

T



MAÍRA STORCH
PRODUCT @KYTE

❤️ TERA-LOVER DESDE 2017



AGENDA

- **PMs & Devs**
- **Agilidade na prática**
- **Como definir o backlog?**
- **Por onde começar?**



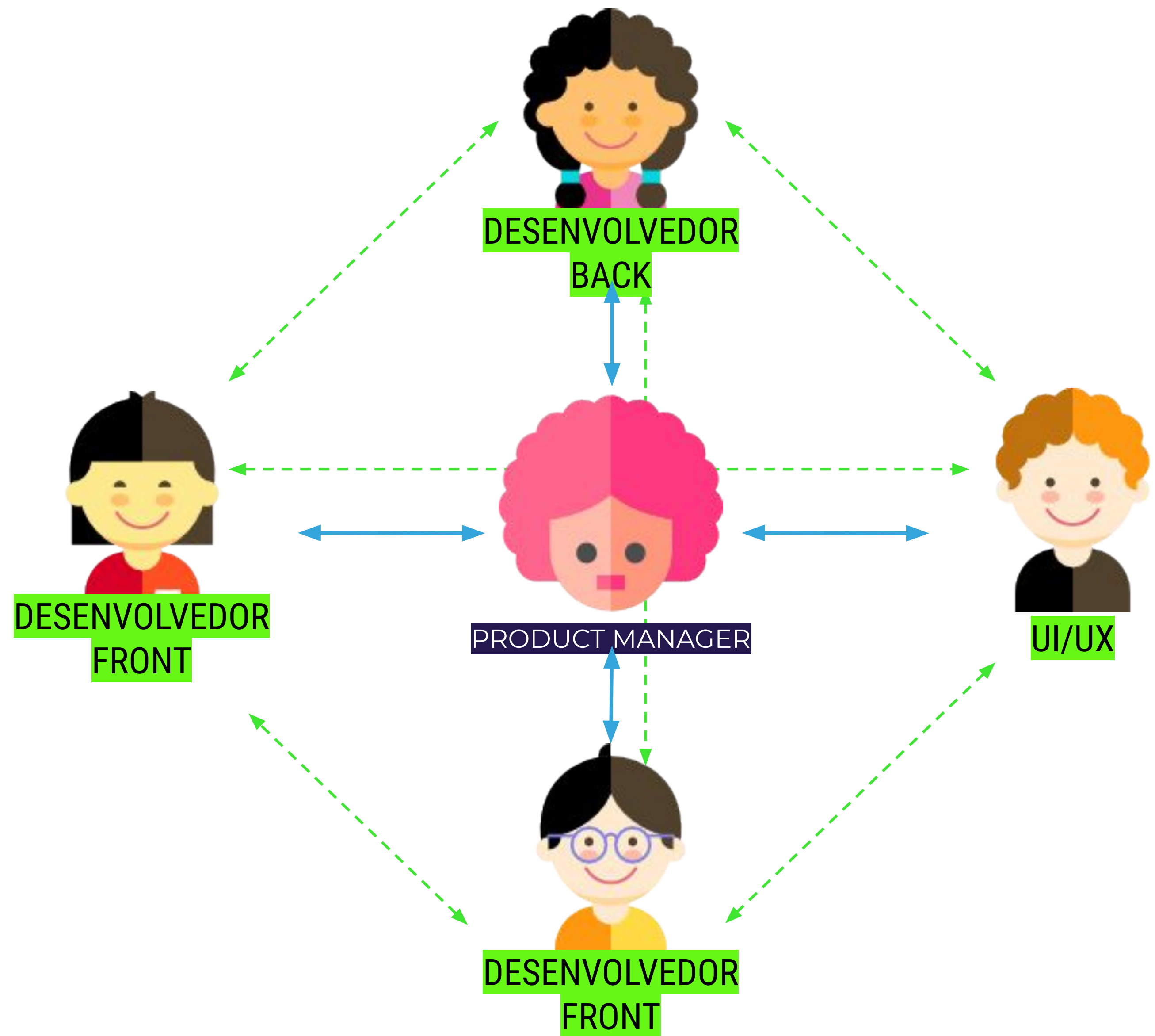
missão da **aula**

Compreender os papéis existentes em um time de tecnologia e como se dá a interação entre as pessoas no papel de PM e as pessoas que desenvolvem estes produtos.

PMs & Devs

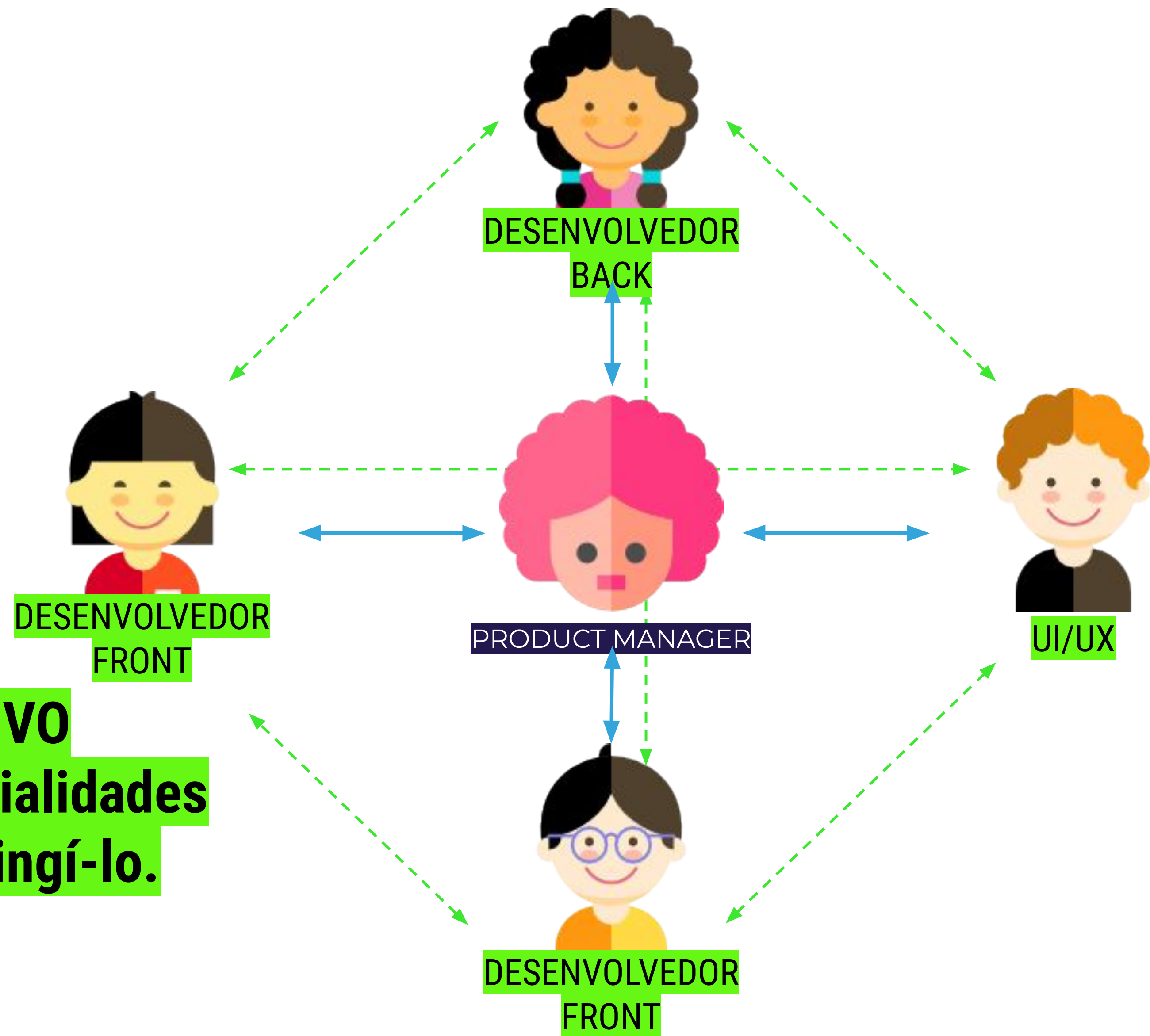
- TIME DE PRODUTO
- PAPÉIS DE PMS E DEVS
- BACKGROUND TÉCNICO VS LIDERANÇA DO TIME DE DESENVOLVIMENTO

COMO É UM
TIME DE
PRODUTO
“TÍPICO”?



**O QUE
REALMENTE FAZ
DIFERENÇA:**

**Time formado em
torno de um OBJETIVO
claro, com as especialidades
necessárias para atingí-lo.**



Papel de PMs no time de produto

- > Liderar e inspirar o time - **não chefiar!**
- > Garantir que o time esteja fazendo as coisas certas, que vão gerar mais valor.
- > Evitar trabalho jogado fora (mas às vezes acontece)
- > [Frequentemente] ajudar a organizar o time

Papel de PM no time de produto

Problem space

(outras aulas cobrem bem!)

- Trazer objetivos e **propósito**
- Trazer a **visão** dos clientes, usuários e stakeholders
- Definir claramente o **problema** a ser trabalhado
- Definir as **métricas** relevantes (e acompanhar)
- Comunicar e **explicar** as decisões

Solution space

- Conduzir e **facilitar** a definição da solução, buscando clareza (shared understanding)
- Dar visibilidade para **trade-offs** entre:
 - Valor
 - Usabilidade
 - Viabilidade

*Aqui entram os famosos **backlogs**, **user stories**, **critérios de aceite** - mais adiante!*

Delivery

- **Facilitar** a comunicação entre as pessoas do time e stakeholders
- **Conduzir** o planejamento do time, orientado para a **geração de valor**
- **Desempatar** decisões que surjam no desenvolvimento

Pós-Delivery

- Trazer **resultados** pro time!

Como **devs** contribuem
para o time de produto?

Papel de DEV no time de produto

Problem space

- Entender o objetivo
- Entender o problema e por que é importante
- Criar empatia com usuários e clientes
- Perguntar, questionar!

Solution space

- Desenhar soluções (não somente técnicas!)
- Levantar novas tecnologias
- Perguntar, questionar!
- Identificar restrições

Delivery

- Construir
- Testar
- Identificar débitos técnicos
- Garantir qualidade

Pós-Delivery (em produção)

- Monitorar performance
- Corrigir bugs

Precisa de background técnico para liderar desenvolvimento de produto?

Resumindo: **não***, mas...

Precisa topar **aprender** um pouco sobre:

- Bancos e estruturas de dados (talveeeeeeez SQL)
- Lógica de programação
- Arquitetura
- APIs
- Processo de deploy e release

Na próxima aula ;)

Por quê????

Poder **conversar** com o time sobre as soluções e entender os **trade-offs**.

Se não sabe, pergunte!

Obs:* Exceção para produtos mais técnicos 😊

A vertical bar with a gradient from bright green at the top to light blue at the bottom, positioned to the left of the title.

Agilidade na Prática

- **AGILIDADE**
- FRAMEWORKS ÁGEIS
- O FAMOSO BACKLOG

T

Qual é mais rápido?

BLOCO 2







Manifesto ágil [link](#)

Estamos **descobrimos** maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo.
Através deste trabalho, passamos a **valorizar**:

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

Ou seja, mesmo **havendo valor nos itens à direita**,
valorizamos mais os itens à esquerda.

”



o manifesto ágil [...] não foi pensado para um modelo de produto de tecnologia, e sim para um modelo de **serviço de desenvolvimento** de produtos.



Jeff Patton

Podcast Getting to the bottom of Agile

Manifesto ágil nos dias de hoje

> Colaboração com o cliente?

Manifesto ágil nos dias de hoje

- > Colaboração com o cliente 🤔
- > Principal medida de **sucesso** é software funcionando?

Manifesto ágil nos dias de hoje

- > Colaboração com o cliente 🤔
- > Principal medida de **sucesso** é software funcionando?

Manifesto ágil nos dias de hoje

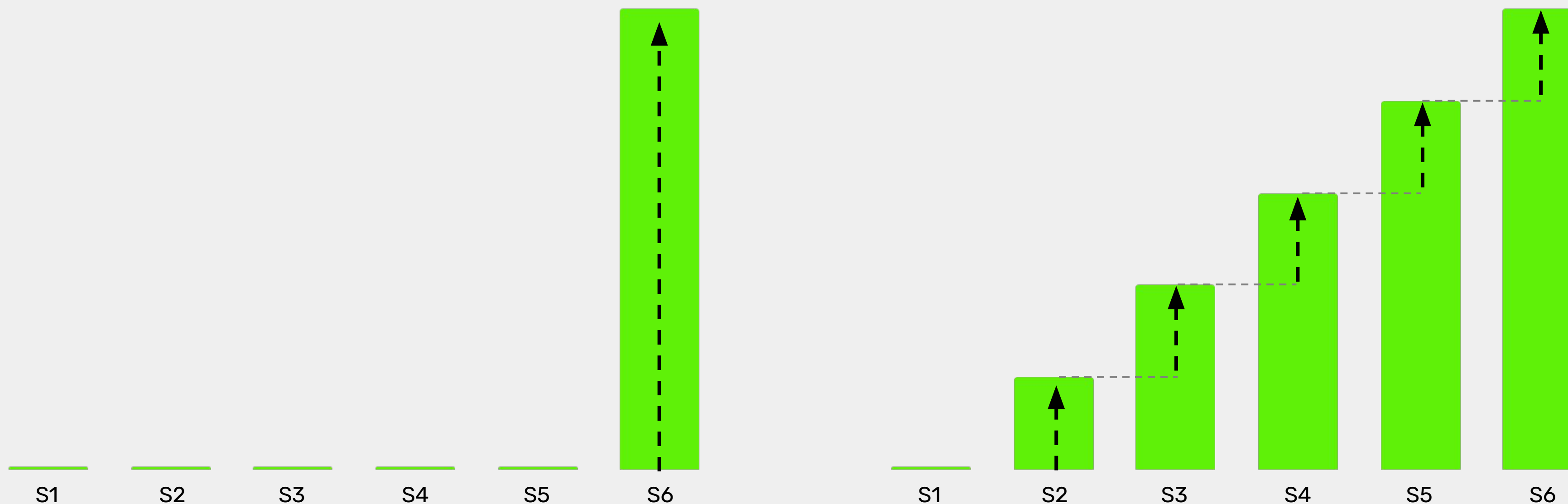
- > Colaboração com o cliente 🤔
- > Principal medida de **sucesso**
~~é software funcionando?~~
são resultados!

