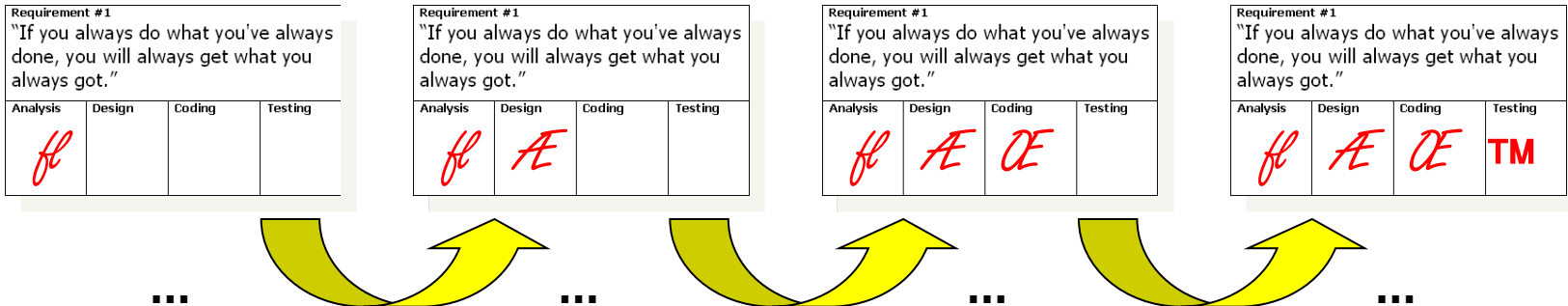
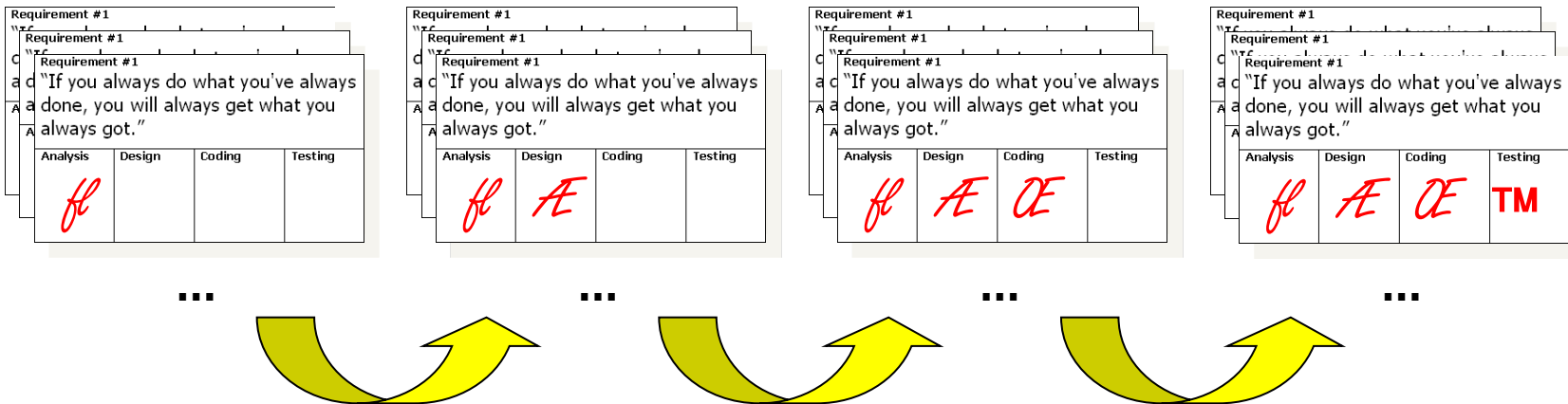


Requirement #1 "If you always do what you've always done, you will always get what you always got."			
Analysis	Design	Coding	Testing
<i>fl</i>	<i>AE</i>	<i>CE</i>	

Pequenos Lotes



Grandes Lotes



- Qual é o arranjo logístico mais rápido?
- Qual equipe apresentou o maior esforço por projeto?
- Quais as vantagens de cada abordagem?
- Quais as desvantagens de cada abordagem?
- Qual a justificativa para manter os grandes lotes?

Sumário

- Cenário
- Lean Thinking
- Lean Startup
- Lean Inception
- Lean Software Development

PROCESSES THAT ARE APPLIED TO PRODUCING E.G.



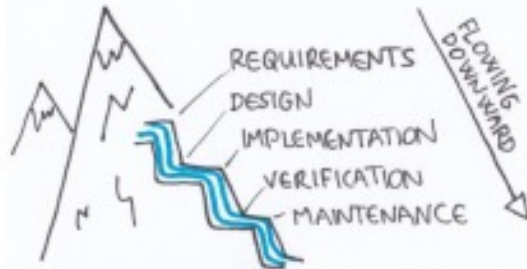
ARE NOT EFFECTIVE FOR PRODUCING DIGITAL PRODUCTS & SERVICES ...

