

Nesse ano, Henrik Kniberg e Anders Ivarsson publicaram um artigo “Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds”<sup>15</sup> com a proposta de como lidar com várias equipes em uma organização de desenvolvimento de produtos. Essa proposta ficou conhecida como o modelo Spotify.

O modelo é resumido pela imagem a seguir, que demonstra os seguintes pontos:

- » Uma tribo é formada por algumas equipes que trabalham em áreas relacionadas (ex: tribo de infra de *backend*);
- » Um *squad* é similar a uma equipe Scrum, uma equipe multifuncional com especialistas em uma área do produto (ex: *squad* de Indexação avançada);
- » Um capítulo é formado por pessoas de um mesmo cargo (exemplo: capítulo de desenvolvedoras web);

- » Uma guilda é uma comunidade de interesse, formada por pessoas de cargos distintos (exemplo: guilda de agile coaching).<sup>16</sup>



## » OBJETIVOS E RESULTADOS-CHAVE - OKR

Os objetivos e resultados-chave (ou OKRs, do inglês *objectives and key results*), são uma estrutura de definição de metas criada pela Intel nos anos 70 que é utilizada com muito sucesso pelos investidores do Vale do Silício, portanto, adotada em grandes empresas como Google,<sup>17</sup> Oracle, Twitter, LinkedIn e Dropbox. Atualmente, os OKRs estão sendo aplicados em todo o mundo, desde startups até grandes organizações.

A ideia principal dos OKRs é que a organização defina um conjunto de objetivos em cascata. Por exemplo, OKRs devem ser definidos primeiramente para a organização. Em seguida, OKRs são definidos para os departamentos e unidades de negócios, e, depois, para as equipes. E assim por diante, em cascata até atingir os indivíduos, configurando seus próprios OKRs. Cada entidade (organização, departamento, indivíduo) deve ter de quatro a seis objetivos e até cinco resultados-chave por objetivo.

O seguinte é um modelo para OKR compartilhado por Felipe Castro.<sup>18</sup>

**Nós vamos (objetivo),  
conforme medido por (este conjunto de resultados-chave).**

Felipe então explica que um OKR tem dois componentes, o objetivo (o que queremos alcançar) e um conjunto de resultados-chave (como sabemos se estamos chegando lá).

O objetivo pode ser qualitativo, enquanto os resultados-chave devem ser quantitativos. O objetivo define uma meta por um período de tempo definido, geralmente um trimestre. Os resultados-chave indicam se o objetivo foi atingido até o final do tempo.

## OKR – exemplo

Segue um exemplo do Felipe Castro sobre um OKR de uma empresa de tecnologia que deseja aumentar o envolvimento e a satisfação do cliente.

### **Objetivo:**

Encantar nossos clientes.

### **Principais resultados::**

- › Visitas recorrentes: média de 3,3 visitas por semana por usuário ativo.
- › Alcançar NPS (*net promoter score*) de 90.
- › Tráfego não pago (orgânico) de 80%.
- › Engajamento: 75% dos usuários preenchem um perfil completo.

## » SOBRE ALGUMAS PRÁTICAS ESPECÍFICAS

As práticas descritas a seguir foram utilizadas no dia a dia do time. Elas aparecerão mais adiante no livro e, por isso, também estão explicadas nesta seção.

### Sobre o estilo da Daily Scrum

Diariamente o time seguindo o Scrum realiza uma reunião para verificar o andamento das tarefas de trabalho. Essa reunião é a Daily Scrum.

#### Daily Scrum tradicional ou Daily Round-Robin

De pé, o time responde às três perguntas listadas a seguir. Elas auxiliam o time a se auto-organizar, buscando o alinhamento diário em relação ao trabalho da Sprint:

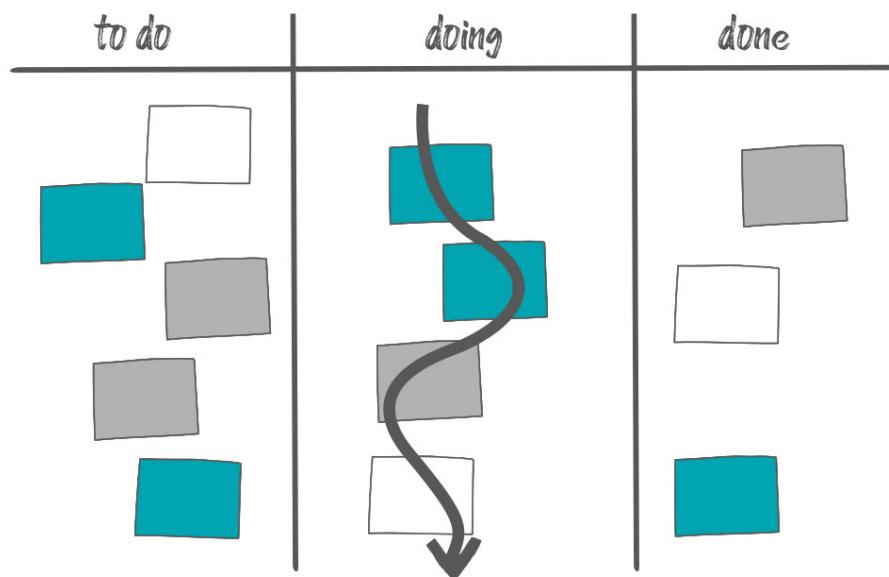
- » O que fiz ontem?
- » O que vou fazer hoje?
- » O que está impedindo o progresso do meu trabalho?

Chamamos essa reunião de Daily Scrum tradicional ou Daily Round-Robin.<sup>19</sup>

Round-Robin é o algoritmo que atribui frações de tempo para cada processo em partes iguais e de forma circular. E é assim que são as Daily Scrum tradicionais: o time forma um círculo e cada pessoa tem uma fração de tempo para responder às três perguntas.

### Daily Scrum top down

Muitas equipes dão visibilidade do seu trabalho (tarefas, histórias, qualquer que seja o item de trabalho da equipe) via alguma lista *top down* de itens de trabalho representados em um quadro visual (por exemplo a coluna *doing* no quadro *to do – doing – done*).



Esses quadros visuais podem ter vários nomes (inclusive o nome de ferramentas on-line). Os mais comuns são: kanban, quadro de tarefas, quadro Scrum, parede de cartões, Trello e Jira.

Na Daily Scrum *top down* a equipe segue o quadro visual conversando sobre cada um dos itens na etapa *doing*, de cima para

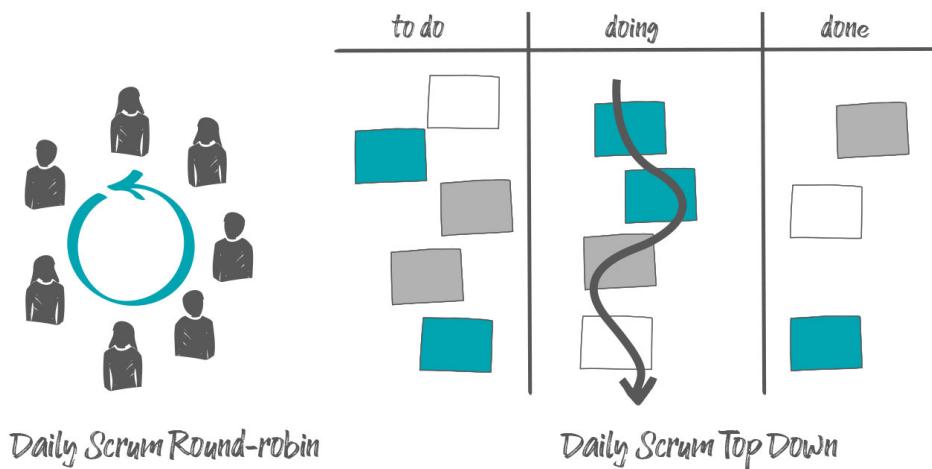
baixo. Nesse estilo, as três perguntas da Daily Scrum devem ser:

- » O que fizemos nesse item ontem?
- » O que vamos fazer nesse item hoje?
- » O que está impedindo o progresso desse item?

O item no quadro visual geralmente identifica quem está responsável pelo mesmo. Nesse caso, essa pessoa dá o update sobre a tarefa e a Daily Scrum segue na ordem *top down* dos itens da equipe.

Qual o estilo atual da sua Daily Scrum?

É muito comum uma nova equipe adotar um desses dois estilos. E também acontece que, de tempos em tempos, a equipe vai alterá-lo.



Recomendamos que reconheça e decida qual estilo estão usando e evite misturar ambos ao mesmo tempo. Siga a ordem Round-Robin das pessoas ou a *top down* da lista de itens em *doing*.

Usar ambos os estilos ao mesmo tempo é como tentar colocar uma perna em uma bicicleta e outra em um skate: não vai dar certo!

## Planejando uma retrospectiva

A retrospectiva é uma reunião de aproximadamente uma hora que acontece com frequência (geralmente uma por semana ou uma vez por Sprint para times Scrum), na qual as pessoas do time conversam sobre o que foi bom no período, o que pode ser melhorado e o que devem fazer para melhorar.

A retrospectiva é essencial para unir o time em um objetivo comum e para evitar que desentendimentos se elevem a níveis de desconforto interpessoal, que, por sua vez, levam à redução da comunicação e à ruptura do grupo.

Ela também cria o ambiente para as pessoas refletirem e escutarem sobre como todos estão se sentindo, para conversarem sobre um assunto específico (relacionado ao contexto e ao momento da equipe), e para buscarem o Kaizen – a melhoria contínua.

A seguir estão algumas perguntas que você deve responder ao planejar sua retrospectiva:

- > **Momento:** Onde você está na linha do tempo? Você está iniciando um time? Você está tentando aprender com o passado? Você está se preparando para um próximo lançamento?
- > **Contexto:** O que você está buscando? Você está procurando focar em itens de ação? Você precisa aumentar a moral do time? É tempo de se conhecerem? Você deseja obter uma métrica da felicidade do time?

Com base nesses fatores, você precisa escolher qual atividade (ou sequência de atividades) é mais apropriada de ser executada.

Além de pensar na retrospectiva adequada para o contexto e momento da sua equipe, muitas equipes têm a retrospectiva como uma reunião recorrente. Por isso buscam alternativas para evitar a monotonia, causada por seguir sempre a mesma agenda, com as mesmas atividades. Ao variar a agenda da retrospectiva e as atividades, a equipe pode olhar para diferentes ângulos e perspectivas, gerando novos insights.

No livro e site *FunRetrospectives*<sup>20</sup> é sugerida uma agenda de sete passos para retrospectivas eficazes. Ao longo desta obra você encontrará alguma dessas retrospectivas. Elas foram baseadas no site, que também foi usado para criar a agenda e os quadros para as retrospectivas remotas.

## 1. Contexto

Por que esta retrospectiva está sendo realizada? O que ela vai cobrir?

Retrospectivas, assim como qualquer outra reunião, são mais eficazes se os participantes alinharem suas expectativas antes de participarem. Você pode começar com um contexto pré-definido ou defini-lo em tempo real com os participantes. Mas deve ficar claro para todos qual é o contexto dessa retrospectiva.

## 2. Diretiva Primária

No livro *Project Retrospectives: A Handbook for Team Reviews*,<sup>21</sup> Norman Kerth e Gerald Weinberg apresentam a diretiva primária

para retrospectivas, uma declaração destinada a preparar o cenário para a retrospectiva:

Independentemente do que descobrimos, nós entendemos e realmente acreditamos que todos fizeram o melhor trabalho possível, dado o que era conhecido na época, suas habilidades e competências, os recursos disponíveis e a situação em questão.

A diretiva primária é essencial para definir o tom da reunião. Torne-a visível para todos os participantes e a leia em voz alta no início da sua retrospectiva.

Essa é realmente uma reunião de retrospectiva? Uma reunião que olha para trás"? Caroli e Tainã, autores do livro *FunRetrospectives*, reconheceram que alguns contextos das reuniões de "retrospectivas" não eram sobre o passado, mas sim sobre formação de equipes ou sobre uma perspectiva futura. Por isso eles criam e disponibilizaram outras duas diretivas primárias, respectivamente para *team building* e *futurespectives*.

### 3. Energizador

Atividades energizadoras, ou quebra-gelo, são simples e rápidas para aquecer a equipe e promover a interação do grupo. Essas atividades ajudam a criar um ambiente amigável e tornam as pessoas mais confortáveis em participar das demais atividades. São essenciais para criar um clima mais leve e divertido

(retrospectivas não precisam e não devem ser reuniões pesadas e chatas!).

**Exemplos de atividades:** Ping Pong e Que coisa mais louca.

#### 4. Check-in

Check-ins são atividades curtas; pense nelas como um aperitivo para abrir o apetite de todos pelo prato principal e, ao mesmo tempo, fornecer feedback à pessoa facilitadora sobre o envolvimento dos participantes.

**Exemplos de atividades:** Verificação de segurança e Uma palavra.

#### 5. Prato principal

O prato principal é a principal atividade de uma reunião que busca promover a melhoria contínua. É aqui que você coleta dados, verifica o moral da equipe, fala sobre coisas positivas, reconhece pessoas e busca melhorias.

Escolha o prato principal para o momento e o contexto em questão. Esta é a principal atividade da sua reunião; as informações que você reunir e discutir definirão o tom da melhoria contínua.

**Exemplos de atividades:** Linha do tempo com altos e baixos e Pequena estrela-do-mar.

#### 6. Filtragem

Após o prato principal, talvez você tenha muitos dados à sua frente. É importante ter critérios bem definidos para decidir o que será discutido. No tempo limitado da reunião, é possível que você

tenha que deixar alguns tópicos de fora. Se esse for o caso, defina os critérios de filtragem com sua equipe.

**Exemplo de filtros:** grupos com maior número e votação por pontos.

## 7. Check-out

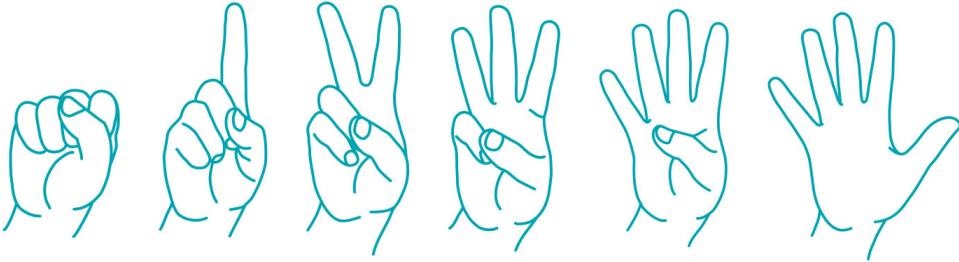
A reunião está quase no fim e você precisa fechá-la. As atividades de check-out ajudam nesse processo de fechamento. Essas são atividades muito curtas. Pense nelas como um check-out eficaz de hotel, onde o hóspede (participante) sai dele rapidamente, com a sensação de ser ouvido e dando um feedback valioso ao hotel. Ele pode até carregar uma nota ou um item de ação em sua bagagem.

**Exemplo de atividades:** Que-quem-quando e Anotação pessoal.

Todos os exemplos listados estão no capítulo “Atividades de retrospectivas”, na seção Anexos.

## Confidence Voting

É uma prática utilizada para dar visibilidade do quanto confiante um time está em relação a algo que está sendo proposto como, por exemplo, entregar a Sprint. Além de remover influências, também funciona como um *safety check* e ajuda a evidenciar quando algo não está legal e precisa ser ajustado antes que o time possa seguir adiante.



Essa prática consiste em pedir para que todas as pessoas do time mostrem um número, ao mesmo tempo, que represente o seu nível de confiança entre 0 e 5.

- 5: Estou completamente segura de que posso fazer isso.
- 4: Não consigo pensar em nenhum problema agora, acredito que posso fazer.
- 3: Existem poucos riscos com baixa probabilidade e/ou impacto. É factível.
- 2: Tenho algumas preocupações, mas estou disposta a tentar.
- 1: Tenho grandes preocupações com o que está sendo proposto. Posso seguir a definição do time, porém preciso de mais informações para me sentir confiante.
- 0: Não me sinto nada confiante e não gostaria de seguir adiante com isso.

É importante que ocorra uma conversa caso alguém tenha votado em 0 ou 1, ou se a maior parte dos votos for abaixo de 3. Nesses cenários, possivelmente faltam informações, conhecimentos ou recursos (como, por exemplo, tempo) para que o time possa fazer o que está sendo proposto. A conversa terá como objetivo fornecer informações ou garantias de que os conhecimentos e recursos necessários estarão disponíveis para o time. Se isso não for

possível, a proposta deverá ser revisada para que a equipe se sinta confiante e possa se comprometer.

## TDD

TDD (Desenvolvimento orientado a testes ou *test driven development*, em inglês) é uma forma de desenvolvimento de software na qual o desenvolvimento é guiado pelos testes, como o nome indica. TDD é uma das práticas do eXtreme Programming (XP), método e livro escrito por Kent Beck, publicado em 1999.<sup>22</sup>

TDD se baseia na repetição de um ciclo curto: codificar o teste, codificar o código funcional para fazer o teste passar, melhorar o código. Essa forma de trabalhar incentiva o desenho – design – mais simples e inspira confiança para criação, alteração e melhorias no código.

## » ASSESSMENT DE DEVOPS

O *assessment* aqui apresentado é baseado na versão publicada no artigo da *InfoQ Brasil*<sup>23</sup> escrito por Bruno Tavares e Paulo Caroli, a qual foi inspirada no livro *Entrega contínua* de Jez Humble e David Farley. Cabe ressaltar que o maior benefício em se aplicar tal *assessment* está nas conversas e conselhos dados pela pessoa com bastante experiência prática e teórica em DevOps.

Entrega contínua fornece uma linguagem padrão para as atividades de construir, testar e implantar software no caminho de produção. Além disso, faz referência à Integração contínua como um ponto de partida, mas estendendo a mesma para atender as necessidades do negócio, como um todo, provendo um ciclo contínuo de lançamentos de software.

A avaliação de DevOps é uma conversa baseada numa série de perguntas que ajudam a identificar visualmente oito domínios de maturidade no processo de entrega (contínua) da equipe em uma organização:

1. Processo de entrega;

- 2.** Controle de qualidade;
- 3.** Controle de configuração;
- 4.** Ambientes e deployments;
- 5.** Armazenamento de dados;
- 6.** Arquitetura técnica;
- 7.** Alinhamento organizacional;
- 8.** Visibilidade.

## Processo de entrega

Capacidade de construir e testar uma base de lançamento do produto para garantir que as mudanças individuais são compatíveis com outras alterações feitas, em sincronia e continuamente. Isso permite que a equipe gerencie o ritmo do seu trabalho e da entrega de um produto de alta qualidade, reproduzível sob demanda.

### Referência – Nota 0

- › Descoberta de erros cometidos devido à incompatibilidade das mudanças acontecem em produção.
- › Não há normas ou práticas para verificar a compatibilidade entre as partes do sistema.

### Referência – Nota 3

- › A prática reproduzível e gerenciada é aplicada de forma consistente (tanto manual quanto automaticamente), utilizando padrões e práticas definidas que são compreendidas por todos.

- A expectativa é que a equipe opere de uma forma consistente e
- › previsível, mas as métricas se concentram em falhas e não no sucesso.

## Referência – Nota 5

- › Cada *build* bem-sucedido gera um candidato para liberação (*release candidate*).
- › O foco está em fazer commits mais frequentes, com o aumento da confiança na qualidade do produto.
- › CI<sup>24</sup> (*continuous integration*) cria ambientes para permitir a escalabilidade dos testes.
- › Testes são executados em paralelo ou em várias máquinas.
- › Pipeline de entrega estende-se até a produção.

## Controle de qualidade

Conceito de descobrir sistemática e rapidamente problemas durante o processo de entrega do produto com ciclos de feedback mais frequentes e mais curtos para garantir a qualidade. Descobrir defeitos mais cedo no ciclo desenvolvimento é menos custoso e mais fácil de corrigir. Os problemas não são expostos para o cliente por serem identificados e resolvidos antes de chegar à produção.

## Referência – Nota 0

- › Não há responsáveis por testes.
- › O produto é entregue diretamente à produção não testado.

- › A função de suporte lida com os defeitos.

### Referência – Nota 3

- › Desenvolver e testar são funções colaborativas.
- › Testar é parte da responsabilidade de todos na equipe de entrega.
- › Práticas estão previstas para encurtar os ciclos de feedback e mover testes para o começo do ciclo.
- › Consciência para construir com qualidade ao invés de reagir ao encontrar algo quebrado.
- › Automação é empregada, mas não de forma abrangente ou sustentável.

### Referência – Nota 5

- › Sistemas “imunológicos” de produção detectam falhas de implantação e as corrigem automaticamente.
- › A equipe tem consciência avançada de práticas de testes e seleciona aqueles que fornecem maior benefício para o produto e também resposta rápida a mudanças.
- › Novas metodologias, técnicas e abordagens são exploradas e aplicadas para melhorar a qualidade do produto.

## Controle de configuração

Capacidade de rastrear as alterações feitas nos artefatos que afetam o comportamento de um sistema, e gerenciar múltiplas contribuições para um único artefato. Isso inclui o código-fonte,

bibliotecas, arquivos de configuração, testes, descrições de ambiente, bibliotecas dependentes, estrutura de banco de dados, documentação de suporte e qualquer outra coisa relacionada à entrega do produto.

### Referência – Nota 0

- › Mudanças são feitas por vários membros da equipe, simultaneamente, sem qualquer esforço para manter as versões ou rastrear quem fez a mudança e quando.
- › Se existe controle de versão, é geralmente realizado por indivíduos que precisam de estrutura para organizar as suas atividades.
- › Não há formas de uma equipe reverter as mudanças para uma versão de trabalho anterior.

### Referência – Nota 3

- › Os itens necessários para configurar todos os ambientes são identificados.
- › Um único conjunto de ferramentas para gerenciamento de configuração do produto foi determinado e há um esforço para mover todos os artefatos de entrega para o sistema de controle de versão.
- › Scripts de teste, bibliotecas e dependências são gerenciados.
- › Todos os membros da equipe fazem check-ins de forma frequente e regular.

### Referência – Nota 5

- › Há apenas um fluxo de contribuição no projeto para todos os envolvidos, que é validado constantemente através de entregas.
- › A equipe muda as práticas frequentemente à medida que o produto evolui.
- › Novas ferramentas de controle de versão são avaliadas e implementadas para atender às necessidades de evolução das práticas de entrega do produto.
- › O desenvolvimento é todo feito no *branch* principal e a equipe consegue fazer as entregas e a adição de novos recursos sem recorrer a *branches* de longa duração.

## Ambientes e deployments

A disponibilidade de ambientes adequados para o desenvolvimento e testes para garantir que o produto funcionará como esperado em produção. A capacidade para entrar em produção com o mínimo de trabalho e sem interrupção das operações e usuários finais.

