Uma Introdução ao Desenvolvimento de Software Lean

Eduardo Katayama
Alfredo Goldman
Claudia Melo

IME-USP

Software Metrics and Agile group at USP

AgilCoop

DCC, IME - USP
Minicurso SBQS'2012

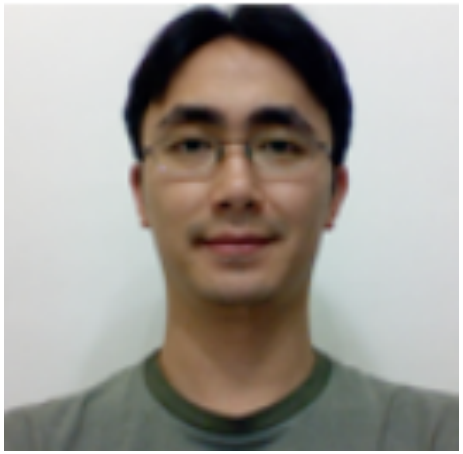
Sobre o curso

- Motivação
- Histórico
- Valores
- Princípios
- Práticas
- Pergunte aos Poppendieck
- Um pouco de Lean Startup

Sobre nós 1/2

- Todos membros da AgilCoop
  - Criada para inicialmente para fomentar Métodos Ágeis
  - Hoje mudou um pouco o foco:
    - “objetivo contínuo de se manter na vanguarda de desenvolvimento de software e fornecer informações de qualidade que ajudem a disseminar conceitos inovadores na comunidade e organizações”.

Sobre nós 2/3



Eduardo Teruo Katayama

- Mestre pelo IME - USP
- Dissertação sobre Lean e Theory of Constraints (ToC)

Sobre nós 3/3

- Grande experiência na indústria
- Mestre pelo IME - USP
- Doutoranda pelo IME - USP
  - Produtividade de Times Ágeis

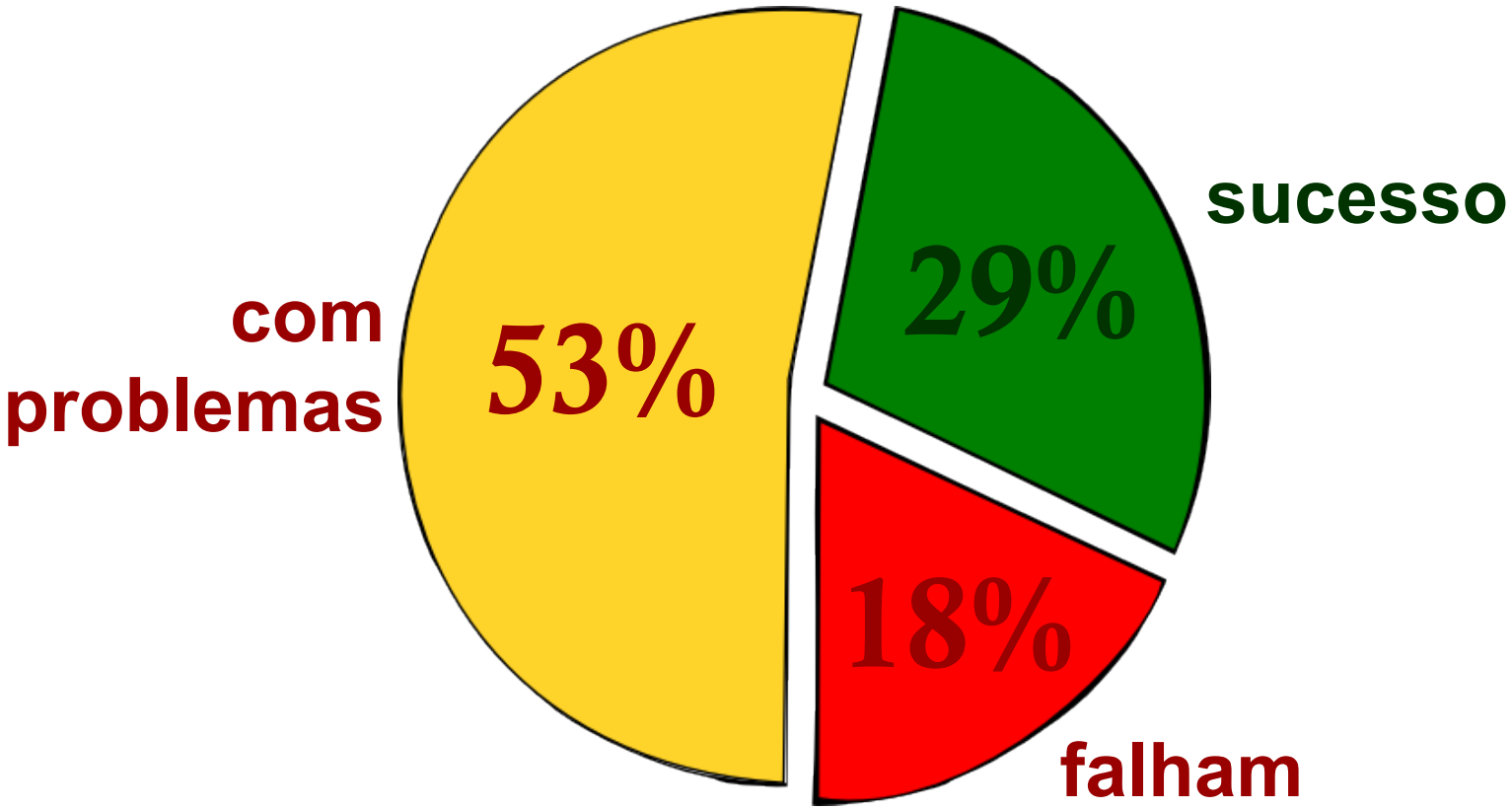
Claudia Melo

Alfredo Goldman

- Professor do IME - USP desde 93
- Trabalha com métodos ágeis desde 2002
  - Orientou 5 mestrados e orienta 2 doutorados

Motivação

Problemas com Software



CHAOS report, 2004

Cenário geral

	1994	1996	1998	2000	2002	2004	2006	2009
Successful	16%	27%	26%	28%	34%	29%	35%	32%
Challenged	53%	33%	46%	49%	51%	53%	46%	44%
Failed	31%	40%	28%	23%	15%	18%	19%	24%

http://www.projectsmart.co.uk/the-curious-case-of-the-chaos-report-2009.html

Slide "emprestado" do A. Cockburn

What is / isn't software development?

Model Building	**(Jacobson)**
Engineering	**(Meyer)**
Discipline	**(Humphreys)**
Poetry	**(Cockburn)**
Math	**(Hoare)**
Craft	**(Knuth)**
Art	**(Gabriel)**

If you know what it is,
you can apply known solutions.

Jacobson, 08/2007

Subscribe to JOT's newsletter

O-O NEWS & EVENTS

Previous column next article

REFEREED COLUMN

Enough of Processes - Lets do Practices

PDF Version

Ivar Jacobson, Pan Wei Ng and **Ian Spence** Ivar Jacobson Consulting

Abstract

All modern software development processes try to help project teams conduct their work. While there are some important differences between them, the commonalities are far greater - and understandably, since the end goal of them all is to produce working software quickly and effectively. Thus, it doesn't matter which process you adopt as long as it is adaptable, extensible, and capable of absorbing good ideas, even if they arise from other processes.

To achieve this kind of flexibility things need to change. The focus needs to shift from the definition of complete processes to the capture of reusable practices. Teams should be able to mix-and-match practices and ideas from many different sources to create effective ways of working, ones that suit them and address their risks.

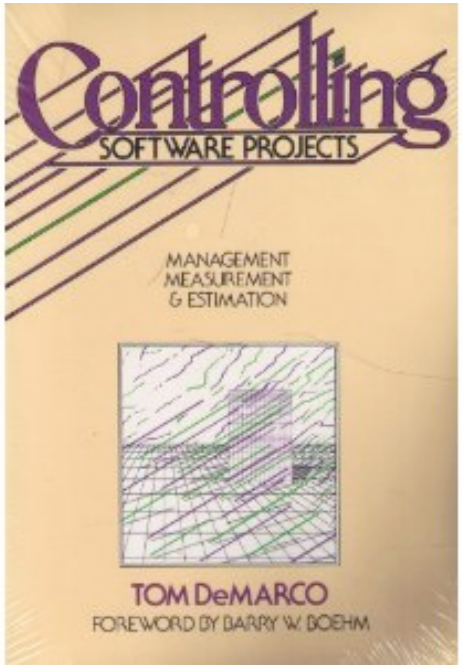
DeMarco, IEEE Software 2009

Software Engineering:

An Idea Whose Time Has Come and Gone?

Tom DeMarco

Rather, I'm advocating a management approach, one that might well steer the team toward agile methods, at least toward the incremental aspects of the agile school.



Manifesto Ágil

- **Indivíduos e interações** são mais importantes que processos e ferramentas
- **Software funcionando** é mais importante que documentação completa e detalhada
- **Colaboração com o cliente** é mais importante que negociação de contratos
- **Adaptação a mudanças** é mais importante que seguir um plano

Algumas frases 1/2

- *"...ability to bring new high quality products rapidly to the market"*
- *"...uses a relatively unstructured development process,..."*
- *"...having each worker, rather than professional inspectors, check the previous worker's result;..."*
- *"...encouraging workers to redesign their own jobs, rather than having trained industrial engineers break the work down and prescribe procedures..."*

Algumas frases 2/2

- *"Detailed process manuals can be cumbersome"*
- *"...is able to combine an emphasis on teamwork, communication and consensus.."*
- *"..general manager of styling comented that they "prefer lots of torpedoes to a single sniper bullet""*
- *"The manager's job is to prevent people from making decisions too quickly"*
- *"...key milestones rather than detailed* breakdowns of activities."

De onde vieram

- De um artigo de 1995
  - *The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster*
- e já havia o primeiro paradoxo :)
  - Just-in-time
  - Cada um é responsável
  - etc

Histórico

Introdução

- Empresas que aplicam conceitos de Lean

Produção / Manufatura

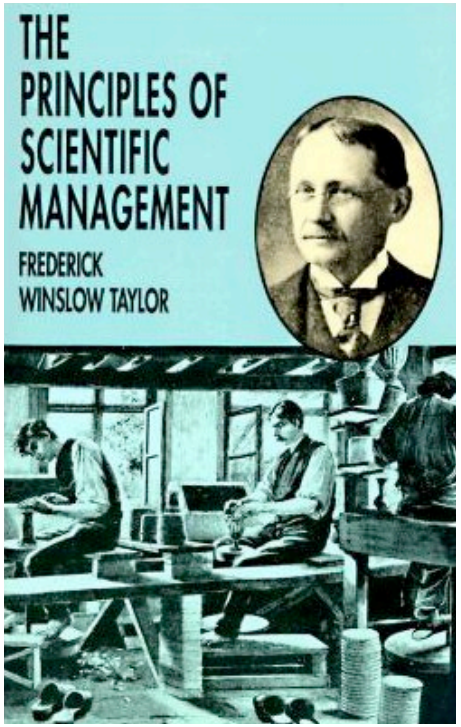
Produção / Manufatura +
Desenvolvimento de Produtos

Cadeia de Suprimento

Desenvolvimento de Software

The One Best Way

- Suposições
  - Trabalhadores irão fazer o mínimo possível
  - Trabalhadores não se preocupam com qualidade
  - Trabalhadores não são capazes de descobrir qual a melhor forma de realizar o trabalho
- Visão sobre eficiência
  - Experts deveriam simplificar ao máximo as tarefas, de forma que cada trabalhador realizasse rapidamente uma parte do processo como um todo.