PRODUCT DEVELOPMENT PROCESSES

UX Knowledge Base Sketch #53

... DIGITAL PRODUCTS & SERVICES ARE:

- COMPLEX SYSTEMS
- UNPREDICTABLE
- IN CONSTANT CHANGE



WATERFALL-MODEL

- SEQUENTIAL, LINEAR
- NOT MUCH ITERATION

USED IN SOFTWARE DEVELOPMENT FROM THE MIDDLE OF THE 20TH CENTURY

- PROBLEMS:**
- NOT FLEXIBLE
 - TOO SLOW
 - KNOWING THE REQUIREMENTS BEFORE THE PROCESS STARTS

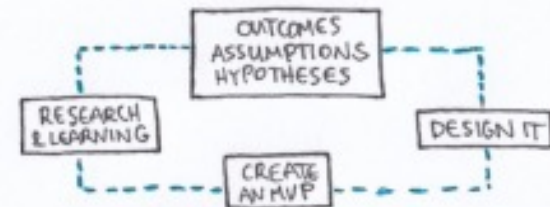


AGILE FRAMEWORK

- E.G. SCRUM

- RAPID & FLEXIBLE
- SHORT CYCLES, SMALLER TIME-FRAMES
- ADAPTIVE PLANNING
- ACCOUNTABILITY & REFLECTION
- CONTINUOUS LEARNING

- PROBLEMS:**
- OUTPUT-BASED: SHIPPING FEATURES
 - GREAT VELOCITY, BUT: IS THE DIRECTION RIGHT?
 - NO DECISION-MAKING FRAMEWORK



LEAN UX

BY JEFF GOTHELF & JOSH SEIDEN

- OUTCOMES INSTEAD OF OUTPUTS: CHANGE IN CUSTOMER BEHAVIOR INSTEAD OF FEATURES
- DOUBT -> CERTAINTY IN SMALL BATCHES (AGILE!)
- REFLECTION: ENOUGH EVIDENCE TO HAVE A CYCLE IN THE SAME DIRECTION?
- REDUCES RISKS & WASTE

- PROBLEMS:**
- IF NOT A MATURE PRODUCT, HARDER TO MEASURE CUSTOMER BEHAVIOR.
 - DEFINING SUCCESS CAN BE CHALLENGING

IMPORTANT INGREDIENTS:



DESIGN THINKING

"HUMAN-CENTERED APPROACH C-3 TO INTEGRATE THE NEEDS OF PEOPLE, THE POSSIBILITIES OF TECHNOLOGY, AND THE REQUIREMENTS FOR BUSINESS SUCCESS." (TIM BROWN)



②



LEAN STARTUP

BY ERIC RIES

BUILD - MEASURE - LEARN FEEDBACK LOOPS
MINIMUM VIABLE PRODUCT (MVP)
GETTING OUT OF THE BUILDING



DESIGN SPRINT

"SHORTCUT TO LEARNING WITHOUT BUILDING AND LAUNCHING." (GV.COM/SPRINT)



DUAL TRACK

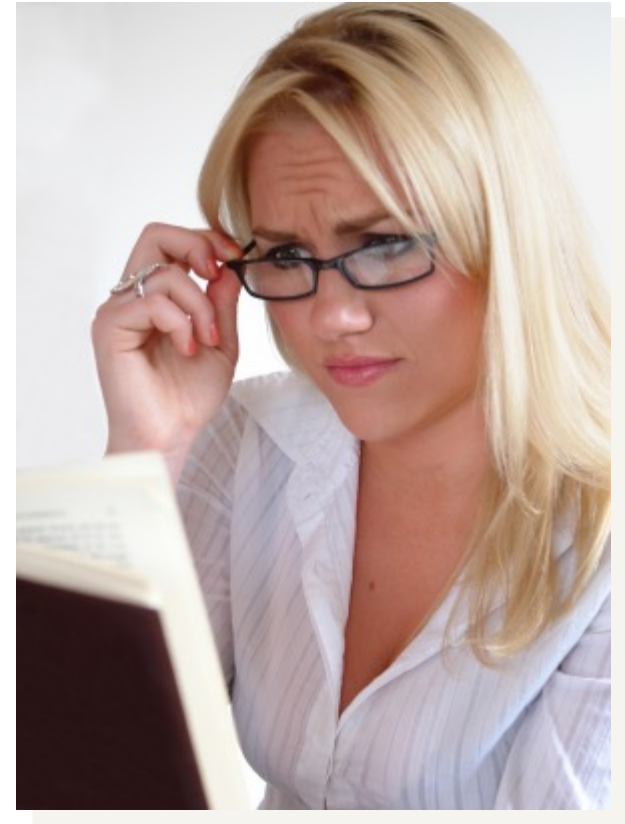
1. GENERATIVE RESEARCH, DESIGN THINKING -> DESIGN SPRINT

2. LEAN UX + AGILE PRODUCTION



O que é desperdício?

- Desperdício é qualquer coisa que deprecie recursos no tempo, esforço, espaço ou dinheiro sem adicionar valor ao cliente.
- Valor é visto através dos olhos daqueles que pagam pelo uso e derivam o valor dos sistemas que nós criamos.



Lean Software Dev.

“Desperdício (***muda*** em japonês) é qualquer coisa além do mínimo de equipamento, materiais, peças, espaço e tempo do operador que sejam absolutamente essenciais para agregar valor ao produto.”

Shoichiro Toyoda,
Fundador da Toyota



Lean Software Dev.

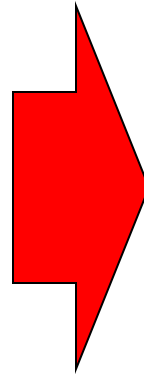
Aumento de produtividade
> 2,5



Lean Software Dev.

Sete Desperdícios da Manufatura

- **Excesso de produção**
Produzir mais do que o necessário para uso imediato.
- **Estoque**
Qualquer estoque de processo que exceda o que é exigido para satisfazer a demanda dos clientes.
- **Extra-Processamento**
Usar mais energia do que é necessário para produzir um produto ou agregar mais valor do que os clientes estão dispostos a pagar.
- **Excesso de Movimentação**
Movimentação desnecessária de pessoal, assim como caminhar, levantar, curvar-se e esticar-se.
- **Transporte**
Movimentos desnecessários de produtos, materiais ou informação.
- **Esperas**
Qualquer atraso entre o final da atividade de um processo e o início da próxima atividade.
- **Defeitos**
Qualquer produção que resulte em retrabalho ou refugo.

