

PRODUCT BACKLOG BUILDING



Concepção de um product
backlog efetivo



Fábio Aguiar
Paulo Caroli

Editora
Caroli
caroli.org

The logo consists of a black circle with a white "C" shape inside it, positioned to the left of the publisher's name.

Product Backlog Building

Concepção de um product backlog
efetivo

Paulo Caroli e Fabio Aguiar

Esse livro está à venda em <http://leanpub.com/pbb>

Essa versão foi publicada em 2019-12-17



Esse é um livro [Leanpub](#). A Leanpub dá poderes aos autores e editores a partir do processo de Publicação Lean. [Publicação Lean](#) é a ação de publicar um ebook em desenvolvimento com ferramentas leves e muitas iterações para conseguir feedbacks dos leitores, pivotar até que você tenha o livro ideal e então conseguir tração.

© 2014 - 2019 Paulo Caroli e Fabio Aguiar

Conteúdo

Disclaimer	i
Sobre Fabio Aguiar	ii
Sobre Paulo Caroli	iii
Sobre a Editora Caroli	iv
Motivação do PBB	1
Um Overview do Scrum	2
Os eventos do Scrum	2
O time Scrum	5
Trabalho em Equipe, Alinhamento Cadenciado e Clareza	6
PBB e Scrum	7
PBB, a prática de construção de backlog	7
Backlog, a única fonte de requisitos	8
Product Owner, o facilitador do PBB	8
Product Backlog Building	13
Benefícios do PBB	13
Estrutura de granularidade do PBB	13
O PBB Canvas e o fluxo de construção do Backlog	14
Contextualize o produto	15
Descreva as Personas	16
Entenda as Features	18

CONTEÚDO

Mapeie os passos de uma Feature - Steps Maps	20
Descreva a ação de cada PBI	24
Por fim, o Product Backlog	25
Organização visual do PBB	26
CORG – a técnica de priorização do PBB	27
Estimativa pelo tamanho da camisa	32
PBB e História de Usuário	34
História de Usuário	34
Critério de Aceite	35
Tarefas	36
Um exemplo	37
Escrevendo Histórias de Usuário com o PBB	38
Modelo ARO	41
Backlog Ready – definindo a qualidade de um PBI	43
Dando um TAPAs no PBB	45
O ciclo HEVA	47
PBB e a Lean Inception	49
PBB, o passo seguinte da Lean Inception	49

Disclaimer

Estamos tratando esse E-Book como um MVP. Se já tem algo minimamente viável para passar a ideia, o conceito, então está disponível.

Por isso neste E-Book você vai encontrar erros de ortografia, imagens em versões iniciais, e texto por escrever.

Mas como um bom MVP, estamos muito interessados no seu feedback.

Abraços,

Paulo Caroli e Fabio Aguiar

Sobre Fabio Aguiar

Product Agility Specialist na Accenture|SolutionsIQ, fez parte da Diretoria da Agile Alliance Brazil e foi Coordenador Geral da Agile Brazil 2017. Com formação de Bacharel em Sistemas de Informação com especialização em Engenharia em Processo de Software. Faz parte do time de instrutores do PM3 - o primeiro curso online para formar Product Managers no Brasil, do corpo docente da FIAP e ICAgile Authorized Instructor - criador do primeiro treinamento Certified Product Ownership no Brasil.



Fabio Aguiar

Acompanhe Fabio Aguiar no seu site e nas redes sociais:



www.fabiogr.com



twitter.com/fabyogr



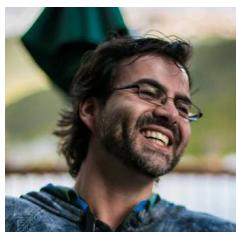
facebook.com/fabyogr



linkedin.com/in/fabiogr

Sobre Paulo Caroli

Consultor principal da Thoughtworks Brasil e cofundador da AgileBrazil, Paulo Caroli possui mais de vinte anos de experiência em gestão e desenvolvimento de software, com passagem em varias corporações no Brasil, Índia e EUA . Em 2000, conheceu o Extreme Programming e, desde então, tem mantido seu foco em processos e práticas de gestão e desenvolvimento ágil. Ingressou na ThoughtWorks em 2006 e ocupou os cargos de agile coach, treinador, e Gerente de Projetos. Possui os títulos de Bacharel em Ciência da Computação e MS em Engenharia de Software, ambos da PUC -Rio. Autor dos Livros ”Direto Ao Ponto; Criando Produtos de forma enxuta” , e “Fun Retrospectives; activities and ideas for making agile retrospectives more engaging”.



Paulo Caroli

Acompanhe Paulo Caroli no seu site e nas redes sociais:

-  www.caroli.org
-  twitter.com/paulocaroli
-  facebook.com/paulo.caroli
-  linkedin.com/in/paulocaroli

Sobre a Editora Caroli

A Editora Caroli nasceu em 2017, quando Paulo Caroli decidiu realizar todos os passos desde escrever um livro a impressão e venda do mesmo. Este livro é O Mistério do Colégio Alipus.

Paulo Caroli segue muito do seu aprendizado no Vale do Silício para a criação de conteúdo e publicação de livros. Com um estilo Lean StartUp (construir fazendo) nasceu a Editora Caroli.

Caroli constituiu uma editora para publicar seus livros e para ajudar a publicar livros de amigos.

WIP (*Writing In Progress*)

A Editora Caroli apresenta uma nova proposta de trabalho, aproximando autores dos seus leitores desde o início da geração do conteúdo. Porque esperar o autor terminar de escrever para ver se o conteúdo é bom? No mundo atual isso não faz mais sentido. Então a Editora Caroli promove o compartilhamento (gratuito sempre que possível) do WIP (*Writing In Progress*) através dos formatos eBook (pdf, mobi e epub). Desta forma, leitores tem acesso rápido as novas ideias, e podem fazer parte da evolução da obra. Para os autores, esse é uma forma efetiva de feedback e motivação para a geração de conteúdo.

Este e outros eBooks estarão disponíveis no site <caroli.org>

Motivação do PBB

Scrum é o framework ágil mais utilizado para a construção de produtos digitais. Scrum tem como ponto de partida o Product Backlog, entretanto não há no mercado respostas claras para as seguintes perguntas:

- Como chegar ao Backlog?
- Como construir algo que tenha valor?
- Como encontrar a real necessidade do cliente?
- Como definir o que é prioridade para o cliente no primeiro momento?

Tentando responder essas perguntas, depois de diversas experiências em vários clientes, Fábio Aguiar documentou a forma mais efetiva de criar o Product backlog, e passou a compartilhar o PBB -- abreviação de Product Backlog Building.

Este E-book compartilha o passo a passo para a utilização efetiva do PBB.

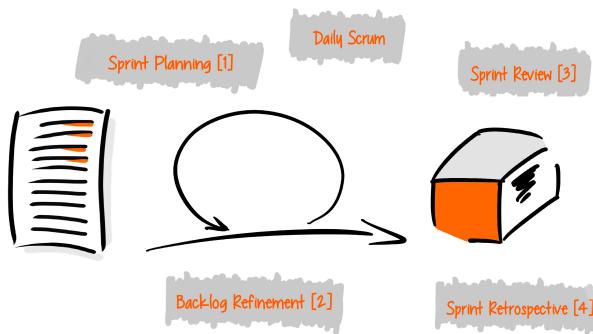
Um Overview do Scrum

Scrum é um framework Ágil para gerenciar projetos/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 chamados de Sprints, que duram, tipicamente, de uma a quatro semanas.

Os eventos do Scrum

O modelo Scrum sugere que cada Sprint comece com uma breve reunião de planejamento, e termine com um evento de revisão do trabalho realizado na Sprint. Estes são os princípios do gerenciamento de projetos Scrum: ciclos curtos e cadenciados com eventos de alinhamento, acompanhamento do trabalho, e melhoria do time.



os eventos do Scrum

Além dos eventos de (1) planejamento e (3) revisão, Scrum sugere mais dois eventos que acontecem a cada Sprint. Essa são: (4) retrospectiva, o evento que promove o momento kaizen, onde o time busca a melhoria contínua em relação ao processo, entrega, e interação entre as pessoas; e (2) refinamento, o evento onde o backlog do produto é revisitado buscando o entendimento dos próximos requisitos, candidatos ao próximo Sprint.

Na Sprint Planning, o time decide os itens para o Sprint Backlog. Esse itens foram previamente refinados na reunião de Refinamento. Diariamente, na Daily Scrum, os participantes do time atualizam e conversam sobre o progresso desses itens. Na Sprint Review o time faz uma revisão do estado de todos os itens do Sprint Backlog. Na reunião de Retrospectiva, o time reflete sobre a Sprint, na busca de melhoria contínua.

O time Scrum busca trabalhar em níveis elevados de rendimento. Muito disso é alcançado pelo entrosamento do time, pelo alinha-

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

A Sprint Planning, ou evento de planejamento do Scrum é uma reunião de planejamento que acontece no início de cada Sprint. Nela, o time esclarece qual a meta da Sprint e os itens do Sprint Backlog. Primeiro todos conversam sobre o que desejam alcançar ao final da próxima Sprint. Em seguida, identificam os itens relacionados que conseguem terminar no período da Sprint. Esses itens foram previamente detalhados na reunião de refinamento.