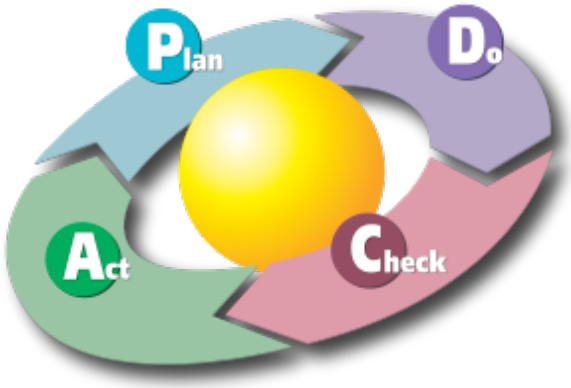


Fordismo

- Baseado em linha de montagem
- Produção em massa
- Produção de somente 1 modelo
- *"Any customer can have a car painted any color that he wants so long as it is black"* -- Henry Ford

Deming

- Sistemas de conhecimento profundo



Ciclo de Shewhart

Sistema de Produção da Toyota

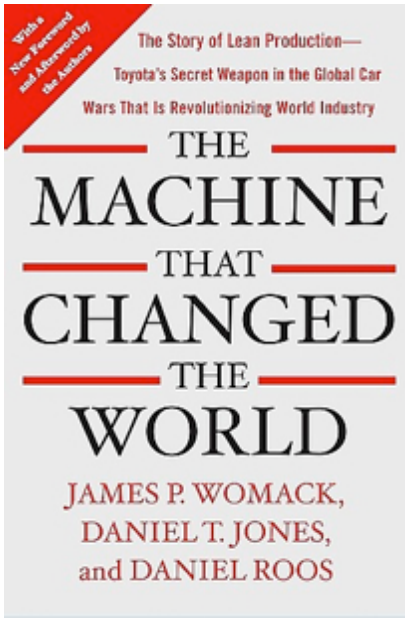
- Taiichi Ohno
  - The Toyota Production System, 1988 (1978)
    - Elimine desperdícios
      - Fluxo "Just-in-Time"
    - Exponha os problemas
      - Stop-the-Line

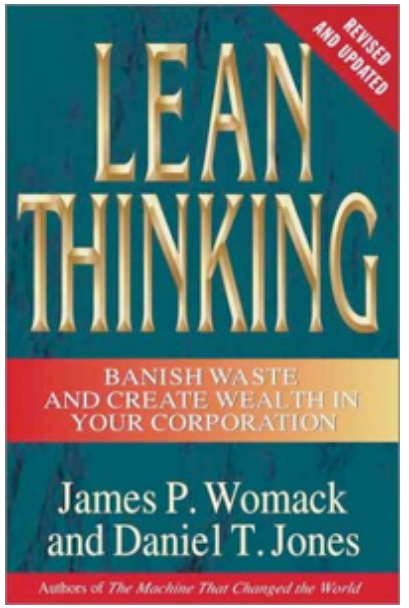
Sistema de Produção da Toyota

- Shigeo Shingo
  - Study of 'Toyota' Production System, 1981
    - Produção sem estoque
      - Setup de apenas um dígito
    - Zero inspeções
      - Mistake-Proof

Princípios de Lean Thinking

- Valor
- Eliminar desperdício
- Fluxo contínuo
- *Pull*
- Busca da perfeição



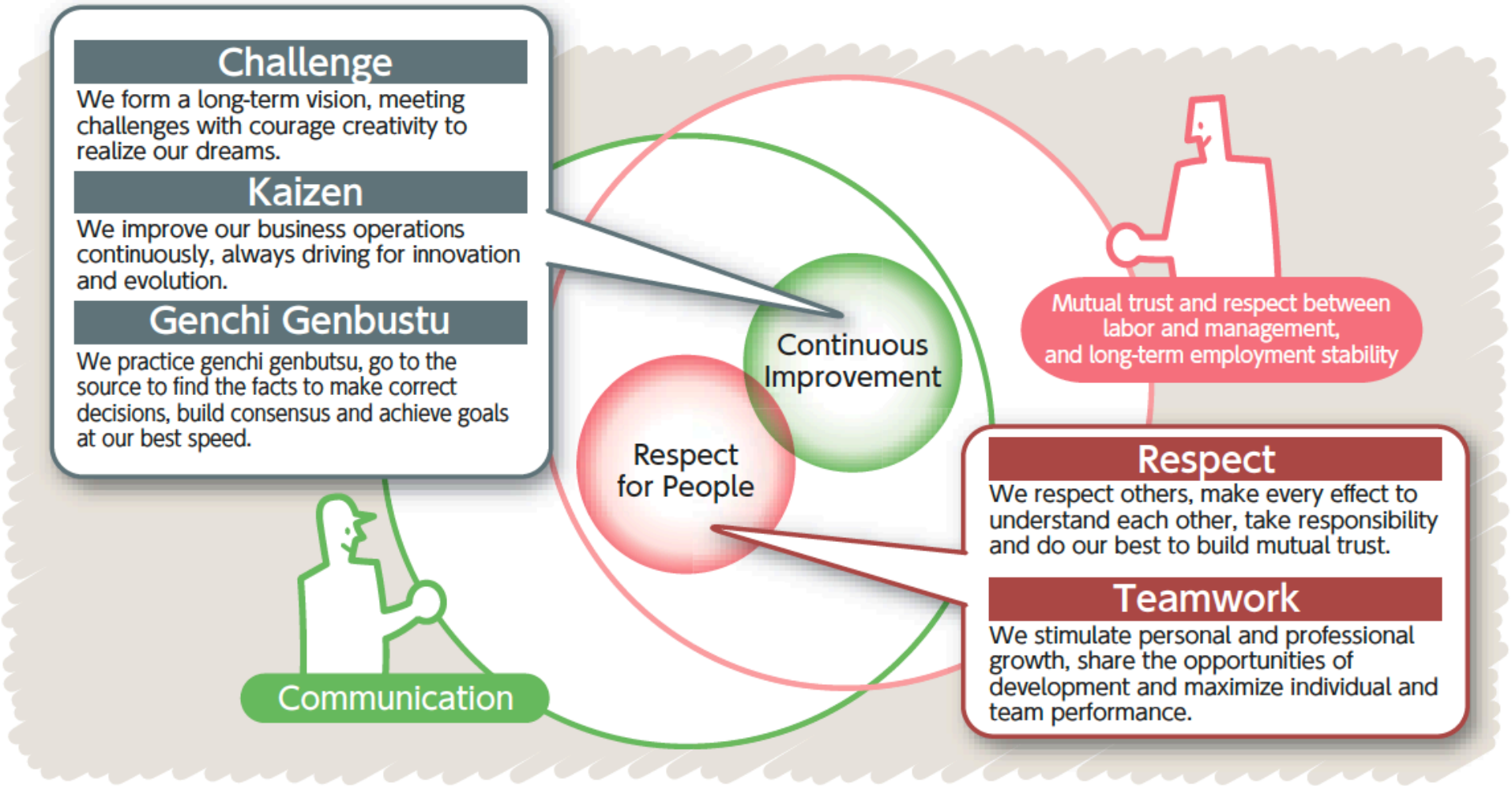


Valores

Abordagem Lean

- Definição do que o cliente valoriza em um produto
- Ações necessárias para a criação do produto são identificadas
- Ciclo constante de inspeção, adaptação e melhoria

Valores de Lean



http://www.toyota-global.com/sustainability/sustainability_report/

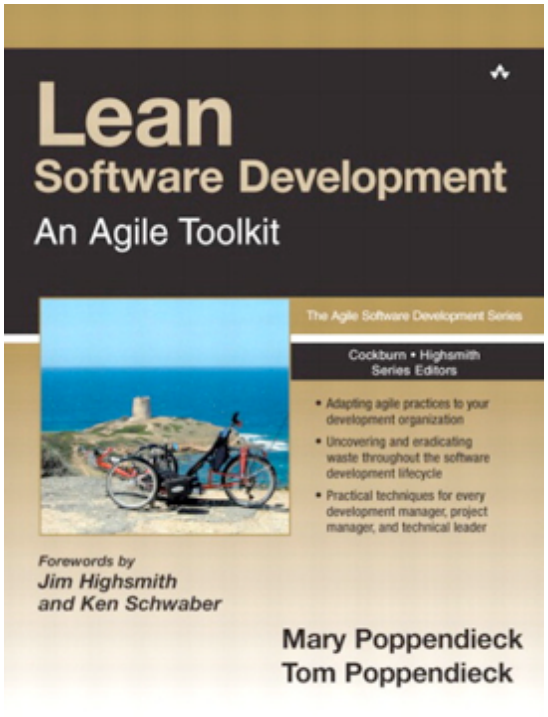
Sistema de Produção da Toyota

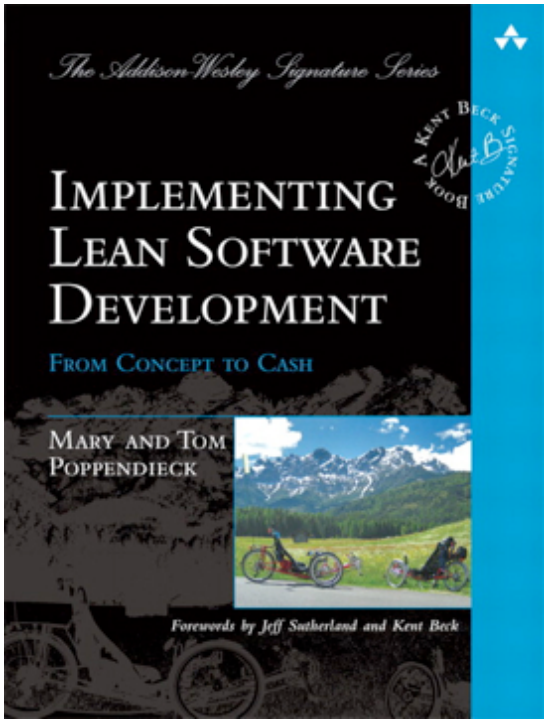
- Os valores foram expandidos para outras áreas:
  - Produção Lean
  - Manufatura / Operações Lean
  - Cadeia de suprimentos
  - Desenvolvimento de Produtos Lean
- Desenvolver software é criar um novo produto!

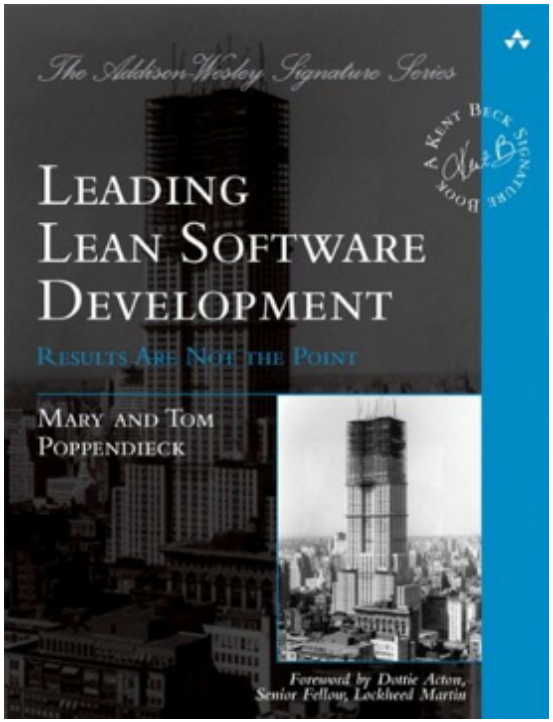
Manufatura x Desenvolvimento

- Por que utilizá-la como referência?
- Aspectos em comum:
  - Objetivo, Inventário
- Aspectos diferentes
  - Custo de mudanças, informações chegam mais tarde, variabilidade

Lean Software Development







Princípios

Princípios

- Elimine desperdícios
- Inclua a qualidade no processo
- Crie conhecimento
- Adie comprometimentos
- Entregue rápido
- Respeite as pessoas
- Otimize o todo

1. Elimine desperdícios

- Desperdício é...
  - qualquer atividade que gasta tempo, esforço, espaço, ou dinheiro e não agrega valor
  - Para enxergar o que é valor é necessário observar o processo através da perspectiva do cliente

1. Elimine desperdícios

- Lucro = Valor do produto - Custo
- Desperdício é qualquer trabalho feito parcialmente
- Funcionalidade que não é necessária agora é desperdício
- Desperdício é fazer algo errado ou fazer errado algo

1. Elimine desperdícios

- Os sete desperdícios de software:
  1. Trabalho feito parcialmente
  2. Funcionalidade a mais
  3. Reaprendizado
  4. Multitarefa
  5. Handoff
  6. Tempo de espera
  7. Defeitos

Muda (無駄)