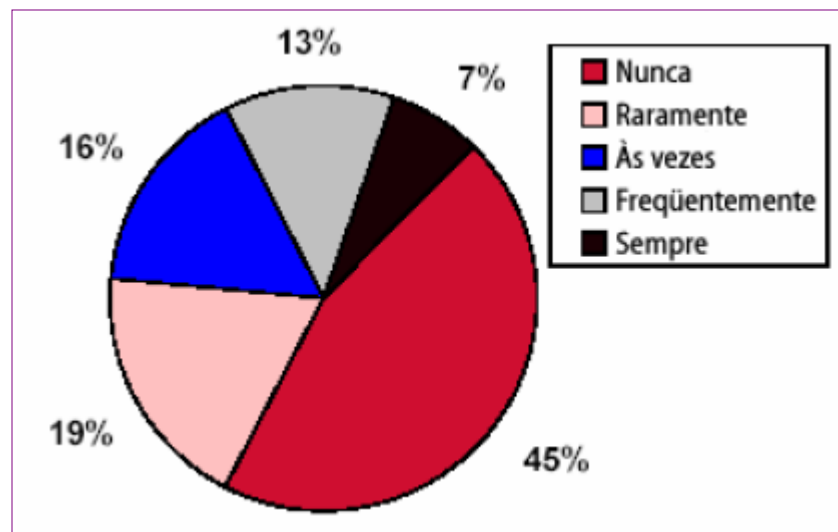


Sete Desperdícios do Software

- **Excesso de Funcionalidades**
Produzir mais do que é necessário para a aplicação do sistema.
- **Trabalho Inacabado**
Qualquer produção intermediária que não possa ser agregada ao produto ou entregue ao cliente.
- **Defasagem Tecnológica**
Usar mais energia do que é necessário para desenvolver o produto de software.
- **Trocas de Tarefas**
Alternar entre atividades antes do trabalho ter sido concluído.
- **Excesso de Manipulações**
Repases desnecessários de tarefas num fluxo unitário de operações.
- **Esperas**
Qualquer tipo de atraso entre uma atividade e outra.
- **Defeitos**
Qualquer falha de produto ou processo que resulte em retrabalho ou descarte.

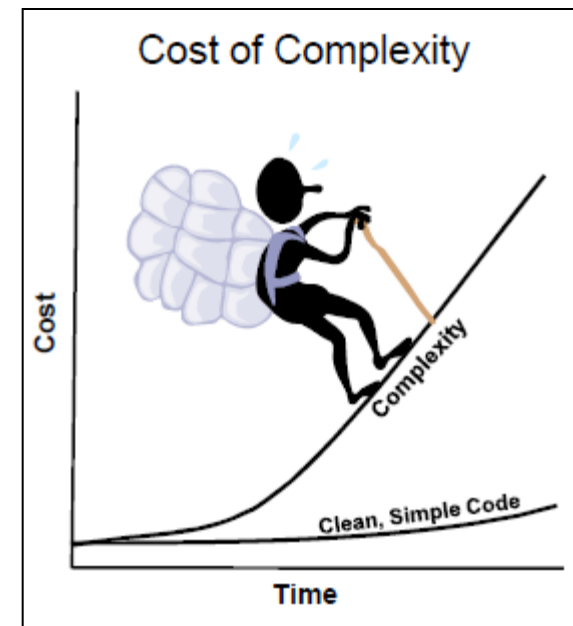
Lean Software Dev.

- ✓ **Estudo do The Standish Group conclui (Chaos Report 2002):**
Pesquisa sobre a utilização das funcionalidades do software ...



Mais de **64%** de um sistema de software quase nunca não é utilizado!

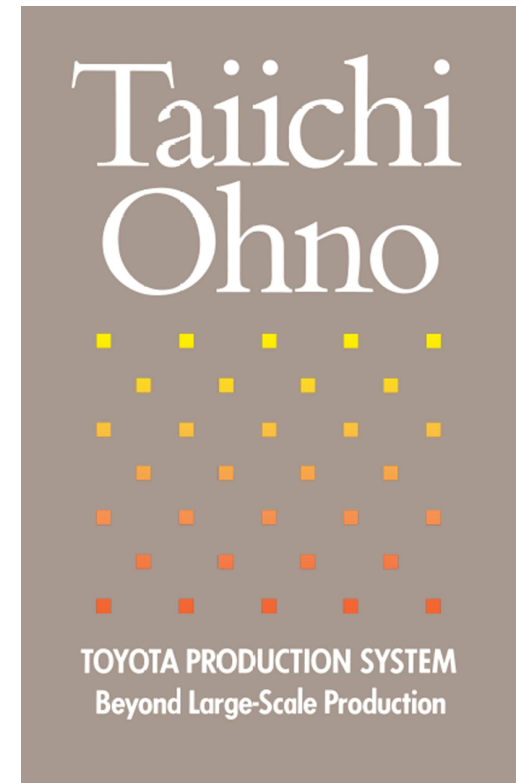
“A forma mais rápida de se aumentar a produtividade é desenvolver menos código!”



O PENSAMENTO LEAN

Origem do Pensamento Lean

- Sistema de Produção da Toyota: Taiichi Ohno e Shigeo Shingo
- Posteriormente, seus valores foram expandidos para outras áreas:
 - Produção Lean
 - Manufatura / Operações Lean
 - Cadeia de Suprimentos Lean
 - Desenvolvimento de Produtos Lean
(desenvolver software é criar um novo produto!)



