

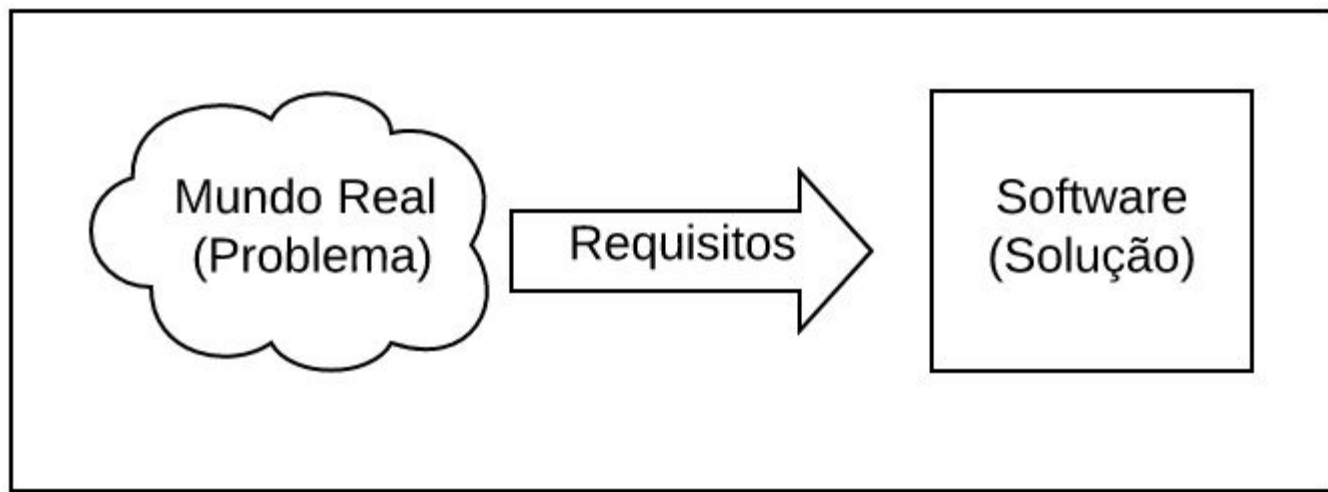
Requisitos de Software

Prof. Pedro Henrique Dias Valle

A parte mais difícil da construção de um software é a definição do que se deve construir -- Fred Brooks

Requisitos

- Requisitos funcionais: o que um sistema deve fazer
 - suas funcionalidades
- Requisitos não-funcionais: sob que restrições
 - Desempenho, segurança, disponibilidade, etc

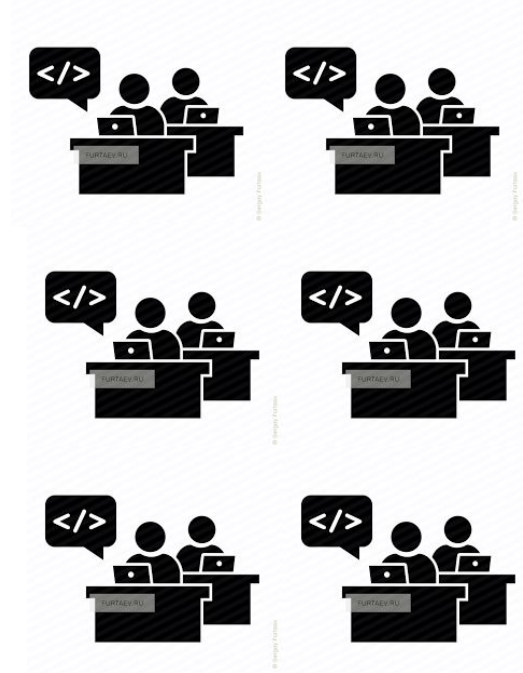


O que vamos estudar?

- Histórias de usuários
- Casos de Uso
- Produto Mínimo Viável (MVP)
- Testes A/B

Histórias de Usuários

Antes ...



Analistas

(PRD: program requirement document)

Programadores
(Fábrica de Software)

Hoje ...



Product Owner senta junto dos desenvolvedores e
"explica" requisitos para eles

Histórias de Usuários = 3C's

- **C**artão + **C**onversas + **C**onfirmação
- Cartão: lembrete para conversas sobre o requisito durante sprint
- Confirmação: cenários que serão usados pelo PO para aceitar a implementação da história (escritos no verso do cartão); também chamados de testes (ou critérios) de aceitação

Exemplo: Loja Virtual

- Cartão: "Fechar uma compra"
- Conversas: PO explica os meios de pagamento; as formas de entrega, formas de parcelamento, etc
- Confirmação:
 - Testar compra à vista e compra parcelado
 - Testar com cartões de crédito A, B e C
 - Testar com modos de entrega X e Y