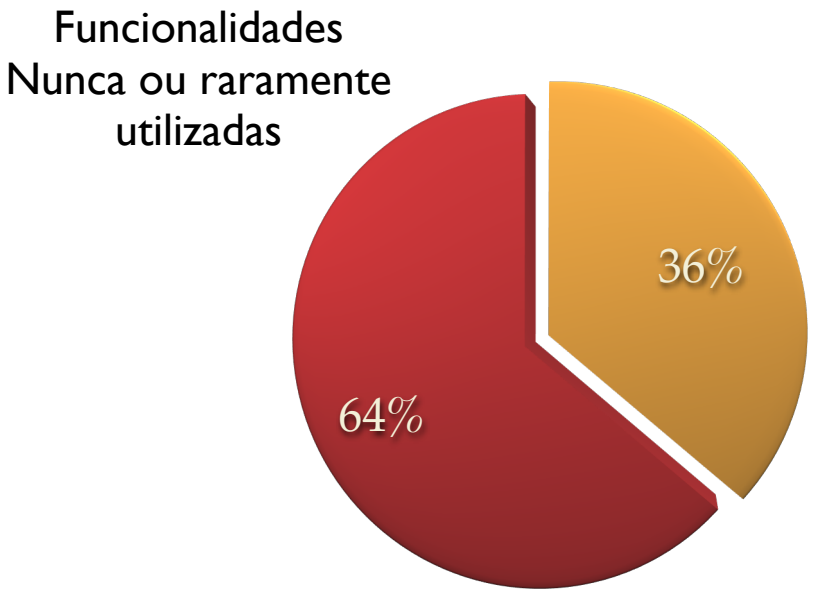
- **Desperdício**
- Toda e qualquer atividade que não agrega valor

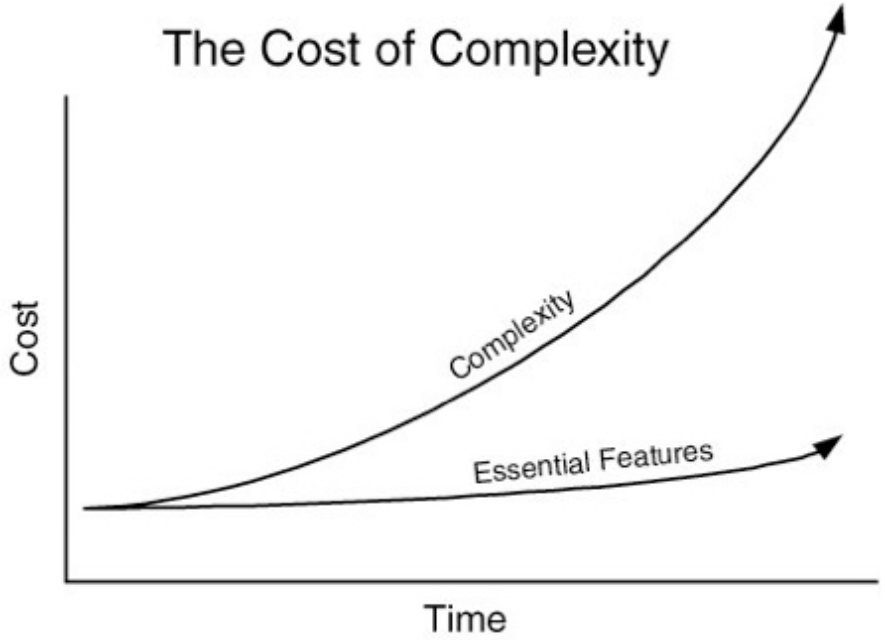
Muda (無駄)

Exemplos

- Carros parados na fábrica
- Pilha de portas de carros no chão da fábrica
- Reuniões sem fim
- Bugs!!
- Funcionalidades que o cliente não usará!!
- Relatórios e documentos que ninguém lê

Funcionalidade a mais





- A maior oportunidade de melhorar a produtividade no desenvolvimento de software é escrevendo menos código.

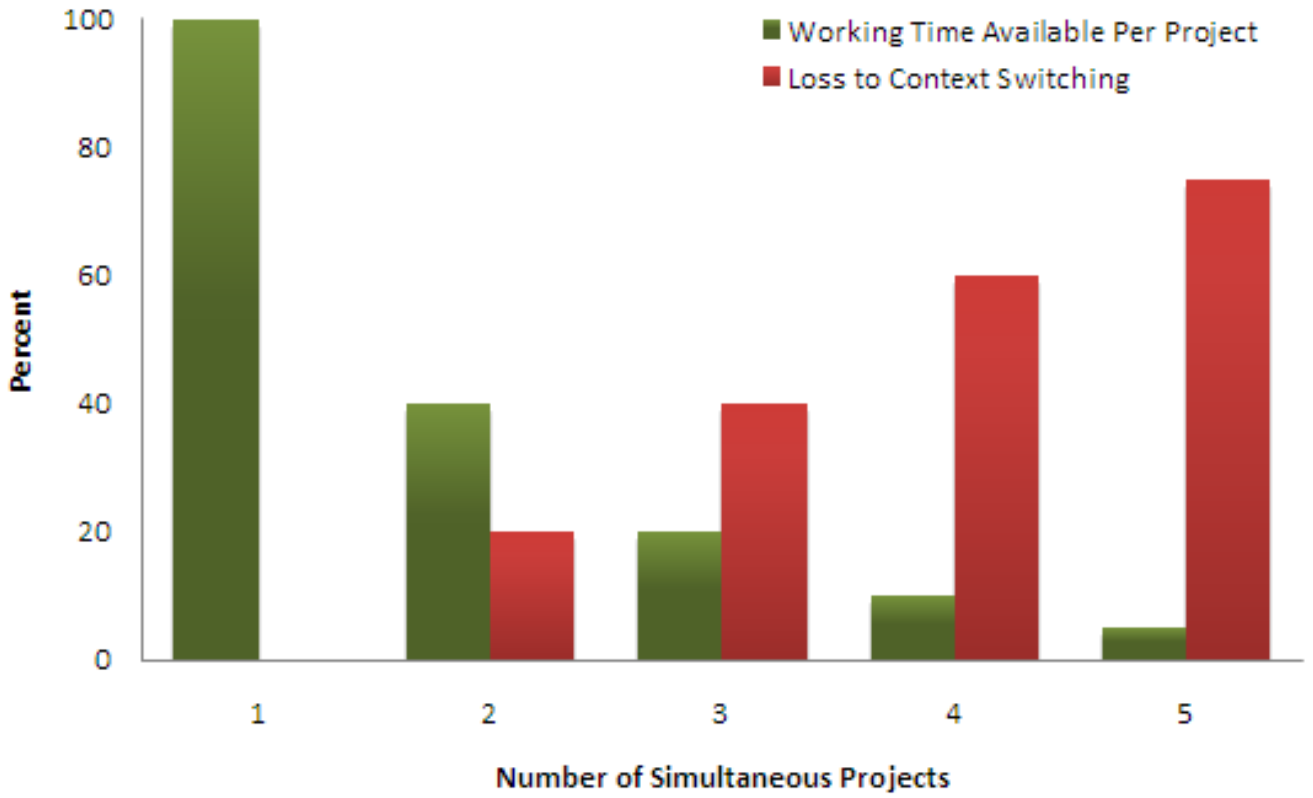
Handoff

- Handoff ocorre quando separamos:
  - Responsabilidade - O que fazer
  - Conhecimento - Como fazer
  - Ação - Fazer
  - Feedback - Aprendizado sobre resultados

Multi-tarefa é desperdício

- Exercício:
  - http://davecrenshaw.com/multitasking-example/

Multi-tarefa



Quality Software Management: Systems Thinking, Gerald Weinberg

Tempo de espera

- Feedback lento
- O produto ainda satisfaz a necessidade do cliente?
- Nenhum cliente está interessado em estar na sua fila!

2. Inclua a qualidade no processo

- "Inspecionar para prevenir defeitos é bom; Inspecionar para encontrar defeitos é desperdício" -- Shigeo Shingo
- Um processo de qualidade inclui qualidade sempre
  - ao invés de deixar para depois para testar a qualidade do produto
  - se você encontra defeitos no fim do seu processo
  - seu processo é defeituoso!

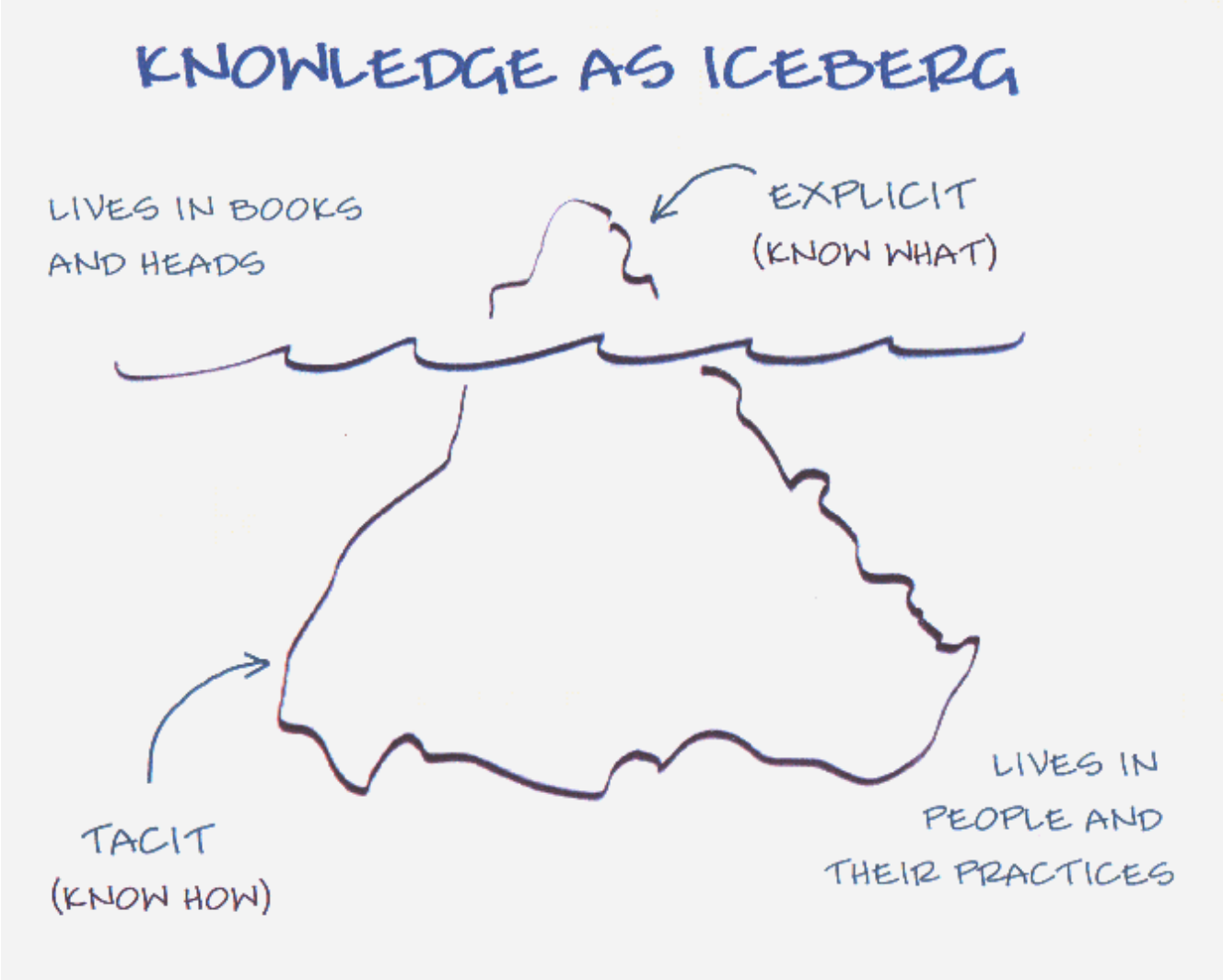
2. Inclua a qualidade no processo

- Crie código que revele a intenção
- Revise seu design/código
- Crie testes automatizados
- PARE se seus testes não passarem
- Use integração contínua
- Análise por que defeitos ocorreram durante o processo

3. Crie conhecimento

- Metáfora: criar vs preparar uma receita
- Incentive o compartilhamento de conhecimento tácito
- Buscar um processo "padrão" engessa
- O processo deve ser continuamente melhorado

Conhecimento tácito x explícito



"Nós sabemos mais do que podemos contar"

M. Polanyi, Knowing and Being. Chicago, IL: University of Chicago Press, 1969.

4. Adie comprometimentos

- Decisões irreversíveis devem ser tomadas o mais tarde possível (*last responsible moment*)
- Real options
- É preciso definir o momento da decisão
- Flexibilidade arbitrária também é ruim

4. Adie comprometimentos

- Exemplo: Toyota Prius
  - 15 meses do conceito ao lançamento
  - 10 opções de motores híbridos desenvolvidos durante os 4 primeiros meses
  - Motores híbridos viraram item opcional

4. Adie compromentimentos

- 60-80% de todo software é desenvolvido após o primeiro release
- Um processo de desenvolvimento que antecipa mudanças irá resultar em um software tolerante a mudanças
- Tome decisões reversíveis a qualquer momento
- Tome decisões irreversíveis no último momento possível

4. Adie comprometimentos

- *Set Based Design*
  - Na incerteza, experimente diversas soluções
  - Agende o momento da decisão
  - Sempre deve haver uma solução que funciona no prazo
  - Paradoxo: Isso não é desperdício

5. entregue rápido

- “Devemos encontrar uma maneira de entregar software tão rápido que nossos clientes não tenham tempo de mudar de ideia”-- Mary Poppendieck

- Empresas que competem com base na velocidade:

  - Possuem um processo com menos desperdício, poucos problemas e um profundo conhecimento das necessidades do cliente

  - Possuem uma taxa de defeitos muito baixa

6. Respeite as pessoas

- Pessoas são recursos?
- Papel da gerência é distribuir tarefas e monitorar?