### Referência – Nota 0

- › Não há ambientes separados para desenvolvimento e testes.
- › Ambientes de desenvolvimento estão superocupados.
- › Firewalls e configurações de redes são muitas vezes empecilhos para o desenvolvimento.

### Referência – Nota 3

- › Provisionamento automatizado com implantações através de scripts.

- › Ainda há dependência de habilidades individuais para garantir que a implantação funcionará em produção.
- › Ambientes de teste estão prontamente disponíveis e podem ser reproduzidos com trabalho manual e coordenação entre as equipes operacionais.

## Referência – Nota 5

- › Existe capacidade de pipeline para entrega contínua.
- › Ambientes são facilmente replicados sob demanda, conforme necessário, utilizando um modelo de self-service para facilitar o ciclo de feedback ideal.
- › Provisionamento e configuração do ambiente são totalmente automatizadas, de preferência usando um sistema baseado em nuvem.
- › Ambientes são revistos regularmente para simplificar e otimizar o uso eficaz da tecnologia.
- › Provisionamento é escalável para atender às demandas flutuantes.

## Armazenamento de dados

Capacidade para que todos possam acompanhar as alterações feitas na estrutura do banco de dados durante cada entrega e também o efeito que essas mudanças trazem. Isso inclui a capacidade de fazer *rollforward* ou *rollback* da versão de todas as alterações.

Alterações de banco de dados devem ser scripts localizados junto a outros artefatos de implantação, e dados de teste reutilizáveis devem ser criados para todos os ambientes.

### Referência – Nota 0

- › A equipe de desenvolvimento desconhece o processo e o funcionamento do banco de dados.
- › Controle completo de dados e banco de dados é realizada por uma equipe externa.

### Referência – Nota 3

- › Alterações de banco de dados são executadas automaticamente como parte da entrega.
- › Os conjuntos de dados são definidos para diferentes finalidades no processo de entrega. Por exemplo: ambiente de desenvolvimento, integração, teste de aceitação de usuário (ou UAT, do inglês *user acceptance testing*), carga e desempenho.
- › Todos os conjuntos de dados são scripts e incluídos no pipeline de entrega.

### Referência – Nota 5

- › Ciclo automático de feedback de desempenho e entrega no banco de dados está presente no processo e é usado para iniciar melhorias.
- › Os testes são realizados usando conjuntos isolados de dados que são bem dimensionados para o propósito de teste (ambientes

anteriores à produção usam conjuntos de dados menores).

## Arquitetura técnica

Pensamento holístico sobre as decisões técnicas e seus efeitos sobre à capacidade de mudança do negócio. Isto é caro e difícil de alterar uma vez que a tecnologia está em uso. O conceito se refere aos mecanismos presentes para tomar decisões de arquitetura e também às decisões em torno dos recursos compartilhados. Modelar componentes em torno das capacidades de negócio para reduzir custos e riscos e facilitar alterações.

### Referência – Nota 0

- › Decisões técnicas são feitas no momento e não há visão de longo prazo ou planejamento técnico.
- › Uso de componentes de alto nível e lógica de negócio em partes difíceis de testar.
- › Dependências não são gerenciadas e nem completamente compreendidas.
- › Dependências não são em torno de capacidades do negócio.
- › Versionamento de dependências é mal gerenciado.
- › Código não tem testes automatizados e a arquitetura não permite o teste unitário automatizado.
- › Arquitetura só permite entregas *big bang*.

### Referência – Nota 3

- › Existem linhas abertas de comunicação entre arquitetos e as equipes de entrega.
- › Equipes de entrega participam de decisões sobre o design.
- › A comunicação entre equipe e arquiteto é informal e frequente.
- › Equipes utilizam práticas modernas para isolar as funcionalidades/serviços para obter habilidade de entregar independentemente.
- › Equipes são maduras o suficiente para decidir quando refatorar e arquitetar o sistema para suportar os novos recursos ou dependências.
- › Existe uma estratégia de versionamento para dependências e como lidar com as mudanças que quebram o design da API, mas ela nem sempre é seguida.

## Referência – Nota 5

- › Existem mecanismos para capacitar a equipe de produto a tomar decisões de arquitetura e tecnologia. Isso porque a visão técnica é clara e transparente.
- › Os arquitetos estão totalmente envolvidos com o negócio para permitir a inovação empresarial.
- › A arquitetura permite medir e ajustar para melhorar o desempenho.
- › A arquitetura pode ser dimensionada de várias maneiras, dependendo das necessidades do negócio.
- › Equipe está habilitada a fazer mudanças tanto no processo de *build*, pipeline quanto nos estágios de entrega durante o ciclo de

vida da arquitetura, para otimizar tanto o todo quanto às áreas locais.

- › Estratégias de versionamento do design da API são bem compreendidas e geridas por todas as equipes.

## Alinhamento organizacional

A capacidade das pessoas da equipe de compartilharem ideias e trabalharem juntas para melhorar os processos e o produto, fornecendo software funcional de uma forma mais rápida e mais segura. Capacidade da equipe para compartilhar conhecimentos e competências e determinar melhorias.

### Referência – Nota 0

- › Nenhum esforço para facilitar a comunicação aberta e transparente.
- › Grandes equipes de indivíduos executando tarefas de forma isolada.
- › O responsável técnico é apenas uma posição nominal.
- › Seções de código são totalmente propriedades de indivíduos.

### Referência – Nota 3

- › Todos os times funcionais são vistos como membros da equipe de entrega do produto e são representados em reuniões regulares de produtos voltados para a melhoria da entrega.
- › Equipes de operações prestam um serviço consultivo para as de entrega.

- › O compartilhamento de conhecimento entre os grupos multifuncionais não é uma prática comum e consistente.
- › Existe um plano para manter a continuidade da composição da equipe entre interações.

### Referência – Nota 5

- › A capacidade do negócio para enviar alterações é o fator que limita o que a equipe de produto está trabalhando.
- › As empresas têm dados ricos com base em padrões de uso e capacidade de lançar novos produtos para selecionar os usuários finais na produção (*canary release*, teste A/B etc.).
- › Todos os membros da equipe são qualificados em todas as áreas técnicas e há pouca especialização dependente de únicos indivíduos.

## Visibilidade

A capacidade de planejar e responder às solicitações de mudança do PO de uma forma que permita um ritmo consistente e previsível de trabalho que também é completamente transparente para todos.

### Referência – Nota 0

- › Não existe uma ferramenta ou forma fácil de verificar o que foi feito, por quem e porquê.

### Referência – Nota 3

- Todas as alterações no produto podem ser rastreadas através de
- › uma ferramenta comum, compartilhada entre as equipes, que inclui rastreamento de aprovação e resultados de testes.
  - › Falhas podem ser facilmente relacionados às mudanças individuais no início do ciclo de vida.

## Referência – Nota 5

- › Existe a transparência completa do que faz parte de cada lançamento.
- › O PO é capaz de determinar quando uma versão vai para produção e não é mais dependente da capacidade da equipe para entregar.
- › Evidências de controle e as decisões tomadas podem ser geradas por meio de conjuntos de ferramentas automatizadas usadas pela equipe de produto.

---

1 Disponível em: <<https://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: 17 mar. 2020.

2 Womack, James P.; Daniel T. Jones; Daniel Roos. *The machine that changed the world*. EUA: Free Press, 2007.

Ohno, Taiichi. *Sistema Toyota de produção*. Porto Alegre: Bookman, 1997.

Womack, James P.; Jones, Daniel T. *Lean Thinking*. EUA: Free Press, 2003.

3 Blank, Steve G. *The four steps to the epiphany*, 5<sup>a</sup> ed. Nova York: K & S Ranch, 2013.

4 Ries, Eric. *A startup enxuta*. Rio de Janeiro: Editora Sextante, 2019.

5 HUMBLE, Jez; FARLEY, David. *Entrega contínua*. Porto Alegre: Bookman, 2014.

6 Adaptação do modelo de Jeff Gothelf.

GOTHELF, Jeff; SEIDEN, Josh. *Lean UX*. Sebastopol, EUA: O'Reilly Media, 2013.

- 7 O arquivo do canvas MVP para impressão está disponível em: <[www.caroli.org/o-canvas-mvp](http://www.caroli.org/o-canvas-mvp)>.
- 8 Caroli, Paulo. *Lean Inception*: Porto Alegre: Editora Caroli, 2018.
- 9 CAGAN, Marty. *Product Market fit*, 2014. Disponível em: <<https://svpg.com/product-market-fit/>>. Acesso em: abr. 2021.
- 10 SCHNEIDER, Jakob; STICKDORN, Marc. *This is Service Design Thinking : basics, tools, cases*. Nova Jersey: Wiley, 2012.
- 11 Anderson, David J. *Kanban*. Blue Hole Press Inc, 2013.
- 12 The Scrum Guide (2018). Disponível em: <<https://www.scrumguides.org/scrum-guide.html>>. Acesso em: mar. 2020.
- 13 Wake, William C. ; Extreme Programming Explored; Boston, EUA: Addison-Wesley Professional, 2001.
- 14 Cohn, Mike; User stories applied. Boston, EUA: Addison-Wesley Professional, 2004.
- 15 Disponível em: <<https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScalini.pdf>>. Acesso em: mar. 2020.
- 16 Imagem original disponível em: <<http://kasburger.blogspot.com.br/2017/01/agile-tribe-leader-is-spider-in-web-of.html>>. Acesso em: mar. 2020.
- 17 Yarow, Jay. This Is The Internal Grading System Google Uses For Its Employees – And You Should Use It Too. Disponível em: <<http://www.businessinsider.com/googles-ranking-system-okr-2014-1>>. Acesso em: mar. 2020.
- 18 Castro, Felipe. The Silicon Valley way of setting goals: an introduction to OKR. Disponível em: <<https://medium.com/the-alignment-shop/introduction-to-okr-9912085830f0>>. Acesso em: mar. 2020.
- 19 Round-Robin na Wikipedia. Disponível em: <<https://pt.wikipedia.org/wiki/Round-robin>>. Acesso em: mar. 2020.
- 20 Retrospectivas divertidas é um livro de Paulo Caroli e Tainã Caetano, publicado pela Editora Caroli em versão digital e com versão impressa prevista para 2020. Disponível em: <[www.caroli.org/livro/retrospectivas-divertidas-atividades-e-ideias-para-fazer-retrospectivas-agile-mais-envolvente/](http://www.caroli.org/livro/retrospectivas-divertidas-atividades-e-ideias-para-fazer-retrospectivas-agile-mais-envolvente/)> Acesso em: mar. 2020.
- Você também pode acessar o site disponível em: <[www.funretrospectives.com/](http://www.funretrospectives.com/)>. Acesso em: mar. 2020. (N.E.).

21WEINBERG, Gerald; NORMAN, Kerth; Project Retrospectives: . Boston, EUA: Addison-Wesley Professional, 2013.

22BECK, Kent. eXtreme Programming Explained. Boston, EUA: Addison-Wesley Professional, 1999.

23Tavares, Bruno; Caroli, Paulo. Como avaliar (e auxiliar) a sua equipe com Entrega Contínua. Disponível em: <<https://www.infoq.com/br/articles/como-avaliar-sua-equipe-com-entrega-continua>>. Acesso em: mar. 2020.

24FOWLER, Martin. Continuous Integration, 2006. Disponível em: <https://www.martinfowler.com/articles/continuousIntegration.html>. Acesso em: abr. 2021.



# O DIA A DIA DE UM TIME EM AÇÃO

## » PLANEJAMENTO DO TRABALHO DE CONSULTORIA ÁGIL

O trabalho do consultor ágil está organizado em seis semanas de trabalho com entregáveis específicos e cada uma delas tem um foco distinto. O intuito é que o consultor ágil trabalhe com a equipe somente por esse período e, ao final, apresente um relatório com recomendações e próximos passos.

A equipe foi escolhida como um exemplo dentre várias outras equipes da organização que trabalham de forma semelhante. Os principais objetivos desse trabalho são: 1. ajudar a equipe; e 2. compartilhar as recomendações para que elas possam ser aplicadas em mais equipes da organização.

### SEMANA 1: Lean Inception

Tarefas:

- › Facilitar a Lean Inception da equipe.
- › Alinhar sobre o MVP e o plano de entrega.
- › Conduzir reuniões com stakeholders.

Entregáveis:

- › Roadmap do produto (em um slide).
- › Plano de trabalho para a consultoria ágil.

## SEMANA 2: Normas da equipe

Tarefas:

- › Acompanhar as reuniões da equipe.
- › Alinhar sobre a forma como o trabalho será desenvolvido.

Entregáveis:

- › Apresentação em PowerPoint sobre a forma de trabalho da equipe.
- › *Print-out* visual sobre a forma de trabalho (documento para ser colocado na parede que demonstre as principais reuniões e cerimônias da equipe).

## SEMANA 3: Excelência técnica

Tarefas:

- › Acompanhar as reuniões sobre arquitetura.
- › Aplicar o *assessment* de DevOps.

Entregáveis:

- › Relatório com o resultado do *assessment*.
- › Documento simplificado sobre a arquitetura do produto da equipe.
- › Documento sobre a estratégia de teste e qualidade técnica.

## SEMANA 4: Foco na entrega

Tarefas:

- › Acompanhar as reuniões sobre o entregável (MVP).
- › Realizar análise de ferramentas e itens de trabalho usados pela equipe (interno da equipe).
- › Realizar análise sobre a forma de comunicação com stakeholders.

Entregáveis:

- › Apresentação sobre o acompanhamento do entregável para os stakeholders.
- › Apresentação sobre o acompanhamento do progresso do trabalho para a equipe.

## SEMANA 5: Consolidação

Tarefas:

- › Consolidar o material criado durante as Sprints (semanas).
- › Documentar dores, gargalos, dependências e situações percebidas neste projeto, sempre relacionadas ao mindset ágil.

Entregáveis:

- › Apresentação com o passo a passo de cada uma das semanas anteriores.
- › Apresentação sobre dores, gargalos, dependências e situações percebidas.

## SEMANA 6: Alinhamento com stakeholders

Tarefas:

- › Conduzir as reuniões com os principais stakeholders (presencial, no cliente).
- › Alinhar sobre a continuidade do trabalho.

Entregáveis:

- › Todos os materiais gerados até o momento, templates e artefatos usados pela equipe com relação ao trabalho.

## » ENTENDA AS SPRINTS NA LINHA DO TEMPO

O capítulo anterior apresentou as seis semanas planejadas para o trabalho do consultor ágil. Essa organização de trabalho influenciou a estrutura do livro e também a forma como vamos lhe contar essa história: Sprint a Sprint, na linha do tempo. Mas, antes de começar, você precisa compreender as Sprints na linha do tempo.



### Lean Inception – Semana 1

Na primeira semana o grupo participou do workshop Lean Inception. Como resultado, a equipe alinhou sobre o cronograma e os entregáveis para as semanas seguintes.

### Sprint Zero – Semana 2

Para times que usam a metodologia Scrum, é comum o termo “Sprint Zero”. Apesar de não começarmos a contar do zero (e sim a partir do um), times Scrum geralmente se referem à primeira Sprint como “zero”, sendo essa a Sprint dedicada às atividades de set-up, as quais não contemplam tarefas diretamente relacionadas a criação de funcionalidades do produto.

Independente de utilizar Scrum ou outra metodologia para o gerenciamento e acompanhamento da criação do produto, sugerimos utilizar esse conceito de dedicar um tempo inicial (às vezes somente a Sprint Zero, outras a Zero e mais algumas Sprints, como foi o que aconteceu com este time, que levou mais tempo do que a Sprint Zero) para realizar as tarefas iniciais preparatórias do ambiente de trabalho. Depois disso, seguem as Sprints com entregas de funcionalidades do produto.

## Sprints de uma semana – Semanas 3, 4, 5 e 6

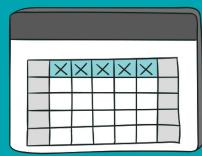
Essa equipe usou o framework Scrum e trabalhou em Sprints de uma semana. Então, no contexto deste livro, por favor considere que uma Sprint é igual a uma semana. Para evitar confusões sobre os números das semanas e das Sprints, coloco-as abaixo, conforme a nomenclatura usada pela equipe descrita no livro:

- » **Semana 1:** Lean Inception.
- » **Semana 2:** Sprint 0.
- » **Semana 3:** Sprint 1.
- » **Semana 4:** Sprint 2.
- » **Semana 5:** Sprint 3.

- Semana 6:** Sprint 4 (Neste momento, o Caroli, que era o » consultor ágil, parou de trabalhar com a equipe, mas seguiu em contato com a Mary e escrevendo este livro).
- » E assim seguiu até a Sprint 20, quando a Mary deixou de trabalhar com a equipe.

## Semanas e dias

Estruturamos as partes a seguir por semanas e, dentro delas, por capítulos que representam os dias. Por exemplo: Semana 2 / Dia 8 – Anotações de uma Daily Scrum). O livro começou no dia 1, que foi uma segunda-feira. Dia 6 e 7 foram, respectivamente, sábado e domingo e a semana 2 começou no dia 8. Cabe ressaltar que a equipe nunca trabalhou nos fins de semana.



# Semana 1

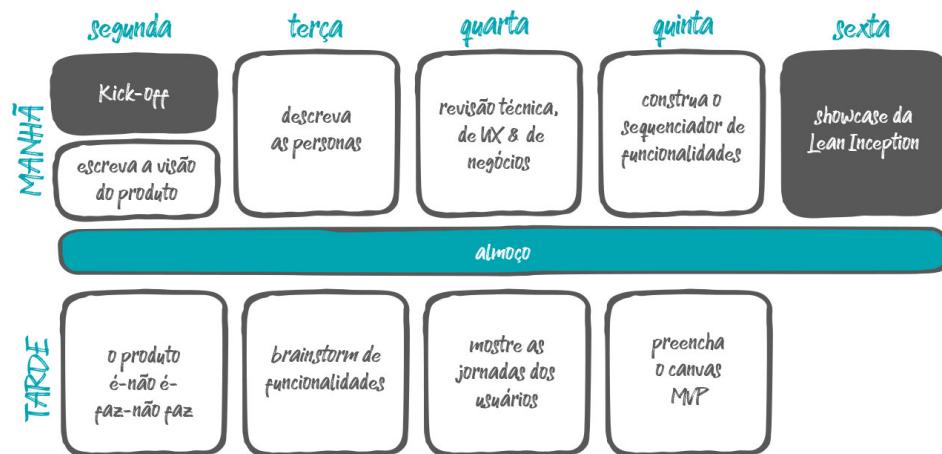
## » DIA 1 A 5 LEAN INCEPTION

Lean Inception é o nome dado ao workshop colaborativo para alinhar um grupo de pessoas sobre o produto mínimo viável a ser construído. Ele utiliza de técnicas de Design Thinking<sup>25</sup> com uma abordagem de Lean Startup.<sup>26</sup>

Antes de começar a fazer o produto, o grupo participa de um workshop colaborativo com uma sequência de atividades para alinhar e definir objetivos, estratégias e escopo do produto.

### Agenda da Lean Inception

Segue a agenda do time na semana da Lean Inception:



Para conhecer o passo a passo das atividades desse workshop colaborativo, recomendamos a leitura do livro *Lean Inception*.

## Resultado do workshop

Os principais resultados obtidos no workshop de Lean Inception foram:

- » A melhoria do conhecimento, relacionamento e comprometimento de todos os envolvidos.
- » Plano de liberação do produto via MVP.
- » Entendimento do MVP.

### Conhecimento, relacionamento e comprometimento

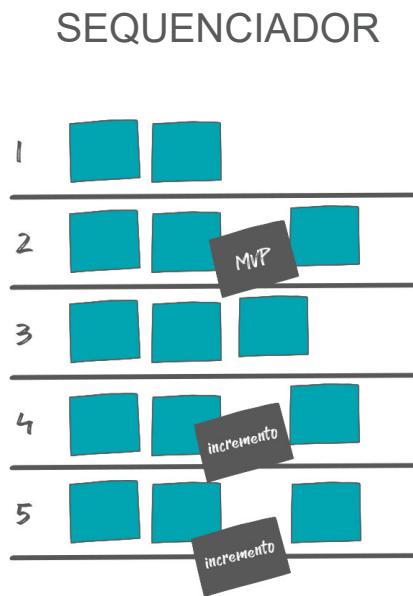
Esses são os três principais benefícios oriundos da Lean Inception. Quem participa da semana de inception adquire um alto nível de conhecimento sobre o produto a ser criado; também cria uma aproximação e um bom relacionamento com as outras pessoas envolvidas na criação do produto, além de assumir um alto nível de comprometimento com as datas e entregáveis acordados na inception.

Entretanto, a negativa também é verdadeira. Imagine alguém que não participou da semana da Lean Inception e, depois de algum tempo – dias ou semanas –, entra para o time que está trabalhando na criação do MVP do produto. Esse novo integrante do time não tem o mesmo nível de conhecimento, relacionamento e comprometimento que os participantes da semana da inception.

É muito importante ressaltar esse aspecto. Por mais que esse novo membro entenda rapidamente os detalhes do MVP a ser construído, ele vai precisar de ajuda e tempo para consolidar conhecimento, relacionamento e comprometimento da mesma forma que o restante do time.

### Plano de liberação do produto via MVP

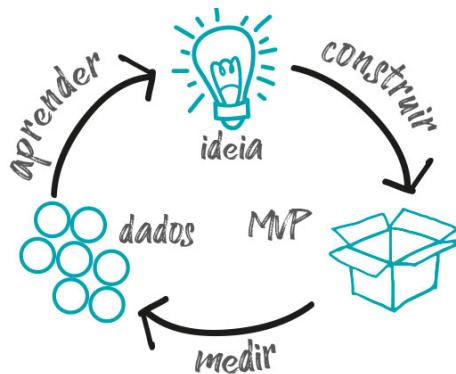
A imagem abaixo demonstra o sequenciador de funcionalidades, um resultado tangível da inception: o plano de liberação do produto via MVP.



### Produto mínimo viável (MVP)

O MVP é a versão mais simples de um produto que pode ser disponibilizada para o negócio. Ele determina quais são as funcionalidades mais essenciais para que se tenha o mínimo de produto funcional que possa agregar valor para o negócio (produto

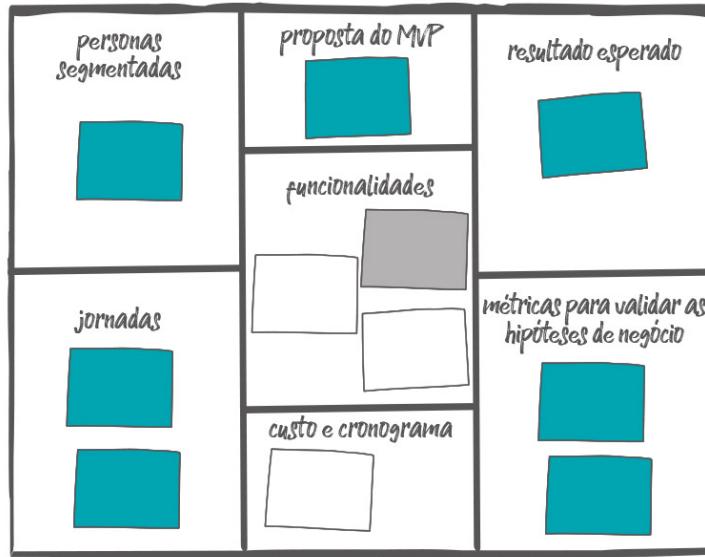
mínimo) e que possa ser efetivamente utilizado e validado pelo usuário final (produto viável).