Manifesto ágil nos dias de hoje

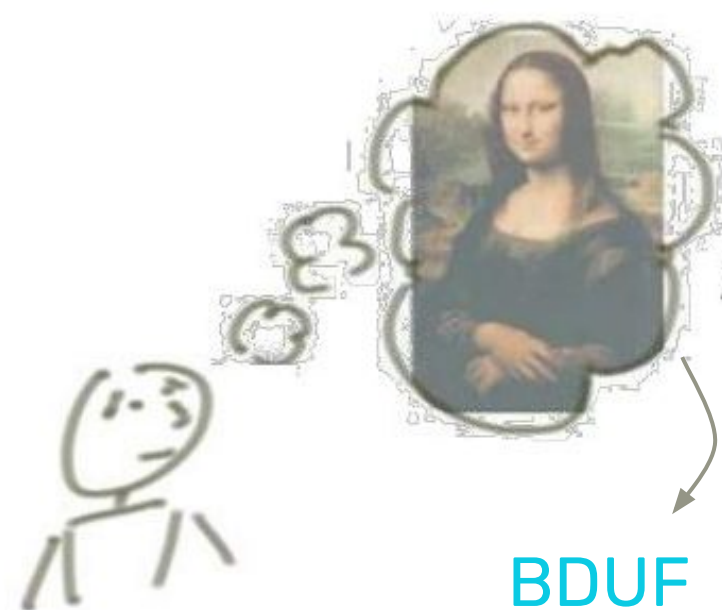
- > Valorizar e **confiar** nos indivíduos
- > Aceitar **mudanças** ao longo do tempo
- > Trabalhar em **ciclos** curtos
- > **Reduzir** o tamanho dos “lotes”
- > Parar e **refletir** sobre o produto e o processo



Agilidade é uma forma de pensar e trabalhar para **reduzir o risco** de investir em algo que não gera valor.



Incremental



BDUF

Big Design Up Front



Iterativo

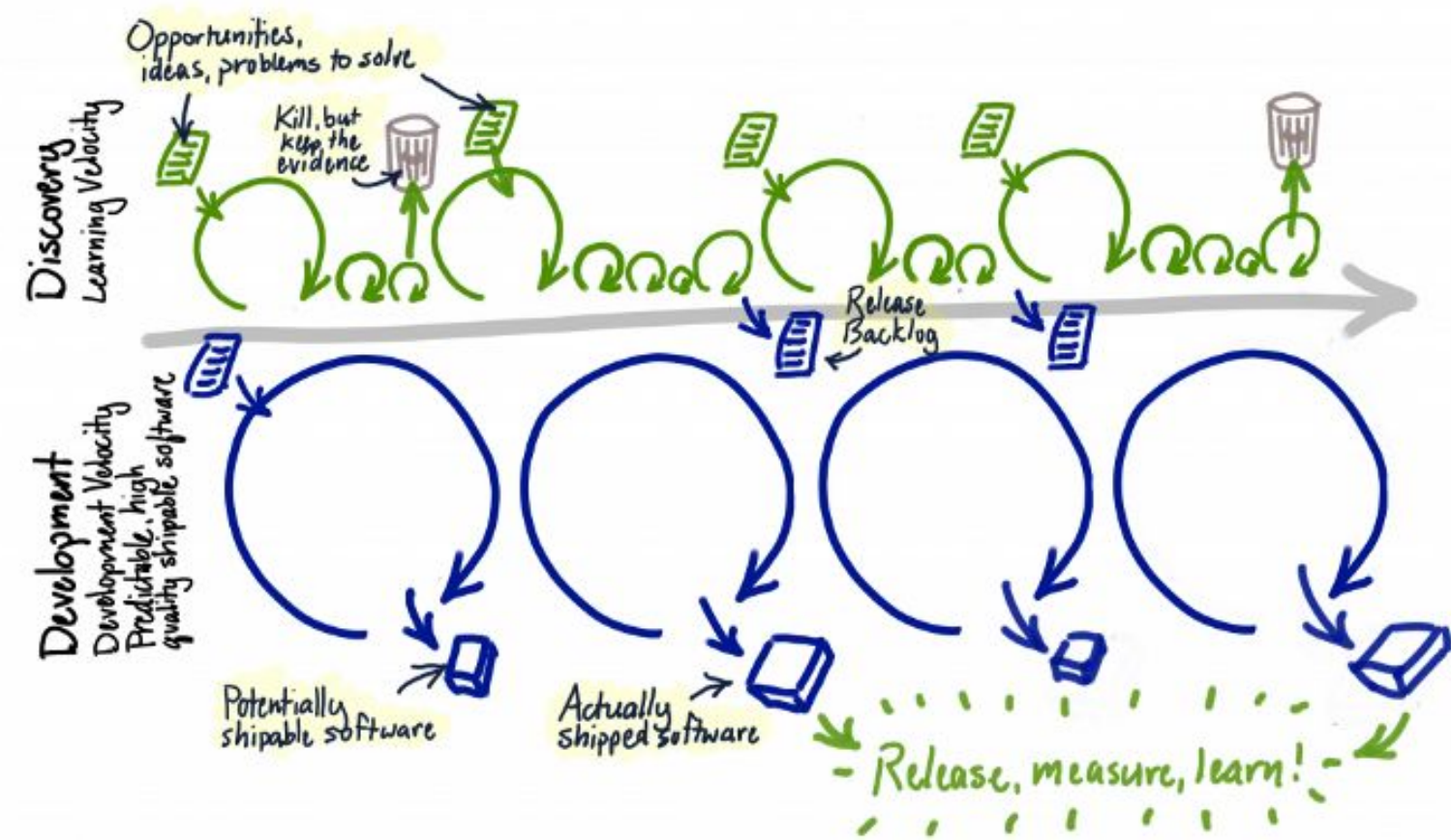


Agilidade na Prática

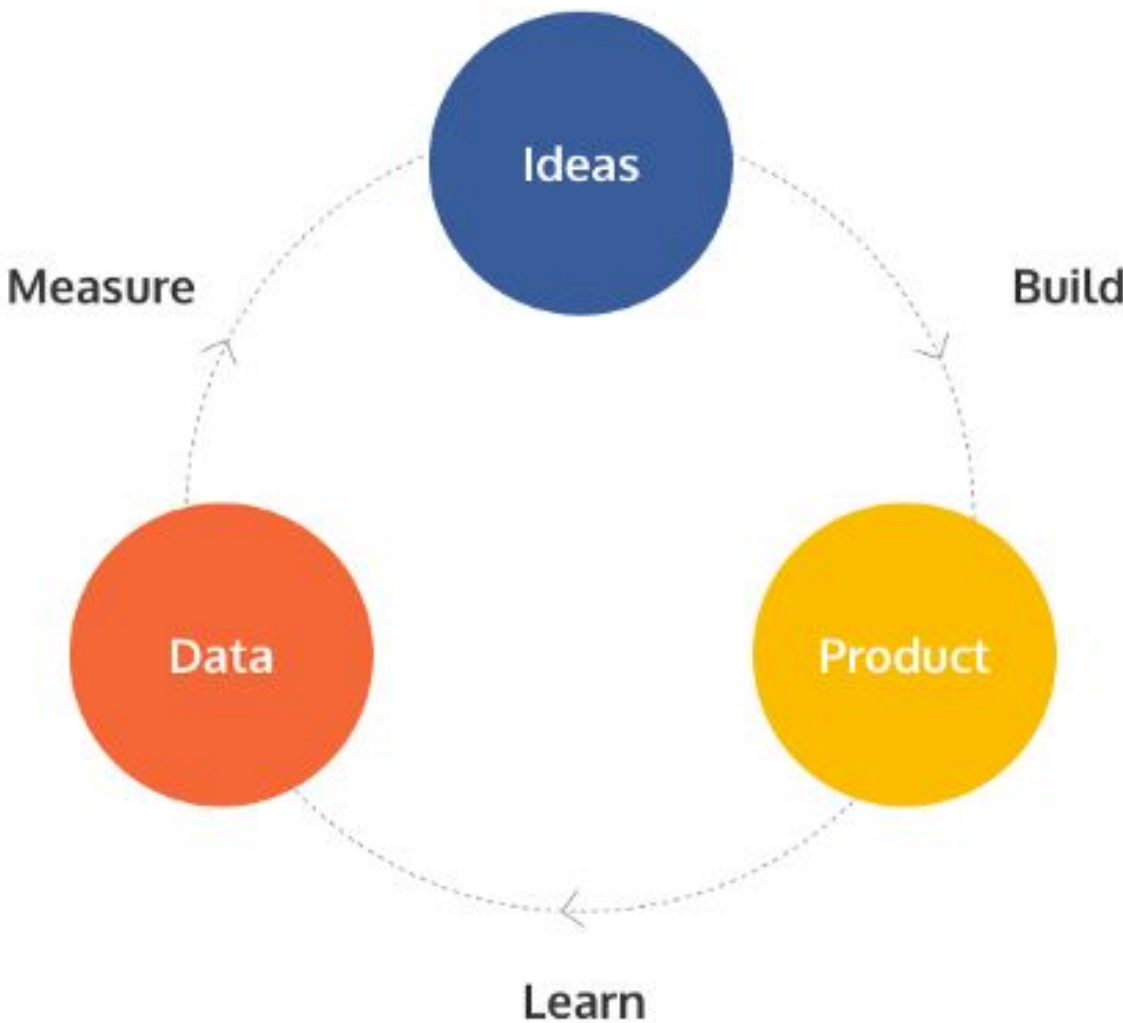
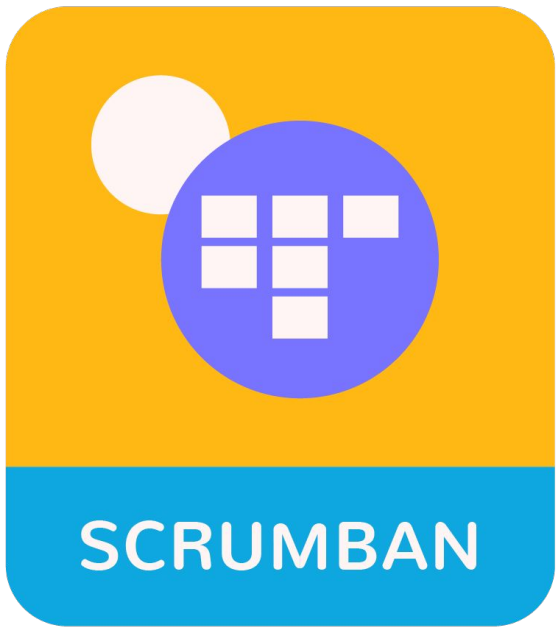
- AGILIDADE
- **FRAMEWORKS ÁGEIS**
- O FAMOSO BACKLOG



A diversidade de frameworks é enorme!

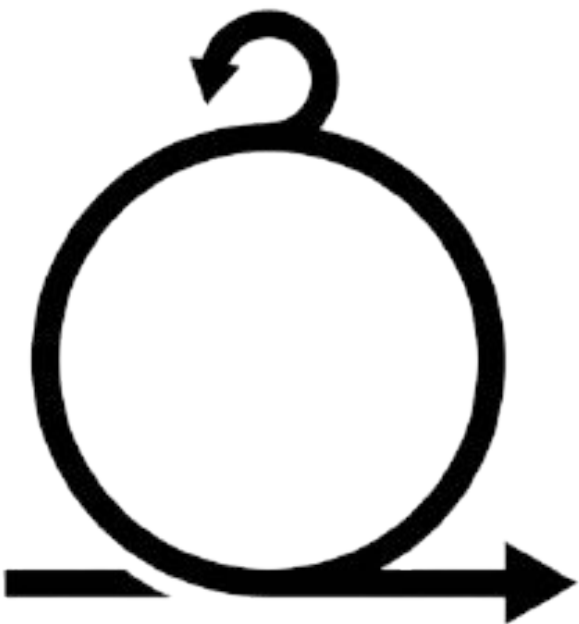
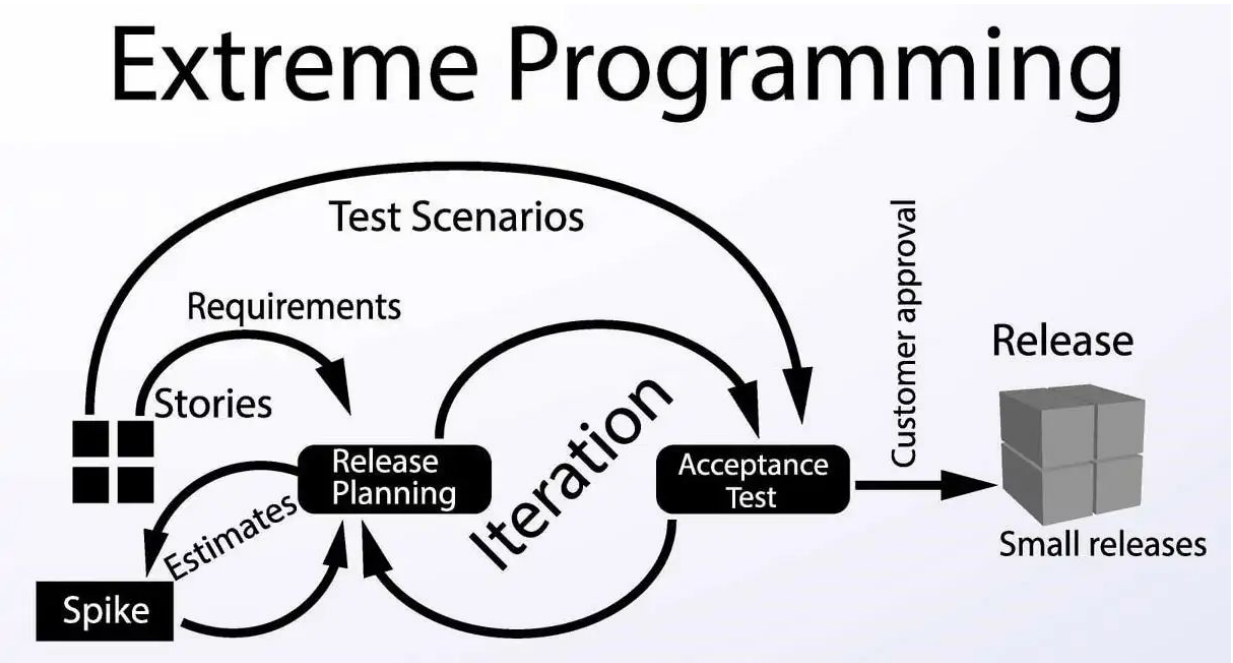