6. Respeite as pessoas

- 3 pilares estão relacionado às pessoas
  - Liderança
  - Força de trabalho com conhecimento
  - Planejamento e controle baseado em responsabilidade
- Liderança
  - Grande conhecimento do cliente
  - Grande conhecimento técnico
- Times completos

6. Respeite as pessoas

- "A verdadeira inovação da Toyota é sua habilidade em usufruir da inteligência dos trabalhadores 'comuns' "
- Mova a responsabilidade e o poder de decisão para o nível mais baixo possível
- Na Toyota as promoções são ligadas ao espírito Lean

7. Otimize o todo

- Círculo vicioso #1 no desenvolvimento de software
  - Cliente pede nova funcionalidade, para ontem
  - Desenvolvedor ouve: Termine isso rápido!
  - Resultado: Mudanças feitas de qualquer jeito no código
  - Resultado: Complexidade do código aumenta
  - Resultado: Número de defeitos no código aumenta
  - Resultado: Tempo para adicionar funcionalidade cresce exponencialmente

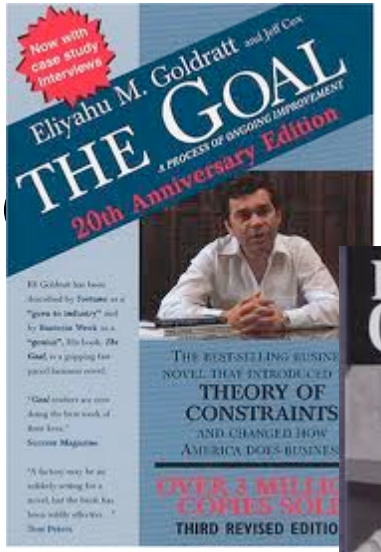
7. Otimize o todo

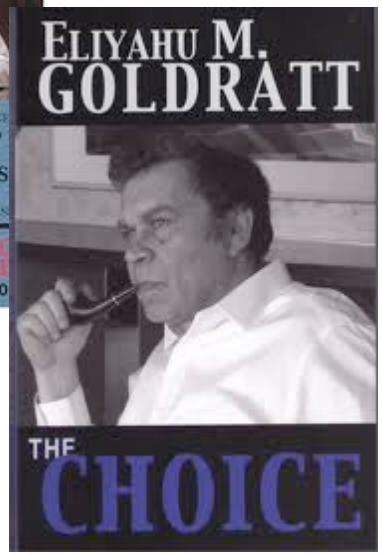
- Círculo vicioso #2 no desenvolvimento de software
  - Equipe de testes sobrecarregada
  - Resultado: Testes são feitos depois da codificação
  - Resultado: Desenvolvedores não receberem feedback imediato
  - Resultado: Desenvolvedores criam mais defeitos
  - Resultado: Equipe de testes tem mais trabalho
  - - ... repita o ciclo
  - Mito: Micro-otimização leva à Macro-otimização

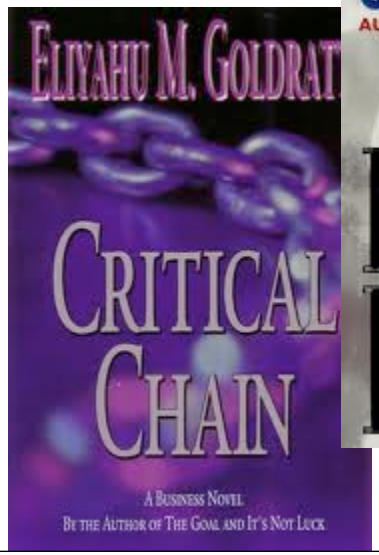
Otimize o todo

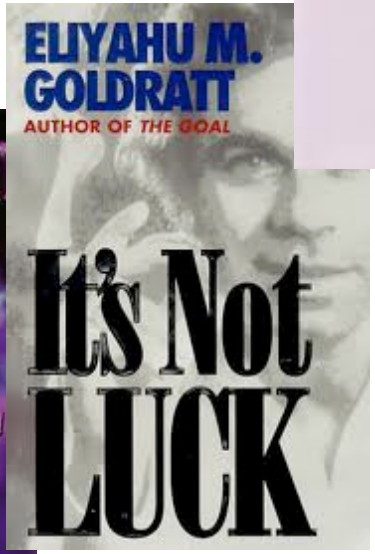
Teoria das restrições

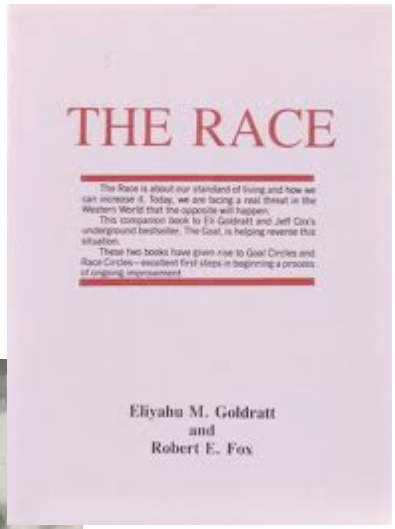
- (











Teoria das restrições

- Alcançar o objetivo através da melhoria da capacidade produtiva da organização
- Pressuposto: qualquer organização é limitada pelo desempenho de um ou poucos elementos
- Melhorias globais: "bons resultados globais não são iguais à soma de resultados locais"
- Identificar os pontos de alavancagem

7. Otimize o todo

- Identificar a restrição do sistema
  - Dica: Utilize o Mapa de Fluxo de valor
- Concentre-se em remover a restrição
  - Cuidado: Melhorar atividades que não são restrição não melhoram o fluxo do sistema como um todo
- Assim que a restrição é eliminada, encontre as acomodações que foram criadas para "suportá-la"
- Volte ao passo 1

7. Otimize o todo

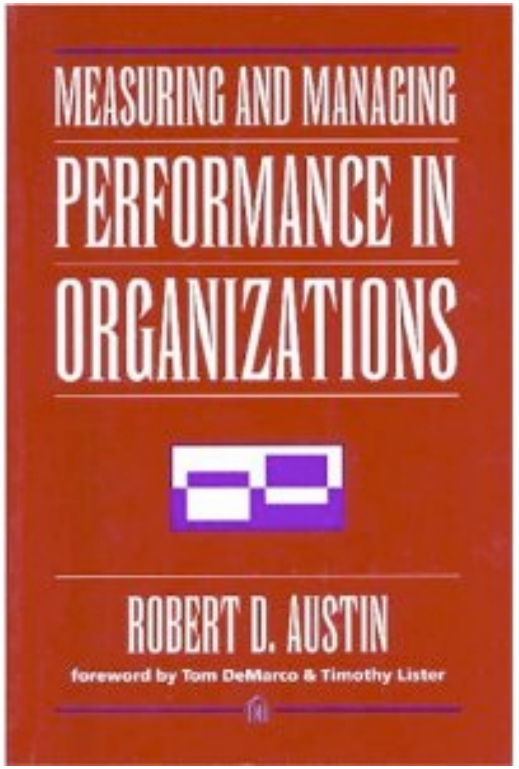
- É preciso olhar para o processo todo
- Não adianta resolver os sintomas
- É preciso resolver a causa

7. Otimize o todo

- “Diga-me como serei medido, que eu te direi como me comportarei” -- Eliyahu Goldratt
- Métricas:
  - Medir informação vs Medir desempenho
- Cuidado!
  - É fácil medir muitas coisas
  - É fácil medir as coisas erradas

7. Otimize o todo

- Diminua o número de métricas de desempenho
- Meça para cima:
  - Medidas no nível mais alto que direcionam para o comportamento correto
  - Medidas em nível de time, não de indivíduos



Recapitulando

- Elimine desperdícios
- Inclua a qualidade no processo
- Crie conhecimento
- Adie comprometimentos
- Entregue rápido
- Respeite as pessoas
- Otimize o todo

Construa o produto certo

- Maior parte dos lançamentos mal sucedidos são decorrentes de falhas na concepção dos produtos - Reinertsen97
- Maior parte das empresas se concentram em reduzir riscos técnicos durante o desenvolvimento do produto
- “Nada é mais inútil do que fazer com eficiência extraordinária algo que não precisar ser feito de jeito algum” -- Drucker

Primeiro, construa o produto certo

Falha Imediata Construir o produto errado Construir da forma correta	Sucesso Imediato Construir o produto certo Construir da forma correta Sucesso a longo prazo
Falha Imediata Construir o produto errado Construir da forma errada	Sucesso Imediato Construir o produto certo Construir da forma errada Perigo a longo prazo

Entenda o negócio

- “A pior coisa possível que você pode fazer é fazer o que seus clientes querem. Você deve entender quais são seus problemas e resolve-los” -- Per Haug Kogstad
- Produtos brilhantes são resultados de uma combinação de modelos mentais entre aqueles que desenvolvem e aqueles que usam

Software é um meio para o fim

- "Desenvolvimento de software é uma cadeia com diversos elos"-- Kent Beck
- Software é um meio para o fim
- Como o software se encaixa no plano mais amplo (big picture)?
- O objetivo no desenvolvimento de software é ser um suporte para o desenvolvimento deum produto (processo) completo que ajude o consumidor a realizar suas atividades

Como é o seu processo de desenvolvimento de software?

- Quem é seu cliente?
- Produção manufatura ou novo produto?
- Gerenciado como projeto ou produto?