Princípios do Pensamento Lean

1. Valor
2. Fluxo de valor
3. Fluxo Contínuo
4. Produção puxada
5. Perfeição

Valor

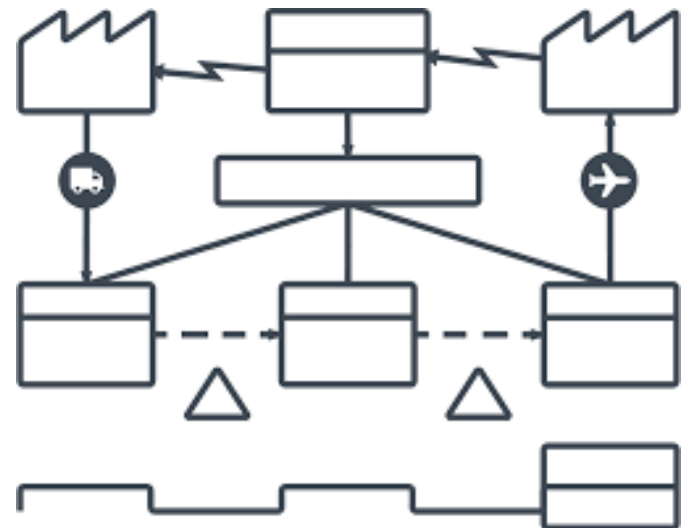


- Valor não é preço
- O cliente definir o que é valor em seu produto
- Aquilo que o cliente não está disposto a pagar pelo produto é desperdício e deve ser eliminado
- Assim como os clientes, os produtos precisam estar em constante evolução, por isso, estudos frequentes são válidos a fim de transformar o produto ao longo do tempo

Fluxo de Valor

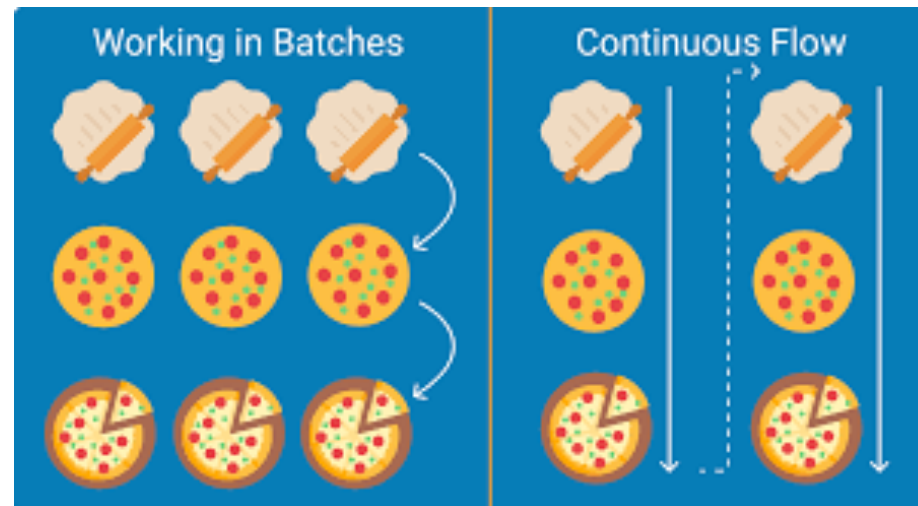
- É o mapeamento dos processos que agregam valor ao produto
- Compreende identificar quais etapas agrega valor ao produto
- Aquelas que não agregam valor (desperdício) devem ser eliminadas, reduzindo automaticamente os custos de máquinas, energia, tempo, etc.

- Mapeamento de Fluxo de Valor



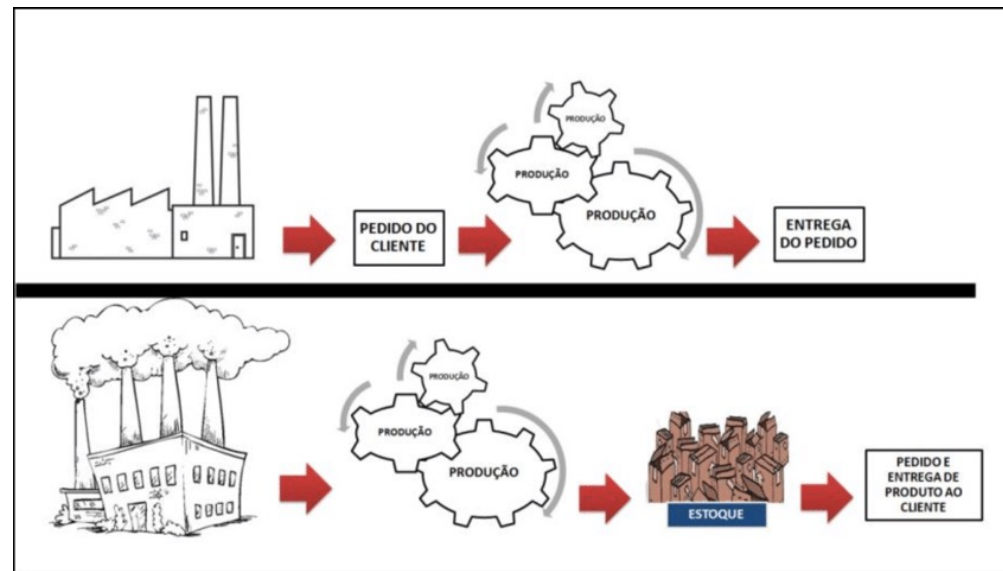
Fluxo Contínuo

- Após identificar apenas as tarefas que criam valor ao produto, deve ser criado o fluxo contínuo, ou seja, produzir sem interrupções
- A ideia é atender as necessidades dos clientes com rapidez, com menor tempo para processar os pedidos e baixo estoque



Produção Puxada

- Produzir apenas o que o cliente quer, reduzindo ao máximo o estoque
- Elimina a necessidade de descontos e promoções com objetivo de acabar com o estoque dos itens que já foram produzidos e encontram-se parados por determinado período