Em fases iniciais de projetos mais tradicionais (ágiles ou cascata), o foco estava em gerar o plano de release do produto, muitas vezes bem completo. Na Lean Inception, o foco é no MVP, no mínimo viável do produto. Pensamos no resultado final, mas somente alinhamos e planejamos o MVP.

Em projetos mais tradicionais planejavamos e criávamos muitas funcionalidades que não eram usadas (e, com isso, muito tempo e dinheiro era desperdiçado!). Na Lean Inception, decidimos somente o MVP e os primeiros incrementos do produto. Depois, com o feedback dos usuários e do negócio, decidimos os próximos passos. O capítulo “Produto mínimo viável” que está no início do livro lhe fornecerá mais detalhes sobre esse conceito.

Um dos principais artefatos gerados pela Lean Inception é o Canvas MVP, que traz os detalhes do MVP. A seguir está um exemplo de Canvas MVP:



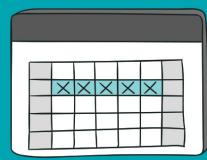
## Aprendizados do time

Ao passarmos cinco dias trabalhando juntos em uma sala, aprendemos lições importantes que poderíamos levar semanas, ou até meses, para identificar. Além dos principais resultados já citados, os importantes aprendizados desse time durante a Lean Inception foram:

- » O cliente acreditava ter certeza do *backlog* de funcionalidades e ter todos os detalhes bem definidos. Ao trazermos pessoas de fora do contexto dele, levantamos questões para as quais não tínhamos as respostas naquele momento e com isso foi possível nos organizar para reduzir a incerteza e risco de forma iterativa.
- » Inicialmente, nosso Product Owner acreditava que não seria necessário estar presente no dia a dia do time de desenvolvimento e que algumas pessoas desenvolvedoras poderiam ser alocadas parcialmente. Após conversarmos sobre o produto e como iríamos trabalhar juntos, foi possível alinhar

as expectativas em relação à presença do PO na rotina do time e à dedicação das pessoas desenvolvedoras.

- » Ao realizarmos a Lean Inception no ambiente do cliente, tivemos diversas interrupções e ausências de pessoas para participarem em outras reuniões. A grande lição aprendida aqui é priorizar um local neutro no qual as pessoas não serão interrompidas.



# Semana 2

## » DIA 8 ANOTAÇÕES DE UMA DAILY SCRUM

Seguem algumas anotações feitas durante uma Daily Scrum.

### TDD<sup>27</sup>

“Fulano está me catequizando sobre TDD. Já ganhou um fiel seguidor”

Um comentário simples que demonstra alguns aspectos bem legais:

- » O time prega pela excelência técnica, por isso o interesse em TDD.
- » Uma pessoa sabe mais sobre o negócio, outra sobre testes automatizados; ambas estão pareando, ambas aprendendo e se desenvolvendo enquanto trabalham.
- » As pessoas do time se conhecem há poucos dias, mas já se comunicam de forma amigável.

### Daily Scrum top down

O time está lendo e dando um update para cada tarefa em andamento. Mais tarde naquele dia, foi enviado um e-mail ao time sobre os estilos da Daily Scrum.

Seguindo a ordem dos cartões no Trello, o time responde às seguintes perguntas:

- » O que fizemos nessa tarefa ontem?
- » O que vamos fazer nessa tarefa hoje?
- » O que está impedindo o progresso dessa tarefa?

## Cadê o UX?

“O cara de UX está por aqui hoje?”, perguntou a desenvolvedora.

“Ele está envolvido em vários outros projetos, vou tentar localizá-lo”, respondeu o PO.

Essa conversa demonstra que o time não tem a pessoa de UX dedicada.

UX é essencial. O melhor cenário seria ter essa pessoa que representa os usuários finais disponível todo o tempo para a equipe. Aliás, essa pessoa deve ser parte da equipe (como no modelo Spotify: ela pertence ao *squad* deste produto e ao capítulo UX)!

Entretanto, como essa pessoa é compartilhada entre várias equipes, ela deveria estar disponível nas Daily Scrum e nas reuniões de planejamento e revisão.

O risco é que a equipe vai seguir trabalhando, independentemente de UX. Isso pode acarretar em duas situações ruins: 1. o trabalho não considera UX; ou 2. haverá retrabalho quando UX perceber um direcionamento equivocado.

## Tech huddle

“Terminei o caso de teste X; temos de decidir se isso é um teste unitário ou teste de UI”, disse uma desenvolvedora.

E isso levou a Daily Scrum a entrar numa boa conversa sobre os testes, onde deveriam ficar, qual nomenclatura usar, qual estrutura etc.

Porém a conversa ficou muito técnica e seria uma boa candidata para uma *tech huddle*, uma reunião mais técnica para conversar sobre algum assunto específico.

## » DIA 12 ÀS 16H46 O CONVITE PARA A RETROSPECTIVA

Abaixo está o convite enviado para a reunião de retrospectiva!



Time,

Vamos usar esse *board* remoto para a nossa retrospectiva:  
<link>.

Por favor trazer o seu laptop, pois vamos usar um para cada participante nessa reunião remota.

Segue a agenda da reunião:

1. **Contexto:** Vamos conversar sobre como lidaremos uns com os outros no dia a dia do nosso trabalho. Coletivamente, vamos definir os princípios da nossa equipe.
2. **Diretiva primária para formação de equipes:** “A cooperação é o ato de trabalhar com os outros e atuar juntos para realizar um trabalho. Time é uma parceria de pessoas únicas que trazem para fora o melhor de si, e que sabem que todos são maravilhosos como indivíduos, mas são ainda

melhores em conjunto. Reunir-se é um começo; manterem-se juntos é um progresso; trabalhar juntos é o sucesso”.

3. **Energizer:** Surpresa.
4. **Check-in:** Uma palavra.
5. **Atividade principal:** Definindo os princípios da equipe (vamos usar o link do FunRetrospectives).
6. **Filtering:** *Dot voting.*
7. **Check-out:** Que-quem-quando  
Abraços,  
Caroli e Mary pareando.

## » DIA 12 DAS 17H ÀS 18H RETROSPECTIVA SOBRE OS PRINCÍPIOS DO TIME

Como esta foi a primeira retrospectiva da equipe, usamos a reunião para definir as normas de trabalho. Foi feita uma atividade de *team building* que tem o propósito de definir essas regras: Definindo os princípios do time.

Segue a agenda da reunião, em sete passos, conforme o post “A 7-Step Agenda for Effective Retrospectives”.<sup>28</sup>

### 1. Contexto

Conversa sobre como lidaremos uns com os outros no dia a dia do nosso trabalho. Coletivamente, vamos definir os princípios da nossa equipe para cada uma dessas áreas:

- » Objetivo e agenda das reuniões;
- » Método preferido de comunicação;
- » Funções e disponibilidade;
- » Processos e metodologia de trabalho.

## 2. Diretiva primária

Lemos a diretiva primária de *team building*:

“A cooperação é o ato de trabalhar com os outros e atuar juntos para realizar um trabalho. Time é uma parceria de pessoas únicas que trazem para fora o melhor de si, e que sabem que todos são maravilhosos como indivíduos, mas são ainda melhores em conjunto. Reunir-se é um começo; manterem-se juntos é um progresso; trabalhar juntos é o sucesso.”

**Diretiva primária para formação de equipes.**

## 3. Energizer

Fizemos a atividade energizante “Ping Pong”. A atividade foi mais difícil dada a existência de participantes remoto, o que é um excelente ponto de reflexão sobre as dificuldades de comunicação encontradas em times remotos.

## 4. Check-in

Cada participante falou uma palavra sobre como estava se sentindo naquele momento, dado contexto da reunião. A atividade utilizada foi a “Uma palavra”.

## 5. Atividade principal

Utilizamos a atividade “Definindo os princípios do time” para ajudar a equipe a definir coletivamente e escrever os princípios.

Utilizamos o FunRetrospectives como ferramenta para a retrospectiva remota.

😊 Instruções: Definindo os princípios do time			
■ Objetivo e agenda das reuniões	■ Método preferido de comunicação	■ Funções e disponibilidade	■ Processos de metodologia de trabalho
Retrospectiva: melhoria contínua do time	Slack para comunicação do dia a dia	Podem ocorrer interrupções da equipe por problemas operacionais de outros produtos em produção	Sprints de uma semana
Daily: alinhamento/status do trabalho	Zoom com vídeo para reuniões	Mary disponível das 10h às 12h e das 13h às 18h	Entregas presenciais
Planning: planejamento da Sprint	E-mail para dúvidas e alinhamentos que não são urgente com pessoas externas ao time	Caroli disponível das 14h às 18h e das 22h às 02h	TDD
Agendar reuniões de uma maneira que não fique interrompendo o time a cada 1h/2h	...	...	Stakeholders presenciais nos showcases
...	...	...	Manter o Trello atualizado
...	...	...	...
<span style="color: #00AEEF;">●</span> Pessoas	<span style="color: #C8C8C8;">●</span> Processos	<span style="color: #000000;">●</span> Ferramentas	

## 6. Filtering

Cada participante teve seis votos no FunRetrospectives. Os cartões estavam ordenados pelos mais votados, então conversamos sobre eles.

## 7. Check-out

Foi enviado um e-mail com *action items* (que-quem-quando) e as anotações da retrospectiva para todos os participantes.

## » DIA 12 ÀS 18H32 E-MAIL COM ANOTAÇÕES DA RETROSPECTIVA



Time,

Segue abaixo algumas anotações da nossa retrospectiva. O link do nosso *board* remoto no FunRetrospectives continua disponível para quem quiser verificar todos os cartões.

### Reuniões:

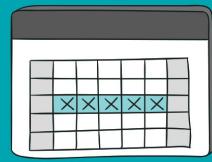
- › Sprint de uma semana;
- › Sprint Planning de trinta minutos: todos devem ler os cartões e escrever na descrição do mesmo;
- › Sprint Planning às segundas, das 11h às 11h30;
- › Sprint Review às sextas, das 16h às 16h30;
- › Retrospectiva a cada duas semanas: às sextas, das 16h30 às 17h30;
- › Daily Scrum todos dias às 11h;
- › Showcase na entrega do MVP e principais incrementos do produto. (agendado com os stakeholders).

## **Comunicação:**

- › Conversas do dia a dia via Slack;
- › Reuniões via Zoom;
- › Dar um joinha no Slack para mostrar que leu a mensagem;
- › Frequência de e-mail: muitos no time somente acessam ao e-mail uma vez por dia;
- › Manter um canal de comunicação aberto: TV e som ligado no Zoom o tempo todo e microfone no mudo. Quando alguém quiser falar, deve desativar o mudo e conversar com a equipe remota;
- › Manter as tarefas do Trello atualizadas com o que e quem está trabalhando.

Abraços,

Caroli e Mary pareando.

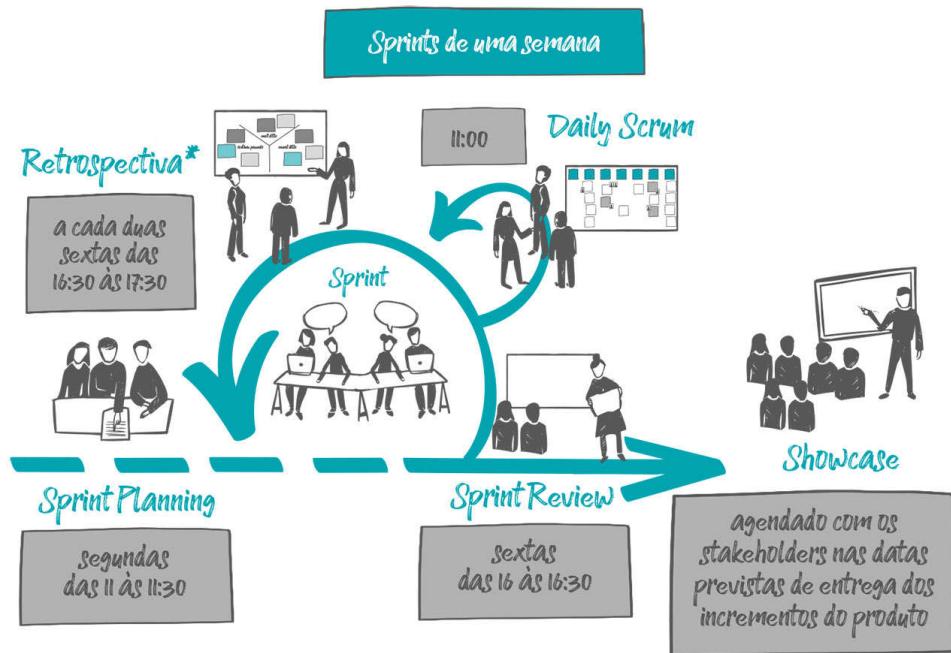


# Semana 3

## » DIA 15 AS REUNIÕES DA SPRINT

O time foi trabalhando em cada item da reunião de retrospectiva sobre os princípios. Durante a reunião, ficou claro que o time utiliza o framework Scrum (para saber mais, consulte o capítulo “Um overview do Scrum”, no início deste livro).

Abaixo está o horário das reuniões da Sprint acordadas pelo time. Essas reuniões foram agendadas pela Mary, que fez o papel de Scrum Master da equipe e esse desenho ficou colado na parede, visível para todos do time.



## Dia 15 – Problemas na Sprint Planning

A Sprint começou com uma reunião de planejamento que não foi muito efetiva: a reunião foi longa e a lista de tarefas no Trello do time não estava bem “aparada”.<sup>29</sup>

Dado o problema, escrevemos o seguinte e-mail para o time:



Time,

Olhando nosso Trello podemos perceber que as histórias/tarefas estão pouco definidas para serem planejadas. Isso é relativamente normal nas primeiras reuniões de planejamento.

Para melhorar isso, temos duas sugestões: 1. Criar um checklist para que uma história/tarefa seja considerada apta a entrar para o planejamento; e 2. Marcar uma reunião de refinamento (antigamente chamada de *grooming*), que deve anteceder a Sprint Planning.

Para a 2, sugerimos manter a eficiência das reuniões de planejamento e revisão: reuniões de 30 minutos. Nós vamos trabalhar no item 1.

Abraços,

Caroli e Mary pareando.

Um dia depois deste e-mail, trabalhamos no item 1, e o resultado foi o “Checklist de refinamento do trabalho”.

## » DIA 16 CHECKLIST DE REFINAMENTO DO TRABALHO



Time,

Segue em anexo o checklist que eu e a Mary criamos.

Vamos utilizá-lo no refinamento que faremos amanhã às 11h30.

Abraços,

Caroli.

### Checklist de refinamento do trabalho (definition of ready)

Cada item de trabalho (história, tarefa etc.) candidato a reunião de planejamento da Sprint deve passar por esse checklist. Uma resposta negativa indica que o item não é apto para entrar no planejamento.



Item pronto para a reunião de planejamento:

Eu tenho a informação necessária para trabalhar?

- Não
- Sim, favor especificar:

Eu sei o porquê deste item ?

- Não
- Sim, favor especificar:

Eu sei como mostrar o término deste trabalho?

- Não
- Sim, favor especificar:

Eu consigo relacionar este item a um épico ou uma funcionalidade esperada?

- Não
- Sim, favor especificar:

Esse item cabe em uma Sprint?

- Não, o item deverá ser quebrado em unidades menores
- Sim

## Contexto do time: Por que o checklist foi criado?

Após a reunião de planejamento do dia anterior, foi possível observar que essa reunião não seria produtiva da forma como estava acontecendo. O time ficava dando voltas em questões que

não tinham como ser respondidas naquela conversa. Ou então, por ser a primeira vez que estavam entrando em contato com um item de trabalho, não sabiam responder perguntas-chave sobre ele. Isso dificultava a quebra em itens menores ou até mesmo a priorização com o intuito de reduzir as pendências e entregar valor mais cedo.

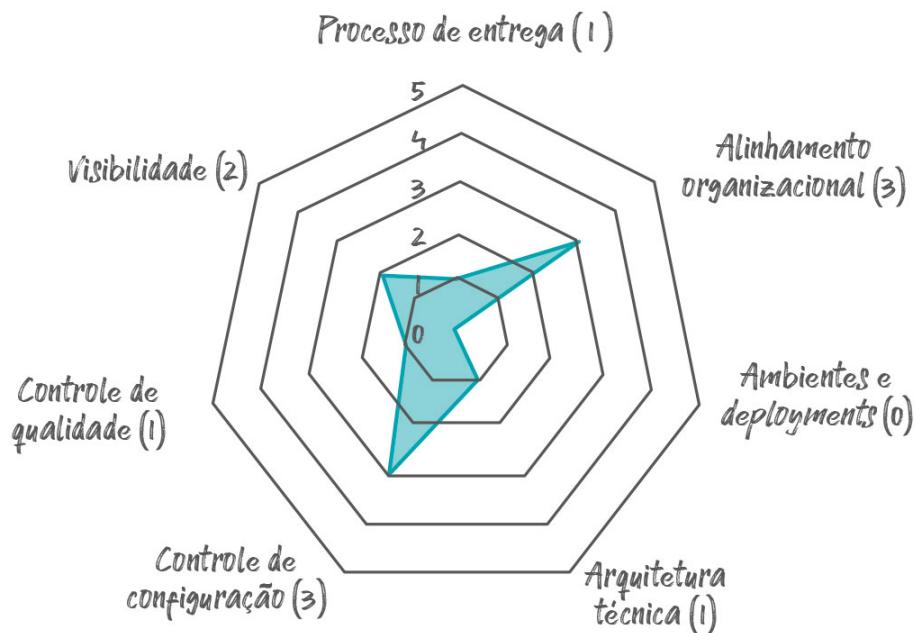
Outra variável que contribuiu para criação do checklist foi a dedicação do PO com a equipe. Como o PO também estava atuando em outros projetos, ele não tinha tempo para obter todas as informações necessárias para contextualizar o time antes da reunião de planejamento.

Nesse contexto, criar um checklist fazia bastante sentido, pois ele serviu como apoio para que o time se preparasse e garantisse que todos os itens no topo do *backlog* tivessem todos os detalhes necessários para não colocar a Sprint em risco. Dessa forma, passou a ser possível estabelecer e se comprometer com a meta da Sprint.

## » DIA 16 ASSESSMENT DE DEVOPS DO LEGADO

Após duas semanas, decidimos aplicar hoje o *assessment* de DevOps do legado,<sup>30</sup> do código que a equipe recebeu. Daqui a quatro semanas pretendemos aplicar novamente para verificar melhorias e decidir próximos passos.

Segue o resultado e o motivo para cada avaliação.



Resultado do assessment de DevOps – Legado

## Processo de entrega

Nota: 1

- › Existia um padrão de *branches* (estilo *gitflow*) e havia um teste manual, quando voltava um ok, os artefatos eram enviados por e-mail.

## Controle de qualidade

Nota: 1

- › Projeto Android não tinha nenhum tipo de teste; iOS tinha alguns poucos testes unitários.
- › Testes manuais (testes de mesa) eram executados a cada liberação do produto.
- › É o suporte que faz o tracking de bugs.

## Controle de configuração

Nota: 3

- › Tanto Android quanto iOS já utilizavam o gerenciador de dependências – Gradle e Cocoapots. Eles já usavam o Git.

## Ambientes e deployments

Nota: 0

- › Não temos um ambiente de teste completo (existe um QA01 para o *backend*, mas não há ambiente de teste mobile).
- › Não temos CI. Todo processo de *deployment* é manual (com vai e volta de APK via e-mail).

## Armazenamento de dados

Nota: não se aplica

- › Nosso produto é somente um middleware, sem base de dados.

## Arquitetura técnica

Nota: 1

- › O código é altamente acoplado. Não temos nada automatizado. É muito difícil fazer teste unitário do código. Mas já existe o gerenciamento de dependências baseado em versões.

## Alinhamento organizacional

Nota: 3

- › Existe um canal aberto de comunicação com os times de produto (que consomem a nossa SDK).

- › Existe um pouco de separação de conhecimento, por exemplo uma pessoa sabe Android, a outra sabe iOS.

## Visibilidade

Nota: 2

- › Temos o histórico de release no Git.
- › O time começou a usar o Trello para dar visibilidade das suas tarefas.

## » DIA 17 SUGESTÃO DE PARKING LOT DA DAILY SCRUM

Segue abaixo o e-mail no qual foi sugerida a inclusão do conceito de Parking Lot para a Daily Scrum da equipe.



Time, parabéns pela Daily Scrum. Pontual, informativa e com todos presentes. Estamos seguindo o estilo Daily Scrum Top Down.<sup>31</sup>

Uma recomendação: quando entrarmos em detalhe mais técnicos de uma tarefa/história (exemplo de hoje: testes unitários, de UI ou de integração. Como decidir?), cogitem colocá-la no Parking Lot <sup>32</sup> (ou mesmo em uma *tech huddle*) ao final da Daily Scrum. Dessa forma, quem não quiser ficar na conversa específica pode sair após o término da reunião.

Se todas pessoas da equipe participarem de todas as conversas, vamos reclamar que estamos sem tempo para trabalhar.

Abraços, Caroli e Mary pareando.

## » DIA 18 AGENDA DA RETROSPECTIVA NA SPRINT 1

A seguir está o e-mail enviado com a agenda da retrospectiva da primeira Sprint:



Time,

Segue a agenda para a nossa retrospectiva de amanhã.

1. **Contexto:** Estamos na primeira etapa, criando o *path to production*. Estamos usando Sprints de uma semana. Vamos olhar para essa última Sprint e conversar sobre o que deu certo e o que podemos melhorar.
2. **Diretiva primária para retrospectivas:** “Independentemente do que descobrimos, nós entendemos e realmente acreditamos que todos fizeram o melhor trabalho que poderiam, dado o que era conhecido na época, suas habilidades e competências, os recursos disponíveis, bem como a situação em questão”.
3. **Energizer:** Vamos repetir a atividade “Ping Pong”. Na última retrospectiva tivemos dificuldade de lidar com essa atividade,

especialmente por causa do fator remoto. Vamos ver se melhoramos desta vez.