Elimine desperdícios

"Desperdício é tudo aquilo que não agrega valor ao cliente"
-- Taiichi Ohno

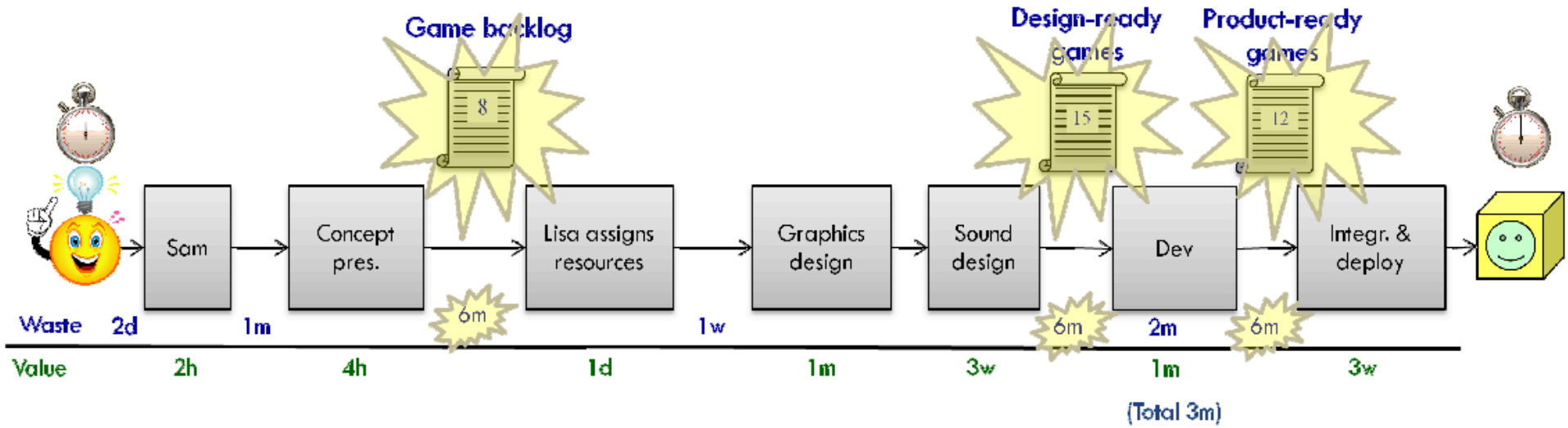
- É preciso aprender a identificar desperdícios
- Coloque-se na perspectiva do cliente

Fluxo de valor

- Tempo de ciclo (*Cycle-time*)
  - Tempo médio entre o início e fim do processo
- Começa e termina com o cliente

Mapa de fluxo de valor

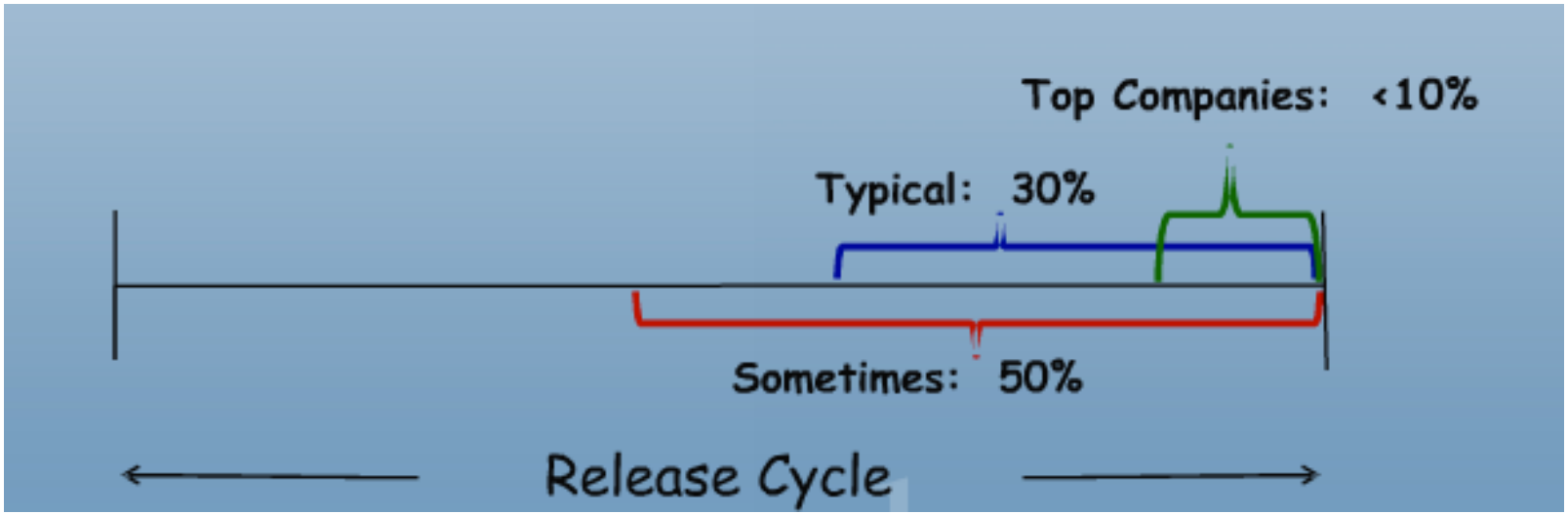
Current Condition



- Process cycle efficiency = 3 months add value / 25 months cycle time = **12%**

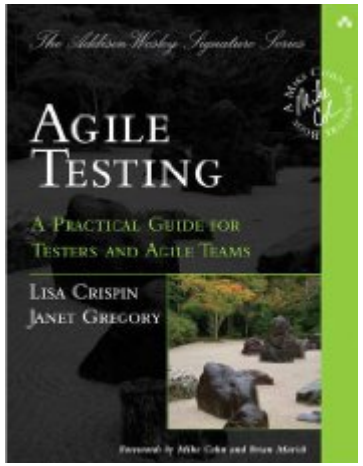
Inclua qualidade no processo

- Quando no ciclo de desenvolvimento você "congela o desenvolvimento" e testa o sistema?
  - Qual a porcentagem que sobra do tempo de ciclo?



Inclua a qualidade no processo

Continuous Integration	Test to Specification	Test to Failure
Technical Design	xUnit Tests	Stress Tests
Product Design	Acceptance Testes	Exploratory Tests
Interaction Design	Presentation Layer Tests	Usability Tests



Pessoas

- O que é um time?
  - Um time é um grupo de pessoas que se comprometem a trabalhar junto para alcançar um objetivo comum
  - Um grupo de pessoas que juntam seus esforços individuais para alcançar um objetivo pode trabalhar em grupo, mas não ser um time

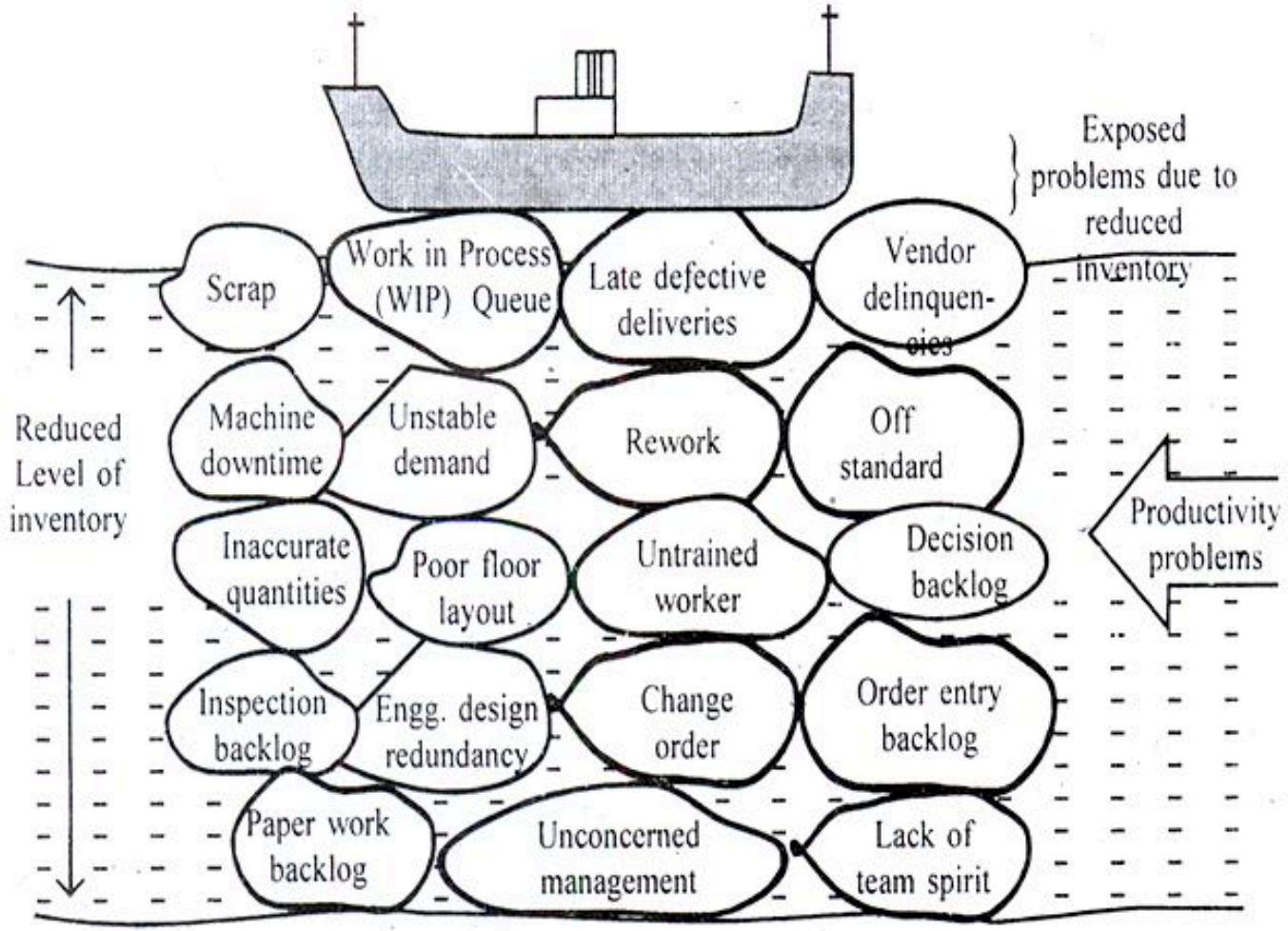
Entregue rápido

- Utilize sistemas pull
- Estabeleça uma cadência regular
- Utilize radiadores de informação
  - Kanban
  - Scrum Meeting
- Utilize pequenas unidades de trabalho

Entregue rápido

- *Lean thinking* está mais preocupado que o trabalho esteja sendo feito corretamente no momento certo, do que quem está fazendo o trabalho
- As pessoas devem trabalhar junto e eles
  - podem trabalhar em velocidades diferentes
  - podem possuir diferentes habilidades
  - mas devem trabalhar sincronizados
- Para isso, o trabalho é organizado por tarefas e a equipe é responsável por trabalhar da melhor forma possível

One Piece Flow



- O objetivo é criar um fluxo estável e contínuo, sem interrupções e sem filas

Kanban

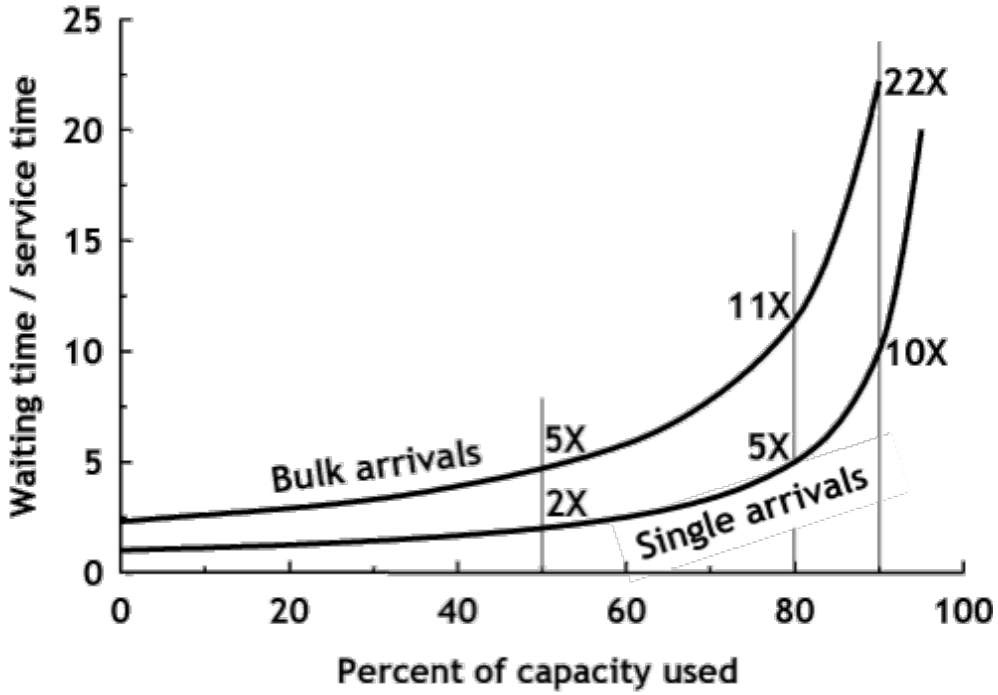
- Visualize e controle o fluxo de trabalho
- Entenda a capacidade do sistema
- Limite o *Work-in-Progress* (WIP)
- Aumente o fluxo de valor
- Inclua a qualidade no processo
- Melhoria contínua