A Sprint é uma iteração, um ciclo de trabalho repetitivo no qual o time Scrum passa por todos os eventos do Scrum: planejamento, revisão e retrospectiva. A Sprint promove uma cadência de uma a quatro semanas, dependendo da preferência do time. Duas semanas é o tamanho de Sprint mais adotado por times de desenvolvimento de Software.

Diariamente o time realiza uma reunião para verificar o andamento das tarefas de trabalho. Este evento é a Daily Sprint. Nesta, basicamente todos os membros do time ficam de pé (para que a reunião não demore demais) e respondem a 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 e o que está impedindo o progresso do meu trabalho. Hoje essas perguntas mudaram, ou melhor, acrescentou a cada pergunta uma palavra bastante importante, a “META”, levando o time a refletir diariamente se está entregando valor, leva o time a outro patamar, não somente a entregar, mas gerar percepção de valor em cada entrega.

Reunião realizada ao final da Sprint, a Sprint Review tem como objetivo de apresentar o estado da arte do produto sendo criado, rever o progresso do trabalho do time, e comparar com o planejamento apresentado na Sprint Planning. Na reunião de revisão do sprint, o time scrum irá rever o trabalho realizado durante o sprint. Neste momento, o time deve revisar e apresentar o trabalho realizado, o

time procura obter feedback da entrega.

O evento de retrospectiva Scrum é um momento para o time refletir sobre como estão trabalhando, e buscar maneiras de melhorar. Tipicamente, o time Scrum realiza uma retrospectiva por Sprint.

No mundo Ágil Scrum, evitamos descrições completas e detalhadas sobre como tudo deverá ser feito na Sprint. O time de desenvolvimento, quando for o momento, é que decide como fazer. Isso porque o time vai saber a melhor forma de resolver o problema em questão.

O time Scrum

Scrum fomenta um time multifuncional e com auto-organização. A eficiência do time depende da capacidade dos membros trabalharem em conjunto, e 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, e essas são questões decididas pelo time como um todo.

Os times Scrum são apoiados por dois papéis específicos. O primeiro é um Scrum Master, alguém experiente com o framework que pode ajudar o time a usar o processo de Scrum para alcançar seus objetivos de alto nível. Os melhores Scrum Masters são aquelas pessoas que sentem mais satisfação de facilitar o sucesso dos outros do que seus próprios. O Scrum Master deve se sentir confortável e seguro com o framework a ponto de dar todo controle em relação ao produto para o Product Owner(PO), e todo controle em relação ao desenvolvimento a seu time.

O segundo papel específico é o Product Owner (PO). O PO representa o negócio, os clientes ou usuários, e orienta o time para a construção do produto certo. O PO deve liderar o esforço de desenvolvimento, através de esclarecimentos e priorizações sobre o trabalho.

Tipicamente, o PO trabalha com o Product Backlog, a lista mestre dos requisitos do produto a ser criado. É sua função priorizar o backlog com base no valor do negócio, e no alinhamento entre as partes interessadas, tanto internas quanto externas o time Scrum. Como tal, o PO deve estar disponível para o time para responder a perguntas e direcionar o time a cada momento ou indagação.

Esta combinação de autoridade e disponibilidade para o time de desenvolvimento faz com que o PO seja peça chave do framework. Scrum valoriza a auto-organização e autonomia do time; portanto, o PO deve respeitar o direcionamento e a capacidade do time para criar o seu próprio plano de ação.

Trabalho em Equipe, Alinhamento Cadenciado e Clareza

o time Scrum – o Scrum Master, o PO e todos membros do time (com suas variadas formações – participam ativamente de todos os eventos com um alto nível de autonomia, transparéncia e comprometimento. Na reunião de planejamento da Sprint, o time decide o Sprint backlog, o qual é acompanhado diariamente e reavaliado na reunião de revisão da Sprint. Através 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 evento.

PBB e Scrum

Scrum é um framework para desenvolver, entregar e manter produtos complexos e consiste de times Scrum com papéis, eventos, regras e artefatos. No capítulo anterior, exploramos bastante sobre os papéis, eventos e regras, mas pouco sobre os artefatos do Scrum.

O principal artefato no Scrum, é o backlog. Entretanto o framework Scrum não defini como construir o backlog. O PBB – Product Backlog Building – complementa o Scrum, auxiliando os times Scrum na elaboração e criação de um Product Backlog efetivo.



PBB, a prática de construção de backlog

O PBB – Product Backlog Building – é um método e um canvas para a elaboração e a criação de um Product Backlog. O PBB esclarece e prioriza as User Stories, os itens a serem adicionados ao backlog do time Scrum.

O PBB tem como principal objetivo ajudar na construção do Product Backlog. Para tanto, o método PBB sugere que o time

Scrum preencha – todos juntos– o canvas PBB. Como resultado, os participantes alcançam um entendimento comum do produto, e obtém o backlog para começar a trabalhar, de forma ágil e eficaz.

Backlog, a única fonte de requisitos

Segundo os criadores do Scrum, Ken Schwaber e Jeff Sutherland [“ScrumGuide”¹](#), o Backlog do Produto é uma lista ordenada de tudo que é necessário no produto. É a única fonte dos requisitos para qualquer mudança a ser feita no produto. O Product Owner é responsável pelo Backlog do Produto, incluindo seu conteúdo, disponibilidade e ordenação.

A definição de Backlog no Guia do Scrum se resume em “Backlog: a única fonte de requisitos”. Mas muitas equipes Scrum trabalham com cenários complexos, onde o grau de incerteza é alto, e falta de conhecimento do que precisa entregar.

“Nem o cliente sabe o que ele quer!” Essa é uma expressão bastante usada no mercado. E, infelizmente, é a realidade do desenvolvimento de produto.

O Backlog pode ser única fonte de requisitos. Mas, antes de “beber” desta única fonte, é necessário alimentá-la, ou seja, compreender e priorizar os itens que devem fazer parte desse backlog. O PBB e este livro te ajudam com isso.

Product Owner, o facilitador do PBB

O Product Owner (PO), ou dono do produto, é responsável por maximizar valor do produto abraçando um processo contínuo de descoberta de produtos, construído em torno de aprendizado e

¹KSchwaber&JSutherland, TheScrumGuide, 2017. Acesso em Junho 2019. Disponível em <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>

experimentação. Product Owner é o dono do backlog do produto, responsável por gerenciar o backlog e dá todo o direcionamento de trabalho ao time.

Como dono do backlog, e como atividade principal é criar o backlog, o Product Owner é o facilitador do PBB. Claro, dependendo do contexto, o Scrum Master (que possui a responsabilidade de garantir que o framework Scrum seja entendido e aplicado, onde seu trabalho parte do scrum através de práticas emergentes para construir um processo aderente a cultura do time) como facilitador pode assumir esse papel, assim como o Agile Coach ou/e Product Manager.



O Product Owner usa o Backlog do Produto para se adaptar a requisitos emergentes, feedback de clientes e mudanças no mercado.

O Product Owner é responsável em criar, priorizar, fatiar e manter o product backlog, ou seja, é o dono do backlog (única fonte de requisitos do produto). O Scrum entrega produto através de ciclos curtos que se chamam sprints, com isso, o Product Owner determina o que é prioridade para o cliente no primeiro momento, buscando sempre

a satisfação do cliente, e antecipa o poder de entregas (lembrando que não é somente entregar, também gerar percepção de valor em cada entrega). Precisa entender que o cliente não está comprando o produto, mas sim investindo no produto, certo que em um curto espaço de tempo terá esse retorno. Com o foco em fazer o cliente tirar vantagens competitivas de mercado, precisa estar aberto a aceitar mudanças. Quem nos disse que um plano não pode mudar? a única certeza que temos em desenvolvimento de produto é que ele vai mudar, mas com os métodos ágeis isso vai acontecer o mais cedo possível, com isso, existe a necessidade de adaptação para responder à mudança sem traumas, lembrando que mudanças sempre são bem vindas. Os requisitos podem mudar a qualquer momento, o que é normal e aceitável, é necessário gerenciá-los conforme suas prioridades. Se os seus requisitos não mudam, tem alguma coisa de errado.

O Product Owner precisa decidir quais funcionalidades são prioridade no primeiro momento, fatiar pequenos incrementos com valor para o cliente. A entrega constante gera a percepção de rápido, pois o objetivo é entregar o mínimo o mais cedo possível para o cliente e gerar o feedback antecipado, gerando assertividade, ou seja, maior frequência de planejamento e maior qualidade das entregas. Com o conceito MVP, tudo isso fica mais claro, pois o objetivo é escolher o mínimo do produto necessário, colocar na mão do cliente e o cliente determina se é viável para evidenciar o aprendizado emergente do produto. Trabalha na concepção do produto para entender o todo e priorizar o que precisa entregar. Nesse momento os detalhes não são importantes (detalhes são refinados durante o tempo) e o esforço inicial não deve chegar a duas semanas, apenas dias. Está sempre engajado e disponível para o time de desenvolvimento, respondendo a quaisquer dúvidas relacionadas ao produto. Durante sessões facilitadas, é importante o envolvimento dos desenvolvedores com as partes interessadas, facilitando o entendimento do negócio. Tem que está muito próximo do time de desenvolvimento e é importante que todos saibam que ele faz parte do time (Scrum). O

Product Owner deve ter o conhecimento necessário sobre o negócio assim como ter toda confiança para que possa fazer seu trabalho, precisa priorizar o que deve ser feito, tem que ter poder de decisão e, principalmente, saber dizer “NÃO” para os desejos do cliente quando não está alinhado com a visão do produto.