Perfeição

- Kaizen: melhoria contínua
- PDCA
- Busca pela melhoria contínua dos processos, pessoas, produtos, etc.
- Tem o objetivo de agregar valor ao cliente sempre
- Identificação permanente de problemas, a fim de melhorar o produto e consequentemente a satisfação do cliente



LEAN STARTUP

Lean Startup

- A Startup Enxuta
- Eric Ries
- Metodologia com origem no Lean Thinking que visa otimizar recursos a fim de criar um ambiente de inovação, seja em pequenas empresas de crescimento acelerado, como as startups, seja em grandes corporações consolidadas
- Apresenta conceitos como o Produto Mínimo Viável (MVP), Pivotar e métodos ágeis de interação com clientes



Os 5 Princípios da Startup Enxuta

1. Empreendedores estão por toda parte
2. Empreender é administrar
3. Aprendizado validado
4. Construir-medir-aprender
5. Contabilidade para inovação

Empreendedores Estão por Toda Parte

- Não é preciso trabalhar numa garagem para estar numa startup
- Startup: uma instituição humana projetada para criar novos produtos e serviços sob condições de extrema incerteza
- A abordagem da startup enxuta pode funcionar em empresas de qualquer tamanho, mesmo numa de grande porte, em qualquer setor ou atividade

Empreender é Administrar

- Uma startup é uma instituição, não um produto, assim, requer um novo tipo de gestão, especificamente constituída para seu contexto de extrema incerteza
- “Empreendedor” deveria ser considerado um cargo em todas as empresas modernas que dependem da inovação para seu crescimento futuro

Aprendizado Validado

- Startups existem não apenas para fabricar coisas, ganhar dinheiro ou mesmo atender clientes: elas existem para *aprender* a desenvolver um negócio sustentável
- Essa aprendizagem pode ser validada cientificamente por meio de experimentos frequentes que permitem aos empreendedores testar cada elemento de sua visão

Construir-Medir-Aprender

- A atividade fundamental de uma startup é transformar ideias em produtos, medir como os clientes reagem, e, então, aprender se é o caso de pivotar ou perseverar
- Todos os processos de startup bem-sucedidos devem ser voltados a acelerar esse ciclo de *feedback*

Contabilidade para a Inovação

- A fim de melhorar os resultados do empreendedorismo e poder atribuir responsabilidades aos inovadores, é preciso focar também em assuntos menos interessantes:
 - Como medir o progresso,
 - Como definir marcos,
 - Como priorizar o trabalho
- Isso requer um novo tipo de contabilidade desenvolvida para startups e para as pessoas responsáveis por elas

MVP - Minimum Viable Product

- Produto Mínimo Viável: uma versão do produto que permite uma volta completa do ciclo construir-medir-aprender, com o mínimo de esforço e o menor tempo de desenvolvimento
- Não apresenta diversos recursos que podem se provar necessários mais tarde, mas deve ser possível medir seu impacto
- Ajuda os empreendedores a começar o processo de aprendizagem o mais rápido possível, com o menor esforço possível
- Diferentemente de um protótipo ou teste de conceito, um MVP é projetado não só para responder a perguntas técnicas ou de design do produto, mas testar hipóteses fundamentais do negócio

Pivotar

- Pivotar: mudar a direção do seu negócio
- Uma mudança de estratégia, sem uma mudança de visão
- Comum em ambientes de startups, principalmente no início do projeto

LEAN INCEPTION

Lean Inception

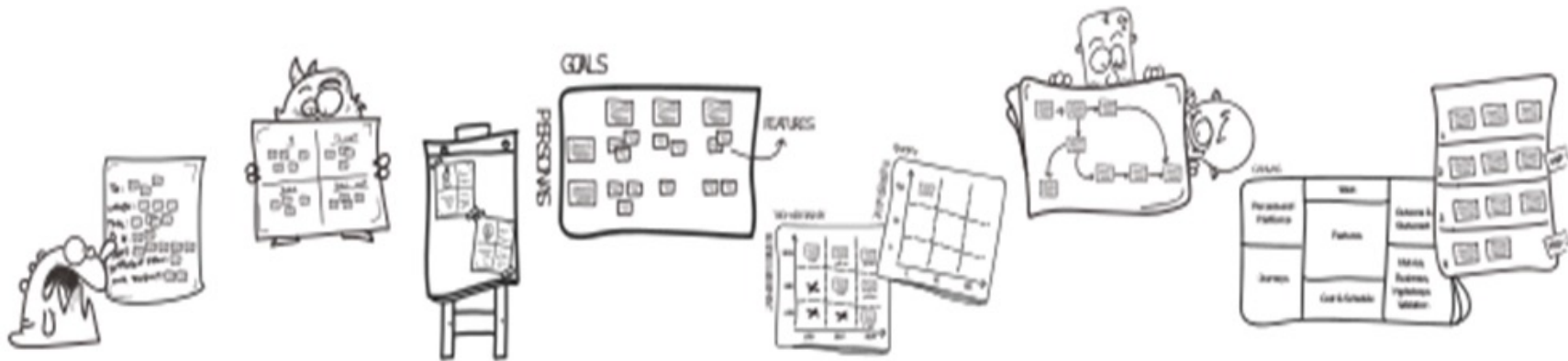
- Paulo Caroli
- Lean Startup + Design Thinking = Lean Inception
- Na origem: um workshop colaborativo com o intuito de alinhar um grupo de pessoas sobre a proposta de solução a ser construída, ou seja, sobre o Produto Mínimo Viável (MVP)



Lean Inception

- Compreende uma sequência de diversas atividades para promover o alinhamento e definição de objetivos, estratégias e o escopo do produto
- Junto a essas atividades, é feito o uso de técnicas de Design Thinking e Lean Startup
- Em resumo, enquanto a Lean Inception trata do alinhamento de um time sobre o produto, a solução a construir, outras metodologias ágeis como o Scrum e o Kanban auxiliam esse time a ser eficiente na forma de trabalhar