Indicadores visuais

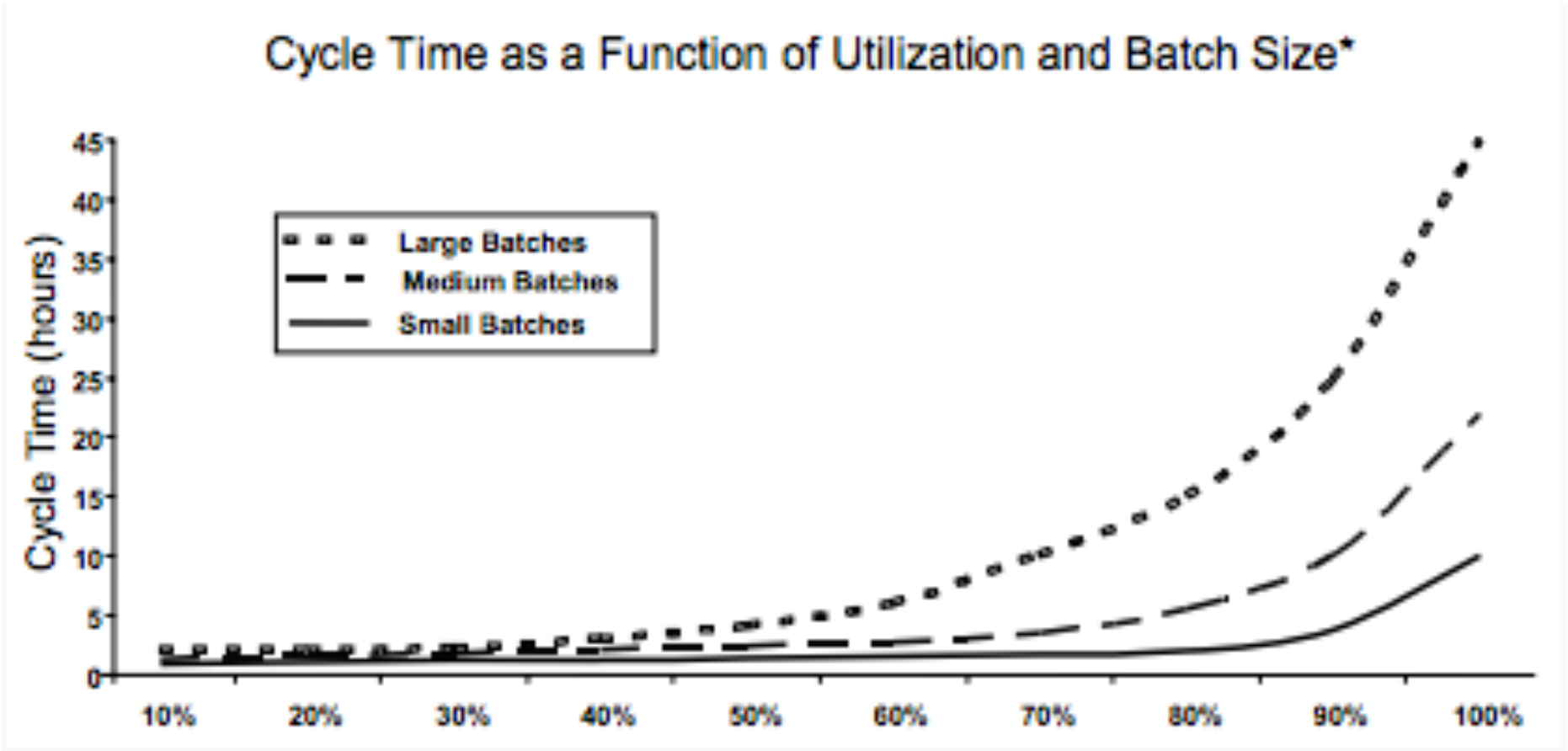
- O Kanban oferece indicadores visuais importantes sobre como está a saúde do projeto
  - Um aumento de WIP hoje, significa um aumento do tempo de entrega no futuro
  - Você não pode entregar mais do que sua capacidade
  - WIP é fácil de controlar
  - A quantidade total de WIP é a soma de todas as partes do seu fluxo

Teoria das filas

- Tempo do ciclo = $\dfrac{\text{\# atividades em processo}}{\text{tempo médio para completar a atividade}}$



Teoria das filas



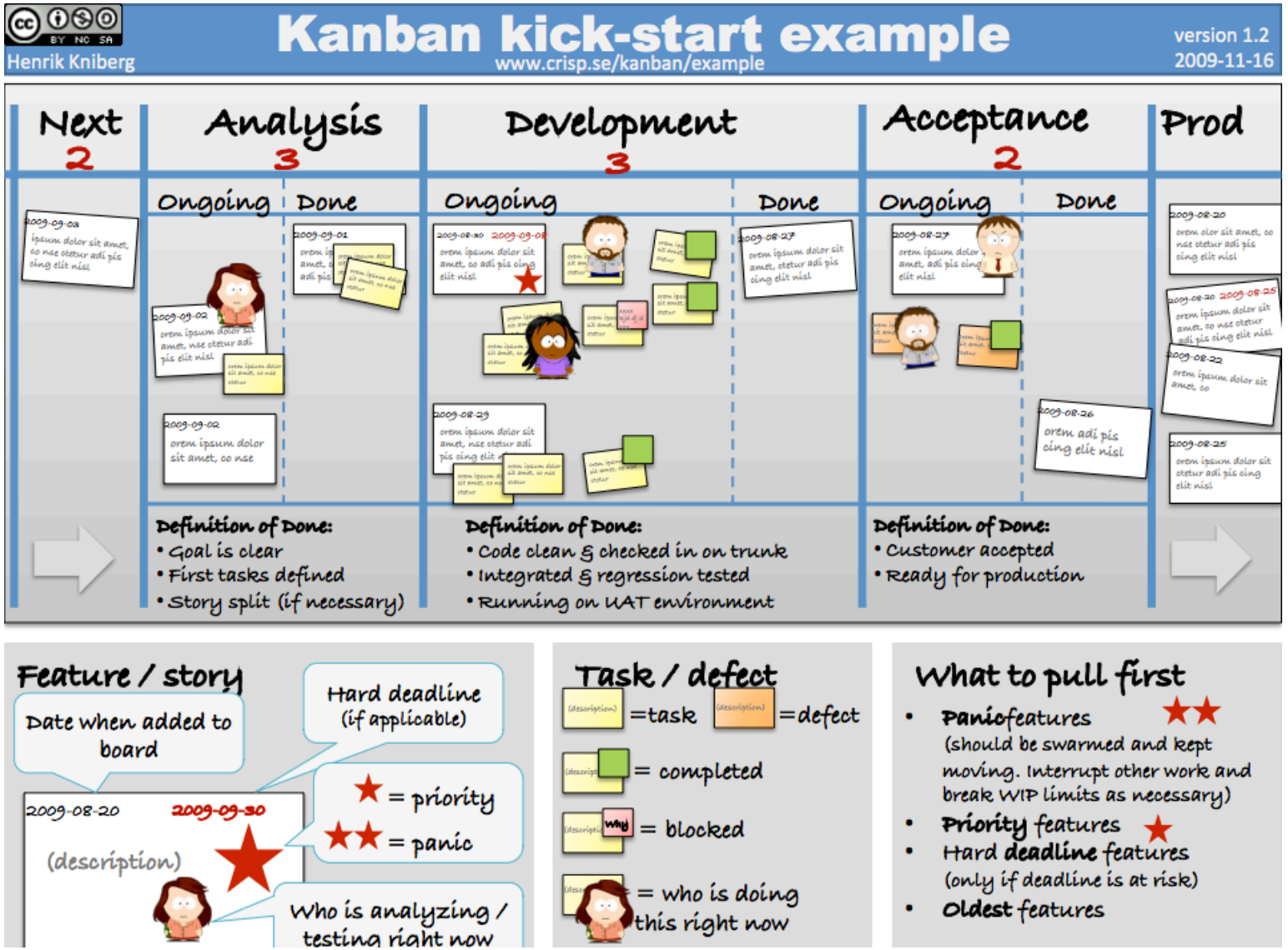
Teoria das filas

- Pequenas unidades de trabalho andam mais rápido
- Possuir alguns recursos inativos diminui o tempo do ciclo

Kanban

- Desenhar o mapa de fluxo de valor
- Para cada etapa identificada uma coluna é criada
- Em cada coluna, indicar a quantidade máxima de tarefas que podem ser simultaneamente
- Criar regras para mover as tarefas entre as colunas
- Medir o fluxo de tarefas do início ao fim

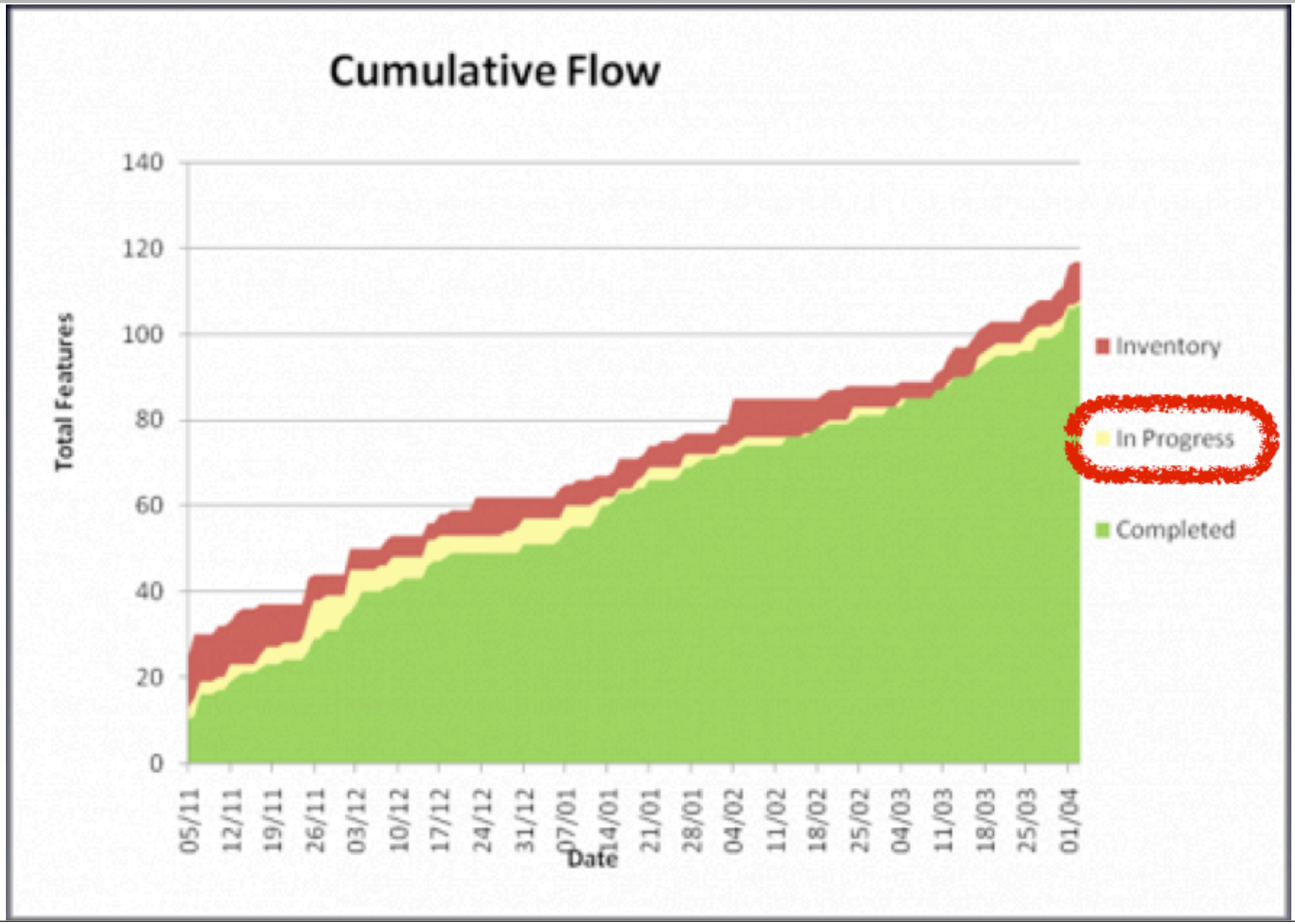
Kanban


Henrik Kniberg
Kanban kick-start example
www.crisp.se/kanban/example
version 1.2
2009-11-16
Next
2
Analysis
3
Development
3
Acceptance
2
Prod
Ongoing
Done
Ongoing
Done
Ongoing
Done
Definition of Done:
• Goal is clear
• First tasks defined
• Story split (if necessary)
Definition of Done:
• Code clean & checked in on trunk
• Integrated & regression tested
• Running on UAT environment
Definition of Done:
• Customer accepted
• Ready for production
Feature / story
Date when added to board
Hard deadline (if applicable)
2009-08-20
2009-09-30
(description)
= priority
= panic
Who is analyzing / testing right now
Task / defect
=task
=defect
= completed
= blocked
= who is doing this right now
What to pull first
Paniofeatures
(should be swarmed and kept moving. Interrupt other work and break WIP limits as necessary)
Priority features
Hard deadline features
(only if deadline is at risk)
Oldest features