Dual Track

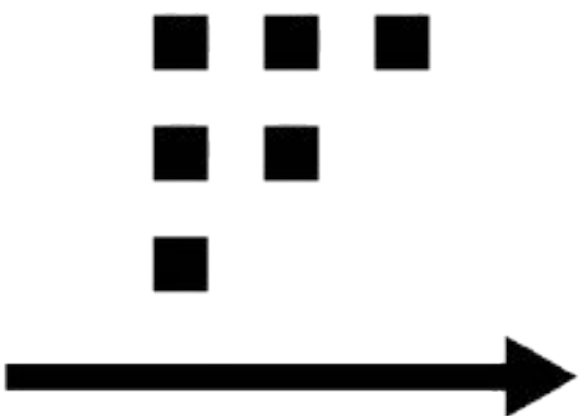


Lean Startup

Extreme Programming



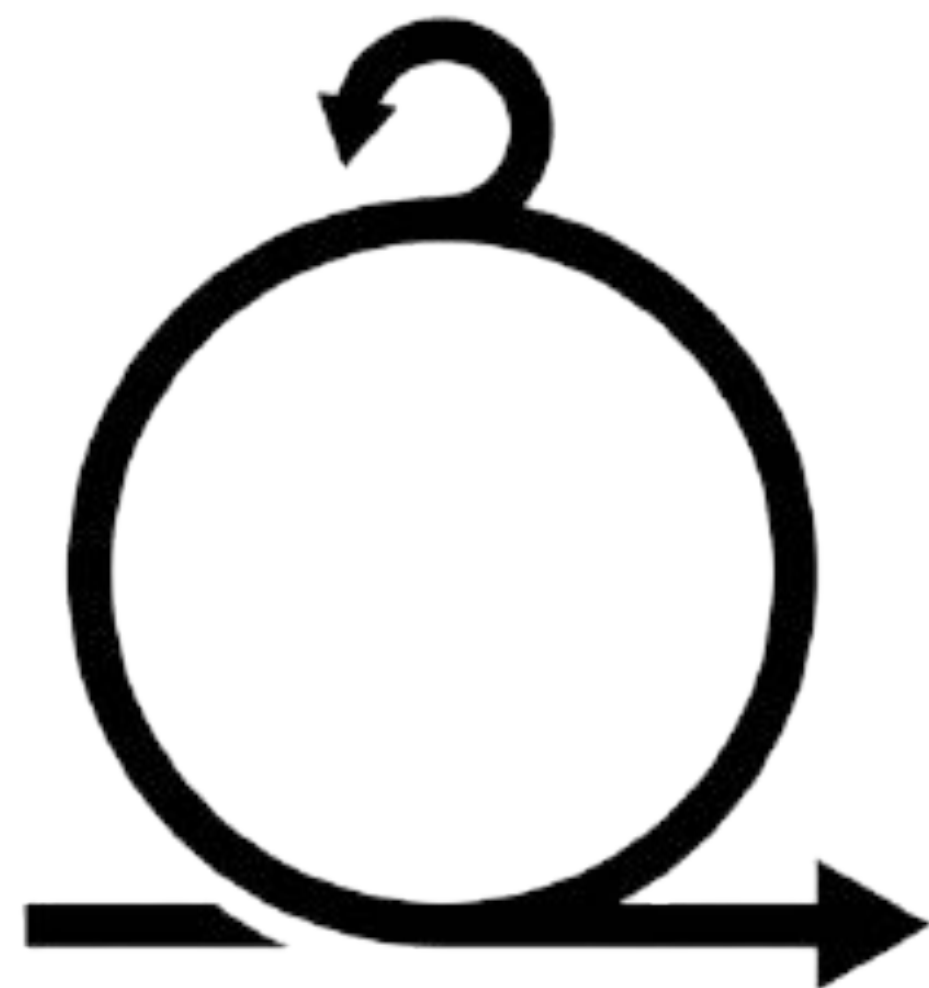
Scrum



Kanban

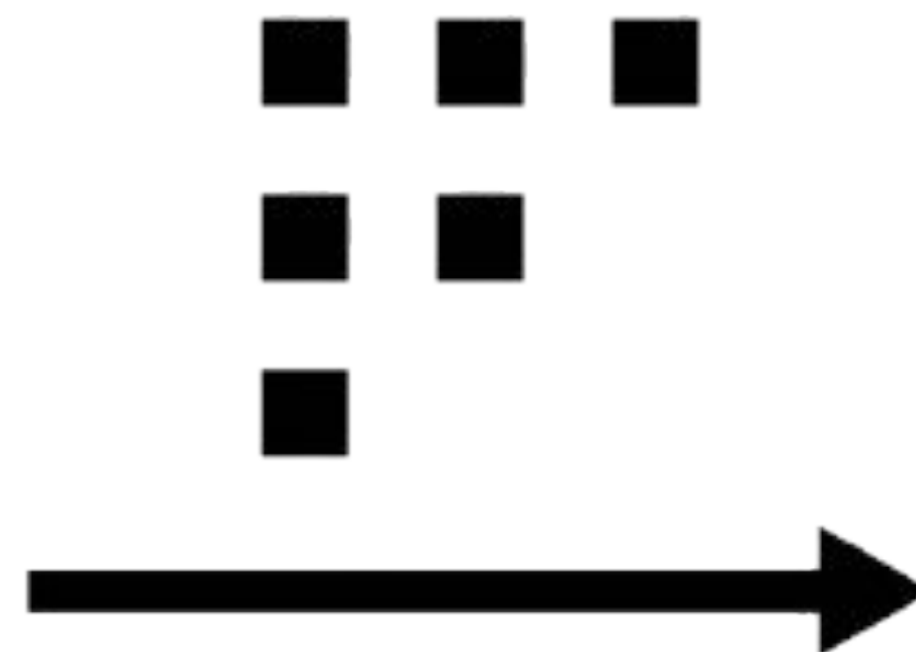


FDD

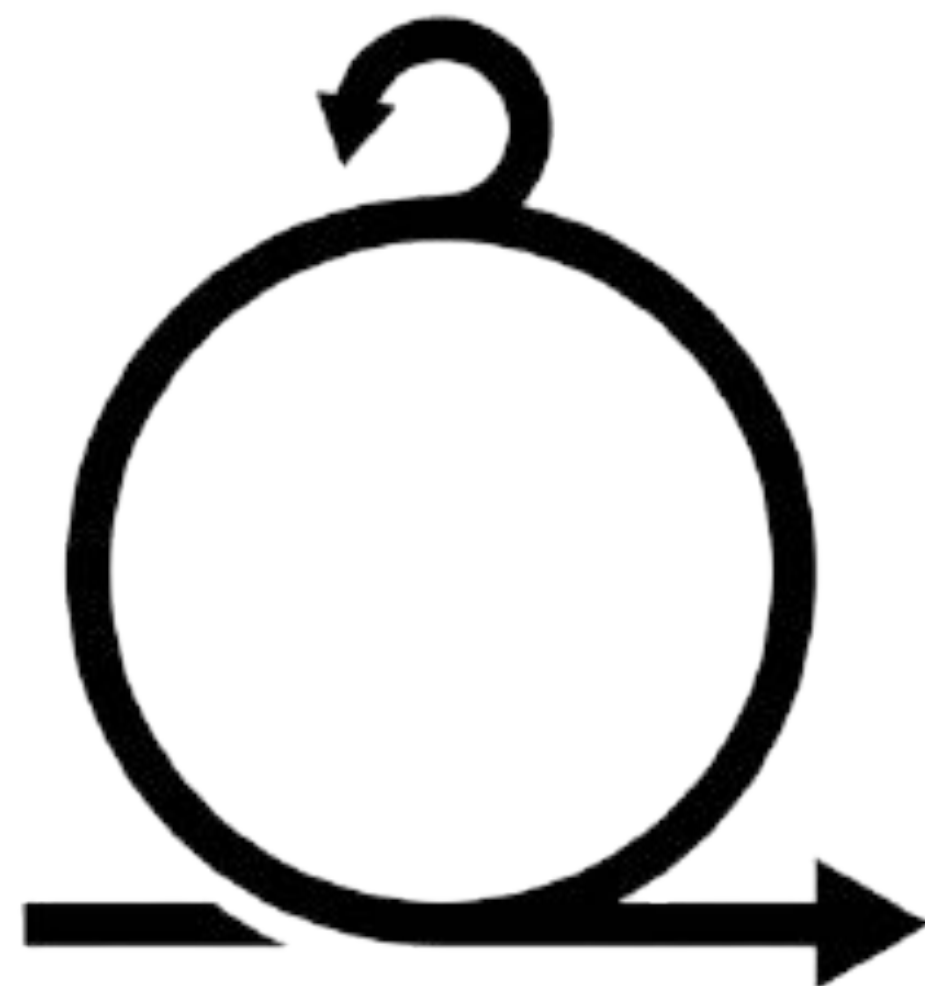


Scrum

vs



Kanban



Scrum

Cadência

Sprints com duração fixa

Estrutura

Papéis, artefatos e rituais bem definidos

Planejamento

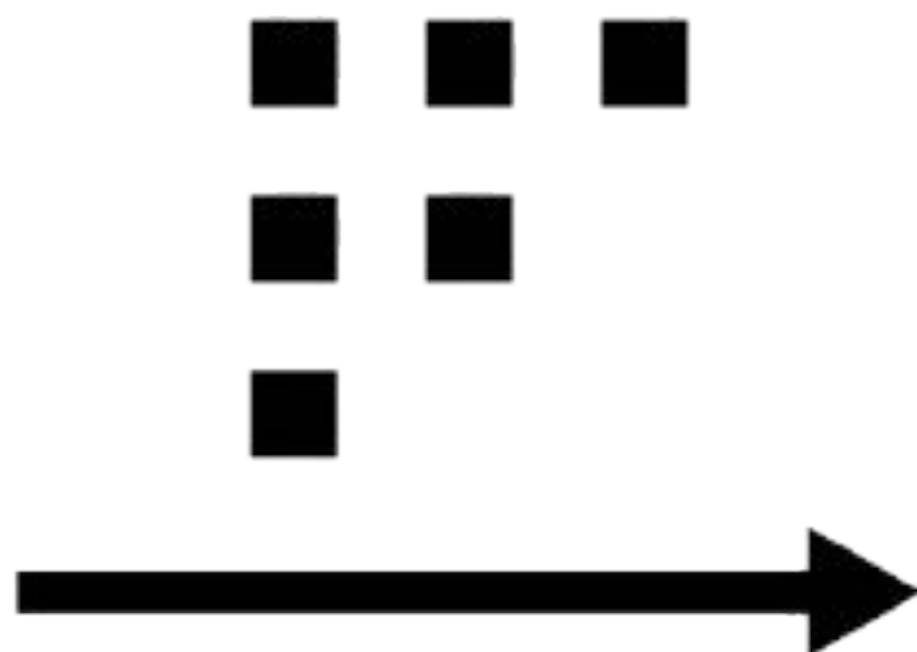
A cada ciclo. Mudanças apenas entre sprints.
Lotes = Sprints

Principais métricas

Velocidade e burndown

Princípios

Inspeção, adaptação e transparência



Kanban

Cadência

Fluxo contínuo, com limite de WIP

Estrutura

Solta, sem papéis ou outras prescrições

Planejamento

Conforme necessário. Mudanças a qualquer momento.
Lotes = itens.

Principais métricas

Empíricas: Lead/Cycle time, Throughput

Princípios

Comece com o que você faz agora

Buscar mudanças incrementais e evolutivas

Respeite os papéis, responsabilidades e cargos atuais

Incentive atos de liderança em todos os níveis



Depende! Nenhum? :D



Agilidade na Prática

- AGILIDADE
- FRAMEWORKS ÁGEIS
- **O FAMOSO BACKLOG**

A maioria tem uma coisa importante em comum: o **backlog**

Lista priorizada de tudo que o time pretende fazer no futuro **previsível**

Responsabilidade do PM/PO: mantê-lo **organizado** e refinado no nível adequado

É um trabalho de **FDP!**

- **F**atiar
- **D**escartar
- **P**riorizar



Como definir o backlog?

- **COMUNICANDO O QUE VAMOS FAZER E O PORQUÊ**
- **DESCOBRINDO A SOLUÇÃO**

Comunicando o que vamos fazer e o porquê

PRDs ou Product Briefs:

- Explicitar as motivações e resultados esperados
- Explicitar os entregáveis
- Explicitar dúvidas e pontos de atenção

Figma

PRD: [Name of Project]

Figma's Product Requirements Doc template

 Yuhki Yamashita

Project: Private embeds ▾

EDIT INFO

Team: @Collaboration

Contributors: , , 

TL;DR: Embed Figma files only accessible to specific individuals or

Status: @Launched

Shipping: 5/18/2020

Problem Alignment

The Problem

Describe the problem (or opportunity) you're trying to solve. Why is it important to our users and our business? What insights are you operating on? And if relevant, what problems are you not intending to solve?

High-level Approach

Describe briefly the approach you're taking to solve this problem. This should be enough for the reader to imagine possible solution directions and get a very rough sense of the scope of this project. (e.g., if "The Problem" was about user engagement in the design process from non-designers, "The Approach" might be a feed which surfaces work that's relevant to them.)

Goals & Success

What does success look like? What metrics are you intending to move? Explain why these metrics are important if they are not obvious.

Problem Alignment

The Problem

High-level Approach

Goals & Success

Solution Alignment

Key Features

Key Flows

Open Issues & Key Decisions

Launch Readiness

Key Milestones

Launch Checklist

Intercom

INTERMISSION NAME

MM/YY

What problem are we solving and why?

Add the problem we're solving (or speculative opportunity we are addressing), why we're solving it, any links to customer conversations or research. Clearly explain all facets of the problem and how it affects different parts of Intercom.

Job Stories

Describe job stories to make the value we deliver to customers clear, and what it will enable them to do: When ____, I want to be able to ____, so I can ____.

How will we measure success?