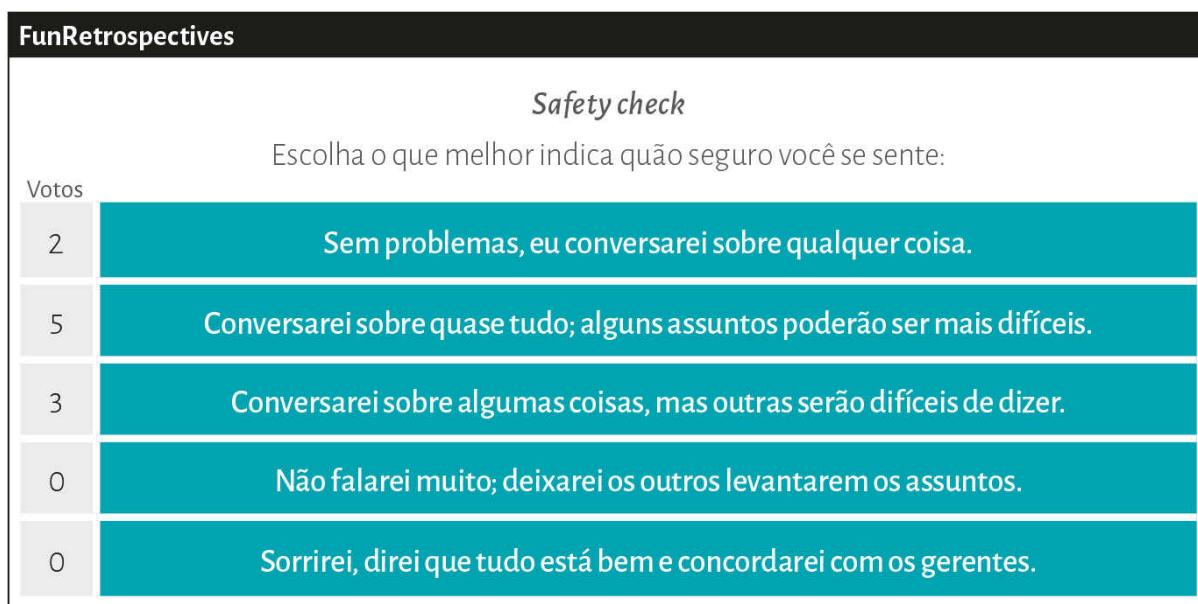
4. **Check-in:** Vamos usar a atividade “Verificação de segurança” no FunRetrospectives.
5. **Atividade principal:** Vamos usar a atividade “Pequena estrela-do-mar” no FunRetrospectives.
6. **Filtering:** Cada participante terá cinco votos no FunRetrospectives. Os cartões serão ordenados pelos mais votados e conversaremos sobre eles.
7. **Check-out:** Conversamos no final da reunião.

Abraços,

Caroli e Mary pareando.

## » DIA 19 RETROSPECTIVA

A retrospectiva começou conforme planejada. A atividade de check-in usada foi a “Verificação de segurança”. Segue abaixo o resultado dela:



Dado o resultado do check-in, não fazia sentido seguir com a agenda planejada para a retrospectiva, na qual a atividade principal seria a “Pequena estrela--do-mar”. Não poderíamos seguir com um contexto planejado, de conversar melhorias técnicas, dado que algumas pessoas na equipe não se sentiam seguras para falar.

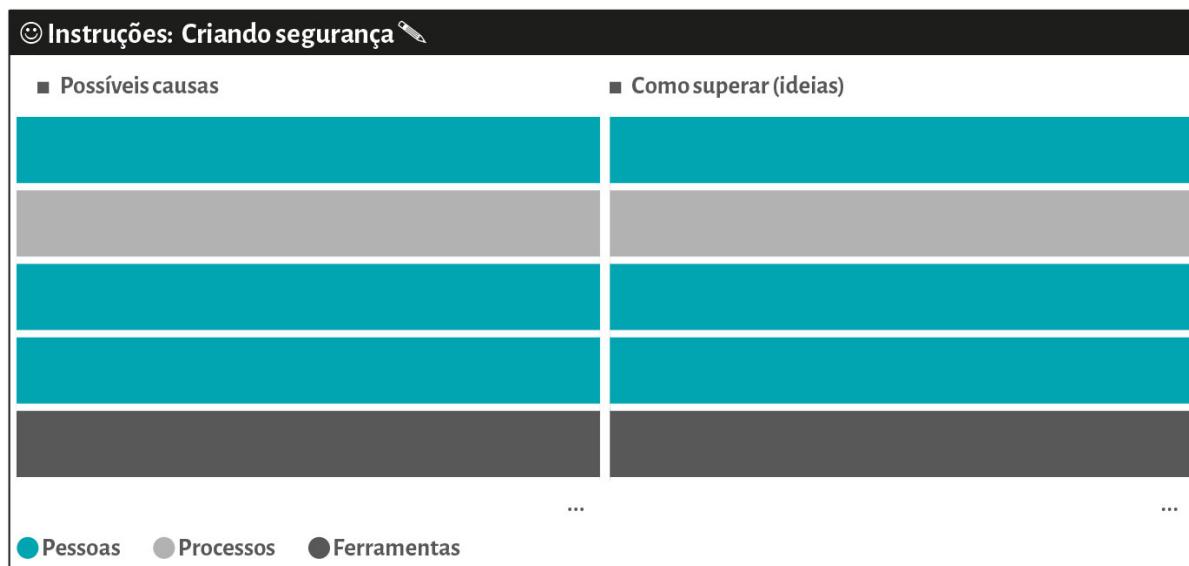
Então decidimos alterar a agenda da retrospectiva: trocamos o *board* remoto da “Pequena estrela-do-mar” para um *board* de

“Criando segurança”.

Segue a nova agenda para a retrospectiva, depois da atividade de check-in:

- » **Atividade principal:** Criando segurança.
- » **Filtering:** Cada participante deu cinco votos nos itens da coluna ideias para superar as possíveis causas da falta de segurança.
- » **Check-out:** Lemos e conversamos sobre os itens levantados e, por fim, todos votaram novamente em um novo *board* de “Verificação de segurança”.

Veja a seguir um exemplo de *board* de “Criando segurança”.



Veja também como ficou o novo *board* “Verificação de segurança” realizado ao final da retrospectiva, depois da conversa sobre “Criando segurança”.

*Safety check*

Escolha o que melhor indica quão seguro você se sente:

Votos	
5	Sem problemas, eu conversarei sobre qualquer coisa.
3	Conversarei sobre quase tudo; alguns assuntos poderão ser mais difíceis.
2	Conversarei sobre algumas coisas, mas outras serão difíceis de dizer.
0	Não falarei muito; deixarei os outros levantarem os assuntos.
0	Sorriréi, direi que tudo está bem e concordarei com os gerentes.

## Refletindo sobre esta retrospectiva

A atividade de check-in da retrospectiva foi muito essencial. Ela demonstrou claramente que o time precisava conversar não somente sobre como trabalhar, mas também sobre como se sentir mais seguro para poder expor ideias e opiniões.

Este é um acontecimento bastante comum em times em formação: sair trabalhando e priorizar conversas sobre o trabalho antes de se conhecer como um time.

Por isso, atividades de *team building* são fundamentais, principalmente quando as pessoas estão começando a trabalhar juntas.

## » DIA 19 E-MAIL DE ACTION ITEMS DA RETROSPECTIVA

Segue o e-mail enviado com os action items da retrospectiva da Sprint 1:



Time,

Muito obrigado pela participação na retrospectiva. Vocês levantaram pontos muito importantes para um excelente trabalho em equipe.

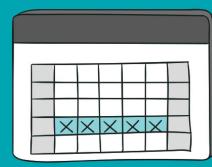
Esse estilo de retrospectiva não gera muito actions items, mas sim uma conscientização e respeito sobre que é importante para cada uma das pessoas do time.

Como *action items*, tenho somente esses dois:

Verificar o melhor dia para a reunião de planejamento; segunda é ruim para o UX: vamos levantar isso no Parking Lot ao final da próxima Daily Scrum que ele participar.

Marcar sessões curtas para as pessoas se apresentarem e se conhecerem: vou usar uma atividade com esse intuito na nossa próxima retrospectiva.

Abraços,  
Caroli.



# Semana 4

## » DIA 22 DAS 11 ÀS 11:30 SPRINT PLANNING

Durante a reunião de planejamento da Sprint, o time conversou sobre o estado atual do trabalho, os próximos itens de trabalho e o comprometimento para a Sprint que estava começando. Debateram também sobre esforço, tempo e comprometimento. Ademais, o time verificou o sequenciador de funcionalidade (resultado da Lean Inception) para relembrar do comprometimento com a data do MVP.

Um item de trabalho foi reescrito. Ele estava “um pouco grande para entrar na Sprint” (o item caiu na malha fina do Checklist de refinamento do trabalho).

O item foi reescrito em outros dois, cada um passou novamente pelo Checklist de refinamento, e foram adicionados ao *backlog* da Sprint.

## » DIA 23 E-MAIL “O QUE” VERSUS “COMO”



Time,

Hoje mais cedo eu estava na reunião de discussão sobre o fluxo de pagamentos e percebi algo que acho importante informar, especialmente para a participação de todos em futuras Lean Inceptions.

Durante a semana da Lean Inception, algumas vezes eu lembrava ao time que naquele momento estávamos interessados em entender **o que** devíamos fazer. E eu dizia ao grupo que, depois da semana da Lean Inception, durante as Sprints, quando fosse o momento, iríamos detalhar o **como** iríamos fazer.

Lembram daquela propaganda: “Eu sou você amanhã”? Quem lembrar está revelando a idade!

Hoje foi um exemplo de uma sessão detalhando o **como** daquilo que ficamos somente no **o que** durante a Lean inception.

Abraços,

Caroli.

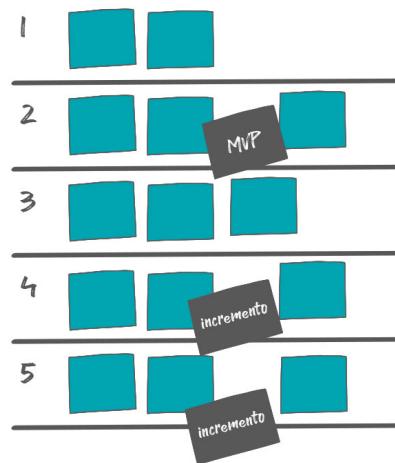
## » DIA 24 AS HISTÓRIAS DAS PRIMEIRAS FUNCIONALIDADES



Time,

Lembram do sequenciador de funcionalidades da Lean Inception?

SEQUENCIADOR



Então, seguindo o nosso planejamento, esta semana começamos com as funcionalidades funcionais na primeira

linha do sequenciador (esquisito escrever dessa forma, não é mesmo?).

Ainda estamos terminando algumas tarefas da primeira fase, do *path to production*, as tarefas não funcionais, o trabalho de melhoria/adaptação/criação da infraestrutura necessária para iniciarmos com o código funcional.

Verificando o planejamento dos épicos e tarefas iniciais, olhando o nosso *board* e ouvindo as conversas sobre o CI, parece que estamos bem, ou seja: sucesso na primeira fase! Sucesso em preparar o terreno para começar a trabalhar nas funcionalidades do MVP.

Agora precisamos refinar as funcionalidades seguintes. São essas aqui:

- › Revisar mensagens de erro (Android).
- › Revisar mensagens de erro (iOS).

Relembro também que já temos um checklist para nos ajudar a verificar se o item de trabalho (história ou tarefa, como preferirem) está apto para a reunião de planejamento.

Na próxima semana teremos uma agenda um pouco diferente. Faremos a reunião de refinamento na segunda-feira e a reunião de planejamento na terça-feira.

Porém, se já quiserem antecipar esse trabalho, como dizem aqui em Porto Alegre, prendam o grito se precisar de ajuda com essas histórias (prender o grito = pode contar comigo)!

Abraços,

Caroli e Mary pareando.

## » DIA 26 ATIVIDADE DE TEAM BUILDING

Segue o e-mail com a agenda para a atividade de *team building*, que tem o objetivo de fazer o time se conhecer melhor.



Time,

Segue a agenda para a atividade de *team building* que faremos daqui a cinco minutos. Esse é um dos *action items* da nossa última retrospectiva.

1. **Contexto:** Queremos nos conhecer melhor.
2. **Diretiva primária para formação de equipe:** “A cooperação é o ato de trabalhar com os outros e atuar juntos para realizar um trabalho. Time é uma parceria de pessoas únicas que trazem para fora o melhor de si, e que sabem que todos são maravilhosos, como indivíduos, mas são ainda melhores em conjunto. Reunir-se é um começo; manterem-se juntos é um progresso; trabalhar juntos é o sucesso.”
3. **Atividade principal:** Vamos usar a atividade *The roles we play* acessando o link do *board* no FunRetrospectives.

Abraços,  
Caroli e Mary pareando.

Nessa atividade, cada pessoa escreveu os seus vários papéis (antes, durante, depois do trabalho e no final de semana) no *board* do FunRetrospectives. Depois lemos todos os cartões, um a um.

Segue abaixo a reprodução do *board* do FunRetrospectives ao final da atividade.

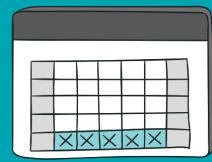
☺ Instruções: ✎

■ Antes do trampo	■ Hora do trampo	■ Depois do trampo	■ Fim de semana
[VT] Pai	[FD] Desenvolvedor	[LR] Jogador de video game	[JB] Churrasqueiro
[ER] Mago	[JB] Product Owner	[YN] Maratonista de séries	[CF] Cachaceiro
[ER] Marido	[VA] Desenvolvedora	[MP] Leitora	[ER] Marido
[FD] Pai	[MP] Scrum Master	[JB] Pai	[FD] Estudante
[JB] Motorista	[YN] Desenvolvedora	...	[MP] Cinéfila
...	[VT] Desenvolvedor		[VT] Pai e marido
	[ER] Desenvolvedor		...
	[CF] UX		
	[GF] Desenvolvedor		
	[PC] Consultor Ágil		
	[LR] Desenvolvedor		

● Pessoas

● Processos

● Ferramentas



# Semana 5

## » DIA 29 MELHORIA PARA O REFINAMENTO: FUNCIONALIDADES E HISTÓRIAS

Segue o e-mail enviado após a reunião de refinamento:



Time,

Tivemos nossa reunião de refinamento hoje, mas as histórias candidatas ao planejamento de amanhã ainda precisam ser melhor descritas. E nós não iríamos escrever este e-mail sem oferecer uma ajuda com isso!

Favor verificar o documento sobre Funcionalidades e histórias do usuário,<sup>33</sup> em anexo neste e-mail. Vamos usar somente uma das três funcionalidades da próxima etapa, como exemplo:

**Funcionalidade:** Revisar mensagens de erro (Android).

**Histórias** (ou melhor, candidatas a histórias):

- › Como um usuário Android, eu quero ler a mensagem de erro (com URL) e clicar no botão “Ok” para fechar a janela – \*favor confirmar o comportamento do botão.

- › Como um usuário Android, eu quero ler a mensagem de erro (com URL) e clicar no “Fale conosco” que leva ao canal correto (a URL) – \*quantos cenários existem?
- › Como um usuário Android, eu quero ler a mensagem de erro sem quebra de linha no título/label (atualmente existe uma quebra no título).
- › Como um usuário Android, eu quero ler a mensagem de erro com as cores e fontes padrões do nosso aplicativo – \*verificar com UX.

Marcamos com um \* as questões que devem ser resolvidas pelo nosso UX.

Abraços,

Caroli e Mary pareando.

## » DIA 29 O PAPEL DO PO E A CAPACIDADE DO TIME

Segue um rascunho de um e-mail que não foi enviado. O assunto poderia ser delicado e decidimos ter uma conversa pessoalmente, ao invés de enviar o e-mail.



Time,

Temos um problema. Porém não vou falar da solução neste e-mail, somente do problema e dos ajustes para as datas e entregáveis.

Quando fizemos a atividade para calcular as datas do MVP na Lean Inception, consideramos que as pessoas desenvolvedoras do time estariam envolvidas principalmente na tarefa de desenvolvimento.

Assumimos que tanto o direcionamento estratégico do produto quanto o detalhamento das funcionalidades em história, seriam um trabalho primariamente realizado pelo cargo do PO (não estou falando de quem está nominalmente como PO desta equipe, mas do cargo de PO em si, essa é uma conversa longa e não somente desta equipe).

Hoje, após a reunião de refinamento, parei para refletir e percebi que as decisões sobre o produto e o detalhamento do mesmo está primariamente sendo realizado pelas desenvolvedoras. Um exemplo disso é a criação de *spikes* para tomar decisões sobre o produto. Dado que esse é o nosso cenário, precisamos rever as datas de acordo com a capacidade da equipe.

Minha sugestão: refazer o cálculo da capacidade de desenvolvimento e revalidar expectativas de datas dos incrementos do produto.

Abraços,

Caroli.

## » DIA 30 REUNIÃO DE REFINAMENTO

Alguns membros da equipe se encontraram para refinar as histórias, a fim de prepará-las para a próxima reunião de planejamento.

O nome dessa reunião variava entre reunião de refinamento ou reunião de pré-planejamento. Independentemente do nome, ela tinha o objetivo de detalhar as funcionalidades em histórias. Outro ponto importante é que a participação de todos da equipe Scrum não era obrigatória (ao contrário da reunião de planejamento que somente acontecia com todos presentes: membros da equipe, PO e Scrum Master).

A seguir está uma representação da reunião na qual a equipe está detalhando uma funcionalidade em história, usando um cartão no Trello, com as histórias candidatas ao *backlog* da Sprint seguinte.

MVP	ONDA	FUNCIONALIDADES	DATA	STATUS
-	0	Infra, CI e testes	21/05	Com
1	1		28/05	Em A
2	2		05/06	Não
3	3		12/06	Não
4	4		19/06	Não
5	5		26/06	Não
6	6		03/07	Não
7	7		10/07	Não
8	8		17/07	Não

Preparar SDK para receber novo contrato da tela de erro do Backend

na lista Sprint Backlog

Etiquetas: Android

**Descrição** [Editar](#)

\*\*DESCRÍÇÃO\*\*  
\*COMO UMA\* pessoa desenvolvedora  
\*QUERO\* ajustar o contrato das telas de erro  
\*PARA\* preencher a tela de erro com os dados vindo do backend

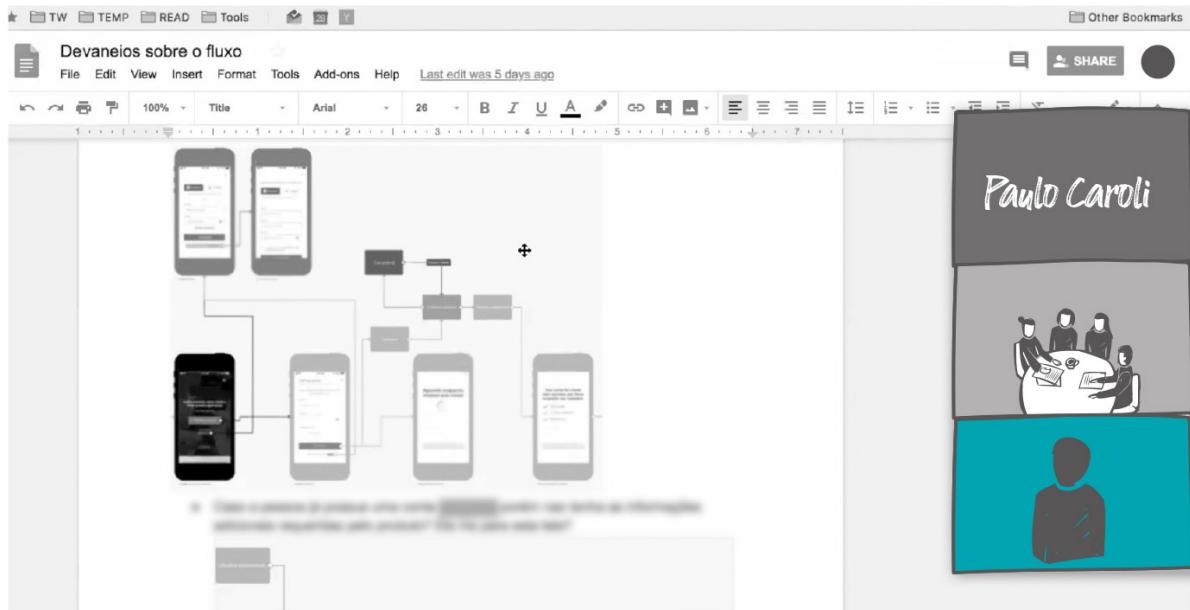
\*\*CRITÉRIOS DE ACEITE\*\*  
\*DADO QUE\* (cenário inicial)  
\*QUANDO\* (ação realizada)  
\*ENTÃO\* (estado esperado)

**Anexos**

**Checklist**

1—Eu tenho a informação necessária para trabalhar  
2—Eu sei o porquê deste item  
3—Eu sei como mostrar o término deste trabalho  
4—Eu consigo relacionar este item a um ônus / funcionalidade

Já na imagem a seguir, observe a equipe conversando sobre o detalhamento – tela a tela – do fluxo da aplicação. Isso gerou muita conversa para verificar histórias e, especialmente, os critérios de aceite (e cenários de teste).



Uma conversa mais detalhada é muito comum no momento de refinamento: detalhes técnicos, do fluxo, sobre cenários de teste, da arquitetura, das tarefas etc. Esse é o momento de detalhar o trabalho a ser realizado. Lembrando que, na reunião seguinte, a de planejamento, a equipe vai se comprometer com o trabalho para a Sprint seguinte: o *backlog* da mesma.

## O template do cartão de história no Trello

Cada cartão de história no Trello começava com o seguinte texto que era reescrito pela equipe:

### Descrição da história

- » **Como um(a) (perfil do usuário/persona)**
- » **Eu quero (ação a ser realizada/funcionalidade específica do produto)**
- » **Para que (valor alcançado)**

## Critério de aceite

- » Dado que (cenário inicial)
- » Quando (ação realizada)
- » Então (estado esperado)

Depois de escrever a descrição, os critérios de aceite e o detalhamento da história, a equipe respondia às perguntas do Checklist de refinamento do trabalho.

Separar o endpoint que lista os produtos disponíveis do endpoint que verifica se a pessoa pode comprar o produto

na lista [Sprint Backlog](#)

**Descrição** [Editar](#)

**ADICIONAR**

Membros

Etiquetas

**ETIQUETAS**

Buscar etiquetas...

Android

Bug

Infra

Blocked

Backend

iOS

Tech Dev

**CHECKLIST DE REFINAMENTO**

Checklist

Ocultar itens concluídos Excluir...

100% [progress bar]

1—Eu tenho a informação necessária para trabalhar

2—Eu sei o porquê deste item

3—Eu sei como mostrar o término deste trabalho

4—Eu consigo relacionar este item a um épico / funcionalidade esperada

5—Esse item cabe em uma Sprint

**CRIAR UMA NOVA ETIQUETA**

**ATIVAR MODO COMPATÍVEL COM DALTONISMO**

## » DIA 31 O PO FAZ (MUITA) FALTA

O PO faz (muita) falta. Sem ele o time é como um barco à deriva.