Workshop de Escrita de Histórias (ou Inception)

- Realizado no início do projeto com os principais usuários
- Objetivos:
 - Definir o que o produto vai fazer \Rightarrow lista inicial de histórias
 - Definir o que o produto **não** vai fazer

Sprint de Setup

- Antes de começar um projeto, alguns times realizam também um “sprint de setup”
- Objetivo: configurar o ambiente e as ferramentas de desenvolvimento

Formato escrita de histórias

- Como um [certo tipo de usuário],
eu gostaria de [realizar algo com o sistema]

Exemplo: sistema de controle de bibliotecas

3 tipos de usuários

- Aluno
- Professor
- Funcionário da biblioteca

Histórias do usuário típico

- Como aluno, eu gostaria de realizar empréstimos de livros
- Como aluno, eu gostaria de devolver um livro que tomei emprestado
- Como aluno, eu gostaria de renovar empréstimos de livros
- Como aluno, eu gostaria de pesquisar por livros
- Como aluno, eu gostaria de reservar livros que estão emprestados
- Como aluno, eu gostaria de receber e-mails com novas aquisições

Histórias de professores

- Como professor, eu gostaria de realizar empréstimos de maior duração
- Como professor, eu gostaria de sugerir a compra de livros
- Como professor, eu gostaria de doar livros para a biblioteca
- Como professor, eu gostaria de devolver livros em outras bibliotecas

Histórias de funcionários da biblioteca

- Como funcionário, eu gostaria de cadastrar novos usuários
- Como funcionário, eu gostaria de cadastrar novos livros
- Como funcionário, eu gostaria de dar baixa em livros estragados
- Como funcionário, eu gostaria de obter estatísticas sobre o acervo
- Como funcionário, eu gostaria que o sistema envie e-mails de cobrança para alunos com empréstimos atrasados
- Como funcionário, eu gostaria que o sistema aplicasse multas quando da devolução de empréstimos atrasados

Características de boas histórias (INVEST)

- **I**ndependentes
- Abertas para **N**egociação
- Agregar **V**alor
- **E**stimáveis
- **S**ucintas
- **T**estáveis

Histórias \Leftrightarrow Requisitos Funcionais

E os requisitos não-funcionais?

Requisitos não Funcionais (RNF)

- Time deve definir RNF com o PO
- Time deve incluir RNF nos "done criteria"

Exemplo

- Suponha que desempenho seja um RNF importante
- Pode-se definir que história para ser considerada **pronta** deve:
 - Passar por uma revisão de código focada em desempenho
 - Passar por testes de desempenho, com carga real

CrITÉRIOS de Aceitação de HistÓrias (ou Testes de Aceitação)	CrITÉRIOS de Conclusão de HistÓrias (Done Criteria)
<ul style="list-style-type: none"> ● Criar um formulário ● Enviar uma ou duas respostas para ele (como usuário normal) ● Fechar o formulário para recebimento de respostas ● Enviar uma nova resposta e verificar se não é possível 	<ul style="list-style-type: none"> ● Cobertura de testes automatizados $\geq 80\%$ ● Sistema de linter executado com sucesso ● UX Designer aprovou as modificações no frontend ● Código revisado por pelo menos um outro membro do time ● Código integrado na branch “development”

Casos de Uso

Casos de Uso

- Documento mais detalhado de especificação de requisitos
- Uso **não** é tão comum com métodos ágeis
- Um ator realizando alguma operação com o sistema
- Incluem fluxo normal e extensões
- Extensões:
 - Exceções (ou erros)
 - Detalhamento

Transferir Valores entre Contas

Ator: Cliente do Banco

Fluxo normal:

1 - Autenticar Cliente

2 - Cliente informa agência e conta de destino da transferência

3 - Cliente informa valor que deseja transferir

4 - Cliente informa a data em que pretende realizar a operação

5 - Sistema efetua transferência

6 - Sistema pergunta se o cliente deseja realizar uma nova transferência

Extensões:

2a - Se conta e agência incorretas, solicitar nova conta e agência

3a - Se valor acima do saldo atual, solicitar novo valor

4a - Data informada deve ser a data atual ou no máximo um ano a frente

5a - Se data informada é a data atual, transferir imediatamente

5b - Se data informada é uma data futura, agendar transferência

Transferir Valores entre Contas

Ator: Cliente do Banco

Fluxo normal:

- 1 - Autenticar Cliente
- 2 - Cliente informa agência e conta de destino da transferência
- 3 - Cliente informa valor que deseja transferir
- 4 - Cliente informa a data em que pretende realizar a operação
- 5 - Sistema efetua transferência
- 6 - Sistema pergunta se o cliente deseja realizar uma nova transferência

Extensões:

- 2a - Se conta e agência incorretas, solicitar nova conta e agência
- 3a - Se valor acima do saldo atual, solicitar novo valor
- 4a - Data informada deve ser a data atual ou no máximo um ano a frente
- 5a - Se data informada é a data atual, transferir imediatamente
- 5b - Se data informada é uma data futura, agendar transferência

Transferir Valores entre Contas

Ator: Cliente do Banco

Fluxo normal:

- 1 - Autenticar Cliente
- 2 - Cliente informa agência e conta de destino da transferência
- 3 - Cliente informa valor que deseja transferir
- 4 - Cliente informa a data em que pretende realizar a operação
- 5 - Sistema efetua transferência
- 6 - Sistema pergunta se o cliente deseja realizar uma nova transferência

Extensões:

- 2a - Se conta e agência incorretas, solicitar nova conta e agência
- 3a - Se valor acima do saldo atual, solicitar novo valor
- 4a - Data informada deve ser a data atual ou no máximo um ano a frente
- 5a - Se data informada é a data atual, transferir imediatamente
- 5b - Se data informada é uma data futura, agendar transferência

Transferir Valores entre Contas

Ator: Cliente do Banco

Fluxo normal:

- 1 - Autenticar Cliente
- 2 - Cliente informa agência e conta de destino da transferência
- 3 - Cliente informa valor que deseja transferir
- 4 - Cliente informa a data em que pretende realizar a operação
- 5 - Sistema efetua transferência
- 6 - Sistema pergunta se o cliente deseja realizar uma nova transferência

Extensões:

- 2a - Se conta e agência incorretas, solicitar nova conta e agência
- 3a - Se valor acima do saldo atual, solicitar novo valor
- 4a - Data informada deve ser a data atual ou no máximo um ano a frente
- 5a - Se data informada é a data atual, transferir imediatamente
- 5b - Se data informada é uma data futura, agendar transferência

Fluxo "Feliz"



Transferir Valores entre Contas

Ator: Cliente do Banco

Fluxo normal:

- 1 - Autenticar Cliente
- 2 - Cliente informa agência e conta de destino da transferência
- 3 - Cliente informa valor que deseja transferir
- 4 - Cliente informa a data em que pretende realizar a operação
- 5 - Sistema efetua transferência
- 6 - Sistema pergunta se o cliente deseja realizar uma nova transferência

Extensões:

- 2a - Se conta e agência incorretas, solicitar nova conta e agência
- 3a - Se valor acima do saldo atual, solicitar novo valor
- 4a - Data informada deve ser a data atual ou no máximo um ano a frente
- 5a - Se data informada é a data atual, transferir imediatamente
- 5b - Se data informada é uma data futura, agendar transferência

Exceções e
Detalhamentos



Transferir Valores entre Contas

Ator: Cliente do Banco

Fluxo normal:

- 1 - Autenticar Cliente
- 2 - Cliente informa agência e conta de destino da transferência
- 3 - Cliente informa valor que deseja transferir
- 4 - Cliente informa a data em que pretende realizar a operação
- 5 - Sistema efetua transferência
- 6 - Sistema pergunta se o cliente deseja realizar uma nova transferência

Extensões:

- 2a - Se conta e agência incorretas, solicitar nova conta e agência
- 3a - Se valor acima do saldo atual, solicitar novo valor
- 4a - Data informada deve ser a data atual ou no máximo um ano a frente
- 5a - Se data informada é a data atual, transferir imediatamente
- 5b - Se data informada é uma data futura, agendar transferência

Erro (passo 2)

Transferir Valores entre Contas

Ator: Cliente do Banco

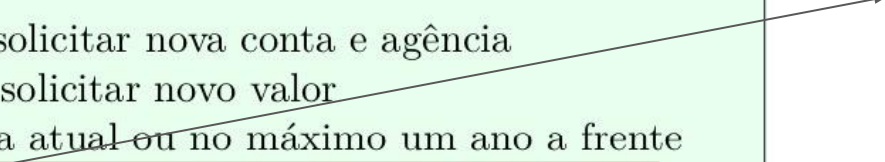
Fluxo normal:

- 1 - Autenticar Cliente
- 2 - Cliente informa agência e conta de destino da transferência
- 3 - Cliente informa valor que deseja transferir
- 4 - Cliente informa a data em que pretende realizar a operação
- 5 - Sistema efetua transferência
- 6 - Sistema pergunta se o cliente deseja realizar uma nova transferência

Extensões:

- 2a - Se conta e agência incorretas, solicitar nova conta e agência
- 3a - Se valor acima do saldo atual, solicitar novo valor
- 4a - Data informada deve ser a data atual ou no máximo um ano a frente
- 5a - Se data informada é a data atual, transferir imediatamente
- 5b - Se data informada é uma data futura, agendar transferência