O Product Owner é responsável em manter o Product Backlog priorizado e com valor, o time de desenvolvimento em entregar o que está se comprometendo em cada Sprint. Com isso, há uma aproximação forte através de conversas ativas e sessões facilitadas, colaborando constantemente para entender os itens do backlog que podem ser apresentados por Histórias de Usuário. Valoriza os momentos de conversas cara a cara para dar o entendido do produto do que está representado através das Histórias de Usuário, lembrando que cada história é apenas um “lembrete” da necessidade do cliente, ou seja, fornece um flash para a comunicação, representa os requisitos (necessidades) mais do que documentá-los. Os melhores documentos nos ajudam a recordar nossas conversas, eles não a substituem. Para provocar a colaboração de todas as partes interessadas, pode-se adotar ferramenta simples, como cartões, post-it, papéis, flip-chart, canvas, paredes ou quadro em branco, certamente incentivará a comunicação ativa de todos interessados no produto.

O Product Owner entrega o que precisa ser desenvolvido para o time, ajudá-lo no entendimento do produto para que o time possa fatiar e estimar. O trabalho de refinamento dos requisitos é um esforço contínuo durante todo o curso de desenvolvimento. Contribui constantemente ajudando o time de desenvolvimento a entender a visão e roteiro do produto, seus objetivos de negócio e as principais features, deixando claro todo contexto de negócio, com isso o time facilmente consegue identificar riscos técnicos, habilitadores arquiteturais, exploração e infra-estrutura, buscando sempre a excelência técnica.

A principal ferramenta do Product Owner é dizer “NÃO” para um item entrar no backlog. Todo o esforço de refinamento do Product

Owner está nos itens de maior prioridade para o cliente, ou seja, saber o que é importante para o produto e saber o momento certo de parar, neste cenário – “menos é mais”. É o responsável pelo backlog e definir sua prioridade, mas em uma entrega todos têm toda condição, habilidades e conhecimento necessário para auto organizar e se comprometer com o trabalho que precisa ser feito. E todo incremento do produto representa uma entrega funcional de valor do produto para o cliente. Sendo responsável por demonstrar esse progresso de valor para todos interessados. O Product Owner reconhece que o seu papel precisa se adaptar e melhorar a medida que o produto evolui e está bem alinhado com todo o processo. Reflexões frequentes conduzem a melhoria contínua e a forma como o time trabalha para maximizar o valor para seus clientes.

Dica: Importante o Product Owner ter em mente toda essa reflexão quando pensar em construir um backlog.

Product Backlog Building

O PBB – Product Backlog Building – é a elaboração e criação de um Product Backlog efetivo e colaborativo, que define as User Stories e o backlog dos times. O PBB Canvas é um quadro visual para a facilitação do PBB.

Benefícios do PBB

- Ajudar na construção de um BACKLOG de forma efetiva e colaborativa.
- Construir um entendimento compartilhado do negócio do cliente, facilitando a descoberta e compreensão do produto.
- Buscar uma maneira de descrever a experiência do usuário com o produto.
- Facilitar a descoberta e escrita de User Stories.
- Priorizar por alinhamento de expectativas e metas.
- Ter como resultado um Product Backlog totalmente alinhado com o valor de negócio do cliente.

Estrutura de granularidade do PBB

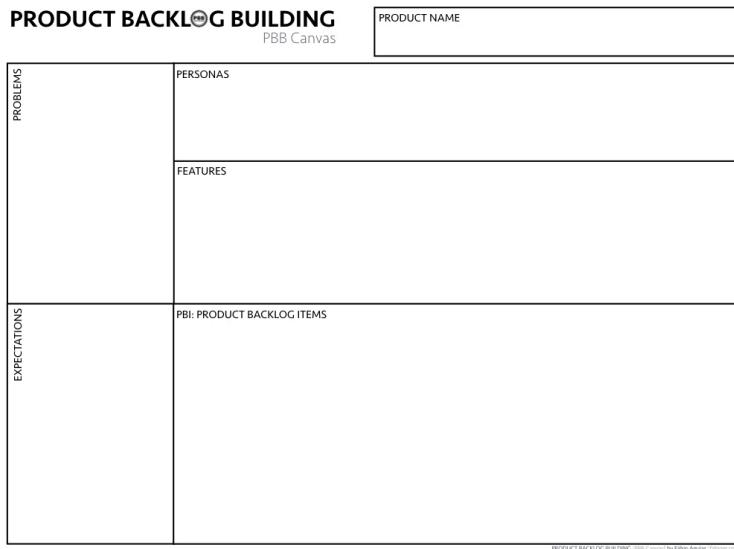
Segue abaixo uma imagem que demonstra a granularidade do trabalho para alguns dos principais métodos e frameworks que influenciam os itens de trabalho de um Product Backlog. Ela serve como referência e alinhamento sobre a forma como o PBB se relaciona com Features e User Stories.



A História de Usuário se divide em três partes, o Tema que é a maior granularidade, é uma coleção ou agrupamento de histórias de usuários no mesmo contexto, o épico que é uma história de usuário abrangente, geralmente quando o time diz que é uma história grande, dificilmente consegue entregar dentro de uma sprint, e as histórias de usuários são descrições curtas e simples de uma necessidade da perspectiva de um usuário ou cliente, é apenas uma representação para fornecer um flash para comunicação.

O PBB Canvas e o fluxo de construção do Backlog

PBB é alcançado através do PBB Canvas, o qual segue um fluxo simples e de fácil compreensão. O canvas facilita o entendimento da necessidade do cliente e a construção do backlog do produto.



o canvas PBB

Segue o fluxo de construção do Backlog: 1. Contextualize o Produto
2. Descreva as Personas 3. Entenda as Features 4. Mapeie os passos
de uma Feature 5. Identifique os itens do backlog

Contextualize o produto

No primeiro momento, é importante esclarecer e entender o que é o produto, entender os problemas e as dores daquilo que precisa ser desenvolvido, de forma a entender as expectativas e clarificar os objetivos desejados.

PRODUCT NAME: Identifique o produto que será construído. Imagine que esse produto estivesse numa caixa; que nome estaria escrito na caixa?

PROBLEMS: Identifique e compreenda o Estado Atual pontuando um conjunto de problemas e dores no contexto em questão. Nesta

etapa, as pessoas de produto devem buscar, de forma colaborativa, a compreensão do estado atual, através da descrição dos problemas. É essencial conhecer os problemas antes de elaborar uma solução.

EXPECTATIONS: Identifique o Estado Desejado, alinhando as expectativas relacionadas aos problemas do estado atual. Evite descrever a solução em detalhes, de como alcançaremos o Estado Desejado; ao invés, liste as expectativas (tipicamente relacionadas a eliminação ou mudanças em relação aos problemas do Estado Atual).

PRODUCT BACKLOG BUILDING		PRODUCT NAME
		Palestras Coletivas
PROBLEMS	PERSONAS	
<ul style="list-style-type: none"> Controle somente no Google Drive Descentralização de temas disponíveis Falta organização Não sabemos quem assistiu 	<ul style="list-style-type: none"> Perda de Temas Processo não é transparente Base de histórico de palestras 	
FEATURES		
EXPECTATIONS	PBI: PRODUCT BACKLOG ITEMS	
<ul style="list-style-type: none"> Portfólio de Palestras Centralização de temas de palestras Programação de grande com palestras cadastradas 	<ul style="list-style-type: none"> Qualquer pessoa monitorar evento Divulgar palestras/ eventos Facilitar a organização de Eventos 	

Exemplo - Problemas e Expectativas

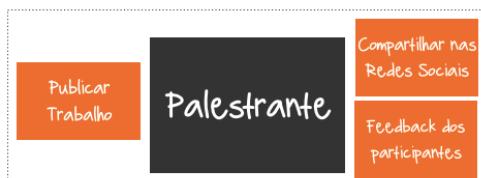
Descreva as Personas

Uma persona representa um usuário do sistema, descrevendo não só o papel, mas também suas necessidades específicas. Isto cria uma representação realística de usuários, auxiliando o time a descrever

funcionalidades do ponto de vista de quem interagirá com o produto final.



PERSONAS: Identifique os usuários, papéis e responsáveis envolvidos no produto; descreva o que fazem e o que esperam do produto.



Exemplo - Persona

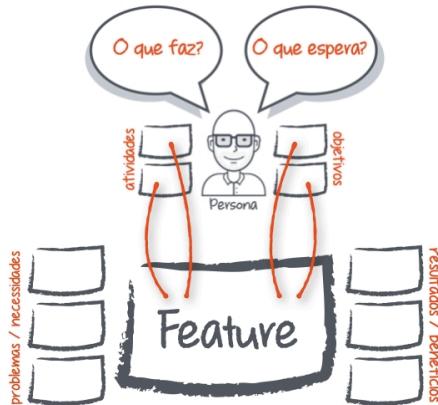
Dica: Represente no post-it o perfil do persona; esse perfil será usado para compor a história de usuário.

Entenda as Features

Feature é a descrição de uma ação ou interação de um usuário com o produto. Por exemplo: imprimir nota fiscal, consultar extrato detalhado, e convidar amigos do facebook. A descrição de uma feature deve ser o mais simples possível. O usuário está tentando fazer uma coisa. O produto dever ter uma feature para isso. Que feature é essa?