Essa frase refere-se ao dia no qual uma reunião foi cancelada pois o PO estava ausente.

A Lean Inception ajuda com o alinhamento sobre os primeiros passos sobre o produto em construção. As pessoas envolvidas na criação do produto (geralmente chamadas de *squad* ou de membros ativos) decidem o MVP e suas funcionalidades.

Tudo lindo e perfeito para começarem a trabalhar, mas isso não é suficiente!

Não, não é apenas a Lean Inception que vai gerar um produto de sucesso. Não foi isso que vimos na maioria dos lugares em que trabalhamos. Mas antes de começar a trabalhar nas funcionalidades do MVP, o time precisa descrever esse trabalho em mais detalhes. É então que começa o verdadeiro Scrum.

Scrum é um termo que vem do esporte rugby. Scrum, uma abreviação de *Scrummage*, que descreve o momento em que os jogadores estão cabeça a cabeça, entrelaçados, tentando ganhar a posse da bola.

É isso que acontece com os times de empresas inovadoras que têm o desafio de criar produtos digitais transformadores. Cabeças pensantes, juntas, todas as pessoas time entrelaçadas para se manter no jogo.

Esse entrelace e entrosamento é essencial para o sucesso da criação do produto. Mesmo tendo começado com uma Lean Inception, e tendo o Norte e o planejamento do MVP bem definido.

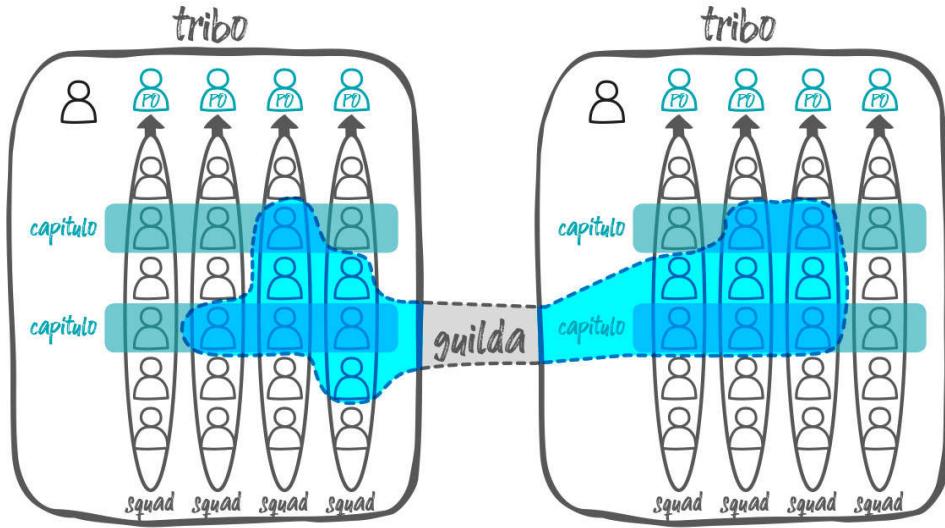
Ao detalhar as funcionalidades do MVP em pedaços menores, mais mastigáveis, mais testáveis, mais factíveis de serem trabalhados; daí sim, a “porca torce o rabo”.

Escolhemos essa expressão pois lembramos daquela piada na qual o porco e a galinha conversam sobre abrir um restaurante que serve café da manhã, chamado “Ovos com Bacon”. A galinha está envolvida (contribui com os ovos), enquanto o porco está comprometido (a sua pele – o bacon – está no negócio).

A questão é que o *squad*, o time de produto, e cada um daqueles comprometidos no dia a dia das reuniões da Sprint representam o porco! E dentro desses comprometidos, um tem papel essencial: o PO.

O Product Owner, a pessoa responsável pelo direcionamento estratégico do produto, é quem fatia, descarta e prioriza o trabalho do dia a dia das reuniões:<sup>34</sup> Daily Scrum, Sprint Planning, Sprint Review, retrospectiva e reuniões de refinamento do trabalho.

Veja nessa imagem como o Spotify (exemplo muito utilizado de boa organização das equipes) se organiza em times. Note o relacionamento entre o time e o PO: não existe um sem o outro.



Um time sem PO é um barco à deriva. Talvez até chegue em algum lugar, mas ninguém garante que esse é o lugar que deveria alcançar.

Se a sua empresa for como um pequeno barco, talvez isso não seja problema: siga navegando e, de tempos em tempos, verifique aonde quer chegar, alinhando o direcionamento dele.

Mas esse não é o caso para empresas grandes, com vários barcos, de vários tamanhos, mas com um destino em comum. Alguém, ou melhor, “alguéns” precisam alinhar todos esses barcos em relação ao direcionamento da empresa.

Não pense em parar todos os tripulantes para decidirem e depois voltarem para seus barcos para navegar. Isso somente funciona em empresas pequenas, com poucos barcos.

Promova o encontro dos POs para alinharem o direcionamento estratégico entre todos os times. Isso provavelmente não é somente uma vez, mas sim encontros frequentes. Afinal de contas, todos continuam a navegar, mas as águas, os ventos e o clima vão mudar

e será necessário repensar o caminho e realinhar os planos de acordo com os acontecimentos mais recentes.

Mas não deixe as equipes sem o Product Owner por muito tempo. Ele pode participar de outras reuniões, mas é o comandante do barco! Se o PO se ausentar por muito tempo, o barco pode seguir sem direção.

Para ajudar com isso, o Scrum tem reuniões bem definidas, e todas requerem a presença do PO. Ele participa ativamente das reuniões da Sprint. Por isso consegue guiar e ajudar a equipe a navegar de acordo com o interesse maior: o direcionamento estratégico da empresa e de todos os times.

## » DIA 33 PRECISAMOS DE UMA RETROSPECTIVA

Até semana retrasada, o time estava envolvido na primeira fase, fazendo todas as tarefas do *path to production*, criando o pipeline de entrega contínua, preparando os ambientes e ferramental necessários para produzir o código funcional.

Nessa semana o time começou a trabalhar nas funcionalidades do MVP. Entretanto, o início foi conturbado:

- » As histórias não estavam bem escritas e compreendidas (reunião de refinamento ineficiente);
- » A reunião de planejamento levou mais de duas horas (o desejado era meia hora);
- » A equipe estava trabalhando em vários *spikes*, já que houveram mudanças nas decisões do trabalho a ser realizado para as primeiras funcionalidades acordadas na Lean Inception;
- » A reunião de retrospectiva da última Sprint foi cancelada.

Tudo isso aconteceu em uma semana. Times ágeis trabalham com um mindset de inspeção e adaptação, o que é abordado sistematicamente na reunião de retrospectiva.

Como essa reunião havia sido cancelada, foi enviado um e-mail com o objetivo de reagendá-la e evitar o cancelamento de retrospectivas futuras.



Time,

Preferem fazer uma retrospectiva hoje ou na segunda-feira?

“Todo time deve fazer uma retrospectiva por semana, a menos que esteja sem tempo. Nesse caso, deve fazer duas”.

Eu sempre repito essa frase. E tento fazer uma retrospectiva por semana em todos times que trabalho. Nunca me arrependi de fazer isso frequentemente, mas me arrependi do oposto: de não fazer retrospectivas com uma boa frequência. Para um time como o nosso, com um contexto complexo e entregas curtas, no estilo MVP, uma por semana é o suficiente.

Semana passada cancelamos uma retrospectiva e me parece que estamos com a sensação de que estamos sem tempo dado a existência de tantas reuniões (ineficientes) e trabalho a fazer. Isso me dá o sinal de que precisamos priorizar a conversa sobre como melhorar a forma que trabalhamos: precisamos de uma retrospectiva.

Abraços,

Caroli.

Após todos responderem o e-mail, marcamos a retrospectiva para segunda-feira.

## Comunicação direta

No e-mail enviado tem uma pergunta direta: Preferem fazer uma retrospectiva hoje ou na segunda-feira?

Note que esse e-mail somente dá duas opções de resposta: hoje ou segunda.

É importante ressaltar que as pessoas recebem muitos e-mails, provavelmente vão ler essa mensagem em dez segundos e vão precisar pular para uma próxima tarefa. Simplifique a vida delas. Seja direta e intencional.

Não comece um diálogo por e-mail com: “oi, a retrospectiva é muito importante, vocês concordam?”, ou “podemos remarcar a reunião de retrospectiva? Que dia fica melhor para você?”.

Vá direto ao ponto. Se você tem um CTA (*Call To Action*), seja objetiva e intencional. Esse e-mail e o resultado dele exemplifica isso!

## » DIA 33 CONVERSA COM AS PESSOAS DESENVOLVEDORAS

Estávamos com a sensação de que a motivação das pessoas desenvolvedoras tinha caído. E precisávamos de algum artefato para levar essa percepção para o PO e para os stakeholders do projeto: uma retrospectiva.

Não uma retrospectiva para toda a equipe, mas somente para as desenvolvedoras. O contexto: um bate-papo sobre como você está se sentindo nesse projeto, em relação ao seu trabalho.

Antes de compartilhar a atividade e o resultado gerado, é importante ressaltar que isso não substitui, de forma alguma, a retrospectiva da equipe. Isso foi somente uma atividade “extra” para validar e gerar insumos para outras conversas.

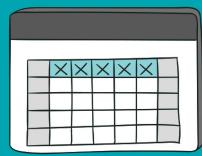
A atividade usada foi “Penhascos e vales” que utiliza uma linguagem visual simples para compartilhar as visões individuais de cada pessoa na linha do tempo antes da Lean Inception, durante ela e nas etapas seguintes. Segue abaixo o resultado da atividade.



Resultado da atividade “Penhascos e vales”

A imagem é uma representação da atividade realizada no Google Drawing, ferramenta utilizada para efetuar a atividade de forma remota.

Cada linha representa como uma pessoa se sentia ao longo do tempo. Ficou mais fácil de demonstrar que as desenvolvedoras gostaram da Lean Inception, estavam “ok” com a primeira fase, mas que a motivação vinha caindo desde que o trabalho funcional começou na segunda etapa.



# Semana 6

## » DIA 36 AGENDA DA RETROSPECTIVA



Time,

Segue a agenda para a nossa retrospectiva de hoje, que acontecerá logo após a Daily, provavelmente das 11h30 às 12h30.

1. **Contexto:** Estamos começando a quarta Sprint. Vamos conversar sobre nosso entregável – o MVP – e como estamos trabalhando.
2. **Diretiva primária para formação de equipes:** “Independentemente do que descobrimos, nós entendemos e realmente acreditamos que todos fizeram o melhor trabalho que poderiam, dado o que era conhecido na época, suas habilidades e competências, os recursos disponíveis, bem como a situação em questão.”
3. **Energizer:** Surpresa.
4. **Check-in:** Uma palavra.
5. **Atividade principal:** Pequena estrela-do-mar.
6. **Filtering:** *Dot voting* (5 votos por participante).

7. **Check-out:** Vamos reler os *action items*, e escrever uma palavra antes de sair: atividade “Uma palavra antes de ir”.

## » DIA 36 RETROSPECTIVA

Seguem mais de detalhes sobre a execução e também as imagens da reunião de retrospectiva.

### Check-in – Uma palavra

Veja na imagem o resultado da atividade de check-in “Uma palavra”. Cada pessoa falou e escreveu uma palavra que reflete como se sentia naquele momento (início da reunião) para o contexto apresentado.

FunRetrospectives
Uma palavra
sereno
confiante
de boa
motivado
confusa
tranquila
tranquilo
confusa
ansioso

## Atividade principal – Pequena estrela-do-mar

Na atividade principal, foi solicitado ao time que escrevessem cartões para as três áreas do respectivo *board* remoto do FunRetrospectives:

- » **Continue (keep doing):** algo que o time está fazendo bem e reconhece o valor.
- » **Mais (more of):** algo que o time está fazendo e acredita que trará mais valor se for feito mais vezes.
- » **Menos/Pare (less/stop):** algo que o time está fazendo e possui algum valor, mas é preferível reduzir aos poucos ou de uma só vez.

🕒 Instruções		
■ Continue	■ Mais	■ Menos/Pare
Refatorar o código	Escrever stories com o PO	Focar muito no futuro e deixar as funcionalidades que precisam ser feitas agora de lado
Testes automatizados	Revisar objetivo da Sprint durante dailies	Reuniões sem a presença de quem define (não efetivas)
Refinamento	Presença física de pessoas da TW no cliente e do cliente na TW	Planning sem PO e UX
Retrospectivas	Reuniões presenciais	Spikes
...	Detalhar <i>user stories</i>	Pegar uma tarefa para fazer que não foi priorizada na Planning (fora do objetivo da Sprint)
	Envolver pessoas da TW em reuniões que acontecem no cliente sobre o produto	...
		...

● Pessoas   
 ● Processos   
 ● Ferramentas

Itens mais votados na atividade "Pequena estrela-do-mar"

## Filtering – Dot voting

Os cartões foram ordenados pelo número de votos. Depois do tempo para escreverem os cartões do “Pequena estrela-do-mar”, relemos e agrupamos todos cartões sem conversar sobre eles. Depois foi pedido aos participantes para votarem nos cartões que gostariam de conversar sobre. Cada pessoa tinha direito a cinco votos (o *board* FunRetrospectives te dá a opção de limitar o número de votos por participante).

## Check-out – Uma palavra antes de ir

E, para terminar a retrospectiva, como atividade de check-out fizemos duas ações:

1. Lemos os *action items* (um documento compartilhado foi criado durante a retrospectiva e os itens eram adicionados durante as conversas sobre os mais votados).
2. Cada participante escreveu uma palavra que reflete como se sentia no momento, ao terminar a reunião de retrospectiva. Usamos a atividade “Uma palavra antes de ir” do FunRetrospectives.



## Comunicação remota

Muitas das imagens compartilhadas neste livro mostram *boards* remotos. Por mais que reuniões presenciais sejam preferidas, o

trabalho remoto é a realidade de muitos times, como foi para a nossa equipe.

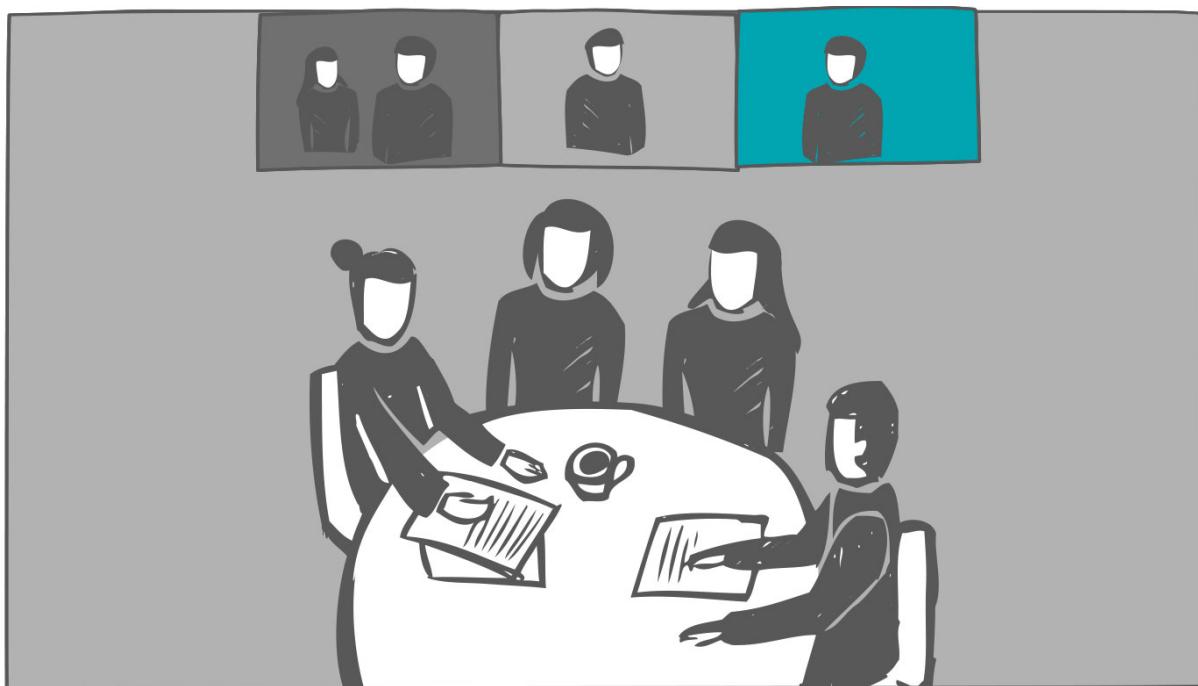


Ilustração criada a partir de uma foto do time em reunião pelo Zoom

Existem diversas ferramentas que facilitam a comunicação remota, mas nada substitui a interação cara a cara entre pessoas. Este time utilizou o Slack para conversas do dia a dia, o Zoom para as reuniões e telepresença e o FunRetrospectives para as retrospectivas remotas. O Zoom ficava o dia todo ligado transmitindo o que estava acontecendo na mesa do time e, quando alguém precisava tirar alguma dúvida, era só desmutar o microfone e falar. Mesmo com essas ferramentas, surgiu como ponto na retrospectiva a necessidade de ter o contato presencial com mais frequência.

## » DIA 38 COFRINHO DA DAILY SCRUM

Isso é uma prática comum em muitos times ágeis. O contexto: aquele típico atraso de cinco minutos para começar uma reunião. O problema é que isso não é nada legal, especialmente numa reunião de dez a quinze minutos na qual as pessoas ficam de pé.

É assim que surgiu a ideia de um cofrinho para coletar a multa (tipicamente uma moeda de um real por minuto de atraso) de quem chegar atrasado para a Daily Scrum.

Segue o e-mail instituindo a prática do cofrinho da Daily Scrum:



Oi, pessoal.

Alteramos o horário da nossa Daily para termos menos atrasos/ausências e, conforme sugerido pelo PO, agora teremos que pagar uma prenda quando nos atrasarmos ou faltarmos em reuniões do projeto. :)

A ideia é revisarmos essa prática e os valores depois de rodarmos um tempo. A presença do *sponsor* será acordada conforme demanda, porém, para o restante do time está valendo para as reuniões de daily, planejamento, revisão e retrospectiva.

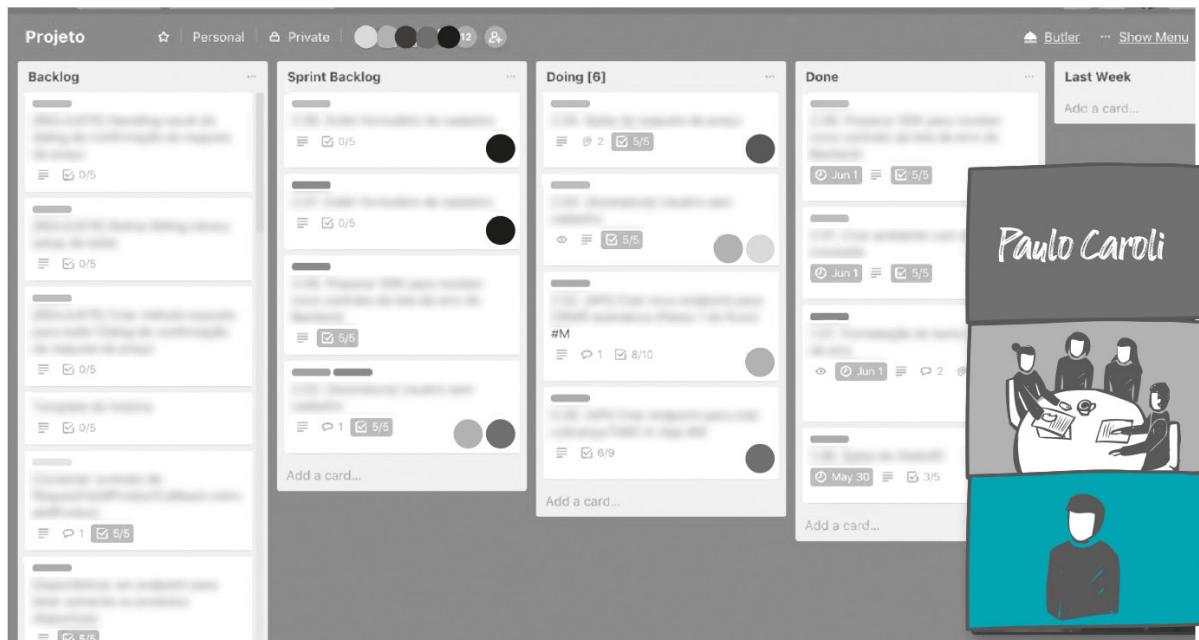
Inicialmente ficarei responsável por fazer o controle dos valores em um Excel compartilhado. Caso tenham alguma sugestão ou problema, me procurem, por favor.

Abs,

Mary.

## » DIA 38 MAIS UMA DAILY - O KANBAN NO TRELLO

Note na imagem abaixo a equipe remota usando o *board* do Trello para seguir a Daily no estilo Top Down das tarefas.



A equipe conversa sobre as histórias na coluna *doing*. Por exemplo, em um dos cartões, a desenvolvedora o abriu e falou: “Ontem eu terminei esta história, aconteceu isso, aquilo e aquilo outro. Mas a história está completa de acordo com os critérios de aceite. Estou movendo-a para a coluna *done*.”

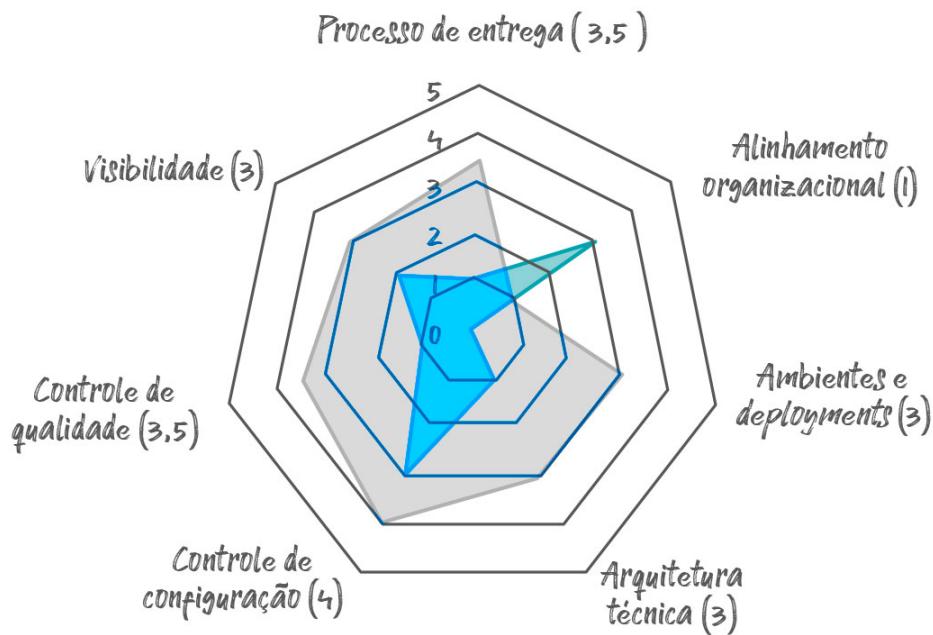
A equipe usa um kanban no estilo *To do > Doing > Done*, no qual o *To do*, ou seja, a lista de trabalho para a Sprint é o Sprint *backlog*, conforme acordado na reunião de planejamento.