Detalhamento
(passo 5)



Importante

- Casos de uso não são "algoritmos"
- Ainda estamos levantando requisitos:
 - Foco: entendimento e delimitação do problema
 - E não em possíveis soluções (i.e., algoritmos)

Exemplo: Venda em Caixa de Supermercado (PDV)

Fonte: Craig Larman. Applying UML and Patterns. Pearson, 2004

Fluxo Normal

1. Cliente chega no caixa com os produtos que deseja comprar
2. Caixa inicia uma nova venda
3. Caixa identifica um produto; por exemplo, usando leitor de código de barras
4. Sistema identifica produto, registra venda, apresenta a descrição do produto e seu preço, bem como o total da compra até o momento
5. Caixa repete passos 3-4 até não haver mais produtos para registrar venda
6. Sistema apresenta total da venda
7. Caixa informa total da venda para o cliente e pede o pagamento
8. Cliente faz o pagamento e o sistema processa o pagamento
9. Sistema registra a venda como concluída e envia informações para o sistema de contabilidade e para o sistema de controle de estoques
10. Sistema gera recibo da venda
11. Caixa entrega recibo para o cliente
12. Cliente encerra a compra, levando os produtos e o seu recibo

Extensões (ou fluxos alternativos): [vamos mostrar para apenas um dos passos do fluxo normal]

7a. Pagamento em dinheiro:

1. Caixa digita o montante de dinheiro que o cliente lhe forneceu
2. Sistema informa o valor do troco e libera a gaveta de notas
3. Caixa deposita o dinheiro na gaveta e retorna troco para o cliente
4. Sistema registra e conclui pagamento com dinheiro

7b. Pagamento com cartão de crédito:

1. Cliente insere cartão na máquina de cartão de crédito
2. Sistema informa para máquina de cartão o valor da compra
3. Cliente informa senha e confirma compra
4. Sistema envia solicitação de pagamento para operadora do cartão
 - 4a. Se erro de comunicação com o sistema da operadora do cartão
 1. Sistema sinaliza erro para o Caixa
 2. Caixa solicita ao Cliente um modo alternativo de pagamento

(continua no próximo slide)

- 5. Sistema recebe resultado da requisição de pagamento
 - 5a. Pagamento negado
 - 1. Sistema avisa o Caixa
 - 2. Caixa solicita ao Cliente um modo alternativo de pagamento
 - 5b. Timeout na espera pelo resultado da requisição de pagamento
 - 1. Sistema avisa o Caixa
 - 2. Caixa tenta de novo ou solicita modo alternativo de pagamento
- 6. Sistema registra e conclui pagamento com cartão de crédito

7c. Pagamento com cheque

....

7d. Pagamento com cartão de débito

....

Exercício

- Escreva um caso de uso para “transferir valor usando PIX”

Transferir valor usando PIX

Ator: cliente do banco

Fluxo normal:

1. Cliente informa a chave PIX do destinatário
2. Sistema mostra nome e banco do destinatário
3. Cliente informa valor que deseja transferir
4. Sistema solicita aprovação da transferência
5. Sistema efetua a transferência

Extensões:

1. Sistema solicita primeiro o tipo da chave: CPF, celular, mail ou outro
3. Se horário noturno e valor maior que R\$ 1000, sistema informa que PIX não é possível e sugere agendamento para 08:00 do dia seguinte

Extensões (cont.):

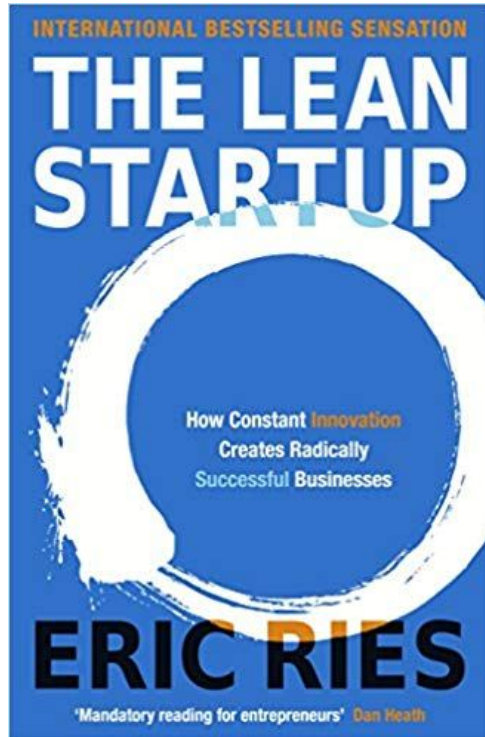
3. Se horário noturno e valor maior que R\$ 1000, sistema informa que PIX não é possível e sugere agendamento para 08:00 do dia seguinte

3a. Se cliente aceitar agendamento, prossiga com horário 08:00

3b. Caso contrário, encerre transferência

Produto Mínimo Viável

Origem



2011

Dois tipos de sistemas

1. Baixo risco e usuários conhecidos
2. Alto risco e sucesso incerto

Exemplo: Sistema de Controle de Bibliotecas

- Sistema conhecido, fundamental em toda biblioteca, etc
- Necessidade e viabilidade desse sistema são óbvias
- Histórias de usuários funcionam bem!