Add how we will determine whether the problem has been solved. Qualitative and quantitative measures.

Scope

Once high level product design is figured out, add the releases below, what is in scope and out of scope, and estimated dates for shipping to beta.

Notes:

- Do not add the solution here.
- An Intermission must always fit on a printed A4 page. If it does not, you haven't a clear enough view of the problem yet. Keep working on it.
- Always have active and upcoming Intermissions printed in your team area or war room.
- Always use plain simple English, no technical terminology or codenames. Write this document as you would describe the problem to a colleague face to face.
- The PM owns the Intermission, but should always solicit input from the full team.

Como definir o backlog?

- COMUNICANDO O QUE VAMOS FAZER E O PORQUÊ
- **DESCOBRINDO A SOLUÇÃO**



Definições da solução

(também conhecido como especificação)

BLOCO 3

Mapa mental

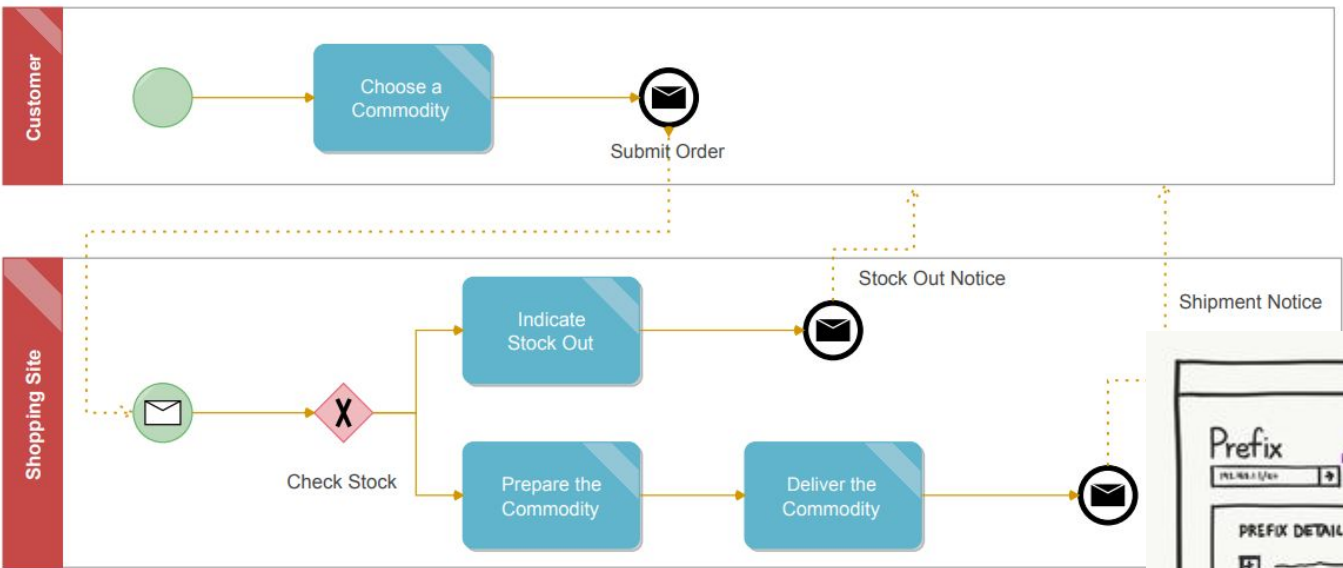
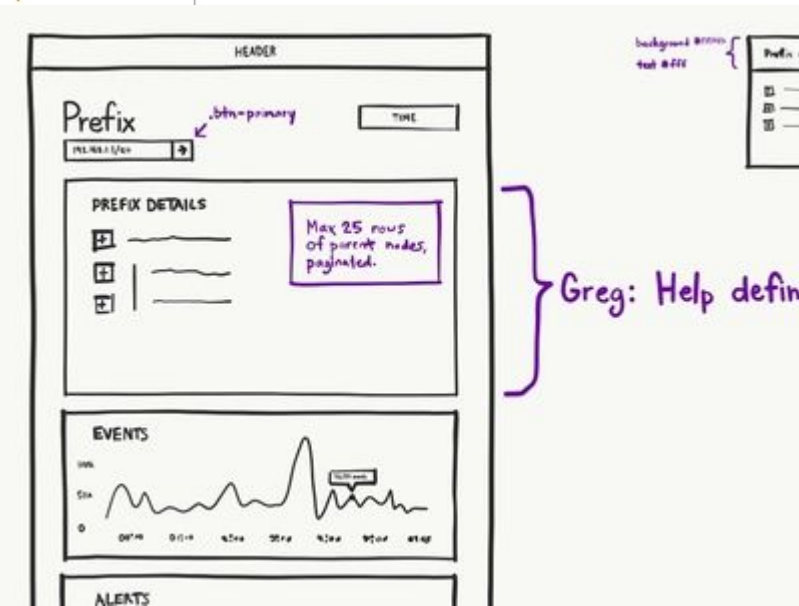
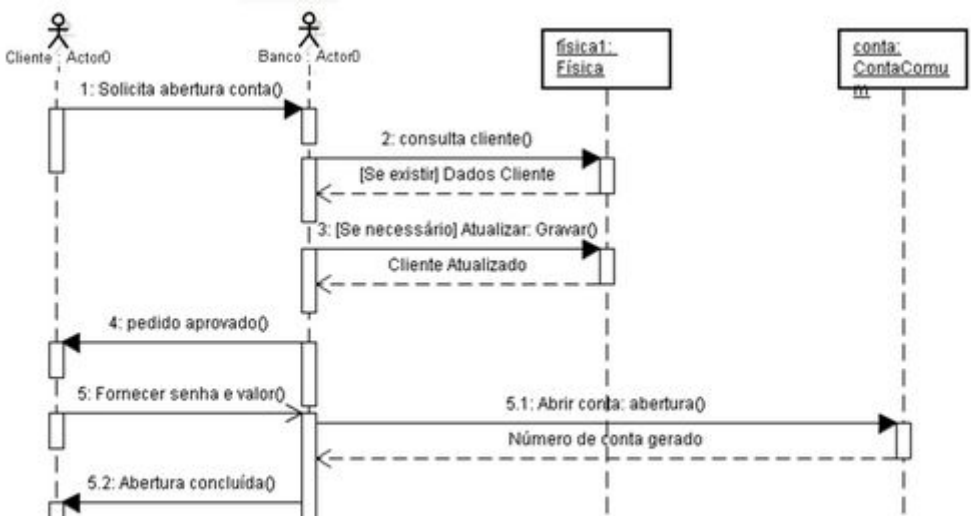


Diagrama de BPMN

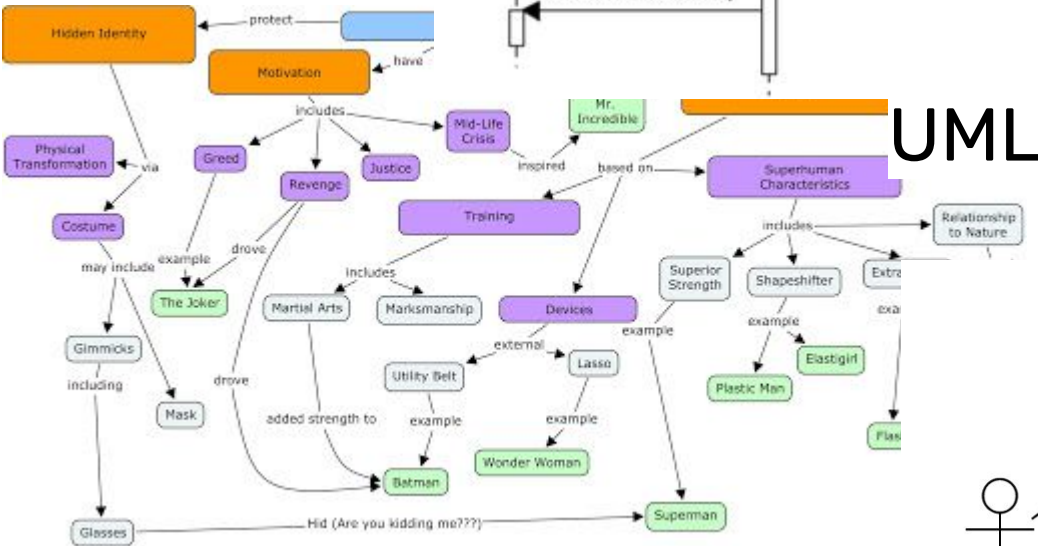


Wireframe anotado

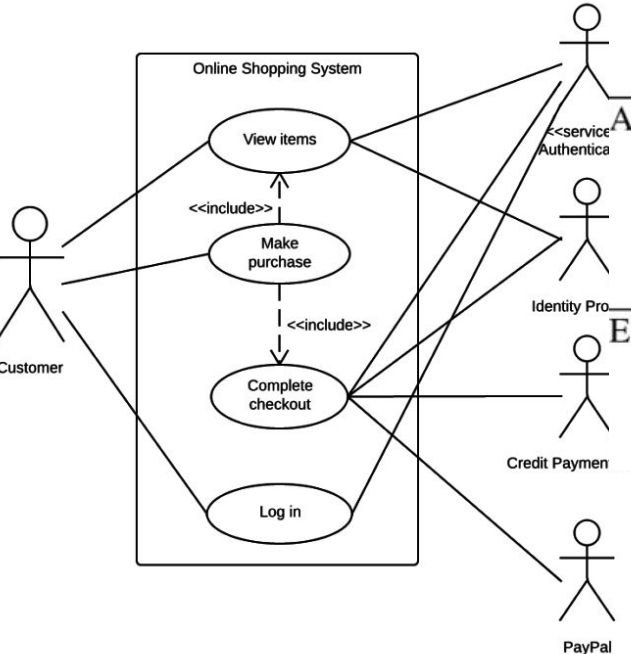
Fluxograma



UML

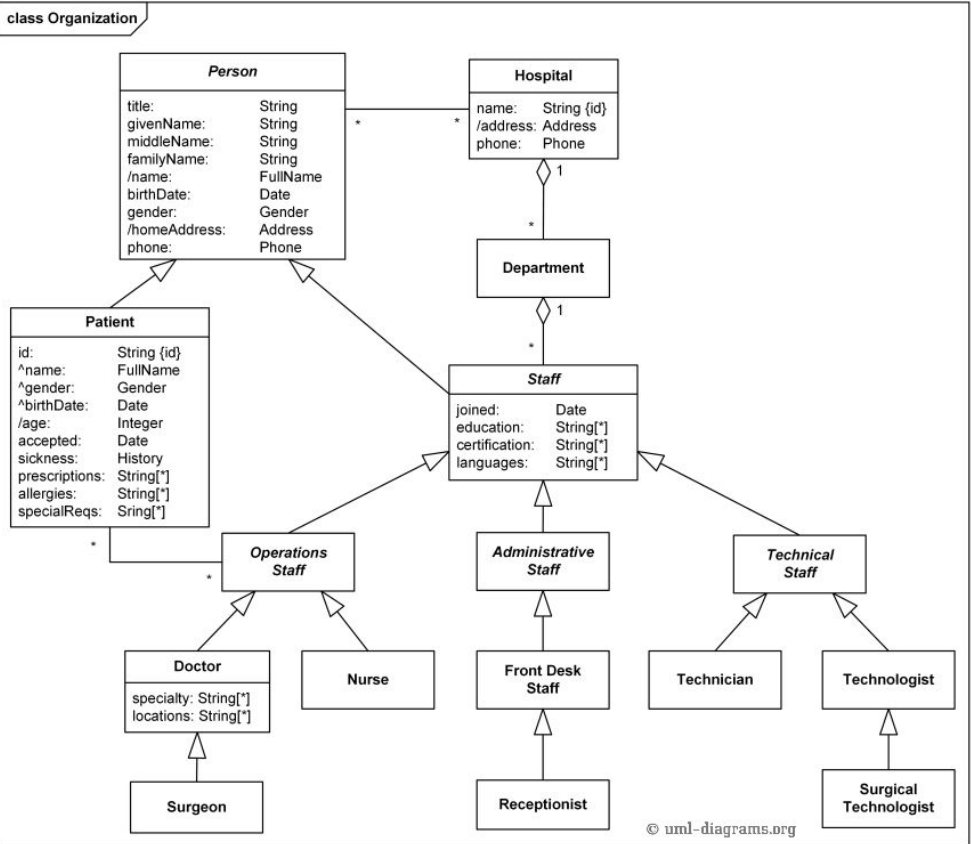


Mapa de conceito



Caso de uso

Há **muitas** formas de especificar!