Lean Inception

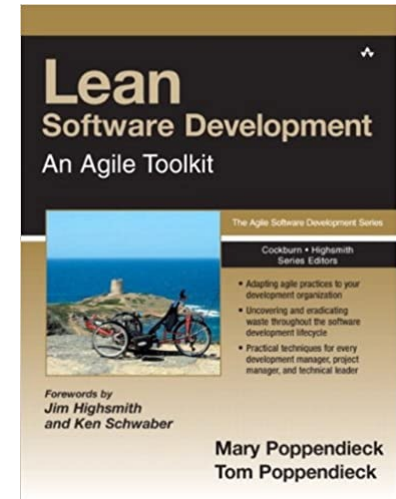


Visão do Produto
É/Não É, Faz/Não Faz
Objetivos
Features
Personas
Funcionalidades
Jornada do Usuário

LEAN SOFTWARE DEVELOPMENT

Lean Software Development

- Mary e Tom Poppendieck – 2003
- Metodologia para auxiliar as equipes de desenvolvimento de software a atingirem um maior nível de maturidade em termos de qualidade e velocidade
- Entender e estudar o processo atual, e trabalhando constantemente na identificação e eliminação de desperdícios
- Oferece ferramentas para que a equipe crie um ambiente de trabalho mais produtivo, onde as pessoas se respeitem e vejam a melhoria contínua como parte fundamental de seu processo



Princípios Lean no Desenvolvimento de Software

1. Elimine Desperdícios
2. Inclua a Qualidade no Processo
3. Crie Conhecimento
4. Adie Comprometimentos
5. Entregue Rápido
6. Respeite as Pessoas
7. Otimize o Todo

Elimine Desperdícios

- Desperdício é tudo que consome recursos, mas não agrega valor ao cliente
 1. Trabalho parcialmente feito: artefatos ou processos inacabados
 2. Processos extras: p. ex., excesso de documentação ou gerenciamento
 3. Funcionalidades extras: desenvolvimento de requisitos não solicitados ou mal definidos
 4. Alternação de tarefas: o tempo para alternar entre diferentes tarefas é tratado como desperdício
 5. Handoffs: a passagem de conhecimento que acontece ao longo do processo de desenvolvimento
 6. Espera: por feedback de outro membro do time, pelo resultados dos testes, pelo feedback do cliente, etc.
 7. Esforços de comunicação: falhas na gestão da comunicação
 8. Defeitos: custo para a correção de defeitos (identificar mais cedo)

Inclua a Qualidade no Processo

- Técnicas que garantem a qualidade do produto devem ser incorporadas ao processo
- Exemplos
 - Teste unitário através de TDD : Escrever o teste / teste falho / escrever o código para passar o teste
 - Refatoração: Melhorar e refinar o código escrito, evitando manter legados de débitos técnicos e fazendo com o que o software continue se comportando da mesma maneira.
 - Integração contínua: Compilação automática do repositório de código, verificando se o código inserido não prejudicou demais funcionalidades do software.

Crie Conhecimento

- O conhecimento adquirido por um, deve ser compartilhado com todos
- Esses conhecimentos são adquiridos por meio de revisões, feedback e reuniões de alinhamentos

Adie Comprometimentos

- Deixar as decisões e comprometimentos para o último momento, permitindo coletar informações e ter experiências para fortalecer a tomada de decisão.
- Realize testes e experimentos antes
- Recursos e funcionalidades devem ser incorporados ao final do projeto, evitando retrabalhos
- O mercado sempre pode mudar

Entregue Rápido

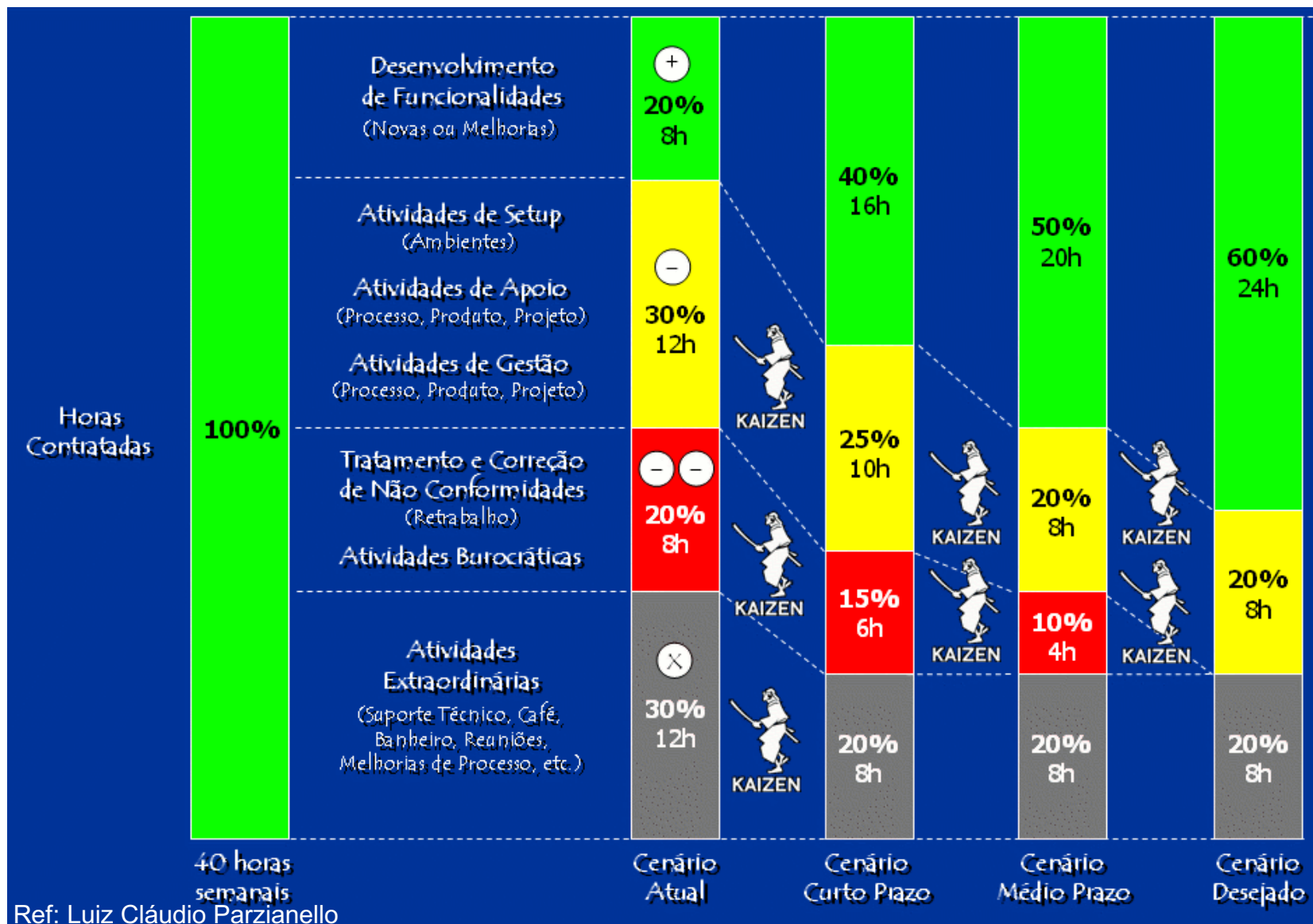
- Os produtos devem ser lançados de forma rápida, o feedback dos clientes será um termômetro para a criação de melhorias
- Se o produto for falhar, deve ser rápido e servir para gerar aprendizado
- Sistema “puxado”