## » DIA 39 ASSESSMENT DE DEVOPS ESTADO ATUAL

Conforme planejado anteriormente, aplicamos novamente o *assessment* de DevOps na equipe. Na terceira semana, ele foi utilizado para verificar a situação do legado, do código que a equipe recebeu. Isso ficou registrado no capítulo “*Assessment de DevOps do legado*”.

É importante ressaltar que tanto as perguntas realizadas quanto as pessoas entrevistadas (uma desenvolvedora e um desenvolvedor da equipe) foram as mesmas.

Seguem uma imagem e uma tabela demonstrando os resultados para cada domínio do *assessment* no início da Sprint 1 (uma semana após a Lean Inception) e na Sprint 4 (quatro semanas após a Lean Inception).



Domínio da avaliação	nota Sprint 1	nota sprint 4
Processo de entrega	1	3,5
Controle de qualidade	1	3,5
Controle de configuração	3	4
Ambientes e <i>deployments</i>	0	3
Armazenamento de dados	NA*	NA*
Arquitetura técnica	1	3
Alinhamento organizacional	3	1
Visibilidade	2	3

\* NA = Não se aplica

A seguir estão as notas e os comentários das pessoas entrevistadas.

## Processo de entrega – Legado

Nota: 1

- › Existia um padrão de *branches* (estilo *gitflow*) e havia um teste manual, quando voltava um ok, os artefatos eram enviados por e-mail.

## Processo de entrega – Atual

Nota: 3,5

- › Agora temos testes automatizados que são executados a cada commit.

## Controle de qualidade – Legado

Nota: 1

- › Projeto Android não tinha nenhum tipo de teste; iOS tinha alguns poucos testes unitários.
- › Testes manuais (testes de mesa) eram executados a cada liberação do produto.
- › É o suporte que faz o tracking de bugs.

## Controle de qualidade – Atual

Nota: 3,5

- › Antes era usado somente um teste manual realizado pelo pessoal do suporte; agora testamos manual e automaticamente antes de chegar em produção.

## Controle de configuração – Legado

Nota: 3

- › Tanto Android quanto iOS já utilizavam o gerenciador de dependências – Gradle e Cocoapots. Eles já usavam o Git.

## Controle de configuração – Atual

Nota: 4

- › Estamos começando a caminhar para *trunk-based*, mas não estamos lá ainda. Todos os testes, scripts de set-up de ambiente estão no Git, nosso controle de versão.

## Ambientes e deployments – Legado

Nota: 0

- › Não temos um ambiente de teste completo (existe um QA01 para o *backend*, mas não há ambiente de teste mobile).
- › Não temos CI. Todo processo de *deployment* é manual (com vai e volta de APK via e-mail).

## Ambientes e deployments – Atual

Nota: 3

- › Temos o ambiente de teste QA01 que continua sem funcionar; o que temos de novo são os ambientes de testes mockados. Atualmente temos releases por meio de script.

## Armazenamento de dados

Nota: não se aplica

- › Nosso produto é somente um *middleware*, sem base de dados.

## Arquitetura técnica – Legado

Nota: 1

- › O código é altamente acoplado, não temos nada automatizado. É muito difícil fazer teste unitário do código. Mas já existe o gerenciamento de dependências baseado em versões.

## Arquitetura técnica – Atual

Nota: 3

- › Não me lembro o que coloquei como nota da arquitetura técnica do legado, mas eu diria que deve ser um 0. Agora é um 3 pois temos conversas frequentes sobre arquitetura. No dia a dia discutimos estratégia de teste, de refatoração e sobre como isolar as funcionalidades. Mas dado o estado atual do código, ainda temos que melhorar.

## Alinhamento organizacional – Legado

Nota: 3

- › Existe um canal aberto de comunicação com os times de produto (que consomem a nossa SDK).
- › Existe um pouco de separação de conhecimento: uma pessoa sabe de Android, outra sabe de iOS.

## Alinhamento organizacional – Atual

Nota: 1

- › Não está claro o alinhamento entre o que estamos fazendo e as necessidades do negócio. O plano inicial que conversamos na Lean Inception mudou, mas não há certeza sobre qual é o plano atual.

## Visibilidade – Legado

Nota: 2

- › Temos o histórico de release no Git.
- › O time começou a usar o Trello para dar visibilidade das suas tarefas.

## Visibilidade – Atual

Nota: 3

- › Conseguimos rastrear quem fez o quê, já que isso está no Trello e no Gitlab. Mas não sabemos quando uma versão vai para produção, pois datas e entregáveis não estão claros para todos da equipe.

## » DIA 39 REUNIÃO PARA APRESENTAR O ASSESSMENT DE DEVOPS

O melhor resultado do *Assessment* de DevOps não é ele em si, mas sim a conversa gerada a partir dele.

Marcamos uma reunião com todos da equipe e os principais *sponsors* do projeto para apresentar o *assessment* e conversar sobre os próximos passos para DevOps.

Além das pessoas da equipe e dos principais *sponsors*, estava presente nessa reunião um DevOps Coach (um consultor sênior com experiência em desenvolvimento e práticas de DevOps). Ele estava presente, pois o cliente expressou o interesse em adicionar esse consultor durante algumas Sprints para parear com desenvolvedores e fazer um coaching nas tarefas de DevOps e desenvolvimento ágil (testes, automação, *continuous delivery* etc).

Segue a lista de anotações/próximos passos para o trabalho deste DevOps Coach:

- » Dar visibilidade à proporção de bugs por release (para averiguar se o aumento de estabilidade reflete na redução de bugs).

- » Adicionar ao servidor de integração contínua um alerta para quando o % de testes reduzir do nível mínimo determinado para cada componente (no nosso caso um SDK).
- » Fomentar a prática de feature flag e desacoplar o release de *deployment* (reduzir a dependência e o sincronismo entre o negócio e a equipe).
- » Fomentar práticas de Dark launch e Blue green *deployment*.
- » Analisar em mais detalhes o armazenamento de dados do nosso produto (SDK), considerando a base local e a necessidade de dados para testes.
- » Criar e dar visibilidade ao versionamento com notificação automática de novos releasable candidates (essencial para nosso time a para todos os times que fazem uso do nosso produto – SDK).
- » Criar um Dashboard de instalação e uso do nosso produto (por exemplo, precisamos saber quando o primeiro usuário usar um aplicativo com o nosso SDK).

## » DIA 40 CONVITE PARA A RETROSPECTIVA



Time,

Seguem os links e a agenda para nossa retrospectiva de hoje.

**1. Contexto:** Estamos na Sprint 4. Vamos conversar sobre conquistas e melhorias, dado nosso aprendizado até o momento.

**2. Diretiva primária para retrospectivas:** “Independentemente do que descobrimos, nós entendemos e realmente acreditamos que todos fizeram o melhor trabalho que poderiam, dado o que era conhecido na época, suas habilidades e competências, os recursos disponíveis, bem como a situação em questão”.

**3. Energizer:** Surpresa.

**4. Check-in:** Vamos usar a atividade “Verificação de segurança”. Por favor usar esse *board* do FunRetrospectives.

**5. Atividade principal:** Vamos usar a atividade “CLFA”. Por favor usar esse *board* do FunRetrospectives.

**6. *Filtering*:** Cada participante tem cinco votos no FunRetrospectives. Os cartões serão ordenados pelos mais votados e conversaremos sobre eles.

**7. Check-out:** Conversaremos no final da retrospectiva.

Abraços,

Caroli e Mary pareando.

## » DIA 40 RETROSPECTIVA CANCELADA

Veja o e-mail enviado após o cancelamento da retrospectiva por falta de quórum.



Time,

Ouvimos e compreendemos a correria do dia a dia de trabalho. Mas cancelar a retrospectiva é um indicador de que estamos operando de modo reativo mais do que proativo. Por exemplo, o corpo de bombeiros que passa tanto tempo apagando incêndio que não consegue trabalhar evitando os futuros incêndios.

Nossa equipe tem uma retrospectiva a cada duas semanas. Logo, cancelando a de hoje, vamos ter quatro semanas entre retrospectivas, um período muito longo para uma equipe ágil, com entrega de MVP e que busca melhoria contínua. De forma alguma queremos trazer mais reuniões ou itens na agenda conturbada de todos, somente estamos constatando que o nosso dia a dia eventualmente está com muitos incêndios.

Provavelmente isso é uma conversa além do escopo da equipe, considerando o escopo organizacional. Deixamos aqui uma pergunta para reflexão: Como mudar o modo operante do nosso corpo de bombeiros?

Abraços, Caroli e Mary pareando.

---

25Design Thinking na Wikipedia. Disponível em: <[https://pt.wikipedia.org/wiki/Design\\_thinking](https://pt.wikipedia.org/wiki/Design_thinking)>. Acesso em: mar. 2020.

26Lean StartUp na Wikipedia. Disponível em: <[https://pt.wikipedia.org/wiki/Lean\\_startup](https://pt.wikipedia.org/wiki/Lean_startup)>. Acesso em: mar. 2020.

27No capítulo "Sobre algumas práticas específicas", na primeira seção deste livro, há uma parte explicativa sobre este assunto (N.A.).

28Artigo escrito por Paulo Caroli em 2015. Disponível em: <[www.caroli.org/a-7-step-agenda-for-effective-retrospectives/](http://www.caroli.org/a-7-step-agenda-for-effective-retrospectives/)>. Acesso em: mar. 2020.

29Em inglês há um termo para isso, que deu nome a uma cerimônia do Scrum: groomed. Aliás, essa palavra em inglês tem outro significado no inglês da Inglaterra, motivo pelo qual o povo do Scrum mudou o nome da reunião de grooming para refining (N.A.).

30Na primeira seção deste livro, há um capítulo explicativo sobre este assunto (N.A.).

31No capítulo "Sobre algumas práticas específicas", na primeira seção deste livro, há uma parte explicativa sobre este assunto (N.A.).

32Essa técnica ajuda a arquivar momentaneamente quaisquer itens, ideias ou questões que são levantadas durante uma reunião ou workshop, mas não são úteis para a discussão naquele momento específico. É uma ferramenta essencial para o facilitador, pois proporciona uma maneira educada de dizer "sim, eu ouvi você. Mas essa conversa ficará para depois".

33Na primeira seção deste livro, há um capítulo explicativo sobre este assunto (N.A.).

34Santana, Magno de. O trabalho de FDP do Product Owner. Disponível em: <<http://knowledge21.com.br/blog/fdp-do-product-owner/>>. Acesso em: mar. 2020.



# PRINCIPAIS PROBLEMAS E RECOMENDAÇÕES

## » O RESULTADO DAS SEIS SEMANAS

O trabalho de consultoria ágil realizado durante as primeiras seis semanas do time tinha como objetivo:

- » Realizar uma Lean Inception para alinhar o time em relação ao produto que seria construído.
- » Facilitar a definição das normas da equipe, ou seja, como o time iria trabalhar junto.
- » Apoiar o time na busca por excelência técnica.
- » Acompanhar a entrega do MVP.
- » Identificar dores, gargalos, dependências e situações percebidas.
- » Realizar alinhamentos com os principais stakeholders sobre os pontos mapeados e como atuaríamos neles.

Os pontos mapeados durante esse trabalho foram apresentados aos stakeholders na sexta semana e os principais problemas e recomendações abordados nessa conversa serão apresentados a seguir.

## » INDEFINIÇÃO/DUALIDADE DOS PAPÉIS

O primeiro ponto discutido na reunião foi a indefinição e dualidade de alguns papéis. Neste caso, os papéis de Scrum Master, Product Owner e gerente de projetos.

### Gerente de projetos ou Scrum Master?

Em Porto Alegre ficavam cinco pessoas desenvolvedoras e uma gerente de projetos/Scrum Master. No cliente, em outra cidade, ficavam mais duas pessoas desenvolvedoras, um UX Designer e um Product Owner.

O papel de gerente de projetos era necessário para a realização de atividades internas como consultoria e o papel de Scrum Master era imprescindível para o time. Ter uma pessoa realizando os dois papéis foi confuso para a equipe, principalmente porque essa pessoa não possuía autonomia sobre o cliente para exercer o papel de Scrum Master efetivamente.

Diante desta constatação, criamos *action items* para garantir que uma pessoa externa teria autonomia para resolver problemas em

que fosse necessário acessar pessoas de outras áreas dentro da estrutura do cliente.

## O que é esperado do papel de PO?

Também foi identificado que existiam expectativas diferentes para o papel de PO. O time esperava uma série de comportamentos e atividades, enquanto o PO tinha outra visão sobre o significado desse cargo. As expectativas do time foram passadas para o PO, ele compartilhou suas dificuldades para atender algumas dessas perspectivas e chegamos a um alinhamento sobre como trabalharíamos nesse contexto.

Como *action item* deste tema ficou definido que faríamos uma atividade chamada *Roles and expectations*, que tem como objetivo mapear as expectativas dos membros do time para cada cargo e, com base nisso, definir o que fará ou não parte de cada papel dentro daquele contexto para aquela equipe.

## » REUNIÕES CANCELADAS E CADÊNCIA INTERROMPIDA

O segundo assunto abordado na conversa com stakeholders foi o cancelamento de reuniões do time e o impacto que interromper essa cadência gerava para o time e para o projeto.

A Daily Scrum deve acontecer sempre no mesmo horário e local por reduzir a complexidade envolvida sem ser necessário descobrir qual é a hora da reunião e onde ela vai acontecer para cada dia da semana.

O problema aqui é que boa parte do time se preparava para participar da Daily, ia para a sala no horário; ligava os equipamentos para a videoconferência, entrava no Zoom e nada do restante do time aparecer. O cofrinho ajudou no início, mas após um tempo o problema voltou a ocorrer.

Além dos atrasos e ausências ocorridos na Daily, isso também acontecia com as demais reuniões do time: planejamento, revisão e retrospectiva. Essas ausências normalmente resultavam em cancelamentos e, em outras vezes, o PO cancelava a reunião devido a outras prioridades.

Já tínhamos alterado os horários das reuniões duas vezes para atender melhor a disponibilidade de todas as pessoas do time e,

nesse momento, buscávamos responder às seguintes perguntas: Por que isso continua acontecendo? O que está gerando esse atraso?

Ao colocar todo mundo em uma sala para discutirmos juntos sobre os problemas, foi possível identificar o que realmente estava por trás dos cancelamentos. Sem saber, o *sponsor* do projeto estava puxando pessoas para outras reuniões no mesmo horário; isso acontecia principalmente com o PO. Por não ter conhecimento do impacto que isso causava no projeto, esse *sponsor* acreditava que essa ação não era um problema e fazia isso sem a menor intenção de prejudicar o time.

Também descobrimos nessa conversa que muitas vezes esse *sponsor* fazia um pedido para as pessoas desenvolvedoras achando que era algo simples e que gastariam poucas horas para resolverem. A realidade era que essas pessoas às vezes passavam dias sem se dedicarem ao projeto e isso também fazia com que as reuniões fossem canceladas.

O *action item* neste caso foi identificar em que situações era necessário “sequestrar” pessoas para outros projetos e reuniões e buscar alternativas para resolver o problema, blindando o time. Também combinamos que daríamos visibilidade ao *sponsor* do impacto que as ausências necessárias gerariam no projeto, de forma que as decisões pudessem ser tomadas de forma consciente.

## » SPRINTS SEMANAIS COM RETROSPECTIVAS QUINZENAIAS

O time decidiu trabalhar com Sprints de uma semana, entretanto com uma retrospectiva a cada duas semanas. Essa foi uma decisão da equipe acordada na segunda semana de trabalho, na primeira retrospectiva, em uma atividade para definir as normas de trabalho.

*“Todo time deve fazer uma retrospectiva por semana, a menos que esteja sem tempo. Nesse caso deve fazer duas!”*

Eu, Caroli, sempre repito essa frase. Até enviei isso para a equipe em algum e-mail sobre a importância das retrospectivas.

Naquela reunião de acordos das normas do time, a equipe decidiu fazer uma retrospectiva a cada duas semanas. Eu me lembro que a Mary e eu conversamos sobre isso e ambos preferimos retrospectivas frequentes, mas, como essa era uma decisão do time, nós acatamos.

Entretanto, poucas semanas depois, nos arrependemos. A retrospectiva, como reunião de melhoria contínua, acontecia com uma frequência muito abaixo da necessidade dessa equipe em formação.

Exatamente pelo time estar em formação, precisava alinhar vários aspectos importantes como os acordos de convivência e o processo de trabalho para entregar com muita eficiência.

Mas não haviam retrospectivas, nem outra reunião ou fórum para conversar sobre melhorias. Não haviam momentos suficientes para estabelecer os acordos do time, ou para reajustar o entendimento dado às novas descobertas.

Entendemos que imprevistos acontecem e, às vezes, reuniões são canceladas. Esse foi o caso com uma reunião de retrospectiva. Nesse caso, o time ia ficar quatro semanas sem ela. Porém, como tínhamos desafios importantes a serem resolvidos, tivemos de marcar reuniões sob demanda para lidar com os desafios que enfrentávamos naquele momento.

Teria sido mais fácil, e com menos intervenção (e-mail pedindo por uma reunião extra, já que não estava na agenda de todos), caso tivéssemos retrospectivas semanais, ao invés de quinzenal e com cancelamento.

Nossa sugestão é que esse time mantenha a cadência da Sprint semanal, mas com uma retrospectiva por semana. Dessa forma, poderemos lidar com problemas enquanto esses ainda forem pequenos e isso também reduzirá a necessidade de reuniões emergenciais. E ainda comporta o caso excepcional de uma retrospectiva ser cancelada (não recomendamos isso, mas, se acontecer, o time não ficará mais de duas semanas sem essa reunião de extrema importância para a melhoria contínua). Então recomendamos realizar uma retrospectiva por Sprint, ou seja, uma a cada semana. O ciclo de entrega é semanal, logo a retrospectiva deve estar neste mesmo ciclo.

## » EXCESSO DE SPIKES E FALTA DE DIRECIONAMENTO DO NEGÓCIO

Durante as seis primeiras semanas do projeto foi observada uma alta quantidade de *spikes* por Sprint e a dificuldade em definir o que buscávamos descobrir com esses *spikes* muitas vezes fazia com que eles gerassem novos *spikes*.

Isso estava causando muita ansiedade no time. O PO achava que o time era lento demais, pois não estávamos fazendo entregas tangíveis. Já as pessoas desenvolvedoras não viam progresso diante de tanta incerteza e se sentiam frustradas com o trabalho.

Mais uma vez tentamos entender a origem do problema, e ter as pessoas certas na mesma sala foi essencial.

Com a ausência do PO, algumas decisões de negócios não eram tomadas e a falta do direcionamento fazia com que o time explorasse todas as opções possíveis para todos os problemas. Essa situação foi potencializada quando o PO delegou essas responsabilidades para um desenvolvedor que estava trabalhando com o produto desde o início. Este desenvolvedor entendia muito bem sobre o como, mas não tinha a visão dos usuários e do negócio.

O resultado disso foi o excesso de *spikes* que buscavam explorar diversas formas de fazer alguma coisa, mas que não se relacionavam a nenhum objetivo de negócio.

Sendo totalmente transparentes, alinhamos as expectativas com o PO e stakeholders sobre quem tomaria as decisões e daria o direcionamento do negócio. Como um *action item* deste tópico, agendamos um workshop em Porto Alegre com todo o time para definir OKRs (o capítulo “Objetivos e resultados-chave - OKR”, no início deste livro, explica mais sobre o assunto) e planejar o escopo dos próximos meses.

## » FALTA DE VISIBILIDADE SOBRE A SAÚDE DO TIME

Ao buscarmos identificar o que estava gerando desconforto no time e o que precisava melhorar, percebemos que um dos problemas era justamente a falta de visibilidade sobre a saúde dele.

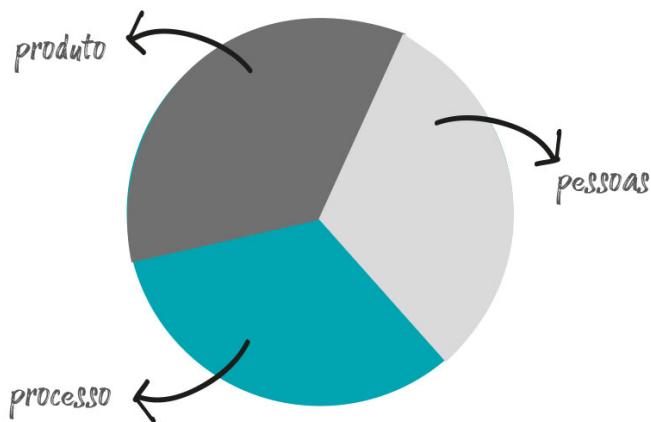
Apesar dos diversos mecanismos que utilizamos para aproximar as pessoas, esse ambiente só permitia que notássemos que o clima não estava legal ou que existia algum problema ali. Ainda faltavam informações sobre o que poderia estar prejudicando a saúde do time.

Além disso, estava muito difícil dar visibilidade aos problemas para o restante da equipe que ficava em outra cidade. Também era difícil sentir como estava o clima por lá. Precisávamos sair do achismo.

Para verificar a saúde do time, sugerimos o acompanhamento periódico (a cada Sprint, se possível) dos seguintes aspectos: pessoas, processos e produto.

Cada um desses aspectos representa um terço da roda da equipe, com as cores verde, amarelo e vermelho, representando respectivamente situação boa, uma que requer alguma atenção, e uma que requer uma ação corretiva com maior urgência.

## A saúde do time



Abaixo segue uma lista de perguntas para cada aspecto. Essas perguntas devem ser usadas para pautar a decisão entre as cores verde, amarela e vermelha.

### Pessoas

- › Precisamos de mais gente no time, seja para complementar com algum conhecimento ou cargo necessário, ou para aumentar a capacidade dele?
- › Está sobrando gente no time, ou seja, estamos com uma capacidade muito ociosa?
- › O *onboarding* de novas pessoas no time foi bem planejado e executado?
- › O *roll-off* de pessoas do time foi bem planejado e executado?
- › Temos problemas de relacionamento entre as pessoas?
- › Todos estão felizes e efetivos com seu papel e trabalho desempenhado atualmente no time?

## Processo

- › Todas as pessoas do time, bem como seus stakeholders e *sponsors*, compreendem e concordam com o processo atualmente utilizado?
- › Todos os participantes e interessados no trabalho do time sabem onde e como buscar informações sobre ele (onde está, como está, para onde vai, por quê)?
- › Todas as pessoas do time compreendem e concordam com o relacionamento entre escopo do produto, datas de entrega e capacidade da equipe?
- › Todos entendem o seu papel no time e sabem como trabalhar com as pessoas que desempenham outros papéis?