Modelo de dados e domínio

User ou job story

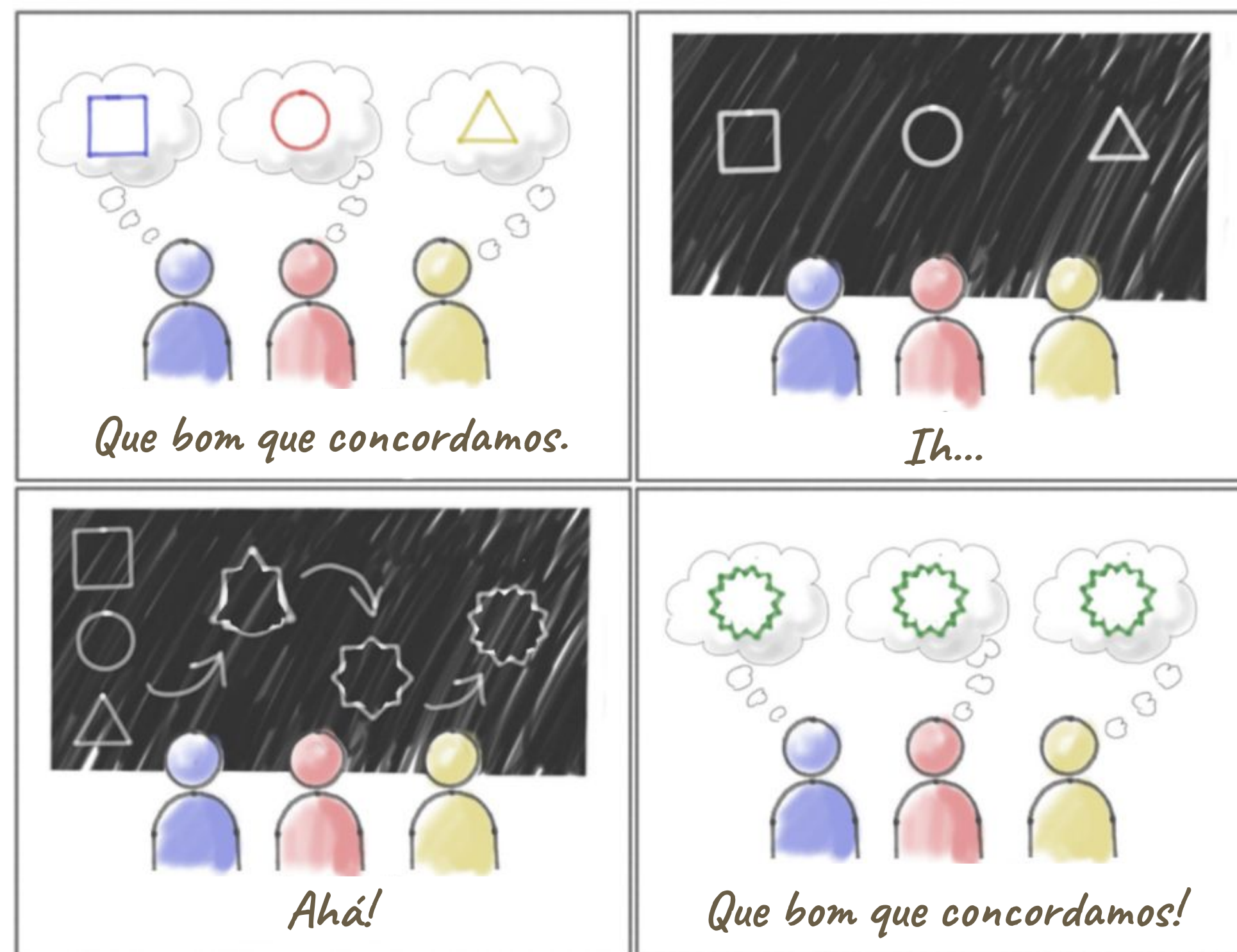
Documento de requisitos e regras de negócio

Descrição do caso de uso

Feature/funcionalidade

Use Case	Draw Image
Pre-condition	<ul style="list-style-type: none">Frame buffer must be clearedReset must switch off
Post-condition	<ul style="list-style-type: none">Image data output to VRAM
Basic path	<ol style="list-style-type: none">This use case starts while Host CPU has already prepared the display list for drawing image.Host CPU writes display list system's FIFO.2D/3D Graphics reads display list from FIFO and starts drawing image according to commands of display list.2D/3D Graphics outputs image data to VRAM.
Alternative path	At step 2 of the basic path, Host CPU writes display list that includes SYNC command to system's FIFO.
Exceptional path	At step 3 of the basic path, 2D/3D Graphics stops after reading SYNC command, and then 2D/3D graphics restarts drawing image after receiving a blank pulse from LCDC. At step 3 of basic path, 2D/3D graphics raises an error interrupt to Host CPU and clears system's FIFO.

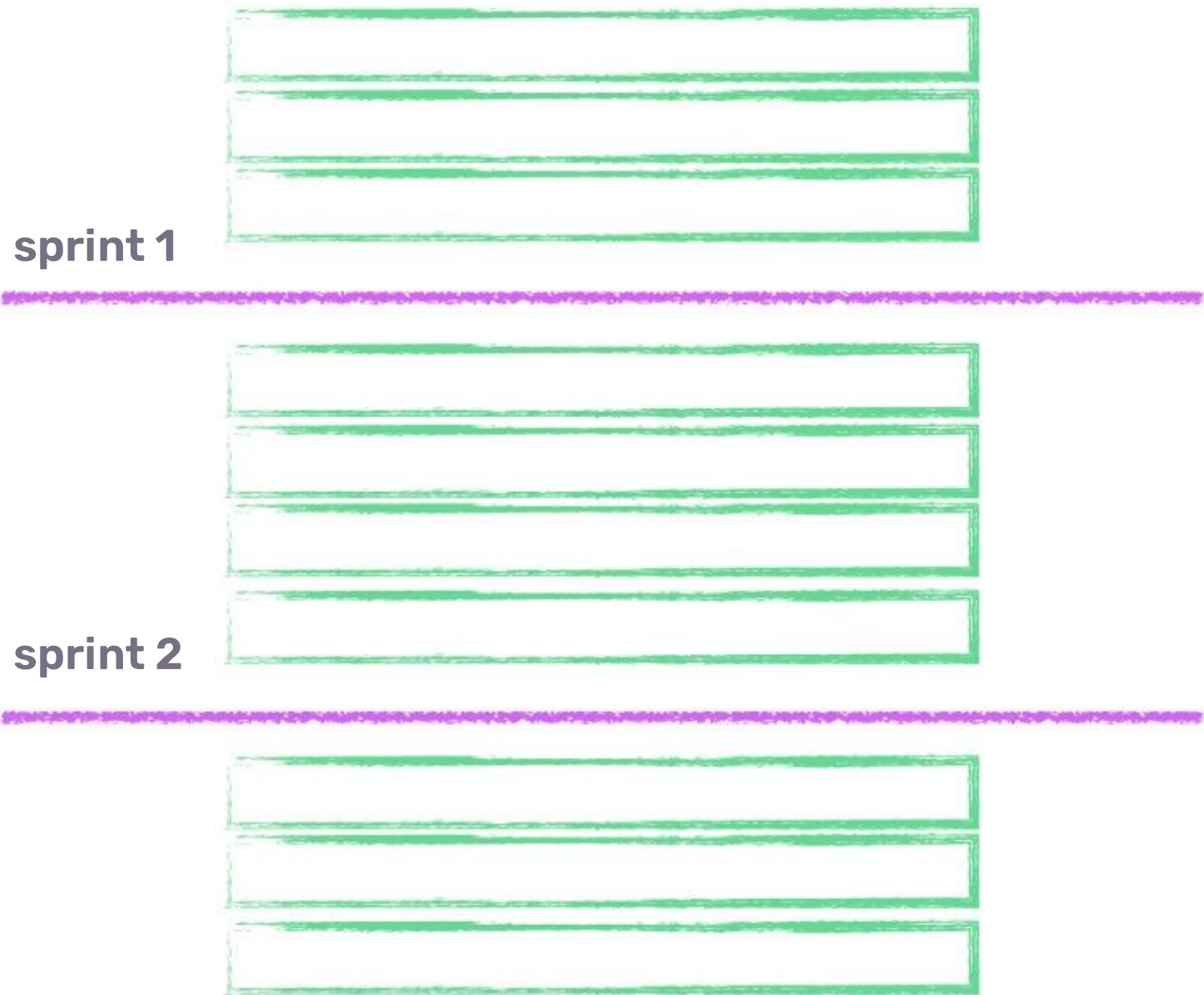
**Objetivo de uma
especificação:
Chegar a um
entendimento
compartilhado**



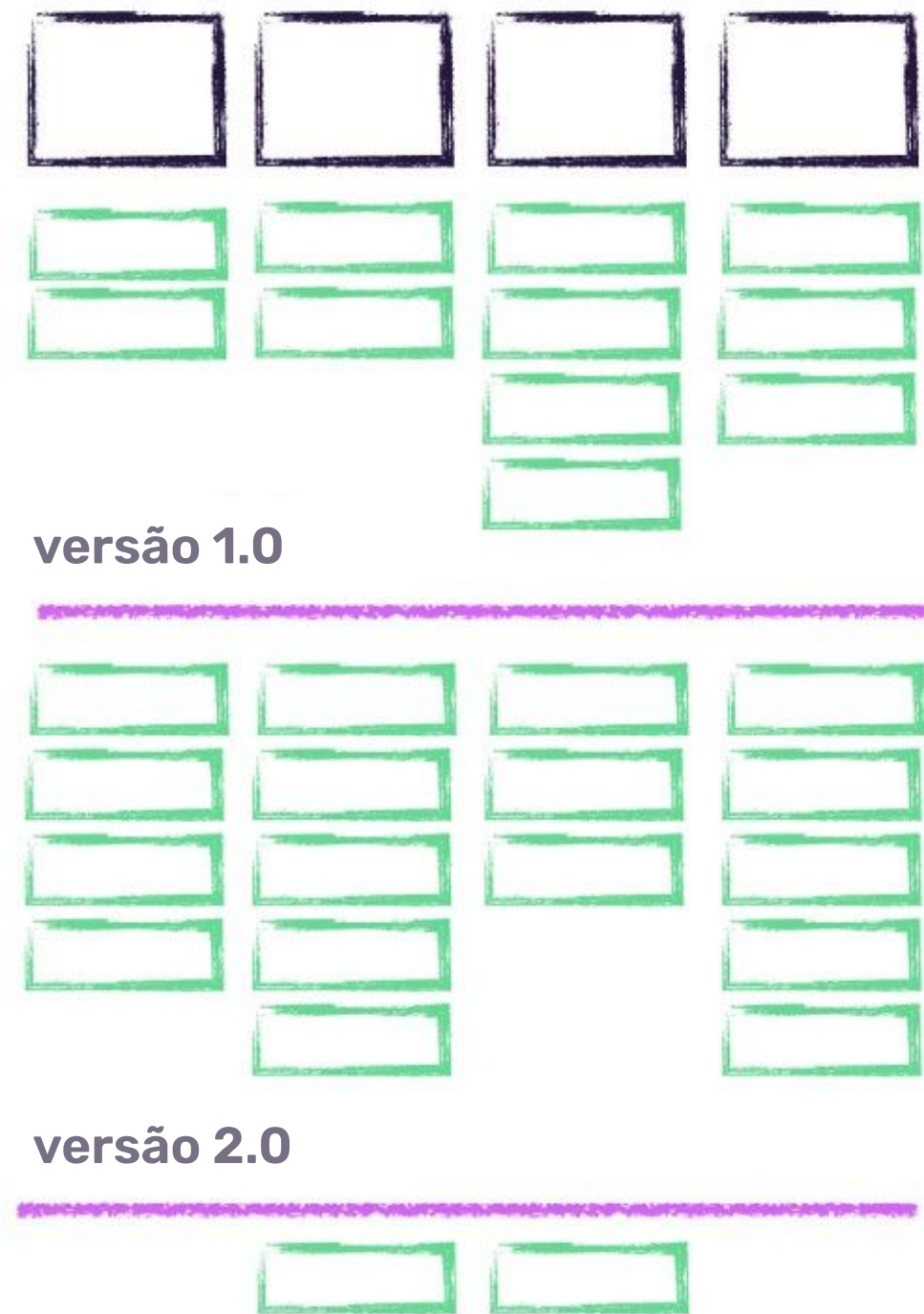
User story mapping

0 backlog bidimensional

backlog normal



story map



User story mapping

Pra que serve?

Serve para...

- Visão do todo!
- **Contar a história** da jornada
- Descobrir histórias e cenários
- Fatiar histórias
- Alinhamento com stakeholders
- Planejamento de releases (Priorização 🧠🧠)



Dica: Template do Miro

User story mapping

Como funciona?



User story mapping

Como funciona?



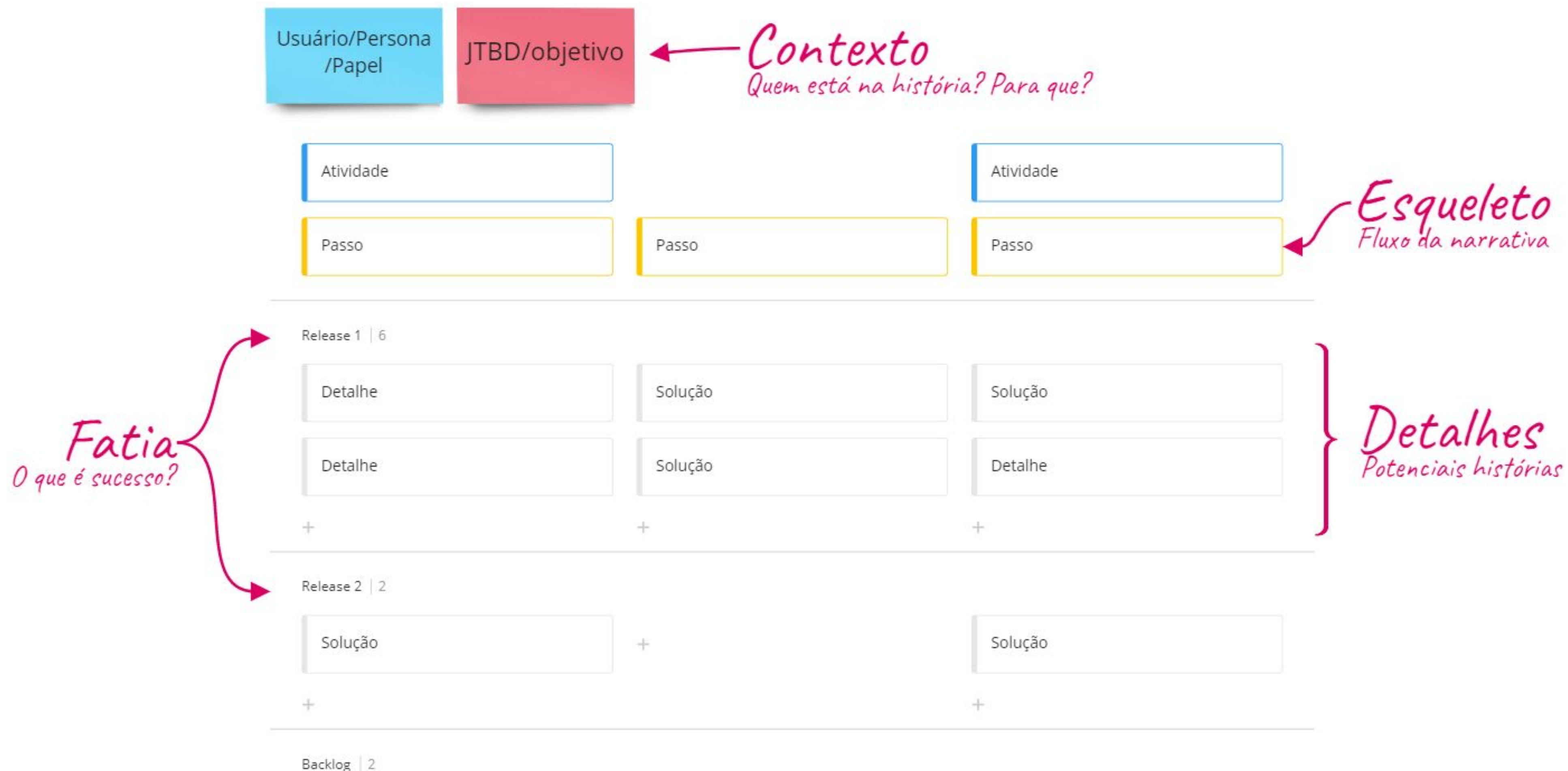
User story mapping

Como funciona?