Exemplo de feature: Realizar Compra Online

Realize as seguintes perguntas para cada uma das personas identificadas: O que tal persona faz? e o que tal persona espera? Tipicamente, as respostas descrevem ações ou interações das personas com o produto: as features.



FEATURES: Identifique as funcionalidades que cada persona realiza no produto, organizado na sequência de uso (da esquerda para a direita). Descreva cada feature com uma breve descrição; na sequência, descreva os “Problemas” e os “Benefícios” relacionados a mesma. Anote “Problemas” e os “Benefícios” em post-it menores e posicione do lado esquerdo e do lado direito da post-it com a descrição da feature.



Dica: Limite o número total de features a um máximo de 10; mais do que isso cria um inventário, não um backlog. Se houver um número considerável de features, aplique um método de priorização para selecionar as mais prioritárias.

Features versus User Stories: As features são tipicamente descritas em um nível mais elevado do que as histórias do usuário. Antes de começar a trabalhar em uma feature, a mesma deve ser analisada e detalhada em suas respectivas histórias de usuário. No Canvas PBB, primeiro identificamos, entendemos e priorizamos as features, para depois detalhá-las em User Stories (que acontece num passo seguinte do Canvas PBB)

Mapeie os passos de uma Feature - Steps Maps

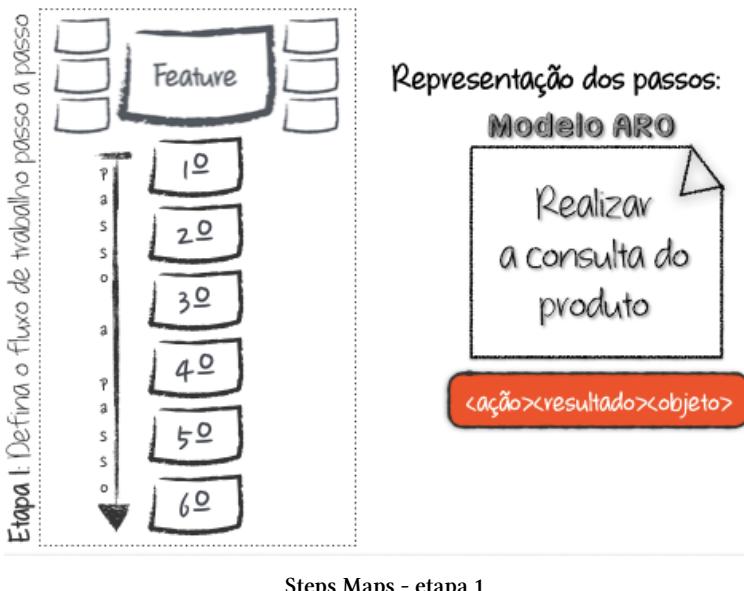
Steps Maps é uma técnica que ajuda a quebrar uma feature em pequenos passos. Cada passo desse será um PBI – Product Backlog Item.

PBIs são elementos que compõem o Product Backlog. Idealmente, os PBIs refletem as necessidades do cliente ou partes interessadas, que podem variar de especificações e requisitos, sendo representado por casos de uso, épicos, histórias de usuário ou até mesmo bugs ou tarefas.

A quebra de feature é feita através do Steps Maps, que representam um mapeamento dos itens de trabalho de uma feature.

Qual o primeiro item de trabalho (ou passo) para esta feature?. Anote em um post-it. Em seguida, pergunte sobre o segundo. Anote em outro post-it. Em seguida, o terceiro e assim por diante, até completar os itens de trabalho da feature. Note que cada item de trabalho representa um passo para a construção da feature.

Por exemplo, considere a feature de exemplo: Fazer compra on-line. Qual é o primeiro passo desta feature? Fazer uma consulta sobre o produto; qual o segundo? fazer a seleção do produto; então para fazer o pagamento do produto e assim por diante. Estes são os Steps Maps da feature.



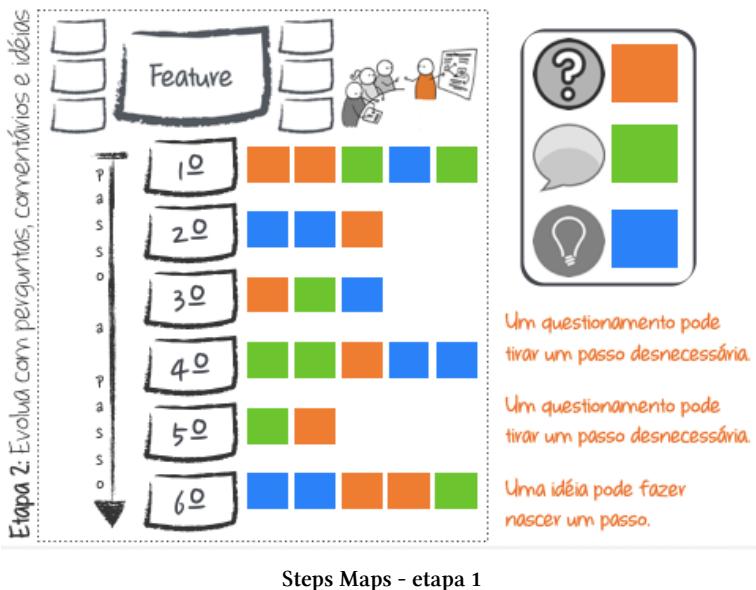
Steps Maps - etapa 1

Na sequência, complemente cada passo da feature com perguntas, comentários e ideias. Perceba que primeiro identificamos “como é” cada passo; para agora complementarmos os passos sobre a perspectiva de “como será”, através de uma sessão colaborativa, via perguntas, comentários e ideias.

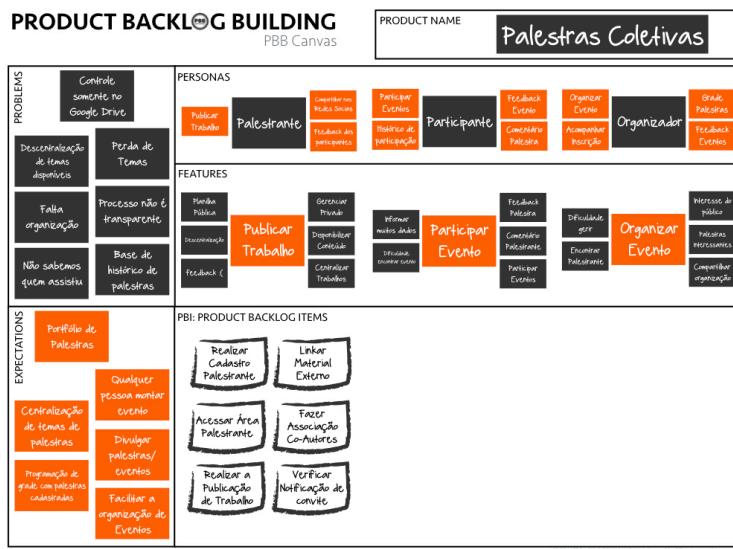
Peça que todos escrevam, individualmente, os questionamentos, comentários ou/e ideias para cada passo. Peça que façam uma anotação por post-it. Sugestão: Use cores distintas para questionamentos, comentários e ideias. Por exemplo: post-it nas cores laranja para questionamentos, verde para comentários e azul para ideias.

Um questionamento pode eliminar um passo desnecessário, assim como, um comentário pode melhorar um passo útil e uma ideia pode fazer nascer um passo novo. Sugiro realizar esta atividade como descrito acima, para evitar que alguém com algum cargo de maior influencia domine a conversa e as decisões. Se esse não for o caso e o grupo estiver colaborando bem, considere fazer a mesma atividade de forma verbal, com descrevendo juntos: as perguntas,

os comentários e as ideias.



Ao final, cada passo no fluxo de trabalho representará um item do backlog, um PBI.



Exemplo - Steps Maps: Mapear os passos de uma Feature

Note que ao final da aplicação do Steps Maps, temos os itens do backlog – PBIs, de uma determinada funcionalidade.

Um Step Map não é uma User Journey. Uma User Journey descreve o passo a passo de um usuário até alcançar um objetivo. Enquanto um Step Map descreve o passo a passo do fluxo de trabalho, do que deve ser construído para completar uma feature. Um Step Map e uma User Journey serão similares quando o passo a passo de ambas refletirem, na mesma ordem: o que o usuário faz e o que precisa ser construído para a feature.



Provoque bastante a colaboração: Um questionamento pode eliminar um passo desnecessário, um comentário pode melhorar um passo útil e uma ideia pode fazer nascer um passo novo.

Descreva a ação de cada PBI

Cada PBI deve representar uma ação de algum usuário no produto. Essas ações serão convertidas em PBIs, os itens do backlog. Essas ações possuem uma descrição textual para: (1) dar contexto e (2) identificar unicamente o item. Por exemplo: “Login básico”, “Consultar partidas próximas a um endereço informado”.

Os exemplos listados trazem duas formas de escrita do item. A primeira é uma breve descrição textual sem seguir um modelo, a segunda segue o modelo ARO.

No modelo ARO <ação|resultado|objeto> – a representação começa com um verbo, seguido pelo resultado desejado, e termina com o objeto dentro do contexto.