## Produto

- › Todos entendem o direcionamento estratégico do produto?
- › Todos entendem as métricas do produto sendo construído e como cada nova funcionalidade está sendo medida?
- › Todos entendem as funcionalidades, histórias e tarefas em criação e como essas se relacionam com o produto e seu cronograma de liberação?
- › O ritmo de completude das funcionalidades, histórias e tarefas está de acordo com o comprometimento do escopo e com a data de entrega para o produto (ou MVP) em questão?

## » PESSOAS COMPARTILHADAS E BAIXA PRIORIDADE DO PROJETO

Dado o contexto do negócio, as pessoas que estavam alocadas do lado do cliente não eram exclusivas deste projeto, ou seja, atuavam em dois ou mais projetos ao mesmo tempo. Desde o início houveram diversas conversas sobre como poderíamos deixar essas pessoas 100% dedicadas a esse produto, mas essa mudança simplesmente não era possível no curto prazo e tínhamos que trabalhar com essa restrição.

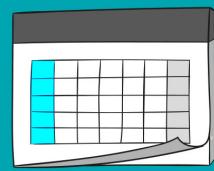
Muitos dos problemas já abordados anteriormente ocorriam ou eram potencializados por este motivo. Dentre todos, esse era o que mais impactava o time, e o fato de que os outros projetos eram tratados com maior prioridade intensificava ainda mais a questão.

O que questionamos durante a conversa com os stakeholders foi: O que determina a prioridade dos projetos? Este projeto realmente tem baixa prioridade?

Descobrimos que, na verdade, o projeto era o mais importante, mas, como esse alinhamento não tinha sido feito internamente, o “gritômetro” estava determinando a prioridade. Conversamos então sobre os impactos que o produto tinha para a organização e buscamos conscientizar os stakeholders sobre essa importância.

Apesar de listado como um problema do projeto, essa situação ia além e era muito mais complexa para ser tratada somente pelo time. Era um problema já conhecido e intrínseco à forma como a organização trabalha, ou seja, fazia parte da cultura da empresa.

Nesse momento, sabíamos que não resolveríamos o problema no projeto, com esse time, mas a transparência foi muito importante para alinhar as expectativas e gerenciar os conflitos.



# OS PRÓXIMOS QUATRO MESES

## » ESTOU TE VENDO, ESTOU TE OUVINDO

Algo simples que muitas vezes passa despercebido: é muito importante ver e ouvir as pessoas. Isso gera empatia e uma comunicação simples e direta.

Entretanto, essa equipe estava dividida entre duas cidades. O lado bom é que eles já estavam acostumados a trabalhar remotamente. Diariamente o time se comunicava por Slack, e-mail, grupo de WhatsApp, *board* no Trello, *board* no FunRetrospectives, reuniões por videoconferência etc.

Felizmente tivemos a Lean Inception, onde todos estavam juntos por uma semana na mesma cidade. O trabalho colaborativo na Lean Inception gerou bastante empatia e um bom relacionamento inicial.

Mas a comunicação no dia a dia, enquanto cada pessoa estava sentada na sua cadeira, podia ser melhorada. E foi!

Uma técnica simples, mas que aproxima ainda mais as pessoas: Uma TV com o vídeo aberto, o áudio ligado e o microfone no mudo. Isso em cada uma das localizações.

Agora, quando uma pessoa quer verificar o porquê a pessoa não respondeu o e-mail, ou se já foi para a sala de reunião, ou se ainda não chegou no trabalho, ela simplesmente olha para a TV e

confirma: “ah, ela não está ali, sentada na sua cadeira”. Mas se ela ver a pessoa ali, “remotamente ao seu lado” (na TV), é só desmutar o microfone e falar com ela.

Simples assim, era possível fazer com que o time que estava no escritório da consultoria na cidade A e o outro que estava no escritório do cliente na cidade B, estivessem “na mesma sala”!

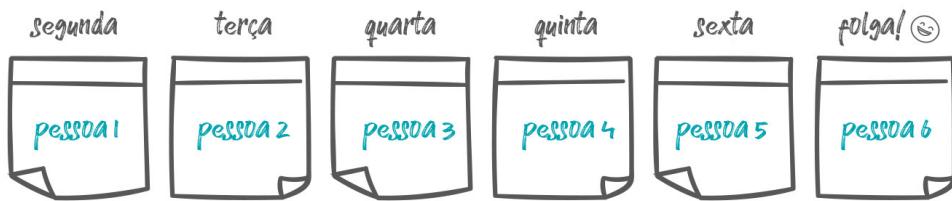
Umas das práticas do XP – eXtreme Programming – defende que o time deveria estar junto, todos sentados em uma única mesa. Essa prática de chamada de *Colocated team* é até comum em empresas mais ágeis, mas era uma ideia muito extrema em 1999.

Já que não podemos estar juntos fisicamente, devemos estar juntos remotamente. Mantenha um canal com vídeo e som aberto para criar o conceito de *Colocated teams* para times remotos.

## » QUEM FACILITA AS REUNIÕES DE SPRINT?

As pessoas desenvolvedoras que ficavam em Porto Alegre já tinham experiência com Scrum e possuíam interesse em desenvolver a competência de facilitação. Essa combinação foi o motivo perfeito para testarmos uma forma diferente de trabalhar quando a oportunidade surgiu.

Durante uma conversa sobre expectativas para o papel do Scrum Master, acordamos que facilitar as reuniões de Sprint não deveria ser responsabilidade somente da pessoa que estava nesse papel e, a partir disso, surgiu a ideia de montarmos uma agenda de facilitação.



A agenda de facilitação funcionava da seguinte forma:

Éramos seis pessoas em Porto Alegre e tínhamos cinco dias da semana para trabalhar. Decidimos, então, que escolheríamos uma pessoa diferente para cada dia da semana e alguém estaria de folga toda semana. Quando virava a semana, rodávamos as pessoas de

forma que quem estava de folga na semana anterior, agora ficava responsável por facilitar as reuniões da segunda-feira. Isso garantia que todas teriam oportunidade de facilitar todos os tipos de reuniões.

Ter uma pessoa de folga ajudava, pois, quando alguém precisava sair de férias, era só combinar a troca com quem estaria de folga da facilitação naquele período. Fazer a gestão dessa agenda e garantir que a pessoa facilitadora sempre estaria presente no seu dia era responsabilidade de todo o time e não somente da pessoa no papel de Scrum Master.

Frequentemente surgiam algumas atividades e *action items* que necessitavam de pessoas desenvolvedoras, mas ninguém se voluntariava para fazer. Alguns exemplos são: montar uma apresentação sobre TDD pro cliente, documentar *guidelines* de testes, garantir que o feedback no fim do pareamento estava acontecendo, entre outras.

Em busca de uma solução para esse problema, a agenda ganhou mais uma finalidade. No dia em que surgisse uma atividade que ninguém queria fazer, a pessoa facilitadora do dia ficaria responsável por ela.

Isso funcionou muito bem durante todo o projeto e, quando alguém estava remoto, era comum a pergunta no Slack do time: “De quem é a Daily hoje?”.

## » A PRIMEIRA ENTREGA EM PRODUÇÃO

Chegou o momento de colocar o MVP em produção!

A primeira dificuldade do time em relação a essa entrega foi o longo tempo que se passou desde o início do projeto até que a primeira entrega realmente entrasse em produção. Isso aconteceu depois de mais de três meses de projeto.

Mas se o time trabalhava com Sprints semanais, ou seja, fazia entregas toda semana, por que a primeira entrega em produção demorou tanto tempo?

Pelos problemas apresentados no capítulo anterior: a indefinição do papel de PO, as reuniões canceladas, o excesso de *spikes*, a falta de direcionamento do negócio, a falta de visibilidade sobre a saúde do time, pessoas sendo compartilhadas com outros projetos, a falta de retrospectivas e, principalmente, a baixa prioridade do projeto. Todos esses pontos influenciaram muito para que a primeira entrega levasse tanto tempo, mas não foi só isso.

A principal lição aprendida sobre nossa primeira entrega foi a importância da integração frequente. Como estávamos trabalhando com um SDK, nós entregávamos o nosso produto para outros times

que eram responsáveis por integrar o nosso SDK com seus respectivos produtos.

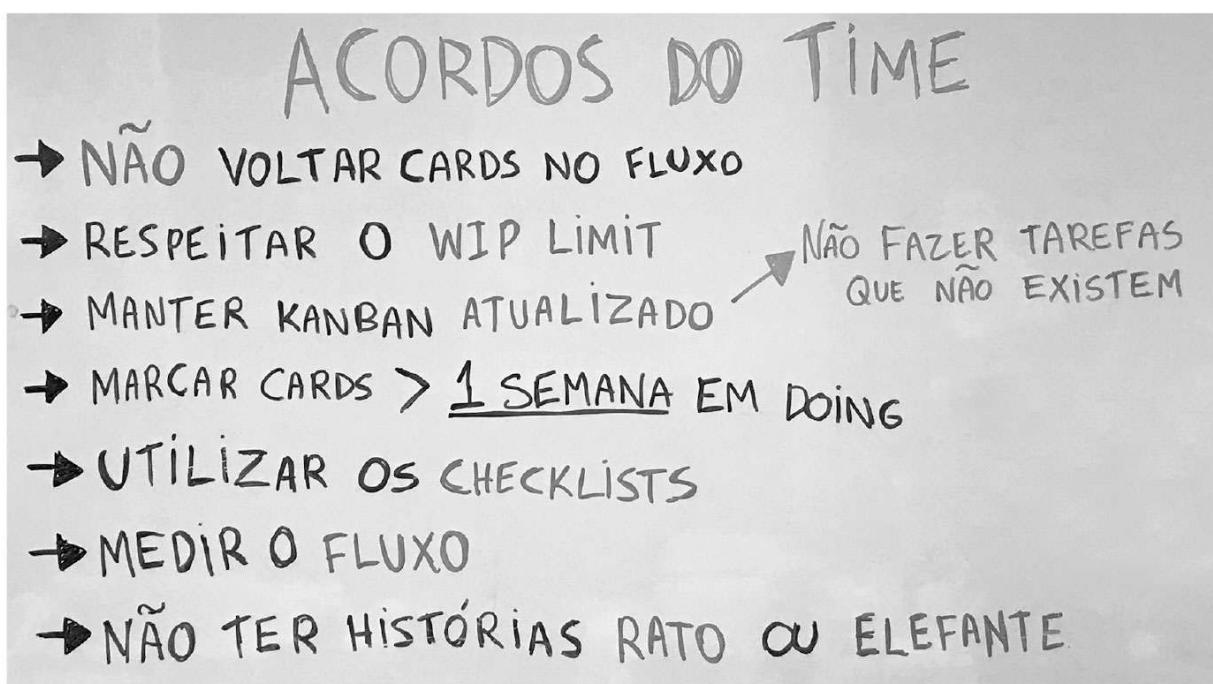
Por mais de três meses fizemos correções e incluímos diversas funcionalidades novas no nosso produto. Vários testes de integração foram feitos, mas o SDK nunca foi realmente integrado com nenhum produto durante todo esse tempo. É óbvio que, quando chegou a hora disso acontecer, ocorreram muitos problemas.

O risco foi levantado em várias *dailies*, reuniões de planejamento, revisões e retrospectivas, mas não tínhamos como priorizar essa integração no *backlog* dos outros times. Às vezes é necessário primeiro sentir a dor para depois fazer algo a respeito, e foi exatamente o que aconteceu aqui.

Após a entrega, fizemos uma retrospectiva de duas horas onde entendemos todos os pontos em que erramos para que os mesmos erros não fossem cometidos nas próximas entregas. O principal *action item* foi: “Definir datas com os outros times para integrações dos produtos antes das entregas.”

## » ACORDOS DO TIME

Depois do MVP entrar em produção, paramos para rever nosso processo e fazer alguns acordos sobre como iríamos trabalhar a partir daquele momento.



Neste ponto já estávamos usando Kanban e não fazíamos mais Sprint Planning e Sprint Review. Continuamos com retrospectivas semanais e a Daily passou a funcionar de outra forma. A principal pergunta agora era: “o que é necessário para que esse cartão passe para a próxima etapa?”.

## » INDIVÍDUOS E INTERAÇÕES MAIS QUE PROCESSOS E FERRAMENTAS

Práticas, técnicas, cerimônias e ferramentas não funcionam sozinhas! As pessoas devem entender para que elas servem e quando utilizá-las. Além disso, com o entendimento dos valores e princípios, é possível adaptá-las ao contexto e tirar maior proveito da sua aplicação. A essência desse trabalho, que ocorreu Sprint a Sprint, estava nos valores e princípios do Manifesto Ágil. E foi isso que possibilitou entender o que não estava funcionando para, então, mudarmos de rumo.

Durante a construção do MVP, foram utilizadas diversas práticas ágeis. Constantemente aplicamos os princípios ágeis e trabalhamos a melhoria contínua dos pilares: pessoas, tecnologia e processos. Entretanto, algo ainda parecia não funcionar muito bem. Na constante busca por melhoria contínua, refletimos sobre o primeiro valor do Manifesto Ágil que diz: “Indivíduos e interações mais que processos e ferramentas”. Ao colocar esse valor em prática, foi possível entender que, além de adotar práticas ágeis, era preciso melhorar a interação entre as pessoas.

Não importa o quanto você tente utilizar os melhores processos e ferramentas, o sucesso do time e, consequentemente do produto,

depende muito mais das pessoas e de suas interações. Diante dessa constatação, foi necessário dar um passo para trás, rever a composição do time e como estávamos interagindo dentro e fora dele. Fazer essas mudanças foi fundamental para transformar uma equipe de trabalho em um time de alta performance!



# ANEXOS

## » TEMPLATES

A seguir estão alguns dos templates que foram criados durante o trabalho apresentado neste livro.

### Checklist de refinamento do trabalho (definition of ready)

Cada item de trabalho (histórias, tarefas etc.) candidato à reunião de planejamento da Sprint deve passar por esse checklist. Em caso de negativa, significa que o item não está pronto para entrar no planejamento.



Item pronto para a reunião de planejamento

Eu tenho a informação necessária para trabalhar?

Não

Sim, favor especificar:

Eu sei o porquê deste item ?

Não

Sim, favor especificar:

Eu sei como mostrar o término deste trabalho?

- Não
- Sim, favor especificar:

Eu consigo relacionar este item a um épico ou uma funcionalidade esperada?

- Não
- Sim, favor especificar:

Esse item cabe em uma Sprint?

- Não, o item deverá ser quebrado em unidades menores
- Sim

## Checklist do Sprint backlog

Use este checklist ao final da sua reunião de planejamento da Sprint. Em caso de alguma resposta negativa, converse com o time sobre os motivos. Se necessário, resolva o problema antes que ele impacte a Sprint.

O time tem conhecimento para trabalhar nos itens que estão no backlog?

- Não
- Sim, favor especificar:

A meta da Sprint foi definita? É factível?

- Não

Sim, favor especificar:

O backlog da Sprint gera um incremento do produto de valor para o usuário?

Não

Sim, favor especificar:

A quantidade de trabalho leva em consideração o limite de WIP, número de pessoas e calendário do time?

Não

Sim, favor especificar:

O time está confiante que entregará a Sprint?

Não

Sim, favor especificar:

Depois de responder a essas perguntas, verifique se os itens da Sprint *backlog* refletem essas respostas (e mantenha controle sobre o *backlog* durante a Sprint).

### Considerações sobre o WIP

Antes de adicionar mais promessas de trabalho (itens no *backlog* da Sprint), vamos verificar o estado atual dos itens em progresso (WIP). Depois de considerar quanto de trabalho acreditamos que ainda está pendente para esses itens, decidimos o quanto adicionar para a Sprint e definimos a sua meta.

## Considerações sobre a definição da meta da Sprint

Antes de decidirmos os itens de trabalho para essa Sprint, vamos primeiro conversar sobre a meta dela.

A meta da Sprint é uma breve descrição do propósito da mesma. Por exemplo: um problema específico do negócio a ser resolvido ou um agrupamento de funcionalidades afins. Tipicamente, o trabalho durante a Sprint é ajustado e alinhado com o objetivo de alcançar a sua meta.

Imagine esse cenário: um diretor chega para um desenvolvedor e pede a ele “Preciso que crie um relatório para demonstrar o número de vendas impugnadas no ano passado por região”. O desenvolvedor deve responder educadamente “Como isso está relacionado à meta da Sprint?”, e encontrar uma forma de chamar a atenção do PO e do Scrum Master para ajudá-lo a manter o foco na meta.

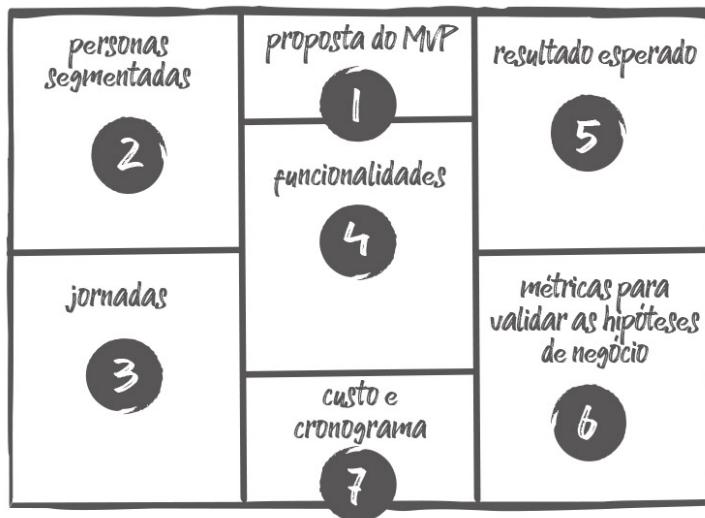
Se o time está com muito WIP (*work in progress*) ainda não terminado da Sprint passada, a meta da atual deve ser algo como: “A mesma da Sprint passada, pois ainda não a alcançamos, que é ABC”.

## Template de validação de ideias

Nós acreditamos que (esse MVP)  
vai conseguir (resultado esperado).

Saberemos que isso aconteceu com base em (métricas para validar as hipóteses do negócio)

## Canvas MVP



Segue a ordem recomendada para preencher (e pensar) os blocos:

1. Proposta do MVP : Qual é a proposta deste MVP?
2. Personas segmentadas: Para quem é este MVP? Podemos segmentar e testá-lo em um grupo menor?
3. Jornadas: Quais jornadas são atendidas ou melhoradas por este MVP?
4. Funcionalidades: O que vamos construir neste MVP? Que ações serão simplificadas ou melhoradas nele?
5. Resultado esperado: Que aprendizado ou resultado estamos buscando neste MVP?
6. Métricas para validar as hipóteses do negócio: Como podemos medir os resultados deste MVP?
7. Custo e Cronograma: Qual é o custo e a data prevista para a entrega deste MVP? Depois de entregue, durante quanto tempo precisaremos coletar os dados para decidir se mudaremos ou prosseguiremos?

## Template do cartão de história

Descrição da história

- » Como um(a) (perfil do usuário/persona)
- » Eu quero (ação a ser realizada/funcionalidade específica do produto)
- » Para que (valor alcançado)

Critério de aceite

- » Dado que (cenário inicial)
- » Quando (ação realizada)
- » Então (estado esperado)

## Template de OKRs

- » Nós vamos (objetivo),
- » conforme medido por (este conjunto de resultados-chave)

## » ATIVIDADES DE RETROSPECTIVA

Neste capítulo estão as atividades de retrospectivas que foram citadas no livro. Este conteúdo é uma cortesia do livro *Funretrospectives*.

### Ping Pong

Esta é uma atividade curta para começar a reunião com um bom clima e deixar os participantes engajados.

Passo a passo da atividade:

1. Comece dizendo o número 1 seguido pelo nome de uma outra participante da atividade
2. A próxima pessoa mentalmente adiciona 1 ao número. Então:
  - › Se o número não é múltiplo de 3 ou 5: diz o número.
  - › Se o número é múltiplo de 3: diz ping.
  - › Se o número é múltiplo de 5: diz pong.E diz o nome de uma outra pessoa
3. Volte para o passo anterior. Entretanto, se alguém errar, comece do número 1 novamente.

Para grandes grupos, é recomendado remover uma pessoa do grupo se ela errar ou acusar alguém erroneamente. Logo, todos estarão rindo e torcendo para os que permanecerem no círculo.

## Que coisa mais louca

Esta atividade é incrível para fazer as pessoas falarem e criarem de modo colaborativo uma história (geralmente engraçada). Ela promove o engajamento e a participação de todos, sendo muito fácil de facilitar, pois é realizada verbalmente.

Passo a passo da atividade:

1. Uma pessoa começa dizendo "Que coisa mais louca" e o nome de outra pessoa.
2. A próxima participante deve continuar a história adicionando três ou quatro palavras e o nome de outra pessoa.
3. E assim por diante até que alguém diga "Fim".

Segue um exemplo:

"Que coisa mais louca"  
"Esse pássaro que voam"  
"Mas eu já vi"  
"Um gato voador"  
"E com superpoderes"  
"Com olhos laser"  
"Que congela aplicativos móveis"  
"Mas não o nosso"

“Por causa da criptonita”

Esta é uma atividade muito divertida e simples. Outra variação é começar com “Era uma vez” e pedir a cada pessoa para adicionar quatro palavras.

## Uma palavra

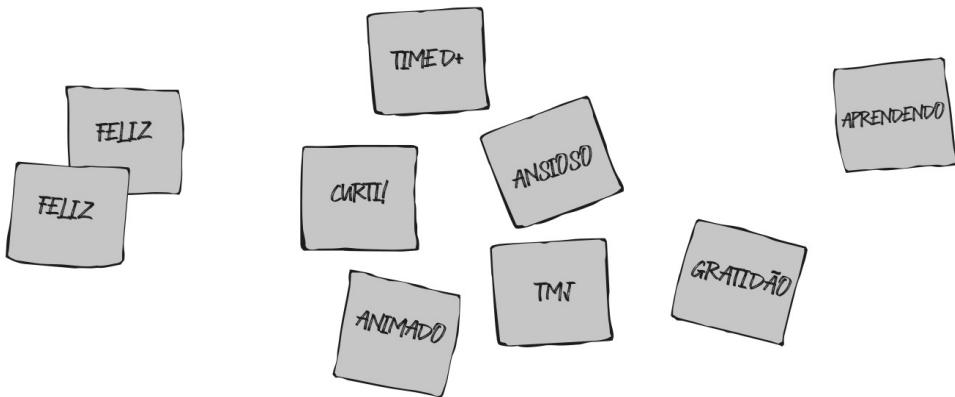
Uma palavra é uma atividade usada para verificar os sentimentos de todos antes de começar ou fechar uma retrospectiva. Normalmente, é um ótimo momento para as pessoas compartilharem seus sentimentos, especialmente quando estão muito entusiasmadas com a retrospectiva.