User story mapping

Como funciona?

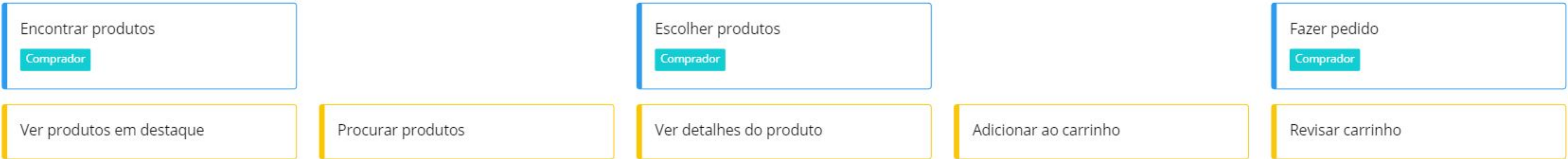


T User story mapping

Como funciona?

Comprador

E-commerce XYZ



Backlog | 33



User story

- Descreve **algo que um usuário vai poder fazer**, a partir da sua perspectiva
- Substitui o “requisito”
- Resume:
 - O que
 - Por que
 - Para quem

Eu, como **aluna do curso de DPL da Tera**,
quero **acessar gravações das aulas** para
poder assistir depois.

Como usuário **[persona]**
quero _____ **[ação]**
Para [que] _____ **[benefício]**

3 Cs

- Card
- Conversation
- Confirmation

Job story

- Descreve uma **necessidade** em um **contexto** - quanto mais, melhor!
- Foca mais na **motivação** do que na implementação

Quando **estou lotada de trabalho**, com **um monte de coisas na cabeça**, e no fim do dia **não tenho condições de prestar atenção** em uma aula de 2h30, quero poder **assistir a aula depois** para **conseguir aproveitá-la quando estiver com a cabeça no lugar**.

Como _____ [persona]
quero _____ [ação]
Para [que] _____ [resultado]

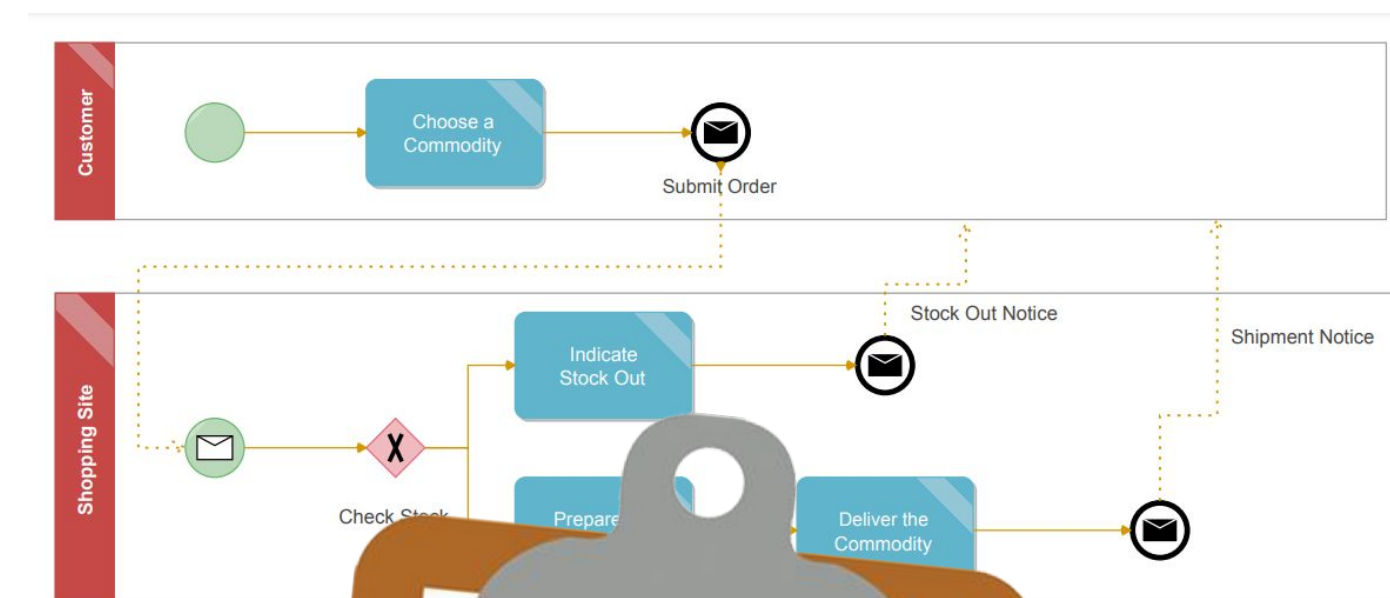
Quando _____ **[situação]**
quero _____
[motivações/forças]
Para [que] _____ **[resultado]**

Critérios de aceite

Enunciados **pequenos e de fácil entendimento**, numa **linguagem informal** e que nossos usuários poderiam entender.

Usados para **determinar quando a funcionalidade produzida está completa e nada mais precisa ser feito** - por isso pensamos neles como uma **checklist** da feature ou user/job story.

Ajudam a criar e confirmar o **entendimento compartilhado** no time - e por isso não são escritos em pedra! Sempre podem ser **negociados** :)



Definindo detalhes

() Parênteses: precisa seguir os **templates** para escrever boas histórias?

Já dizia Jeff Patton na Agile India em 2013...

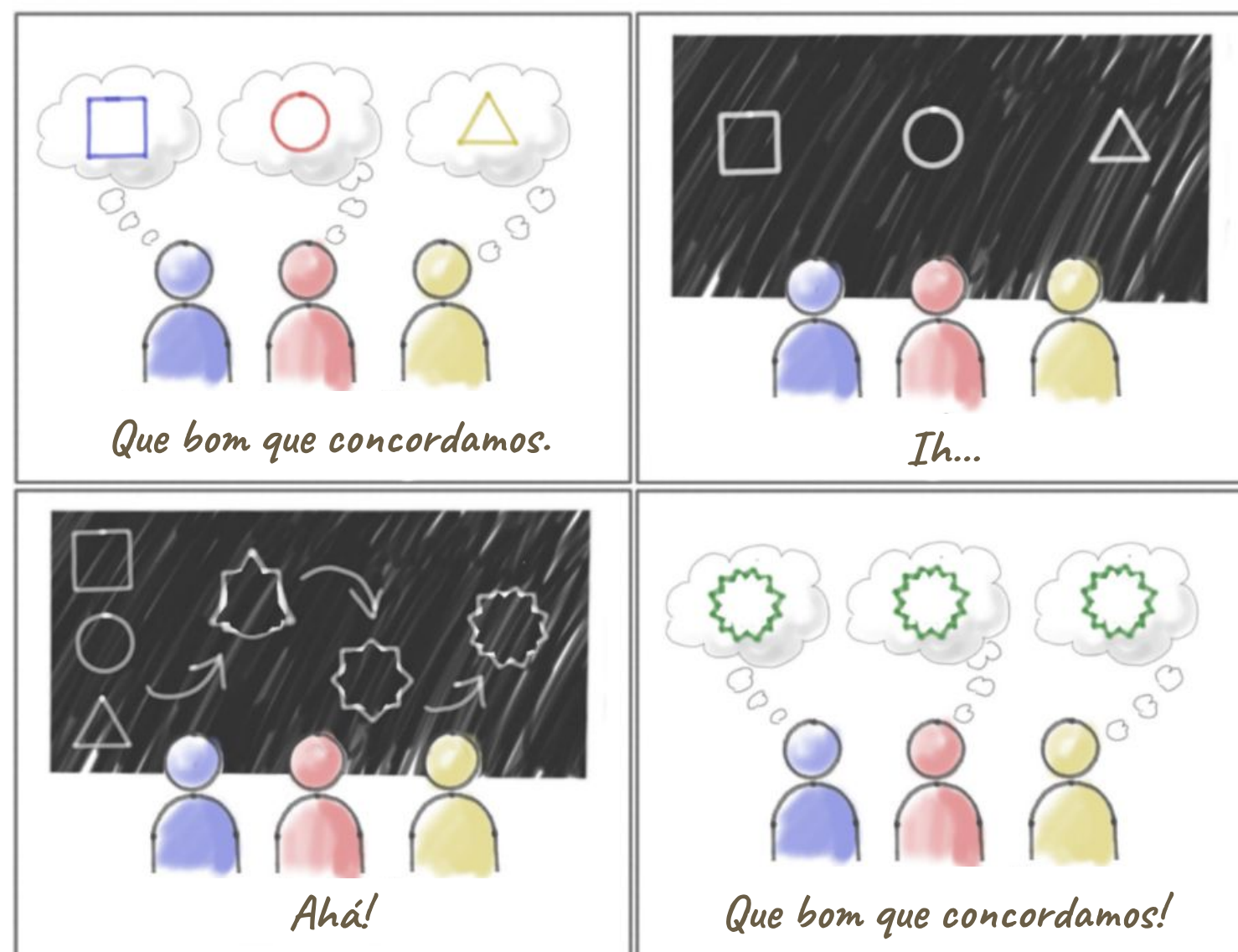
Histórias **não** são um **jeito de escrever requisitos melhores!**

Escrever um monte de requisitos detalhados e entregar para alguém ler **não funciona.**

O objetivo é a **colaboração** das pessoas que precisam de algo com as que o constróem.



Já dizia Jeff Patton na Agile India em 2013...



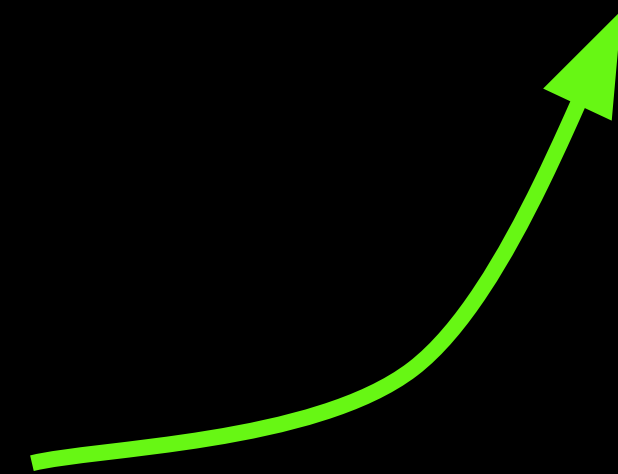
Lembrando dos 3Cs...

Não importa o que você escreve no **cartão**, só escreva alguma coisa pra ajudar a iniciar a **conversa**.

Você conta a sua história, e a pessoa pode fazer perguntas - e *depois* registrar a **confirmação** do entendimento compartilhado.



“Eu digito um CEP e ele preenche automaticamente a cidade e o estado, sem eu ter que tocar um botão!”



A ideia das user stories nasceu de ouvir pessoas **contando histórias sobre novas coisas legais** que o software que usavam estava fazendo



Jeff Patton

Outros PBIs (Product Backlog Items)

» Tarefas e melhorias

Ex.: Tracking, extração de dados, atualização de bibliotecas, refatoração...

» Problemas

Bugs

Débitos técnicos

Atalhos técnicos
escolhidos na
implementação para
acelerar a entrega


Não são histórias, mas também precisam de um **porquê!**

- Custo de não fazer nada
- Criticidade vs. Frequência/Probabilidade

13527 Tracking de eventos no carrinho e checkout

 daniel

 0 comments [Add tag](#)

State	 On Analysis	Area	Kyte\Catalog
Reason	 Moved to state On Analysis	Iteration	Kyte\Product

Description

why

Para poder melhorar o funil de compra do catálogo, precisamos entender onde as pessoas mais saem.

what

Adicionar eventos e propriedades de sessão que nos ajudem a entender o comportamento dos consumidores.

Propriedades de sessão

Referência: https://coda.io/d/Product-Analytics_die6hdn_NEO/Propriedades-de-sessao_sutOe#_lusKT

Propriedade	Descrição	Valores possíveis
catalog_orders_enabled	Indica se o catálogo acessado está com pedidos habilitados.	true/false
catalog_online_payment_enabled	Indica se o catálogo acessado está com pagamento online habilitado.	true/false

Eventos

Referência: https://coda.io/d/Product-Analytics_die6hdn_NEO/Eventos-para-implementar_suQVe#_lukq6

T

Por onde começar?

Prioridades

Priorizar é **fazer escolhas** em relação a um **objetivo**.