No capítulo seguinte, você verá como converter cada PBI em uma User Story, a representação mais comum para descrever o que deve ser feito para atender as necessidades dos usuários finais.

[**ação**] [**resultado**] [**objeto**]

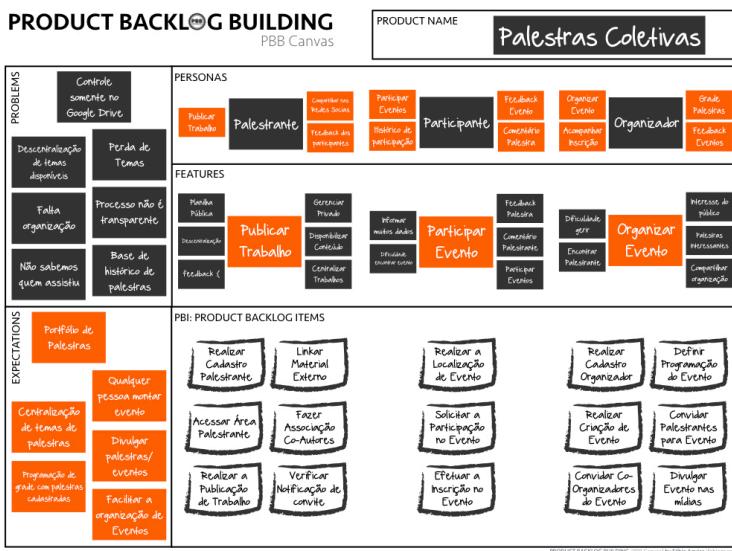
A representação começa com um verbo na sequência o resultado que precisa e termina com o objeto dentro do contexto.

PBIs: Para cada feature, escreva os seus respectivos PBI's. Primeiramente, escreva a descrição textual da ação e em seguida, represente cada PBI como User Story. Organize os itens verticalmente, onde o que está mais acima tem maior prioridade (a prioridade é a próxima etapa).

Por fim, o Product Backlog

Essa é a dinâmica do “PRODUCT BACKLOG BUILDING”: O preenchimento do Canvas PBB por uma sequência de passos simples e efetivos para criar o backlog do produto:

Product Name > Problems > Expectations > Personas > Features > PBIs



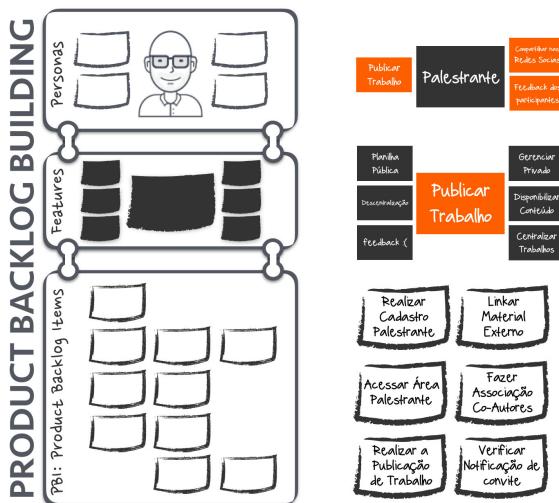
O preenchimento colaborativo do Canvas PBB ajuda:

- a organizar a visão “produtizada”,
- com o compartilhamento tácito de negócio e
- a definir o que o produto irá agregar ao final.

O resultado é um Product Backlog totalmente alinhado com o valor de negócio do cliente.

Organização visual do PBB

Geralmente, uso o tamanho A0 para impressão PBB Canvas, mas dependendo do tamanho do produto, abro mão do canvas impresso e uso uma parede ou uma mesa grande. O que aproveitado canvas, é sua organização visual. Distribuo na parede os post-it's obedecendo o fluxo de construção do PBB, organizo os personas na parte de cima, embaixo posicione as funcionalidade relacionadas a persona, e em formato de coluna cada PBI debaixo de sua respectiva funcionalidade. Praticamente, a mesma organização visual do canvas, mas na parede ou em uma mesa.

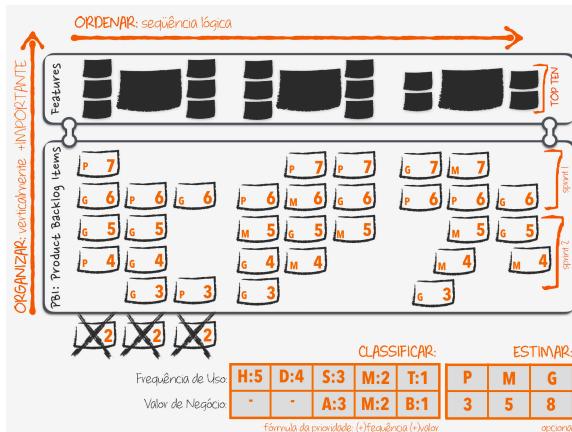


PBB - Organização Visual

CORG - a técnica de priorização do PBB

Após o preenchimento do Canvas PBB, priorize o backlog para direcionar o trabalho do time. Para tanto, aplique a técnica de priorização da sua preferência.

Neste capítulo, compartilhamos uma técnica de priorização de backlog: o CORG. A base da técnica CORG está em Classificar, Ordenar e Organizar.



método de priorização - CORG

Classificar o backlog

O PBB gera um backlog com os PBIs. A primeira etapa do CORG é classificar cada PBI. Para classificar, você precisa criar os tipos de classificação e uma escala para cada classificação conforme o contexto do produto.

Decida com o grupo quais os critérios de classificação com sua respectiva escala, conforme o contexto do produto. É importante que o

grupo alinhe e decida os critérios para classificação. Segue abaixo um exemplo baseado em dois critérios. Ao final da classificação, cada PBI terá uma nota.

Exemplo de classificação

Considere o seguinte exemplo de classificação:

- “frequência de uso”, a frequência que o usuário realiza um PBI (lembra que cada PBI é um passo do fluxo de trabalho de uma feature).
- “valor de negócio”, o valor de negócio gerado quando o usuário utiliza o PBI.

Exemplo de escala

Considere uma escala para o critério “frequência de uso”:

- “hora a hora” = um PBI que realizado mais de uma vez no dia,
- “diário” - um PBI que é realizado pelo menos uma vez ao dia,
- “semanal” - algo realizado uma, duas ou três vezes na semana,
- “mensal” - um PBI que é acionado uma vez no mês ou um pouco mais de uma vez, e por fim,
- “trimestral” - pelo menos uma vez a cada 3 meses.

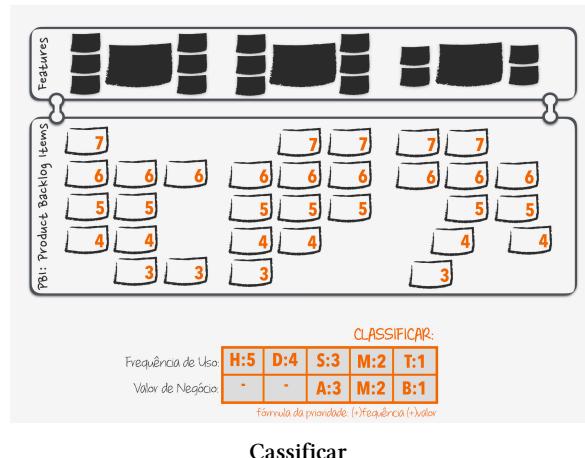
Escolha um valor que faça sentido para cada escala, exemplo, atribua 5 para hora a hora, dessa forma mostra que um PBI que recebe o valor 5 na classificação frequência de uso, que o que é realizado hora a hora é o que tem mais prioridade para ser entregue. Com isso, faça sucessivamente atribuindo 4 para diário, 3 para semanal, 2 para mensal e por último, 1 para trimestral.

Considere uma escala para o critério “valor de negócio”:

- [A] Alto - na visão do usuário esse PBI tem um valor altíssimo,

- [M] Médio - algo que tem relevância mas que não é altíssimo, e por último,
- [B] Baixo - algo que faz sentido mas que não agrupa muito valor nesse momento.

Na sequência, seguindo a mesma aplicação para classificação anterior, faça receber um valor que satisfaça, exemplo, atribua 3 para o alto, 2 para médio e 1 para baixo. Trago dois exemplo para melhor compreensão, claro, que dependendo do seu produto, esses dois tipos de classificação não façam muito sentido.



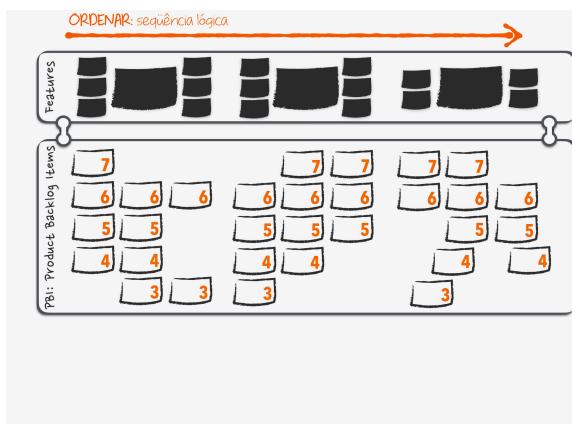
Dica: Escreve o valor total da prioridade no lado direito de baixo do post-it com o PBI.

Ordenar o backlog

A segunda etapa do método CORG é ordenar os itens de backlog em uma sequência lógica, da esquerda para a direita, como se tivesse

contando uma história.