Respeite as Pessoas

- Um ambiente saudável, com uma comunicação proativa e feedback constante tende a gerar uma atmosfera produtiva e colaborativa
- Times auto-organizados
 - Propósito
 - Participação
 - Segurança
 - Competência
 - Progresso

Otimize o Todo

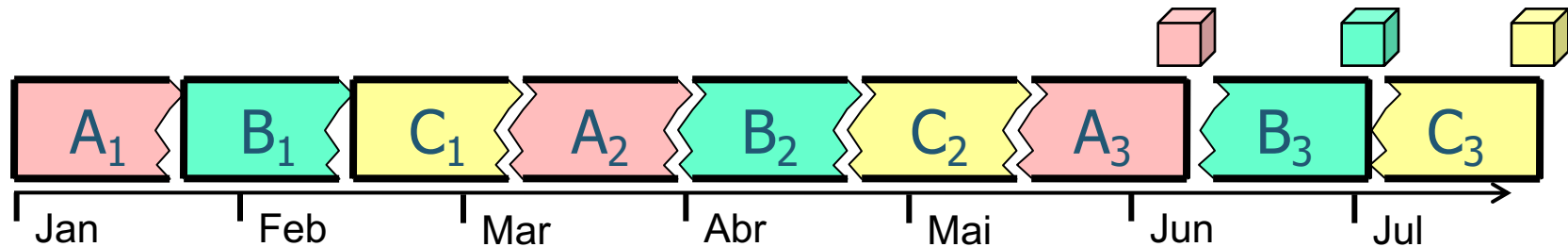
- O processo deve ser avaliado do começo ao fim, com o objetivo de tornar o fluxo o mais eficiente possível
- Não adianta resolver os sintomas, é preciso resolver a causa



O paradoxo da multitarefa

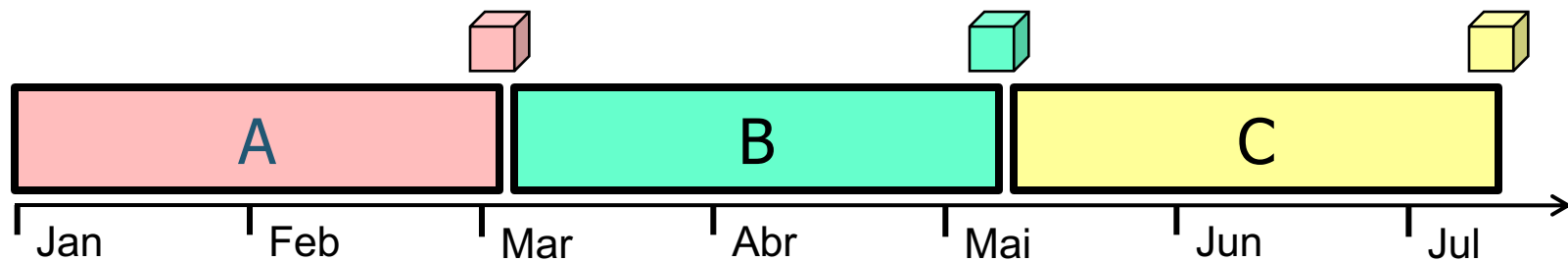
- Visão tradicional

“Tudo é importante, vamos fazer tudo ao mesmo tempo!”