Exemplo: Loja virtual para empréstimo de ebooks com pagamento em bitcoins

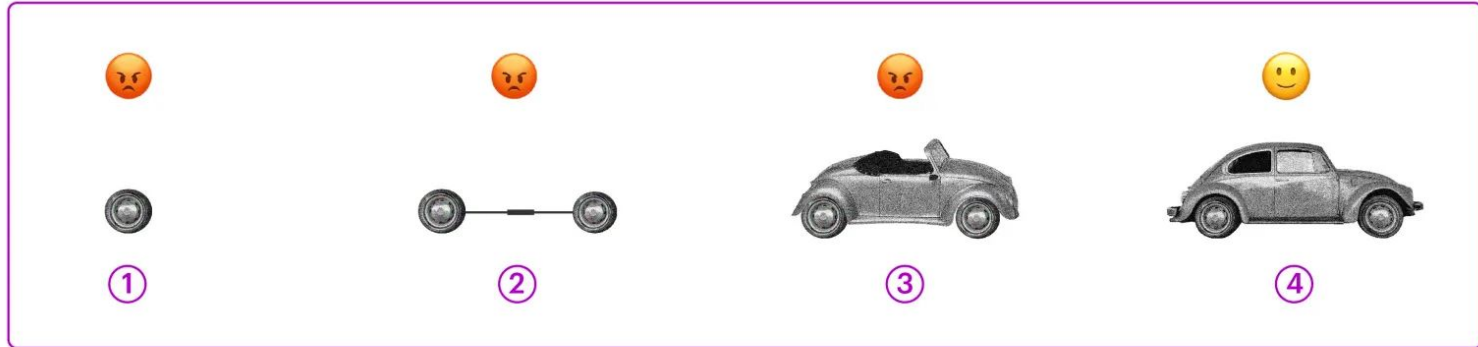
- Sistemas típicos de startups, mas não exclusivos
- Como risco é alto, ideia tem que ser rapidamente validada com usuários reais

Produto Mínimo Viável (MVP)

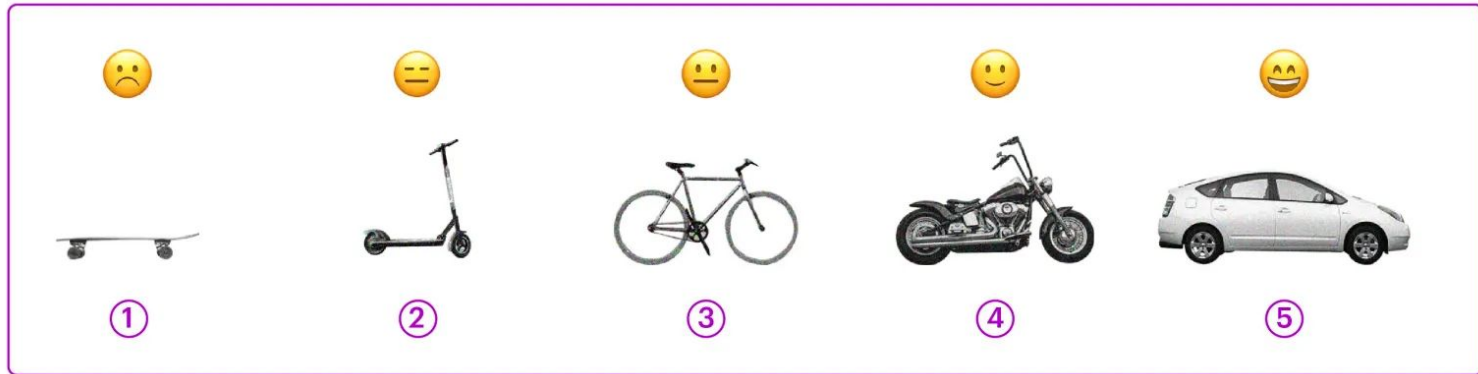
- Produto = pode ser usado
- Mínimo = menor conjunto de funcionalidades (menor custo)
- Viável? = terá mercado?