> Avaliar opções a partir de:

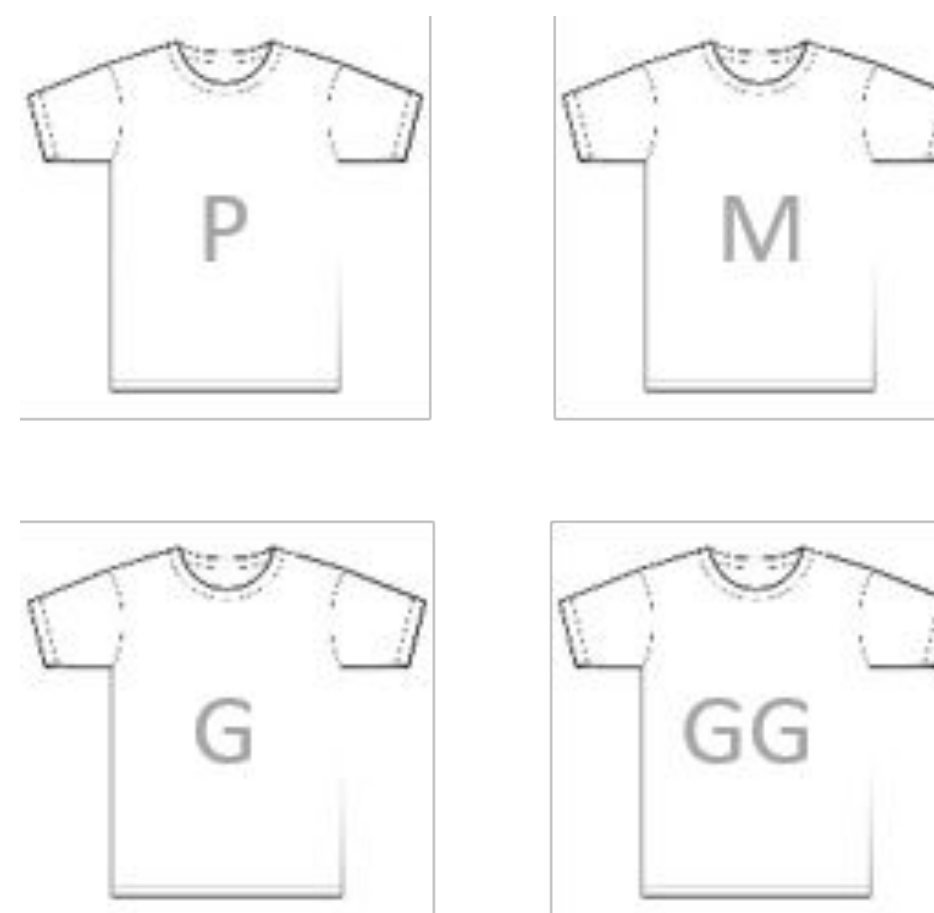
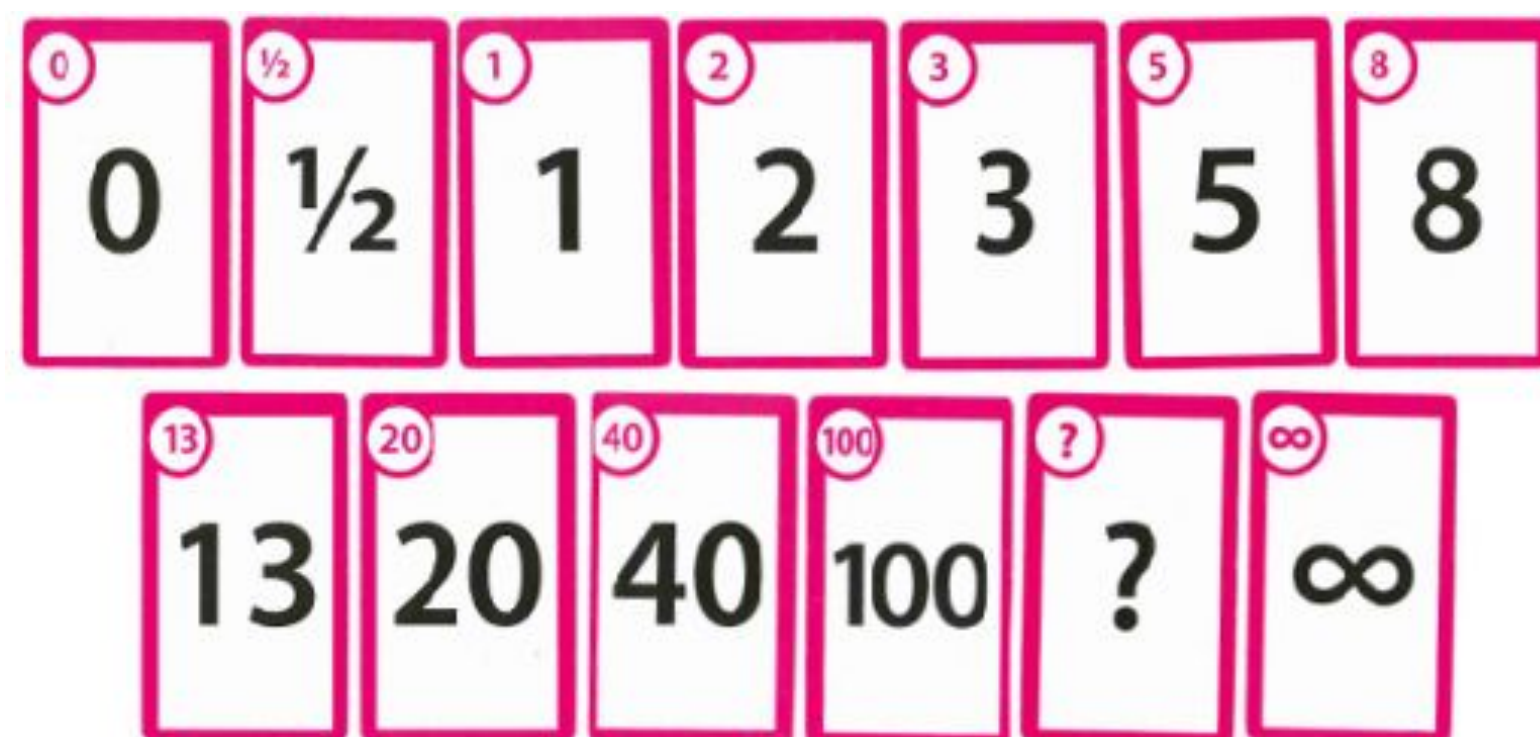
- Critérios
- Dados
- Influências
- Intuição

> Escolher

> **Comunicar** o que vai e o que não vai fazer

Estimativas

- São **estimadas** - não são compromisso!
- Tamanho **relativo** é mais fácil e assertivo
- O mais valioso: as **discussões** em torno dela!