A ordenação começa com as funcionalidades, posicionando no canvas ou na parede, da esquerda para a direita. Depois de ordenar as funcionalidades, ordene os PBIs debaixo de suas respectivas funcionalidade, de cima para baixo. Note que praticamente os itens do backlog já estão debaixo de cada funcionalidade, a própria dinâmica do PBB já conduz dessa forma.

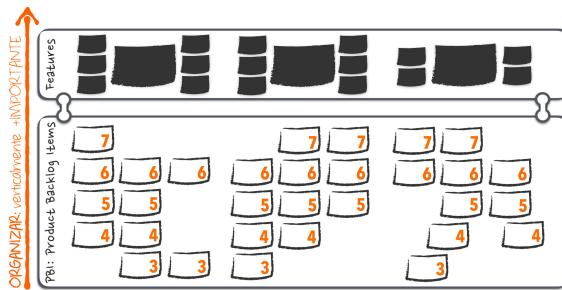


Ordenar

Organizar o backlog

Organizar, essa é a última etapa do método CORG. A organização é realidade de forma vertical, sem alterar a ordenação horizontal das features, mas considerando a nota do PBI.

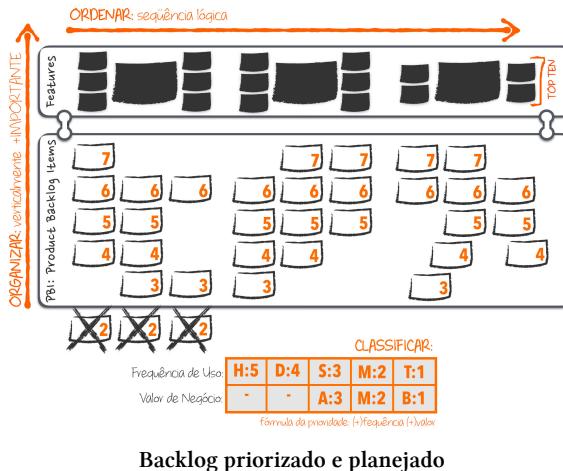
A organização segue da seguinte forma, na primeira linha ficam os PBIs com maior nota, por exemplo, se o 7 for o maior valor, todos PBIs com nota 7 ficando-nos na primeira linhas. Abaixo dessa linha, a linha do 6, depois a linha do 5, e assim sucessivamente. Pronto! Backlog priorizado e planejado! Basta seguir cada linha, da esquerda para a direita.



Organizar

O CORG organiza o backlog e formato de umas matrix. Mas o backlog tem formato de lista. Então voce precisa mapear o COORG para o backlog: o que vai para lista do backlog é a primeira linha, respeitando a sequênciada esquerda para a direita, depois continuando com os itens da segunda linha (sempre da esquerda para as direita), depois da terceira linha, e assim sucessivamente, até o final.

Geralmente, os itens com nota baixa não entram no backlog, pois através da aplicação da técnica fica claro o valor reduzido dentro do contexto do produto. Nesse momento, o Product Owner pode usar seu poder de decisão para dizer “NÃO” esses itens com notas baixas. Desta forma, as primeiras sprints fornecerão os PBIs que tem mais valor de cada funcionalidade, ou seja, literalmente entrega o que tem mais valor no produto, de ponta a ponta.



Estimativa pelo tamanho da camisa

Existe uma atividade extra e opcional: a estimativa pelo tamanho da camisa, que pode ser útil caso tenha a necessidade de saber quantas sprints são necessárias para entregarem os itens do backlog.

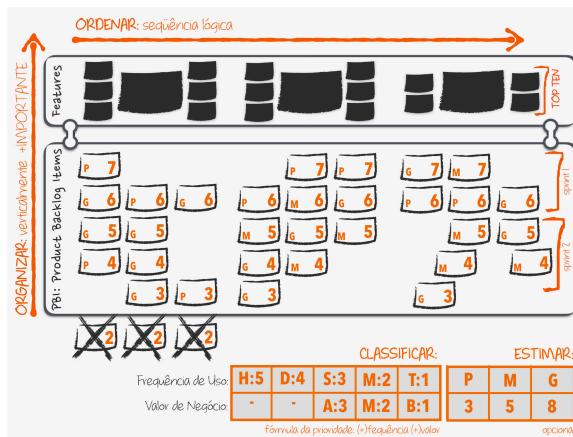
Caso esse seja o seu contexto, então, realize uma sessão de estimativa, pedindo para o time levar em consideração o esforço de desenvolvimento através dos tamanhos: “P” Pequeno - para itens relativamente fácil de entregar, “M” Médio - para itens com esforço considerável e “G” Grande - para itens que vão requerer um esforço grande de desenvolvimento.

Marque todos os PBIs de acordo com o seu ~tamanho de camisa~P, M ou G. Caso algum item seja mais que G, considere rescrevê-lo para que caiba nesses tamanhos. Caso um item seja menor que P, considere agrupá-lo para que caiba nesses tamanhos.

Converta os tamanhos em valores: o P recebe o valor 3, o M recebe o valor 5 e o G receber o valor 8. Some todos os valores.

A soma de todos valores demonstra quantos pontos de esforço precisa para entregar esse backlog.

Caso o time tenha uma base histórica, pode comprar quanto tempo levou no passado e extrapolar quanto tempo levará para entregar o backlog. Se o time não tiver uma base histórica, deve se perguntar quantos pontos acredita que consegue entregar por sprint, e, depois de algumas sprints, verificar se a previsão está de acordo com a realidade.

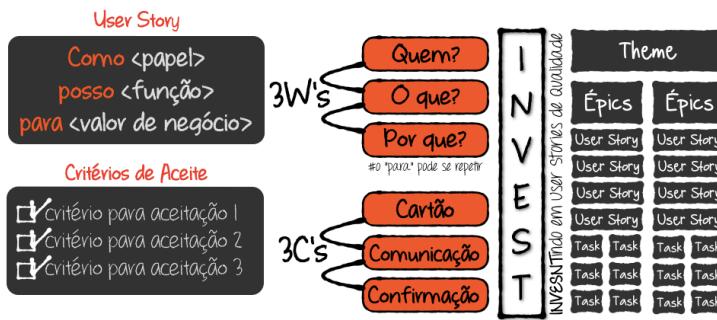


Estimar

Dica: Escreve o valor da estimativa no lado esquerdo debaixo do post-it com o PBI.

PBB e História de Usuário

O Product Backlog Building facilitar a descoberta e escrita de User Stories. Primeiro vamos fazer uma revisão dos conceitos e na sequencia ver a mágica que o PBB faz – Como o PBB ajuda na escrita de histórias de usuário.



Escrevendo User Stories Eficazes

História de Usuário

O guia do Scrum não esclarece como um item do backlog deve ser representado, os criadores do Scrum deixa em aberto, pode representar da forma que desejar. Com isso, pode ser apresentado de forma textual, caso de uso, o modelo aro, ou história de usuário (a forma mais usada por times ágeis). Cada história é apenas um “lembrete” da necessidade do cliente, ou seja, fornece um flash para(sempre) provocar a comunicação, representa os requisitos (necessidades) mais do que documentá-los. Os melhores documentos em métodos ágeis ajudam a recordar nossas conversas, eles não a substituem.

As histórias de usuários são escritas por ou para usuários ou clientes. Tipicamente uma feature com uma granularidade maior é desmembrada em algumas histórias. As histórias de usuário nasceram com o propósito de a própria pessoa afetada escrevesse, com o decorrer dos anos, o Product Owner virou o principal responsável por escrever histórias de usuários e organizá-las em um backlog de produto. Em outros contextos, qualquer um pode escrever uma história de usuário.

História de usuário é um formato textual para a descrição concisa de um requisito que busca responder a três indagações básicas – o acrônimo que representa é o 3Ws: quem? o que? e por que?

Como PERFIL DO USUÁRIO / PERSONA

Posso FUNÇÃO ESPECÍFICA DO PRODUTO

Para VALOR DO NEGÓCIO

O acronônimo INVEST [ref] ajuda a definir a qualidade na escrita de boas histórias. Esta história é... Independente? Negociável? Valiosa para o negócio?, Estimável? Pequena (em inglês, Small)? Testável?

Outro acrônimo que auxilia é o 3Cs [ref]. Em inglês, os três Cs são: card, conversation, and confirmation. Histórias de usuários têm três aspectos críticos. O cartão da história deve seguir um modelo simples (como ... posso ... para que ...). A partir deste modelo simples (card), teremos conversas colaborativas (conversations) para melhor entender e detalhar a história, até que, quando pronta, recebemos a confirmação (confirmation).

Critério de Aceite

Critério de aceite é um formato textual que descreve como testar uma história de usuário, são os critérios definidos para aceitar uma

história como pronta. Tipicamente uma história do usuário terá alguns critérios de aceite.

Dado que CENÁRIO INICIAL

Quando AÇÃO REALIZADA

Então RESULTADO ESPERADO

Os critérios de aceite (ACs, do inglês Acceptance Criteria) fornecem uma lista de verificação que determina quando uma história de Usuário está concluída e funcionando. A partir deles, todos os envolvidos irão saber se a história está completa. Tipicamente isso é verificado junto ao Product Owner, que realiza o aceite da história, ao comprovar que todos ACs estão OK.