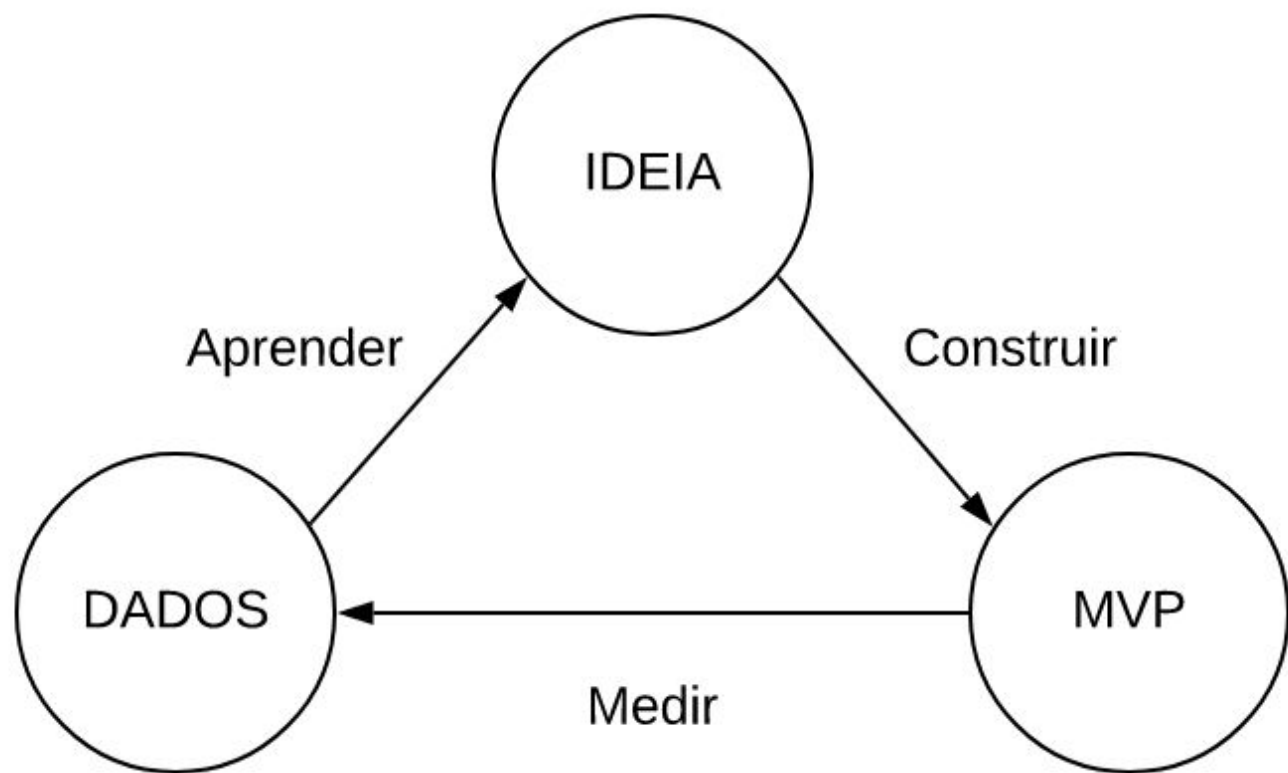
Exemplo: Implementação de um carro

Sem
MVPs



Com
MVPs



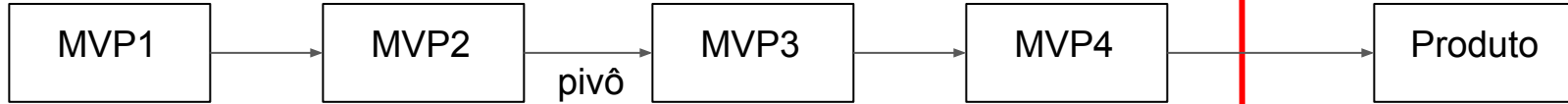


Ao final de um ciclo, pode-se:

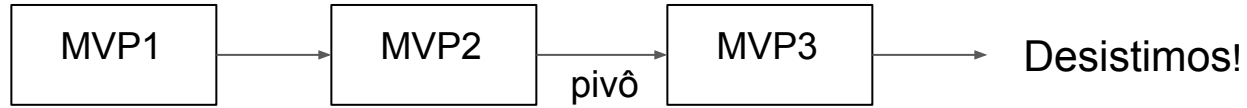
1. Realizar ajustes pequenos e rodar ciclo de novo
2. Pivotar: realizar ajustes grandes e rodar ciclo de novo
3. Desistir (dinheiro acabou!)
4. Deu certo: atingimos o Product Market Fit (PMF) e vamos agora construir um produto robusto

Validação

Market Fit



Validação



Market Fit

Exemplos de MVP

Zappos

- Loja virtual de sapatos, depois adquirida pela Amazon
- As pessoas vão comprar sapatos pela Internet? (em 1999)
- MVP:
 - Sistema Web simples
 - Com fotos de sapatos de lojas físicas da cidade
 - Backend era 100% manual
- Objetivo: apenas validar hipótese de negócio

Zappos.com

the world's largest shoe store!

free shipping
& no sales tax



- customer service
- shopping bag
- account & register

women's

men's

dress

casual

athletic

dress

casual

athletic

kids

Pick a category
to shop from:

Category

Register now &
Save Money

registered customers

username

password

login



Measure Your Foot

The world's largest shoe store!