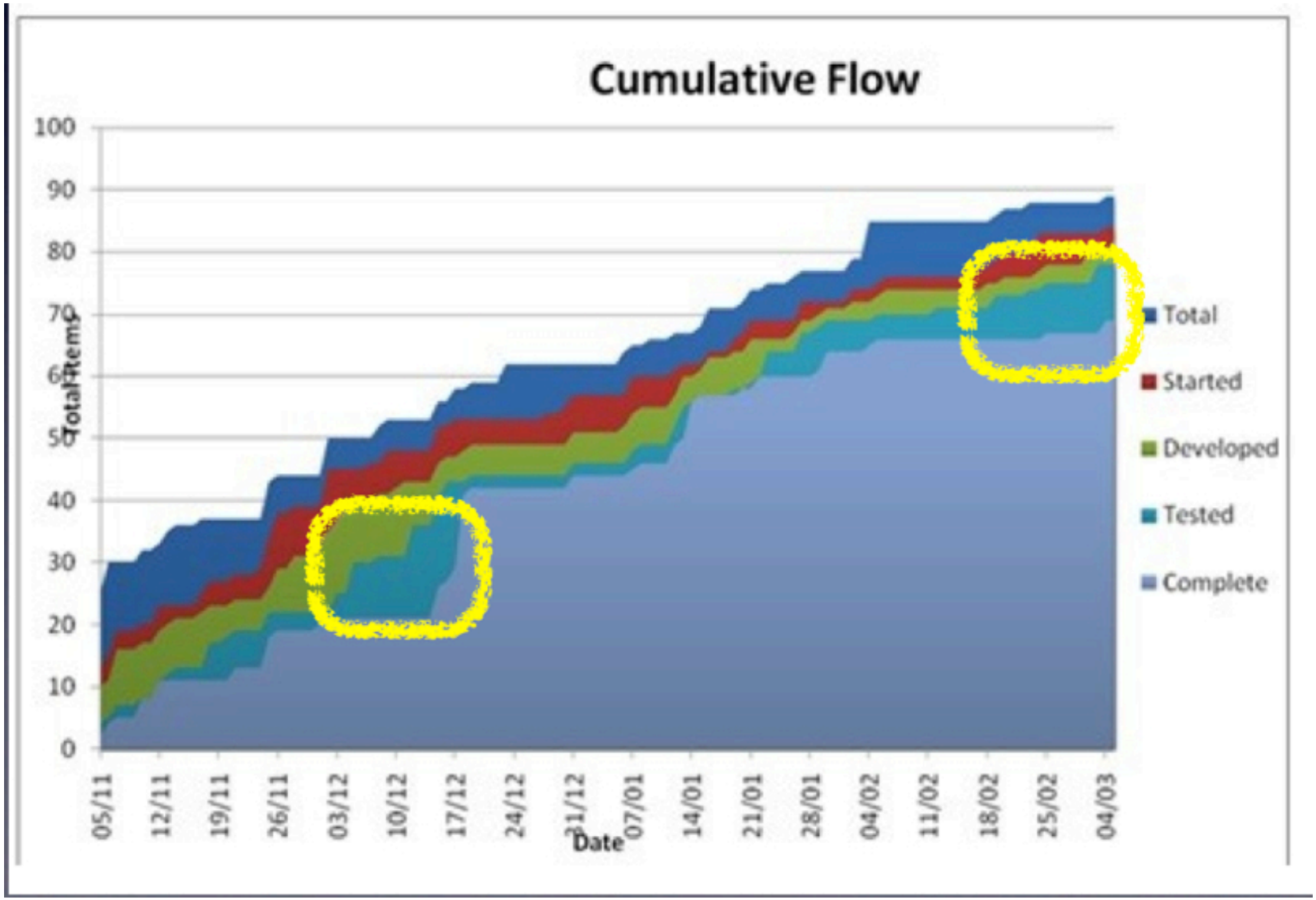
Lead Time e Cycle Time

- *Lead time*: a contagem do tempo começa quando um pedido é feito e vai até quando ele é entregue
  - *Lead time* é o que o cliente observa
- *Cycle time*: a contagem do tempo começa quando o trabalho tem início e vai até ele ser entregue
  - *Cycle time* é uma métrica da capacidade do processo

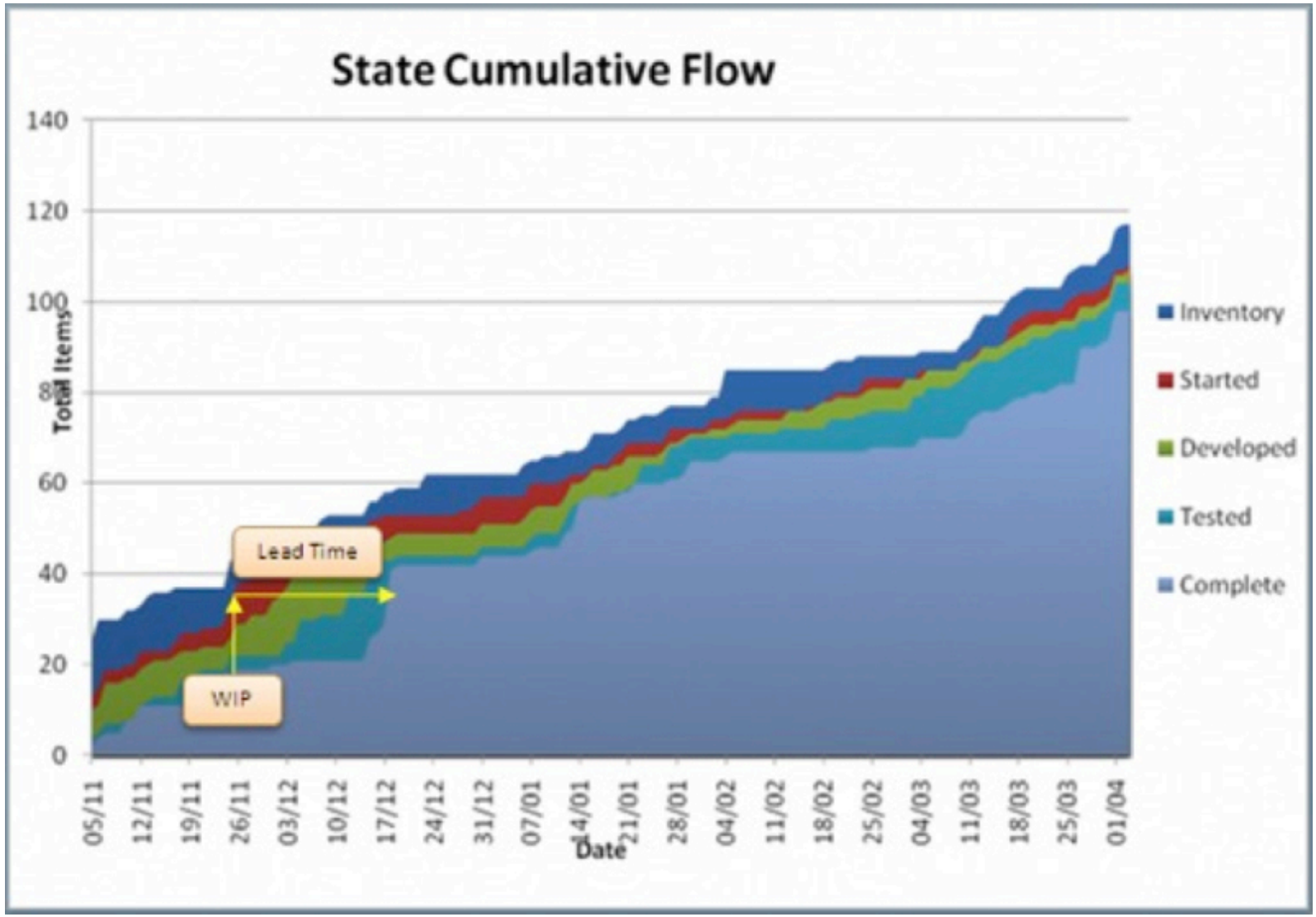
Cumulative flow chart



Cumulative flow chart



Cumulative flow chart



Melhoria contínua

- “Não é o mais forte que sobrevive, nem o mais inteligente, mas o que melhor se adapta às mudanças” -- Charles Darwin

Método científico

- Observe o fenômeno
- Formule uma hipótese para explicar o fenômeno
- Use a hipótese para fazer uma predição
- Teste a predição através de experimentos ou novas observações e modifique a hipótese com base nos resultados do teste

Usando o método científico

- Observe o problema
- Procure por sua causa raíz
- Proponha contra-medidas
- Especifique os resultados esperados
- Faça diversos experimentos rápidos
- Capture os resultados em um relatório A3
- Implemente a melhora contra-medida
- Verifique os resultados

Relatório A3

- Captura não apenas o processo de resolução do problema, mas também sua lógica
- Características:

Proprietário, Mentor e Data	Causa do problema
Tema e Contexto	Contramedidas
Situação atual	Resultados
Objetivo da melhoria	Próximos passos

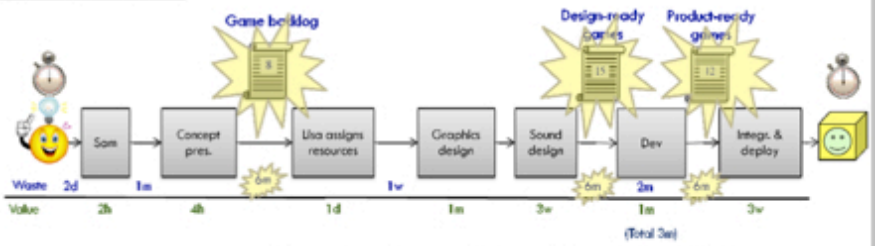
Exemplo de um Relatório A3

Background

Games out of date

- ⇒ Missed market windows – Revenue is declining
- ⇒ Demotivated teams – Key developers about to quit
- ⇒ Overhead costs – Time to develop games steadily increasing due to declining technical quality
- ⇒ Pressure to Work FASTER!

Current Condition



- Process cycle efficiency = 3 months add value / 25 months cycle time = **12%**

Goal / Target Condition

- 8x faster cycle time
- 5x fewer escaped defects
- 20% improvement in revenue

Root Cause Analysis

Owner: Lisa

Mentor: Heinrich

Date: 18 May 2009

Countermeasures

1. **Cross Functional Teams – Graphics design through deployment**
   - ✓ Predict 2x Faster Delivery
     - ⇒ End dependencies – now spend 75% of time waiting/negotiating
2. **Abandon all but most promising 3 games in each queue. Do ONE game per cross functional team at a time.**
   - ✓ 4x faster delivery from reduced task switching
   - ✓ Eliminating queues will cut 1.3 years from schedule
3. **Engage developers in playing games and selecting ideas**
   - ✓ 30% more profit to par with best competitor
     - ⇒ Improved filtering on which games to develop
     - ⇒ More fun games, more popular

Confirmation (Results)

1. **Cross Functional Teams**
   - ⇒ Half as much time waiting
2. **One game at a time**
   - ⇒ Queues eliminated, time to complete game is 4 months (6x)
   - ⇒ Technical Debt decreasing – Escaped defects down by 2x so far
3. **Engage developers in playing games and selecting ideas**
   - ⇒ One team taking time to play is producing more innovative games.
   - ⇒ Impact on profit is TBD.