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, o time de desenvolvimento entra em detalhes técnicos de como os pedaços menores da histórias serão implementados. Diferente das histórias, as tarefas não seguem um formato textual definido. Essas são mais diretas, com uma linguagem bem técnica.

Uma tarefa identifica algo que precisa ser feito, algo necessário para alguma história. Como tal, a tarefa não será necessariamente independente, ou irá demonstrar o valor de negócio. A maioria das tarefas tendem a ser para programadores, descritas com termos utilizados pelos mesmos. Alguns exemplos de tarefas: alterar campos da tabela ABC, criar contas de teste para os usuários X, automatizar os scripts de geração de dados, e assim por diante.

Um exemplo

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

Feature: Consultar partidas sem geo-localização

História 1

Como peladeiro não-cadastrado

Possso consultar partidas próximas a um endereço informado

Para encontrar um jogo próximo ao meu local atual

História 2

Como peladeiro cadastrado

Possso consultar partidas próximo ao meu trabalho

Para encontrar um jogo próximo ao meu trabalho

História 3

Como peladeiro cadastrado

Possso consultar partidas próximo a minha residência

Para encontrar um jogo próximo a minha residência

Critérios de aceite da história 2 (exemplo 1)

Dado que existe uma partida a menos de 10 kilómetros do meu trabalho

Quando procuro por uma partida próxima ao meu trabalho

Então encontro tal partida

Critérios de aceite da história 2 (exemplo 2)

Dado que não existe nenhuma partida a menos de 10 kilometro do meu trabalho

Quando procuro por uma partida próxima ao meu trabalho

Então não encontro nenhuma partida

Tarefas da História 2

- 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

Tipicamente, as equipes de desenvolvimento trabalham com histórias de usuários. O PBB Canvas com Steps Maps nos passo 5 e 6 (Features e PBI) ajuda com o mapeamento de features para histórias do usuário.

Escrevendo Histórias de Usuário com o PBB

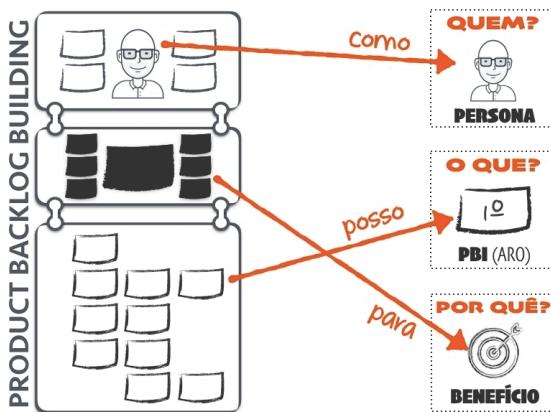
Não é fácil escrever as histórias de usuário. Ou melhor, não era fácil. Mas após preencher o canvas PBB, essas são escritas naturalmente. Perceba a ilustração a seguir.

O Scrum não especifica como representar cada item no Backlog. Mas, o formato História do Usuário é o mais usada atualmente pelos times ágeis para representar um item no Backlog. História de Usuário é uma breve descrição do que é necessário para o cliente ter no produto, que pode representar uma necessidade do usuário ou uma descrição de um item do backlog.

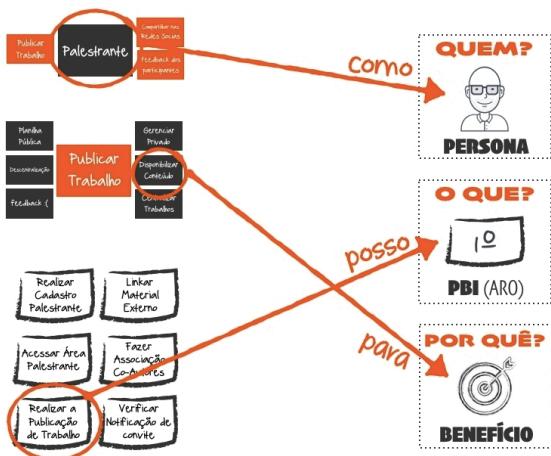
A escrita de uma História de Usuário basicamente responde 3 perguntas: QUEM? O QUE? POR QUÊ?

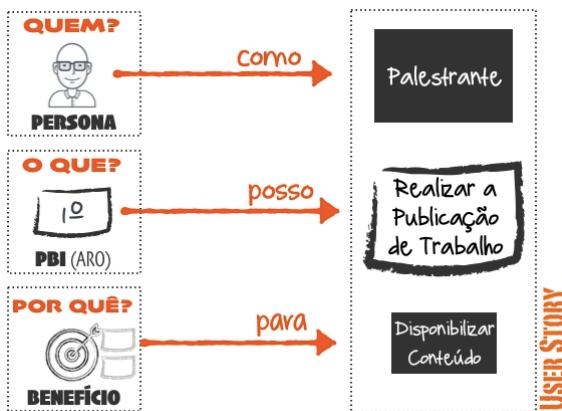


O PBB nos ajuda na escrita das Histórias de Usuário. Como podemos notar no PBB temos o “QUEM?” que é a persona, o “O QUE?” que nesse caso são os PBI’s já representadas em modelo ARO e por último, o “POR QUÊ?” que está nos benefícios que o persona destacou na feature. A figura abaixo exemplifica de uma forma bem simples como fica fácil escrever as Histórias de Usuário com ajuda do Product Backlog Building.



Veja a aplicação prática:





Modelo ARO

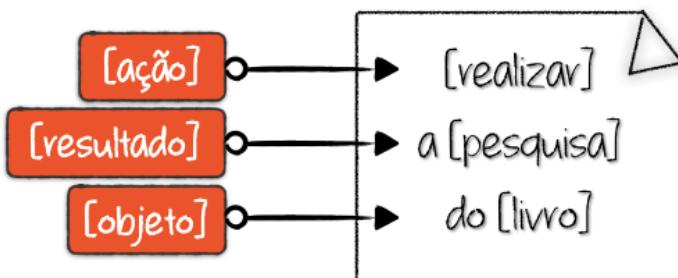
Como já foi mencionado, a história de usuário é a abordagem predileta dos times ágeis para representar um item no backlog. É uma ótima abordagem, quando se tem a pessoa afetada. Quando há um esforço muito grande para encontrar o usuário e em muitas situações é preciso força a escrita, tal como, “Como sistema posso....”, “Como desenvolvedor quero...”, é sinal de alerta que a história não é a melhor opção para representar um item de backlog. Temos vários abordagens para esse cenário, mas vejo o modelo aro como a melhor opção. Outro contexto que se aplica bem o ARO é em itens de backlog técnicos, onde não necessariamente eu tenho uma pessoa afetada.

[ação] [resultado] [objeto]

A representação começa com um verbo na sequência o resultado que precisa e termina com o objeto dentro do contexto.

A representação no modelo ARO é bem mais simples que história de usuário (que já é simples). Segue um formato textual com uma pequena descrição onde o “A” é de “AÇÃO”, “R” de “RESULTADO” e “O” de “OBJETO”, ou seja, a representação começa com um verbo, na sequencia o resultado e termina com o objeto dentro do contexto.

Segue abaixo um exemplo de representação de item de backlog no modelo ARO.



Importante: O PBB Canvas com Steps Maps ajuda com o mapeamento de features em PBIs, antes de escrever em história de usuário, a sugestão é representar no modelo ARO, visto que é um momento de concepção de backlog. A representação em história fica para o passo seguinte, pois o time de desenvolvimento pode definir que a abordagem ARO é o suficiente para puxarem para a sprint.

Backlog Ready - definindo a qualidade de um PBI

Definition of Ready – DOR – (ou, em português, “Definição de Preparado”) é o acordo que indica que a equipe entende o que precisa ser feito e a quantidade de trabalho necessária para concluir a história do usuário ou o PBI. Tipicamente as pessoas dizem algo, tipo: “A História está READY para começar o trabalho”.

Em contrapartida, Definition of DONE (DOD) é o acordo que demonstra a qualidade da história do usuário ou do PBI produzido, onde “Done” comprova que a história atende a todos ACs da sua respectiva lista de verificação.

Este livro trata de ajudar as equipes a ter seus “Backlogs Ready”. Não é escopo desta obra detalhar como as equipes fazem para levar suas Histórias de “Ready” para “Done”.



Apesar da definição ficar em aberto para o time definir, alguns critérios são usados como padrão para times ágeis, por exemplo, um Product Backlog Item está ready quando:

- Está representado em User Story (3Ws)
- Está preservando o conceito INVEST

- Está claro os critérios de aceite
- Possui valor de negócio
- Está em uma granularidade pequena

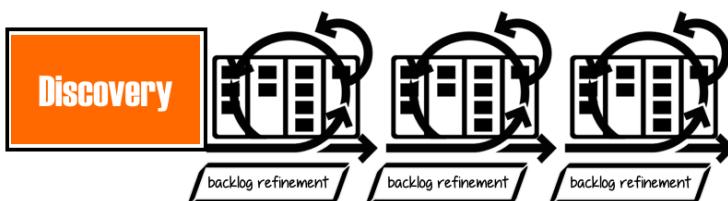
O Product Backlog Building deixa o “Backlog Ready”.

O Product Owner é responsável por colocar as features e histórias no backlog. No entanto, o time deve trabalhar com o Product Owner durante sessões refinamento do Backlog para ajudá-los a colocar as histórias preparadas para o time se comprometer a entrar em um ciclo curto de desenvolvimento. Somente então o time será capaz de estimar quanto de esforço de trabalho uma história vai levar e quantos pontos eles podem levar para uma Sprint.