WHAT WE'RE HEARING!

Welcome to Zappos.com - the shoe store! We have a selection of over 100 brands to shop from! We offer FREE SHIPPING (U.S. orders only) and NO SALES TAX.

Live Customer Service!
Mon. - Fri. 10am-6pm PST
Sat. 10am-5pm PST
[Click here!](#)

Free Shoes!



Congratulations to the November 24 winner,
Adriana Rodriguez of Oakland, CA

We will be giving away a **FREE PAIR OF SHOES** (up to \$150 in store credit) every Wednesday until the year 2000!

To enter become a registered user or send an email to freeshoes@zappos.com with your name and email address. A winner will be notified by email every Wednesday!

featured brands.



Zappos Special:

**Free Shipping &
No Sales Tax**

home

women's dress

women's casual

women's athletic

men's dress

men's casual

men's athletic

kids

MVP ≠ 1a versão de um produto



MVP é um experimento (logo, pode falhar)

- Ou seja:
 - Se existe um mercado certo
 - Se o cliente já te contratou e vai te pagar
 - Se você tem competência para desenvolver
 - Se o cliente sabe muito bem o que quer
 - Logo, não existe risco e não precisamos de MVP
- Se já sabemos que vai dar certo, não é um experimento

MVP & Engenharia de Software

- MVP não precisa usar todas as melhores práticas de ES
 - Testes de unidade, refatorações, arquitetura complexa, etc
- Se a ideia der certo, pode-se depois reimplementar o sistema
- Certos requisitos, principalmente NF, podem ser importantes
 - Desempenho, usabilidade, estabilidade, etc

Quanto tempo leva para construir um 1o MVP?

- Depende e varia muito, mas tem que ser rápido
- Por exemplo, duas semanas...
- “Se você não tiver vergonha do seu MVP, você demorou demais para lançá-lo” (Reid Hoffman)

Tipos comuns de pivôs

- Zoom-in (uma feature específica vira um produto)
- Segmento de clientes
- Aplicação para Plataforma
- Tecnologia

Zoom-in: Slack

- Caso parecido com o do Flickr
- Slack é um aplicativo de mensagens usado por empresas
- Também "saiu" de um jogo de RPG online

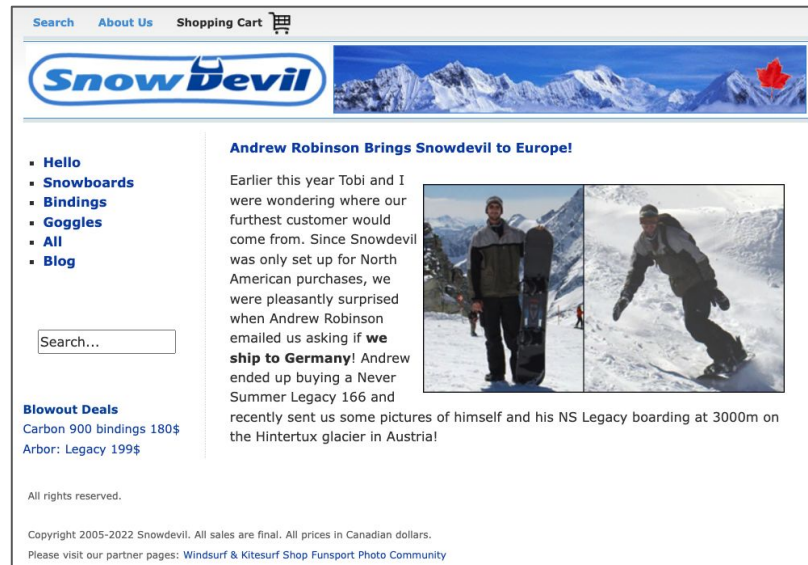
Segmento de Clientes: Twitch

- Objetivo: transmissões ao vivo (streaming)
- Público inicial: amplo (justin.tv)
- Depois: gamers (twitch.tv)



Aplicação para Plataforma: Shopify

- De: uma loja online para alugar produtos para esquiar
- Para: uma plataforma para hospedar lojas online



Tecnologia: Android

- De: sistema operacional para câmaras
- Para: sistema operacional para celulares