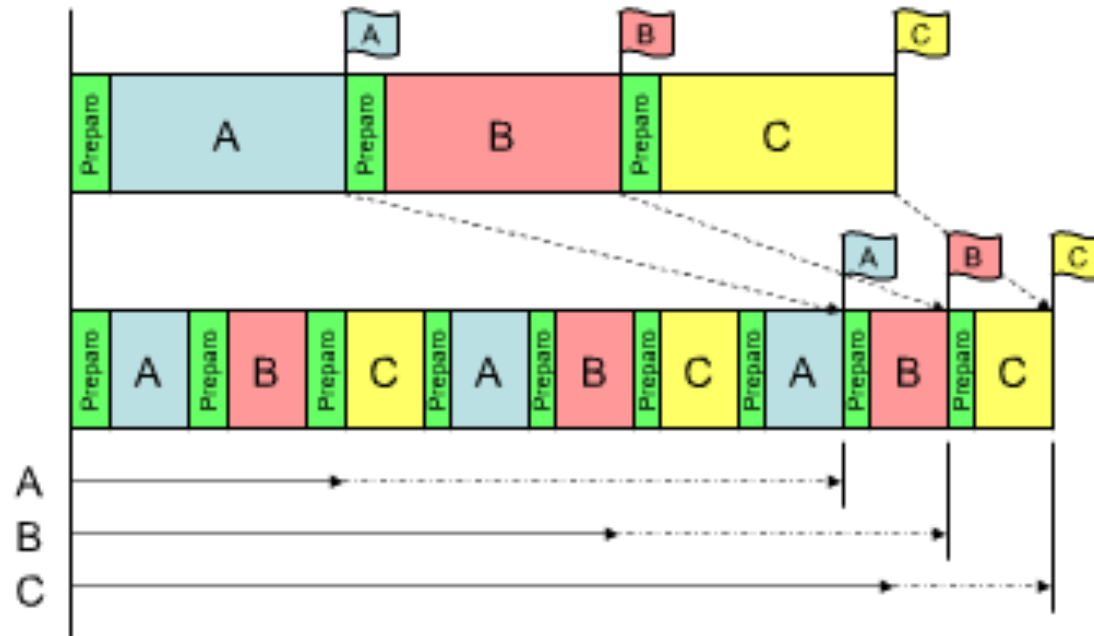
- Visão ágil

“Prioriza e foca naquilo que é mais importante!”



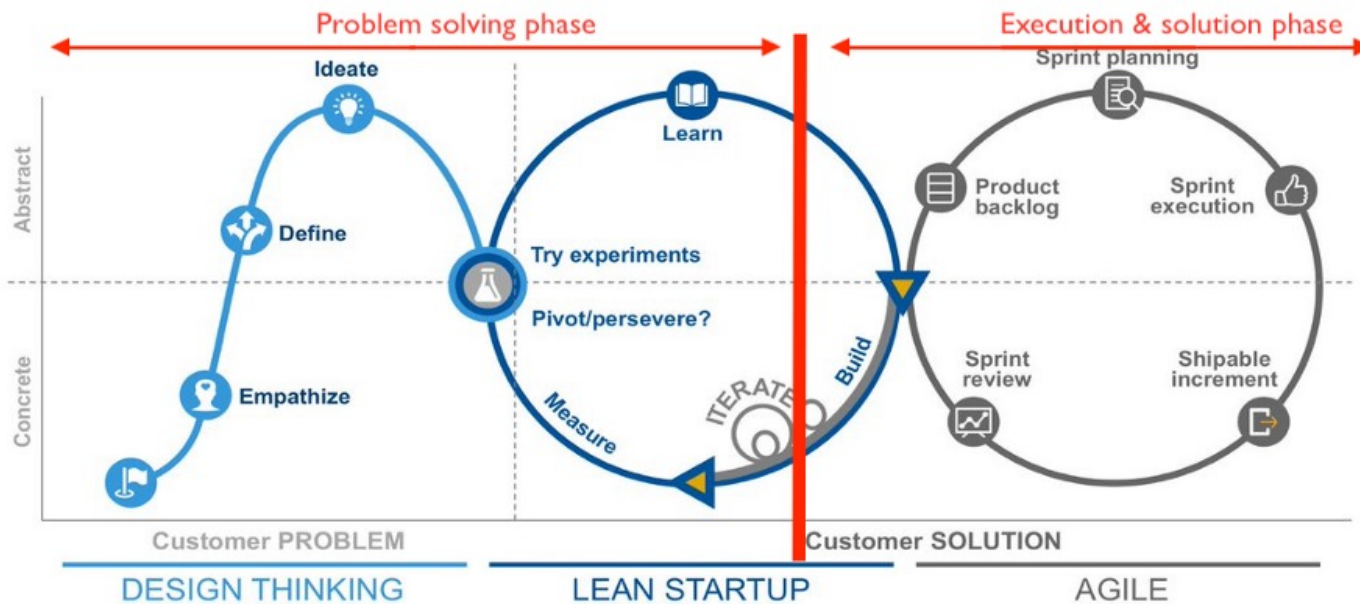
O paradoxo da multitarefa

- Na realidade...



Deixar pessoas
ociosas as vezes é
mais econômico do
que fabricar estoque
de trabalho
inacabado...

Combine Design Thinking, Lean Startup and Agile



#GartnerSYM

21 CONFIDENTIAL AND PROPRIETARY | © 2016 Gartner, Inc. and/or its affiliates. All rights reserved. Gartner and ITXpo are registered trademarks of Gartner, Inc. or its affiliates.

Gartner

Referências

- Womack, J.P.; Jones, D.T. "Lean Thinking: Banish Waste and Create Wealth in Your Corporation". Free Press, 2003.
- Ries, E. "A Startup Enxuta: como os empreendedores atuais utilizam a inovação contínua para criar empresas extremamente bem-sucedidas". Texto Editora. 2012.
- Caroli, Paulo. "Lean Inception. Editora Caroli, 2018
- Mary e Tom Poppendieck, "Lean Software Development: An Agile Toolkit", Addison-Wesley, 2003
- Mary e Tom Poppendieck. "Implementando o Desenvolvimento Lean de Software: Do Conceito ao Dinheiro". Bookman, 2011.