A prática do PBB cobre todos os critérios que são padrão para times ágeis. Dessa forma, o resultado final do PBB, os PBI's estarão escritos em Histórias do usuário(3Ws), com o conceito INVEST, e prontos para o time estimar e puxar para próxima sprint, ou seja, “BACKLOG READY”.

Lembrando que “Requisitos READY se transformam em entregáveis DONE”.

Dica: Não deixe nada entrar na sprint que não esteja READY, e não deixe nada sair da sprint que não esteja DONE.



O uso de “Ready” não se limita somente ao Scrum, pois é uma prática muito útil também quando trabalhando com Kanban e outros métodos.

Dando um TAPAs no PBB

O Product Backlog Building (PBB) facilita a descoberta e escrita de user stories, após o preenchimento do PBB Canvas.

O TAPAs ajuda os times a um bom entendimento e aprendizado da user stories através de um joguinho de cartões que permite um conversa para decidir se uma user stories é: Tangível, Atômica, Preciosa, Acessível (no original em inglês Tangible, Atomic, Precious and Affordable).

Esses conceitos são baseados na metáfora de que uma user story pequena é similar ao famoso estilo de comida espanhola. Isso significa que uma user story deve ser algo pequeno, que cabe na mão, não muito caro para fazer e seja capaz de resolver no mínimo uma pequena parte do problema (aliviar um pouquinho a fome, por exemplo) .

Ok!!! Mas a onde entra o TAPAs no PBB???

O PBB ajuda a construir o backlog dos times e a esclarecer as user stories e o TAPAs entra justamente para dar um tapa final no backlog. Com o backlog priorizado e com os itens representados como user stories, o time decidi se cada user story é, Tangível? (Tátil, Concreto e Específico), Atômica? (Pequeno e Autônomo), Preciosa? (É capaz de resolver o problema de um usuário) e Acessível? (Permite uma implementação viável e direta), ou seja, “TAPAs”.

O TAPAs tem uma aplicação prática no refinamento do backlog, mas funciona muito bem em uma sessão final no PBB, validando os itens prioritários do backlog, ou seja, os possíveis itens de backlog que o time irá se comprometer para primeira sprint.

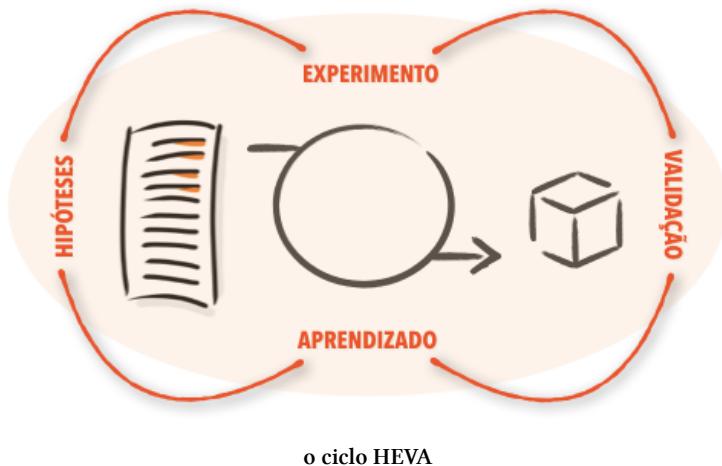
Literalmente, dando um TAPAs no PBB!!!

Fonte: <https://medium.com/@Elabor8/making-tapas-with-your-user-stories-94b013541b38>

O ciclo HEVA

Todos os requisitos estarão no backlog. Mas não se pode afirmar que são requisitos. No âmbito da engenharia de software, um requisito consiste da definição documentada de uma propriedade ou comportamento que um produto ou serviço particular deve atender. Métodos Ágeis agregam mais valor em cenários complexos, onde o grau de incerteza é alto e falta de conhecimento do que precisa entregar. Se ouve muito uma expressão bastante usada no mercado – “Nem o cliente sabe o que ele quer!” – é a realidade do desenvolvimento de produto.

Diante desse cenário complexo, não tem como afirmar que os itens de backlog são requisitos, na verdade, com a falta de conhecimento e o grau de incerteza, em vez de requisitos tem um backlog de hipóteses, onde precisa de um ciclo curto de desenvolvimento para colocar o mais cedo possível na mão do cliente e validar, se é viável ou não para o produto, se está indo no caminho certo. Se tem hipótese e precisa da validação do cliente, o ciclo curto de desenvolvimento será um experimento. E com o feedback se é viável ou não, fecha o ciclo com o aprendizado contínuo do produto.



o ciclo HEVA

Quando se pensa em construir um backlog precisa ter como ferramenta de trabalho esse ciclo, nomeado pelas iniciais das palavras chamado de “ciclo HEVA” – a principal ferramenta do Product Owner, o facilitador do PBB.

PBB e a Lean Inception

A Lean Inception ajuda no entendimento do MVP e suas principais funcionalidades. O PBB ajuda na criação de um backlog de histórias. A combinação das duas técnicas tem se apresentado como uma forma muito efetiva de alinhar estratégia e entrega de produtos digitais para equipes Scrum.

PBB, o passo seguinte da Lean Inception

Criar um backlog de histórias sem compreender o MVP pode ser arriscado: o time pode trabalhar muito, entretanto com baixa eficácia, já que não está alinhado sobre o mínimo viável para verificar o direcionamento do produto. Por isso recomendamos fortemente que antes de aplicar o PBB, que seja realizado as atividades da Lean Inception, segundo o livro Lean Inception: como alinhar pessoas e construir produtos certo.

Ao terminar uma Lean Inception, o time Scrum sabe o que fazer, mas ainda precisa definir o backlog de histórias. O PBB é a ferramenta para ajudar o time a detalhar o que fazer em histórias do usuário, o item de trabalho do dia a dia do time.

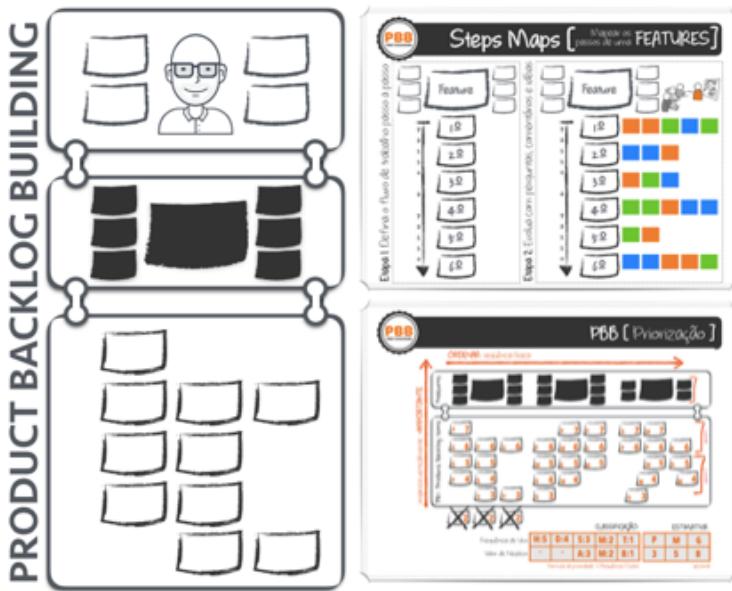
A Inception gera um plano para a criação e evolução do produto digital via MVPs. Sugerimos que o PBB seja aplicado somente para um MVP, no caso, o MVP que o time vai começar a trabalhar. Após a Lean Inception, o time provavelmente vai passar um dia para construir um backlog. Além do sequenciador de funcionalidades, a Lean Inception também gera outro artefato importante: o Canvas MVP. Note que os personas e funcionalidades vindo da Lean

Inception aproveita para popular o PBB Canvas. A saída da Lean Inception é o canvas MVP e a entrada do PBB é o PBB Canvas.



O backlog do produto é a saída da descoberta do produto

O passo seguinte da Lean Inception com o PBB, visto que já temos as personas e as funcionalidades, segue a partir do Steps Maps e depois aplicação do método de priorização – CORG para ordenar o backlog e ter um plano de entrega a nível de sprints, ou seja, quantas iterações terá para entregar o MVP definido no MVP Canvas.



A Lean Inception tem uma agenda de 5 dias úteis, o PBB como passo seguinte da Lean Inception, é no dia seguinte.

	1º dia	2º dia	3º dia	4º dia	5º dia	dia seguinte
KICK OFF						
VISÃO DO PRODUTO	PERSONAS	FEATURES	JORNADAS	CANVAS MVP		PBB
OBJETIVOS	FEATURES	NIVELAMENTO FEATURES	SEQUENCIADOR DE FEATURES	SHOWCASE		

Note que a Lean Inception ajudou a entender e alinhar sobre o MVP.

Mas o Product Backlog Building vai retomar algumas conversas importantes e ajudar a detalhar o MVP com suas funcionalidades em histórias do usuário e com plano de entrega a nível de sprint.

Como podemos perceber o grande poder do PBB é a facilitação e a colaboração que provoca com todos os envolvidos na construção de um Backlog, sempre levando todos a um entendimento compartilhado do contexto de negócio e a descoberta de itens do Backlog totalmente alinhado com o valor de negócio do cliente.