Passo a passo da atividade:

1. Peça para os participantes descreverem seus sentimentos em uma palavra em um post-it e colocá-lo no quadro comum.

“Por favor, compartilhe uma palavra que descreve como você está se sentindo agora.”

2. Pergunte se alguém deseja compartilhar mais sobre a palavra escolhida.



Esta atividade é muito simples, inspirando as pessoas a falarem em voz alta sobre como se sentem naquele momento da retrospectiva.

## Verificação de segurança

Este é um modo rápido e eficaz de medir a participação do grupo nas atividades de retrospectiva que seguirem.

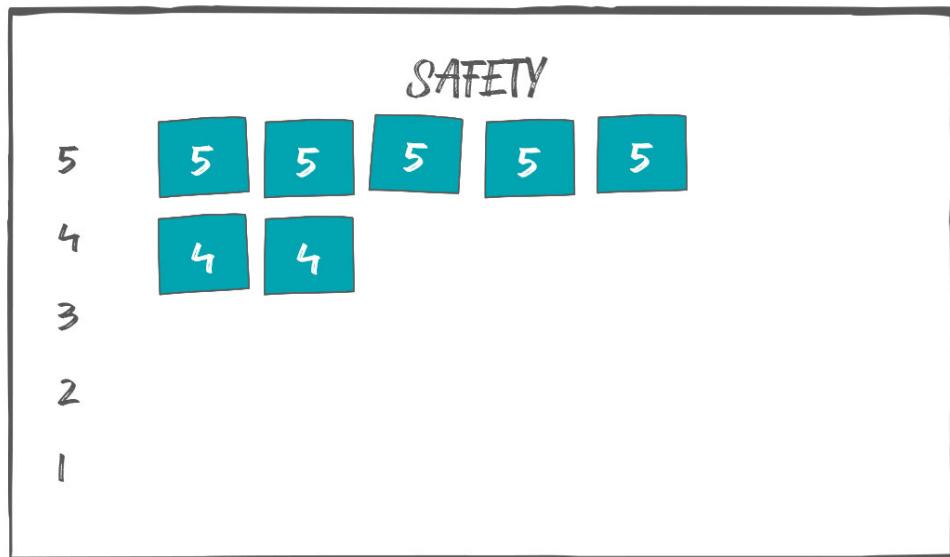
Passo a passo da atividade:

1. Peça aos participantes para escolherem um número entre 1 e 5 que indica quão seguros eles se sentem dentro do grupo, e que os escreva em um post-it (os números devem ser colhidos anonimamente, então todos devem usar o mesmo post-it e canetas da mesma cor). Abaixo, você pode conferir o significado de cada número:

5: Sem problemas, eu conversarei sobre qualquer coisa.

4: Conversarei sobre quase tudo, alguns assuntos poderão ser mais difíceis.

- 3: Conversarei sobre algumas coisas, mas outras serão difíceis de dizer.
- 2: Não falarei muito; deixarei os outros levantaram os assuntos.
- 1: Sorrirei, direi que tudo está bem e concordarei com os gerentes.
2. O facilitador coleta os post-its de cada participante (use um chapéu ou algum recipiente para manter a atividade anônima).
3. Deixe visível para todo o grupo o resultado da verificação de segurança.
4. Confirme os resultados e decida o que vem a seguir.



verificação de segurança

O que fazer com os resultados?

O que pode ser dito quando:

- › O nível de segurança é alto: "Parece que muitos aqui na sala estão dispostos a falar sobre muitos assuntos, então é válido

irmos para a próxima atividade, que vai despertar muita conversa.”

- › O nível de segurança é médio: “Como visto nos resultados, alguns participantes não estão dispostos a falar sobre todos os assuntos. Vamos manter isso em mente e eu, como facilitadora, não perguntarei a opinião de todo mundo, dado que alguns de vocês não estão confortáveis para falar sobre tudo”.
- › O nível de segurança é baixo: “Como visto nos resultados da verificação de segurança, vamos usar o tempo restante e fazer uma atividade que pode nos ajudar a aumentar o nível de segurança do grupo.”

## Criando segurança

Esta é uma boa atividade de follow-up quando o nível de segurança é baixo ou médio. Ela permite que os participantes falem sobre qualquer coisa que pode deixá-los desconfortáveis. Um nível de segurança baixo compromete qualquer reunião, então não se prenda à sua agenda se isso acontecer – em vez disso, trabalhe para criar segurança.

### Passo a passo da atividade

1. Peça insights sobre o que poderia estar diminuindo a segurança: “Então, coloque-se no lugar de alguém que não esteja se sentindo seguro para falar sobre alguns assuntos. Quais podem ser as causas disso? Por favor, escreva-as em um post-it amarelo e coloque-o no quadro.” Essa frase é muito poderosa. A verificação

de segurança é anônima, mas a frase abre as portas para assuntos serem tratados de forma sutil. A pessoa não precisa dizer “Eu me sinto inseguro nisso ou naquilo”. Em vez disso, questões são levantadas sem uma primeira pessoa. Isso deve revelar as causas que diminuem a segurança.

2. Agrupe as causas no quadro, por similaridade.
3. Peça para os participantes que deem ideias sobre como fazer as pessoas se sentirem seguras, dadas as causas no quadro: “Pense sobre as coisas que você pode fazer para ajudar a sanar essas causas no quadro (nos post-its amarelos), escreva-as em um post-it laranja e coloque-o do lado da causa.”
4. Leia todas as notas e guie (cuidadosamente) uma conversa sobre elas. O facilitador terá que usar seu julgamento ao ler as notas e conduzir a conversa. Você deve ter em mente que algumas pessoas podem ficar desconfortáveis com alguns assuntos e ideias, então não se apresse nas conclusões ou coloque pessoas na berlinda.
5. Faça a verificação de segurança de novo.

Se tudo der certo, o nível de segurança vai aumentar. O importante é que os resultados da verificação de segurança tenham sido levados em consideração e os participantes tenham a oportunidade de falar sobre isso com segurança.

## Definindo os princípios do time

Esta atividade ajuda o time a definir coletivamente e escrever os princípios do time que querem seguir. Ela promove uma conversa

honesta e democrática sobre os princípios do time.

### Passo a passo da atividade

1. Apresente a atividade, acrescentando mais áreas de discussão, se necessário: “Vamos conversar sobre como lidaremos uns com os outros continuamente. Coletivamente, iremos escrever os princípios do time para cada uma destas áreas”:
  - › Estrutura e programação das reuniões;
  - › Método preferido para comunicação;
  - › Funções e disponibilidade;
  - › Processo.
2. Divida o quadro entre o número de áreas identificadas (neste exemplo, quatro).
3. Peça anotações individuais. “Escreva suas anotações sobre os princípios do grupo para cada área identificada. Você deve usar o post-it colorido de acordo com a anotação:
  - › Limitações (cartões com a inicial C).
  - › Forte recomendação (cartões com a inicial R)”.
4. Conversem entre si e façam anotações (considere escrever itens de ação em post-its de outra cor).

### Pequena estrela-do-mar

Esta é uma adaptação de uma atividade conhecida como Estrela-do-mar. A tela da Pequena estrela-do-mar é dividida em três áreas, ao invés de cinco. É uma ótima opção para recolhimentos de dados

e para incentivar a reflexão sobre práticas e o valor que o time vê nelas. Ela ajuda os membros do time a entenderem o valor percebido uns nos outros em tais práticas.

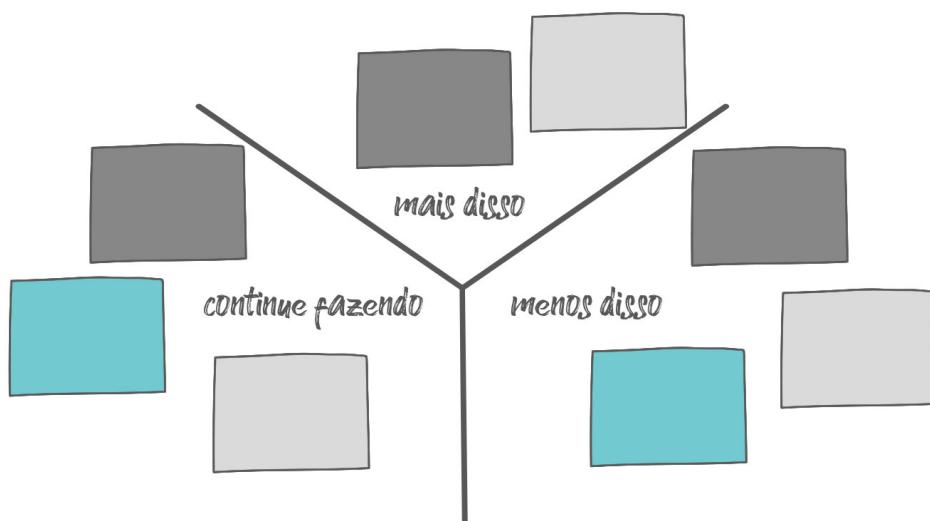
Passo a passo da atividade:

1. Divida a tela em três áreas:

- › Continue fazendo: algo que o time está fazendo bem e você reconhece o valor naquilo.
- › Menos disso: algo que já foi feito, você vê algum valor, mas prefere diminuir um pouco.
- › Mais disso: algo que já foi feito, e você acredita que trará mais valor se feito ainda mais.

2. Peça para os participantes escreverem, individualmente, suas anotações em post-its para cada uma das áreas.

3. Discuta com o grupo acerca dos cartões alocados.



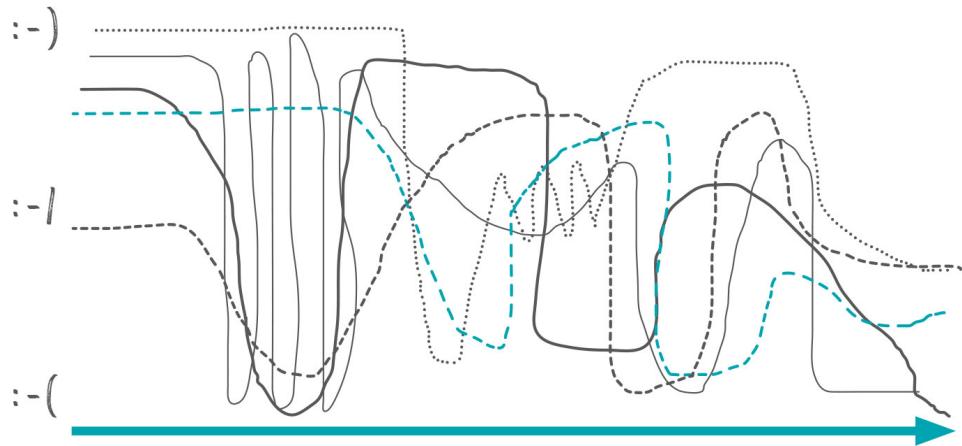
Pequena estrela-do-mar

## Linha do tempo com altos e baixos

Esta atividade também é conhecida como “Penhascos e vales”, *Peaks and valleys* e como “Sismógrafo emocional”. Ela promove uma linguagem visual simples ao compartilhar visões individuais sobre os altos e baixos de uma dada linha do tempo. É um modo eficaz para um grupo grande visualizar e descobrir eventos e suas interconexões.

Passo a passo da atividade:

1. Desenhe uma seta horizontal representando a linha do tempo na parte inferior da tela. Defina o começo e o fim da linha do tempo.
2. Decida sobre o gradiente vertical da tela. Por exemplo, se for o gradiente de felicidade, o topo é o mais feliz e a base é o mais triste.
3. Convide um participante a desenhar sua linha do tempo com picos e vales: “Começando no início da linha do tempo, e mantendo a caneta na tela o tempo todo, por favor, compartilhe seus altos e baixos nela. Você deve fazer isso desenhando a linha e falando sobre ela ao mesmo tempo”.
4. Convide mais participantes para desenharem sua linha do tempo (um de cada vez). Considere usar cores diferentes de caneta, ou padrões diferentes para a linha.



Linha do tempo com altos e baixos

Ao final desta atividade, o grupo todo terá uma visão das linhas do tempo com altos e baixos. É uma boa atividade para os participantes se sentirem mais à vontade para futuras conversas e é também muito útil para correlacionar eventos com o gradiente escolhido e reconhecer contradições – como altos e baixos para o mesmo evento.

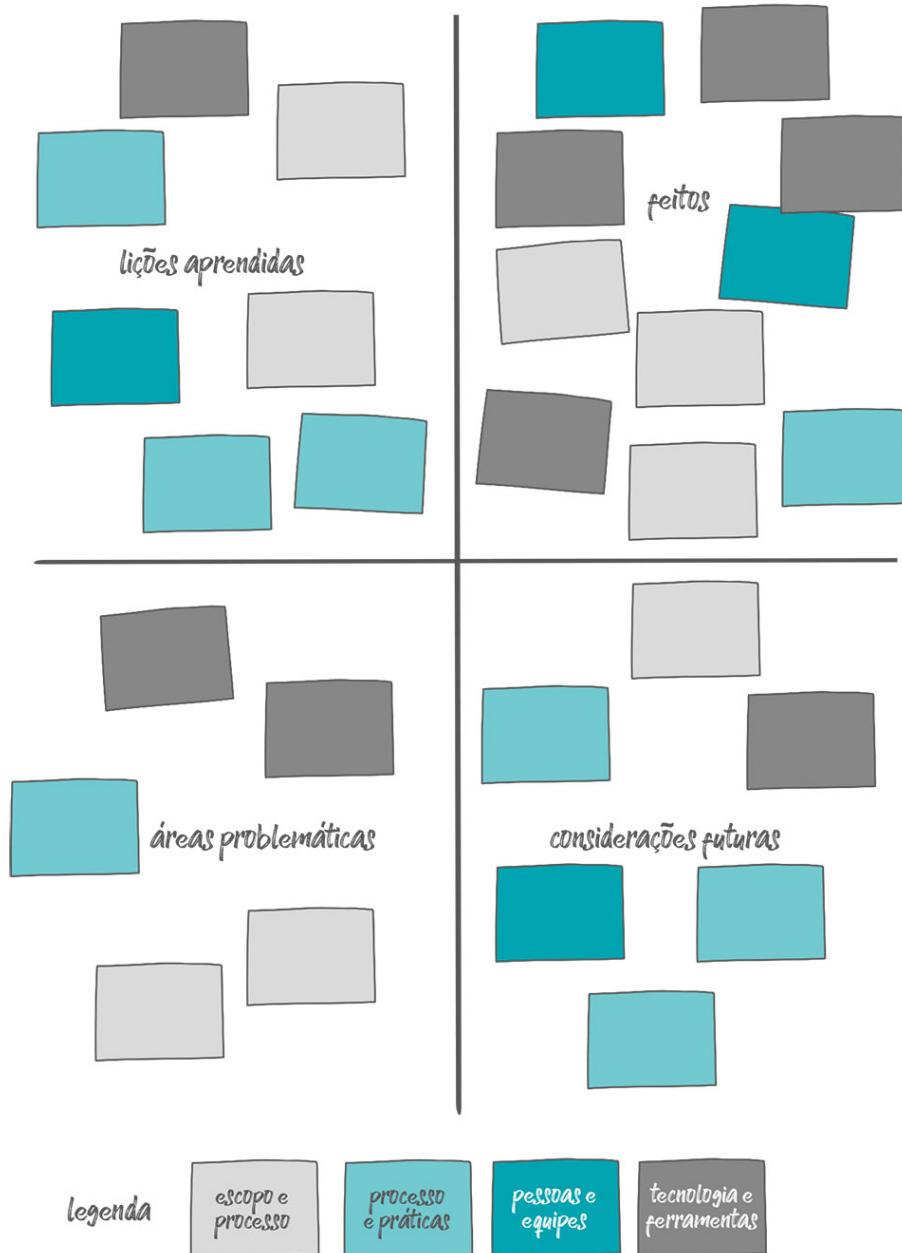
## CLFA: considerações futuras, lições aprendidas, feitos e áreas problemáticas

Este exemplo é uma ótima opção para o pós-projeto/pós-fase. Você deve fazê-la o mais perto possível do fim do projeto/fase – mas não espere muito, ou todo mundo se esquecerá do que aconteceu.

Passo a passo da atividade:

1. Prepare e explique os quadrantes do quadro CLFA:
  - › Considerações futuras: escreva todas as futuras considerações em relação ao projeto/fase.

- › Lições aprendidas: escreva as lições e ideias-chave do projeto/fase.
  - › Feitos: escreva os feitos-chave do projeto/fase.
  - › Áreas problemáticas: escreva as áreas com problemas identificadas durante o projeto/fase especificado.
2. Peça aos participantes que façam suas anotações. Considerem colocar códigos para cores. Por exemplo, você pode usar quatro cores diferentes para cada uma das áreas: Processos e práticas; Tecnologia e ferramentas; Escopo e programação; Pessoas e equipes.
3. Discuta com o grupo e compartilhe os resultados.



## Que-quem-quando

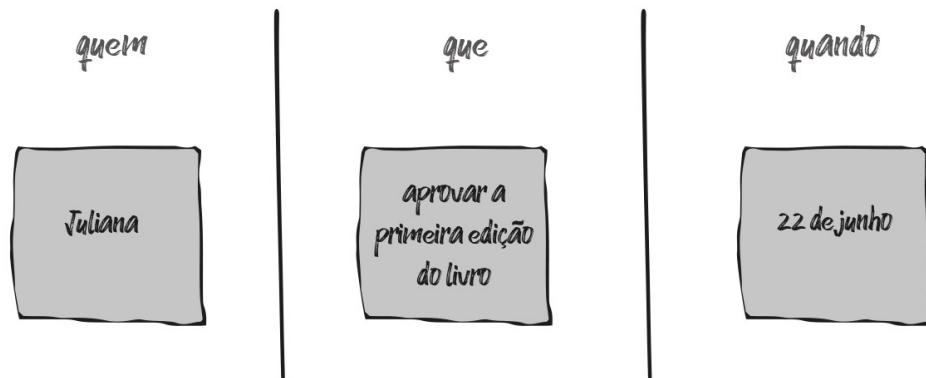
Esta atividade ajuda o grupo a definir compromissos e ações de acompanhamento nas reuniões. Muitas reuniões terminam com uma discussão pouco clara sobre os “próximos passos” ou “itens de

ação". Que-quem-quando ajuda com o comprometimento e o prazo de tarefas.

Passo a passo da atividade:

1. Crie uma estrutura de tabela que descreve que-quem-quando como títulos de coluna.
2. Peça aos participantes que selecionem (da atividade anterior) ou escrevam um post-it com o **que** eles se comprometem, e complementem com **quem** (seu nome) e o **quando** (prazo para demonstrar).

Ao focar a discussão em um formato que-quem-quando, você pode conectar as pessoas com ações claras que elas definem e se comprometem. Isso permite que os participantes tenham clareza sobre seus compromissos e responsabilidades, tornando visível para todo o grupo **quem** vai fazer o **que** até **quando**. ANEXOS ›



## Anotação pessoal

Anotação pessoal é uma atividade de check-out muito simples. Nela a pessoa se concentra em si mesmo, sem compartilhar o compromisso com outras pessoas. Portanto, é uma boa opção após conversas em grupo sem itens de ação para o time.

Passo a passo da atividade:

1. Peça aos participantes que peguem uma caneta e um post-it.
2. Peça para eles refletirem sobre a discussão que acabou de terminar.
3. Peça a cada pessoa que escreva uma nota para si mesma sobre algo que deseja fazer e coloque em seu laptop como um lembrete para o futuro.

Embora simples, escrever uma nota e depois lê-la mais tarde é uma maneira poderosa de fazer uma pessoa pensar e então ponderar (novamente) sobre algo. As reflexões são muito pessoais e nem tudo precisa ser uma discussão ou itens de ação em grupo. Consideramos a nota pessoal muito poderosa para fomentar pensamentos e reflexões individuais.



**ONDE  
ENCONTRAR  
MAIS**

**N**este livro compartilhamos erros e acertos na transformação de um time ágil. Alguns desses acertos se deram depois de criar e utilizar alguns materiais, como os checklists.

Na página do livro você terá acesso a alguns dos materiais apresentados aqui (em formato pdf, para facilitar o download e a impressão).

[www.caroli.org/livro/sprint-a-sprint](http://www.caroli.org/livro/sprint-a-sprint).





**PARTICIPE DO  
TREINAMENTO  
SPRINT A  
SPRINT**

**A**tendendo a pedidos, passamos a realizar treinamentos complementares ao livro. O treinamento Sprint a Sprint aborda os diferentes estágios de um time e apresenta práticas para que ele possa ser bem-sucedido em alcançar seus objetivos. Confira a agenda de treinamentos em:

[www.caroli.org/treinamento/sprint-a-sprint](http://www.caroli.org/treinamento/sprint-a-sprint).





# SOBRE A EDITORAS CAROLI

**P**ara leitores e autores que buscam e compartilham conhecimento de ágil, a Editora Caroli é uma editora-butique – todos os livros são escritos, lidos, editados e/ou revisados por Paulo Caroli, que auxilia na produção, divulgação e distribuição de livros e e-books. Diferentemente das editoras tradicionais, a Editora Caroli dá acesso ao conhecimento na sua essência, disponibilizando o texto das novas obras via e-books gratuitos, além de apoiar eventos e entidades de ensino, presenteando-os com livros impressos.

Em [www.caroli.org](http://www.caroli.org) você encontra este e outros conteúdos de qualidade. Aproveite, pois os livros e e-books em WIP estão disponíveis gratuitamente.

## **WIP (WRITING IN PROGRESS)**

A Editora Caroli apresenta uma nova proposta de trabalho, aproximando os autores dos seus leitores desde o início da geração do conteúdo. Por que esperar o autor terminar de escrever para saber se o conteúdo é bom? No mundo atual, isso não faz mais sentido. Por isso, a Editora Caroli promove o compartilhamento (gratuito sempre que possível) do WIP por meio dos formatos e-book (pdf, mobi e epub). Dessa forma, leitores têm acesso rápido a novas ideias e podem fazer parte da evolução da obra. Para os autores, é uma forma efetiva de feedback e motivação para a geração do conteúdo.



**DEIXE A SUA  
AVALIAÇÃO**

**S**e você você chegou aqui, é porque você acabou de ler o livro. Parabéns! Muito obrigado por compartilhar e por ler o conteúdo que compartilhamos aqui.

Autores vivem e morrem pelas avaliações que recebem dos leitores. Então, se você gostou deste livro, ficaremos extremamente felizes em receber uma avaliação 5 estrelas pela sua leitura!

Caso você tenha alguma crítica, por favor envie um e-mail para [paulo@caroli.org](mailto:paulo@caroli.org) e me diga o que melhorar. Assim conseguimos adicionar conteúdo e corrigir eventuais erros para que a sua experiência de leitura seja cada vez melhor.

Muito obrigado!

[www.caroli.org/avale-o-livro-sprint-a-sprint/](http://www.caroli.org/avale-o-livro-sprint-a-sprint/).