Testes A/B

Testes A/B

- Existe o cenário onde duas implementações de requisitos "competem" entre si
- Exemplo: loja virtual
- Sistema recomendação: quem compra P também compra X,Y,Z
 - Versão original
 - Nova versão, proposta por alguns devs
- Vale a pena migrar para nova versão?
- Testes A/B: deixar os dados decidirem

Versões de controle e tratamento

- Versão A: controle
- Versão B: tratamento
- Novos usuários:

```
version = Math.Random(); // número aleatório entre 0 e 1
if (version < 0.5)
    "execute a versão de controle"
else
    "execute a versão de tratamento"
```

No final do experimento

- Analisam-se os dados, isto é, alguma métrica
- Exemplo: taxa de conversão de visitas em compras
- Versão B introduz ganhos estatisticamente relevantes?
 - Sim, vamos migrar para ela
 - Não, continuamos com o algoritmo original

Comentários finais

- Testes A/B requerem amostras grandes
- São usados por todas as grandes empresas da Internet

Exercícios

Possível resposta:

(a) História épica: Como professor, eu gostaria de poder aplicar provas online.

(b) Decomposição em histórias menores:

- Como professor, eu gostaria de criar um banco de questões
- Como professor, eu gostaria de criar e configurar uma prova online
- Como aluno, eu gostaria de fazer uma prova online
- Como professor, eu gostaria que uma prova fosse corrigida automaticamente
- Como aluno, eu gostaria de ver o resultado da correção de uma prova que fiz
- Como professor, eu gostaria de baixar um arquivo com as notas de uma prova

3. Seja um sistema para criação e preenchimento de formulários online, semelhante ao Google Forms.

Seja a seguinte história de usuário desse sistema:

“Como criador de um formulário, eu gostaria de fechar o mesmo para recebimento de novas respostas”

Escreva os testes ou critérios de aceitação para essa história.

Possível resposta:

História: Como criador de um formulário, eu gostaria de fechar o mesmo para recebimento de novas respostas

Testes de aceitação:

- Criar um formulário
- Enviar uma ou duas respostas para ele (como usuário normal)
- Fechar o formulário para recebimento de respostas
- Tentar enviar uma nova resposta e verificar se não é possível

4. Suponha que você é o líder técnico de um time.

E ficou responsável por escrever os critérios de conclusão de histórias (“done criteria”) desse time.

Liste então pelo menos três critérios que recomendaria com certeza.

Possível resposta:

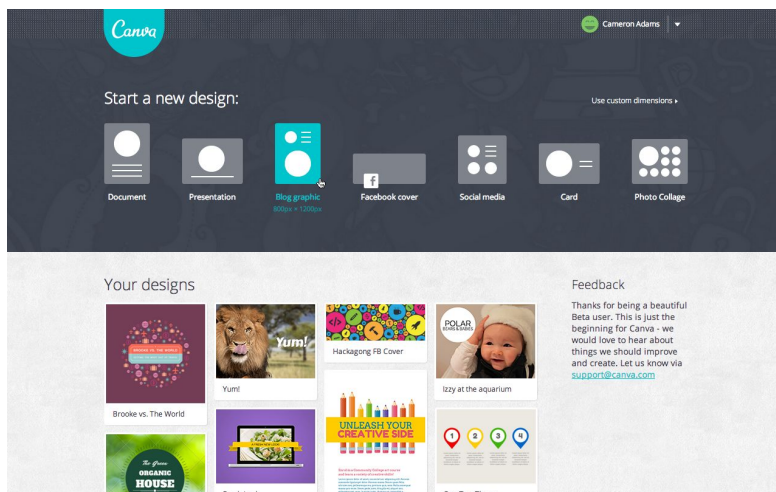
1. Cobertura de testes automatizados $\geq 80\%$
2. Sistema de linter executado com sucesso
3. UX Designer aprovou as modificações no frontend
4. Código revisado por pelo menos um outro membro do time
5. Código integrado na branch “development”

1. Escreva três histórias de usuário para o serviço de PIX de um banco
2. Suponha que você trabalha em sistema de ensino a distância, semelhante ao Moodle.
 - (a) Proponha uma história épica para esse sistema
 - (b) Quebre essa história em histórias menores.

1. Qual a diferença entre um MVP e uma pesquisa de mercado?
2. Suponha que você teve a ideia de fazer um sistema para organizar caronas para alunos da USP.
 - Como você implementaria um MVP para esse sistema sem escrever código?
 - Supondo que seu primeiro MVP falhou, descreva um possível pivô que você poderia tentar.
3. Descreva um tipo de domínio (ou aplicação) para o qual é mais difícil e desafiador criar um MVP. Justifique sua resposta.

Exemplo: Canva

- Primeira versão levou um ano para ser lançada
- Provavelmente, não deveria ser chamada de um MVP



Requisitos de Software

Prof. Pedro Henrique Dias Valle