Dinâmica pro time estimar

A

B

C

D

P



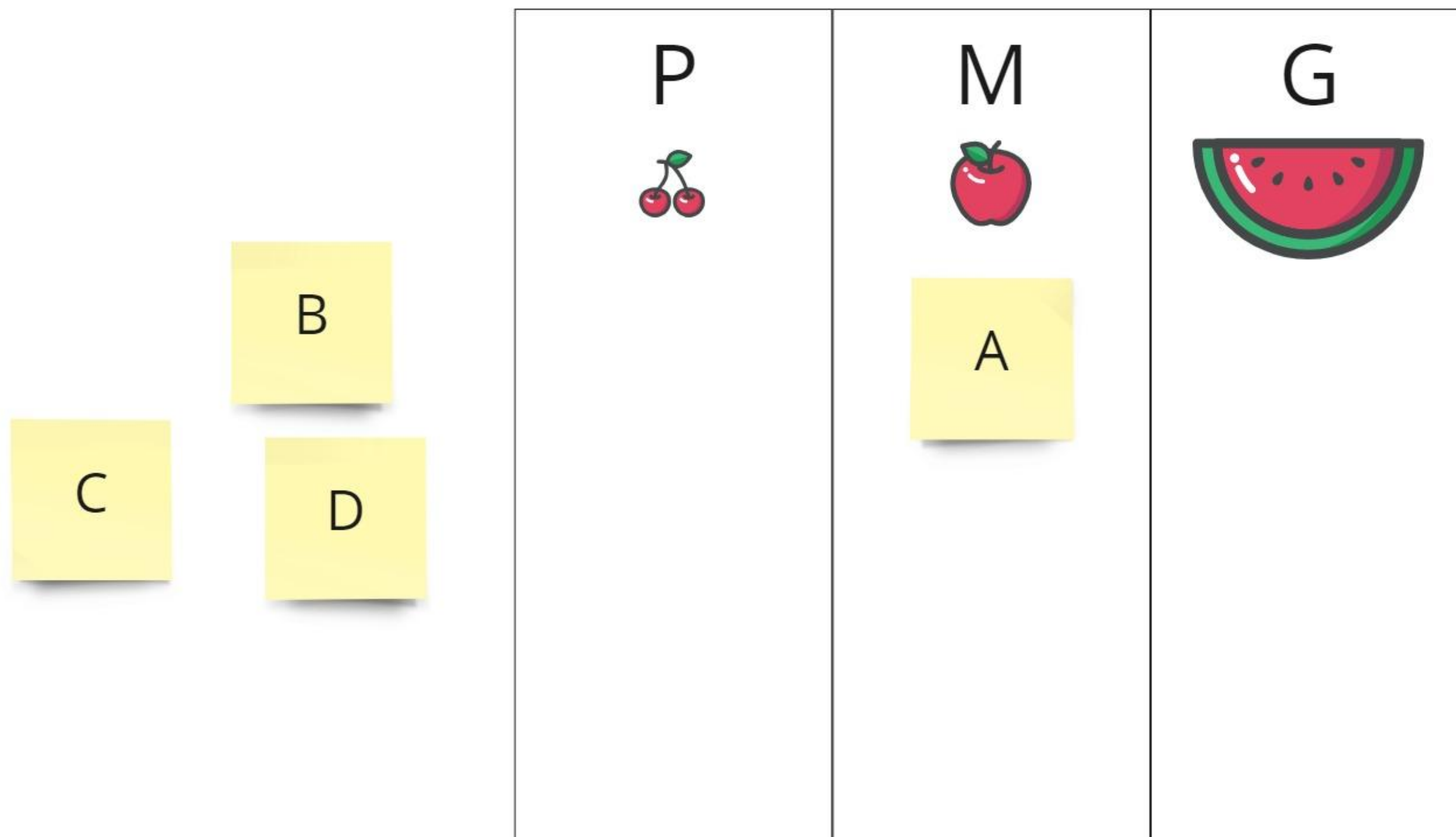
M



G



Dinâmica pro time estimar



Dinâmica pro time estimar



P



B

M

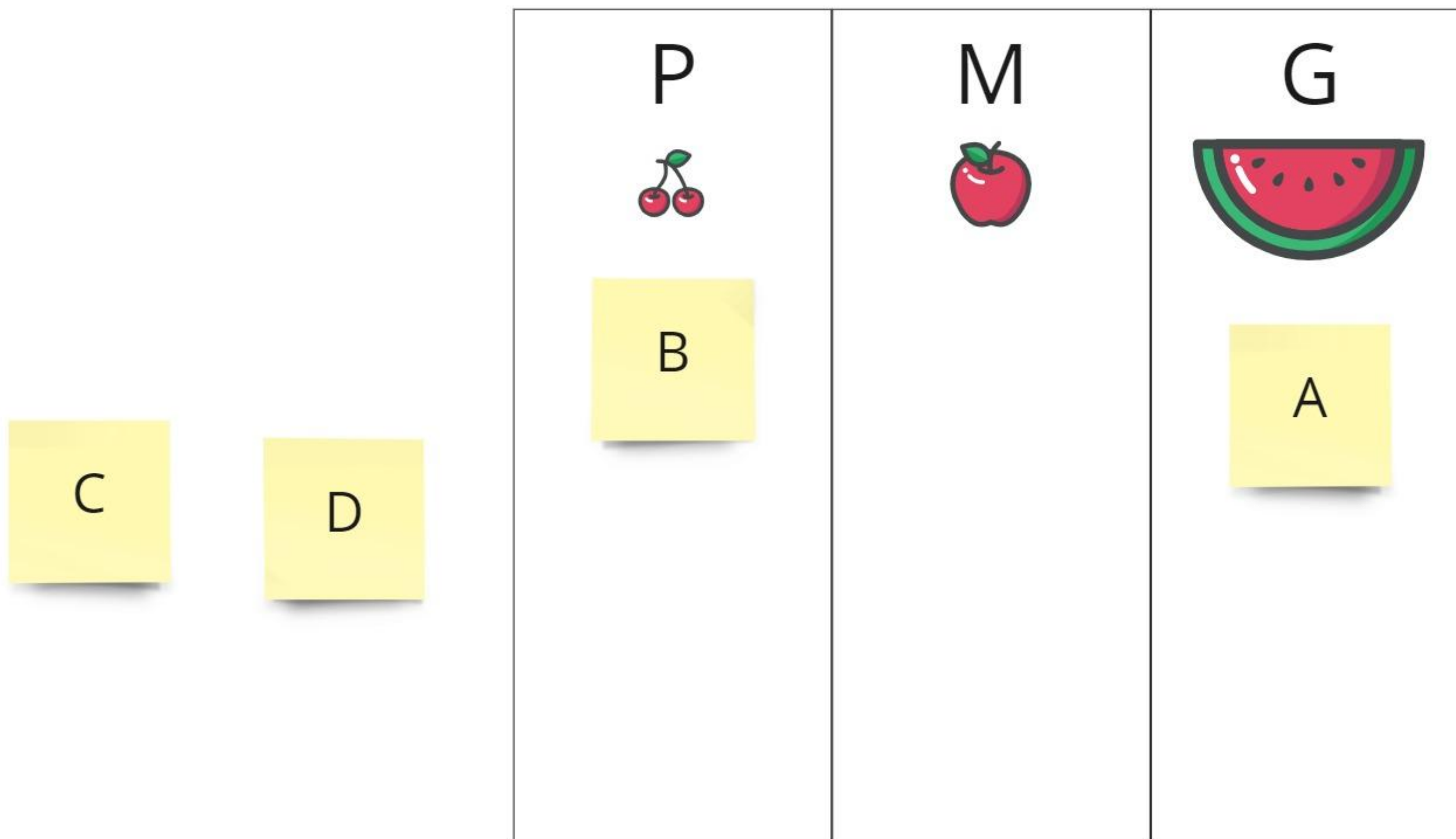


A

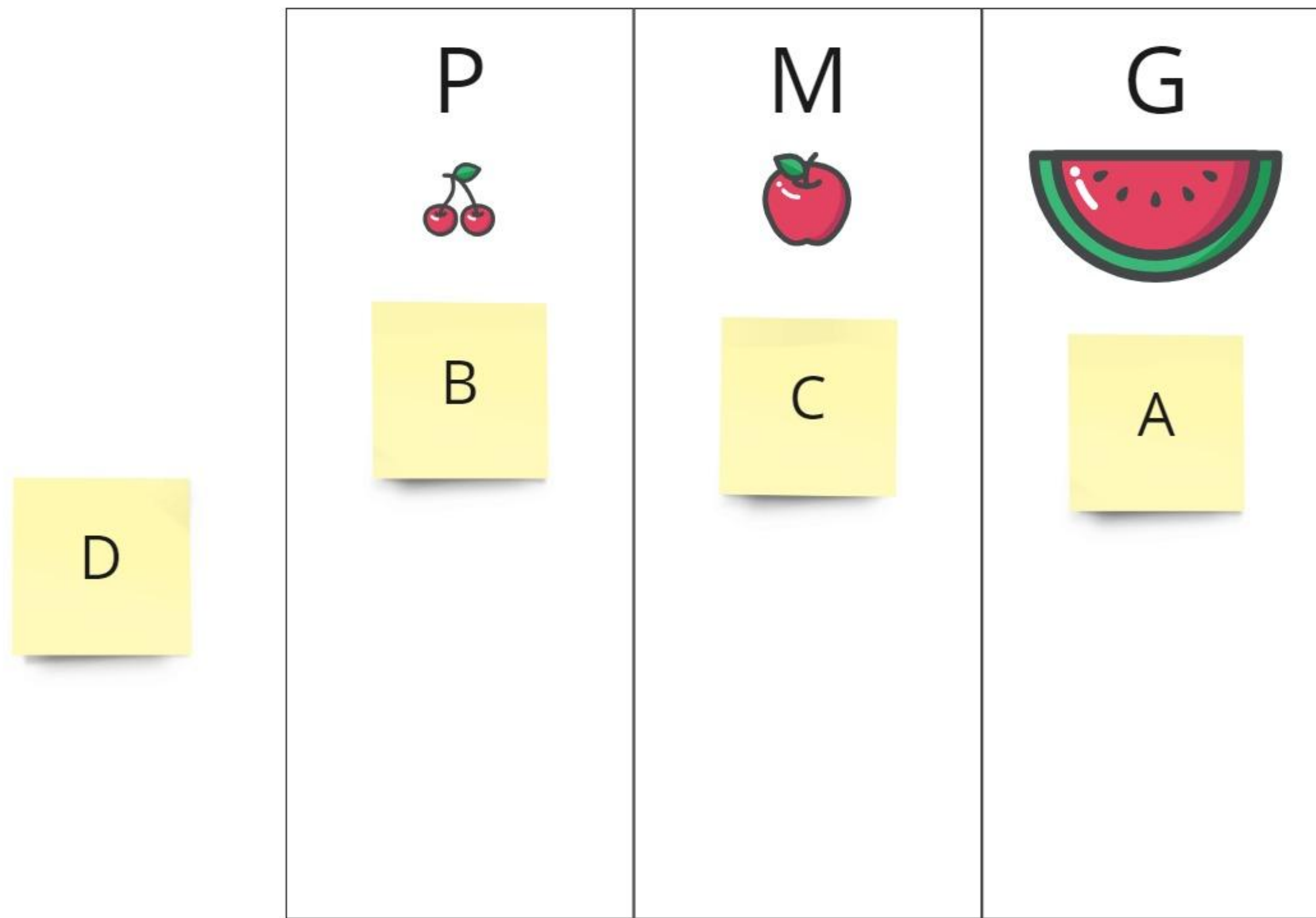
G



Dinâmica pro time estimar






Dinâmica pro time estimar



Dinâmica pro time estimar



P	M	G
		
<div>B</div>	<div>C</div>	<div>A</div>
		<div>D</div>

Atalhos de priorização: frameworks

Frameworks são estratégias para simplificar o processo de decisão.

Qualitativos

- MoSCoW
- Kano (~quanti também)
- Feature Buckets
- Ordenação de lista
- ...

Quantitativos (!)

- Valor x Custo/Esforço
- Score ponderado
- RICE
- ICE
- Projeção de ROI
- Cost of delay
- Opportunity Scoring (JTBD)
- ...

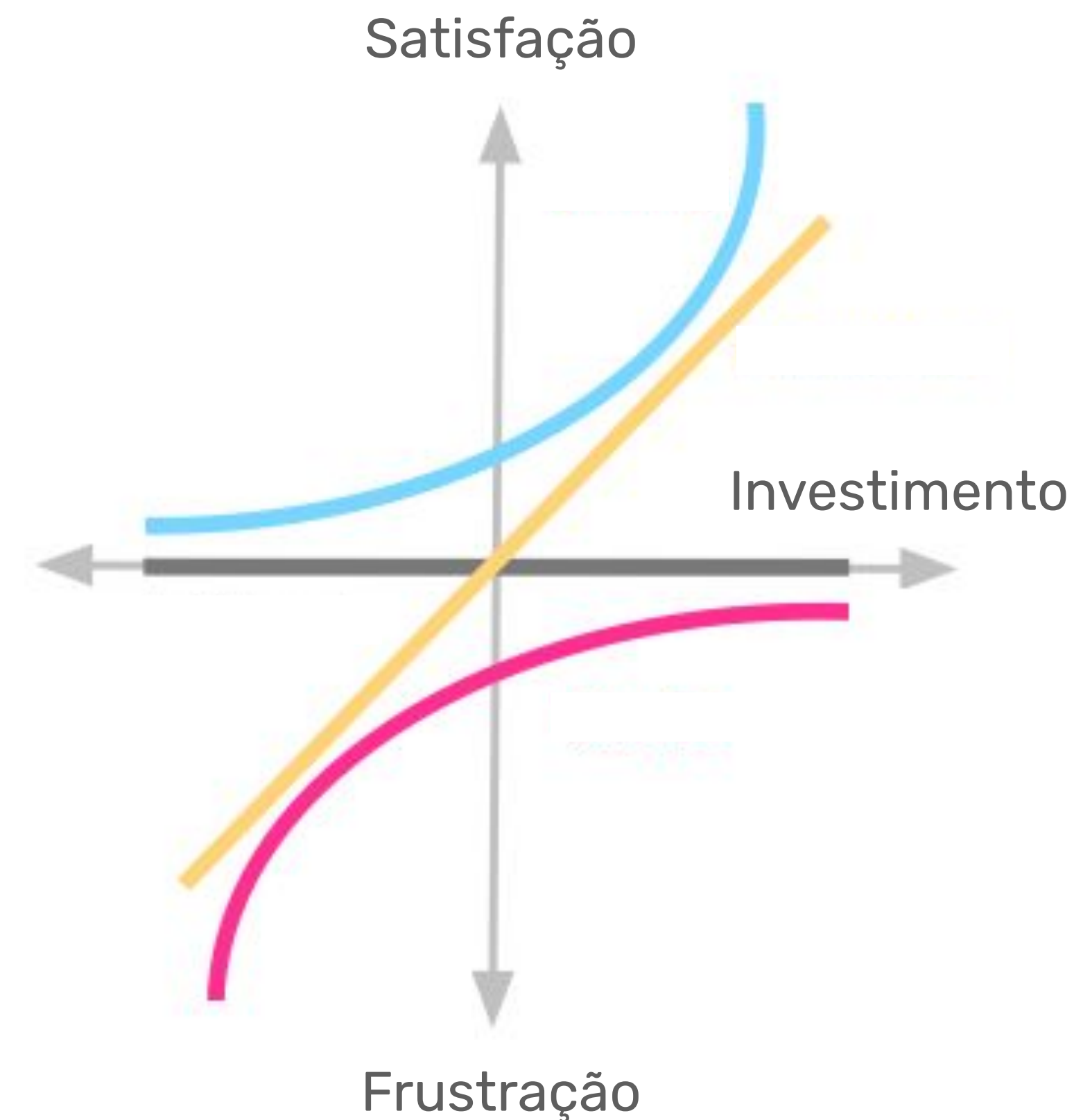
Colaborativos

- Buy a feature
- Prune the product tree
- Story Mapping
- Team Backlog Ranking
- Sequenciador de features
- ...

Kano model

Percepção de valor de clientes e usuários

- **Básica**
Não ter gera frustração
E investir mais não faz diferença
- **Performática**
Quanto mais investe, melhor!
- **Encantadora**
Se não tiver, não faz falta; mas se tem... 🥰
E nem sempre precisa de muito investimento!
- **Indiferente**
Não importa quanto invista, não gera valor





Nenhum framework vai cuspir uma ordenação pronta para ser seguida.

- Eles se **complementam**
- Visão e experiência **contam!**
- Tudo depende do seu **objetivo**
(que pode mudar)

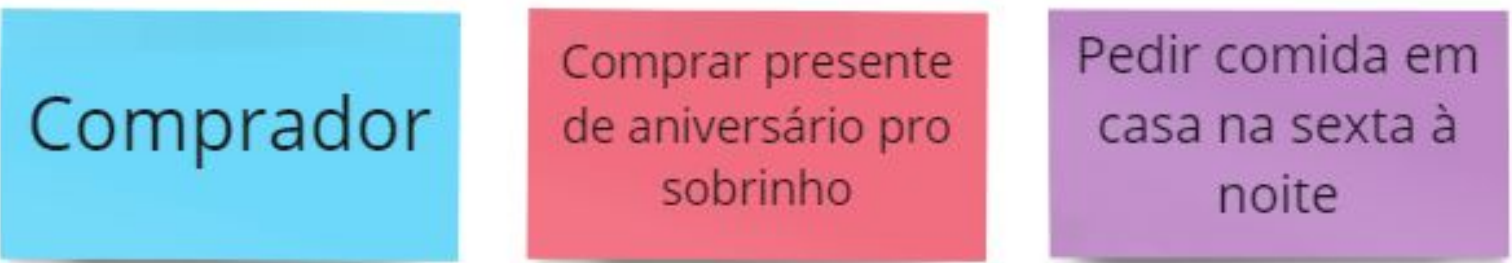
Desenvolvimento iterativo

Também precisa ser considerado na priorização...

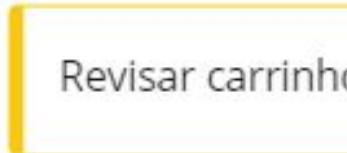
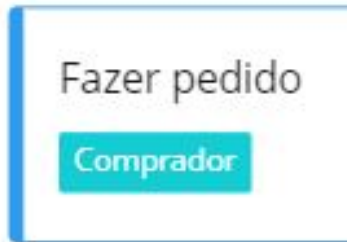
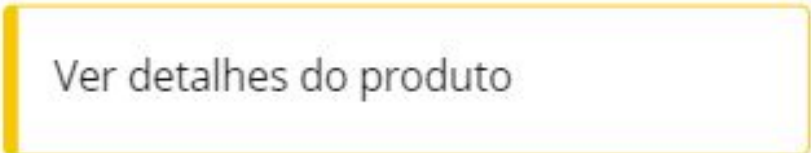
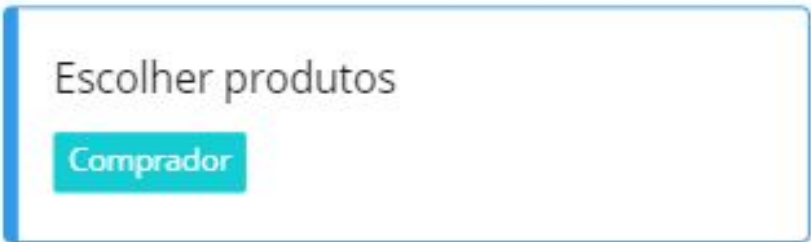
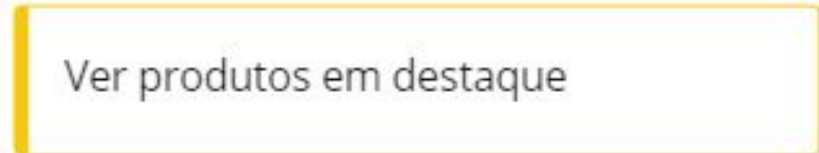


... e a prioridade **sempre** depende do objetivo 😊

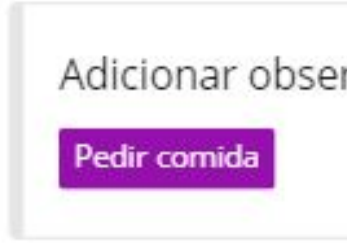
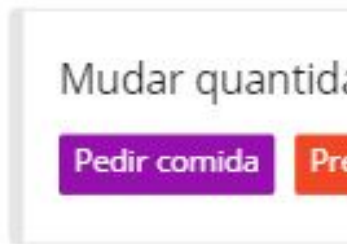
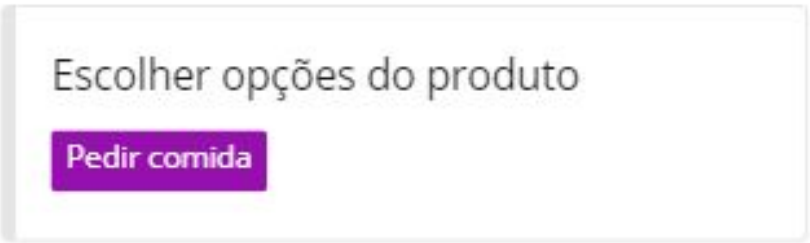
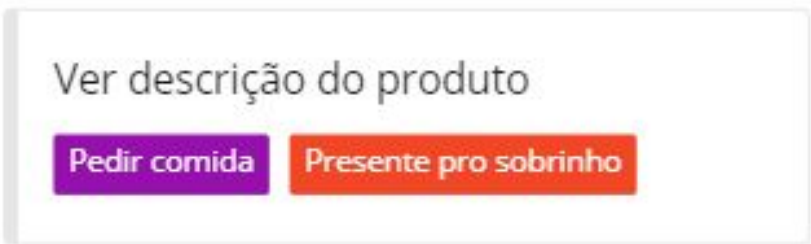
Priorizando por objetivos



E-commerce XYZ



Backlog | 32



Priorizando por objetivos

Comprador

Pedir comida em casa na sexta à noite

E-commerce XYZ

Encontrar produtos

Comprador

Escolher produtos

Comprador

Fazer pedido

Comprador

Ver produtos em destaque

Procurar produtos

Ver detalhes do produto

Adicionar ao carrinho

Revisar carrinho

Release 1: Replicar experiência de cardápio | 6

+

Buscar produtos por categoria

Pedir comida

Presente pro sobrinho

Ver descrição do produto

Pedir comida

Presente pro sobrinho

Colocar produto no carrinho

Pedir comida

Presente pro sobrinho

Mudar quantidade de produto

Pedir comida

Presente pro sobrinho

+

+

+

+

+

Escolher quantidade

Pedir comida

Remover produto

Pedir comida

Presente pro sobrinho

Release 2: Facilitar escolha de produtos | 2

+

Buscar produto por nome

Pedir comida

Presente pro sobrinho

Ver fotos do produto

Pedir comida

Presente pro sobrinho

+

+

+

Priorizando por objetivos

Comprador

Pedir comida em casa na sexta à noite

E-commerce XYZ