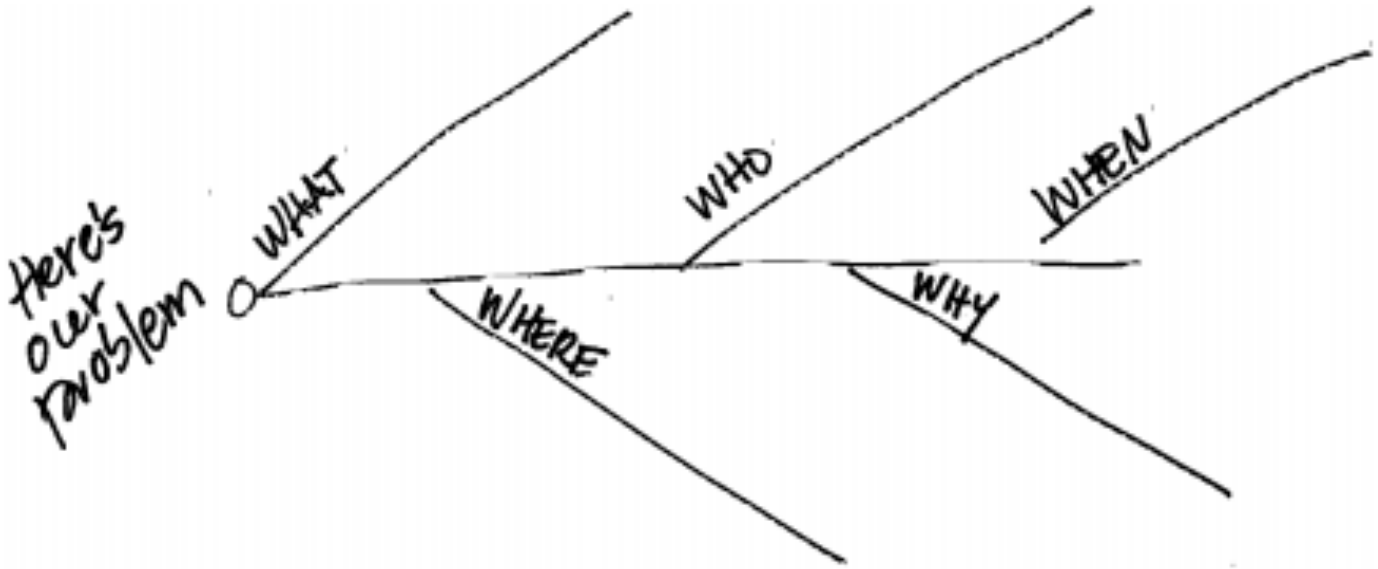
Follow-up

1. Consider more cross training of team members to reduce waiting for expertise
2. Reduce difficulty of integration and deployment steps
3. Improve processes for generating and selecting game ideas
   a. Recruit talent if identifiable/available
   b. Improve skills/process of best people already in company
   c. Broaden both participation in selection and game playing experience of everyone in the company.
4. Continue improvement of reused game components/engines to improve development throughput and reduce defects.

5 porquês

	Problema
Por quê?	Ao final da iteração as histórias não estão finalizadas
Por quê?	Porque nós subestimamos quanto tempo as hist'rias demorariam
Por quê?	Porque as histórias escolhidas eram muito complexas
Por quê?	Porque José não dividiu a história em pedaços menores para que nós conseguissemos entender
Por quê?	Porque José não sabe o que precisamos - ou porque é muito difícil dividir a história em pedaços menores
Por quê?	Porque nós não possuímos um critério de pronto para iniciar o desenvolvimento / os desenvolvedores não estão trabalhando junto com José para dividir as histórias em pedaços menores

Fishbone



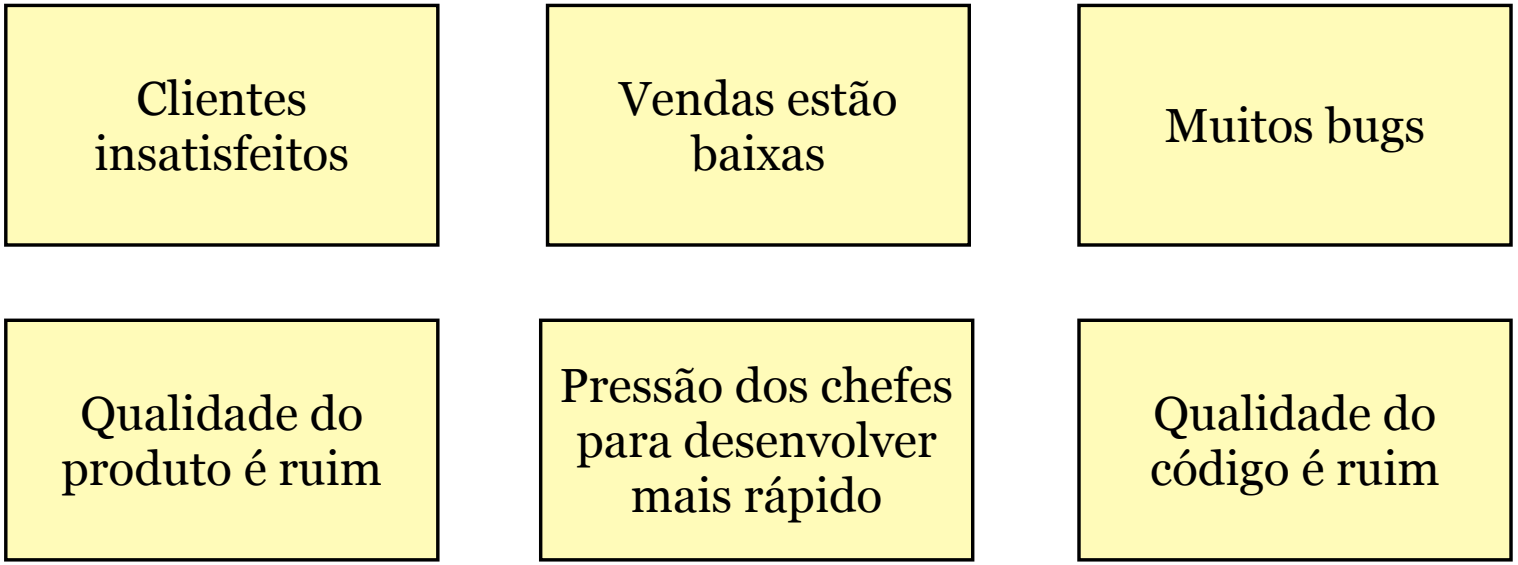
Agile Retrospectives: Making Good Teams Great!, Esther Derby and Diana Larsen

Árvore da Realidade atual

- Árvore da realidade atual
  - Condições necessárias e suficientes
  - Descreve causa e consequência entre os diversos problemas
  - Observar de forma clara os principais problemas de uma organização

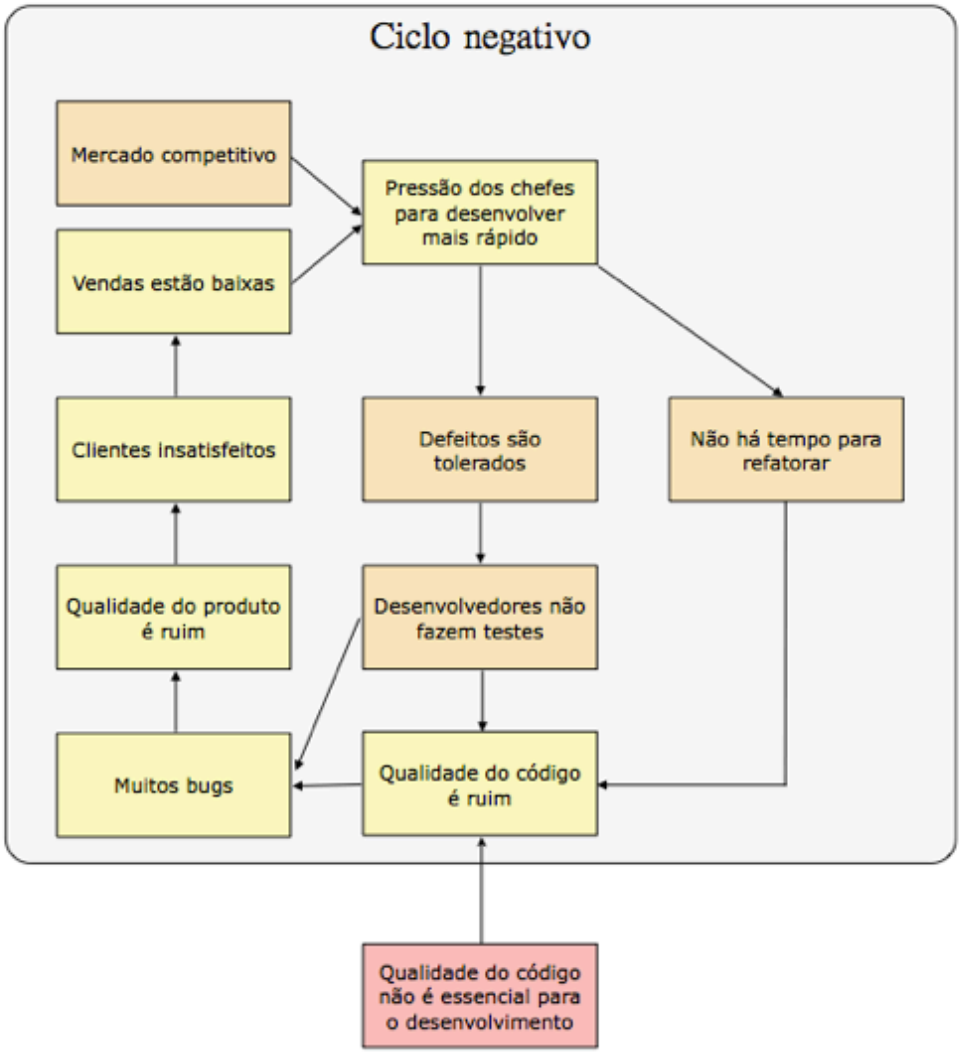
Árvore da Realidade Atual

- Lista de sintomas que estão afetando negativamente a organização



Árvore da Realidade Atual

- Relação entre os sintomas observados



Pergunte aos Poppendieck

Será que Lean é para todos ?

- Perguntas criadas pelos Poppendieck
- Motivação ver se os valores de Lean estão sendo respeitados

Pergunta 1

- A atual economia de mercado requer uma redução de custo agressiva. Qual é a melhor forma ?
  - Reduzir os custos de cada depto
  - Reduzir os custos entre os deptos

Pergunta 2

- Qual é o seu ponto de vista sobre comprometer-se a um plano e cumprir o compromisso?
  - O plano é o compromisso
  - Planos podem ser adaptados

Pergunta 3

- Qual o seu ponto de vista sobre as pessoas sempre terem trabalhos atribuídos ?
  - Para uma alocação eficiente todos devem estar trabalhando
  - O uso na capacidade máxima gera problemas

Pergunta 4

- Qual o propósito da padronização?
  - Com padrões é possível que qualquer um faça o trabalho
  - Padrões devem servir como base para melhoria

Pergunta 5

- Qual é a melhor forma de alcançar os objetivos da empresa?
  - Criar metas e métricas correspondentes
  - Melhorar a capacidade de entregar software funcionando

O início de uma jornada

- Comece onde você está
- Encontre sua maior restrição
- Visualize sua maior ameaça
- Avalie sua cultura
- Treine
- Resolva seu maior problema
- Remove acomodações
- Meça

Lean startup

Baseado na Aula 01 - Fundamentos Verão Ágil'2012

Conceitos básicos

O que é uma startup?

Uma organização temporária criada para **buscar um modelo de negócio repetível e escalável**

(Steve Blank)

Por que Lean?

- O termo "Lean" em Lean Startups possui dois significados.
  - O significado mais comum, "sem gordura"
  - "Lean" como Manufatura Lean.
    - Eficiência do capital
    - Puxe, não empurre
    - Melhoria contínua
    - Definição de valor centrada no cliente

http://www.infoq.com/br/news/2012/01/the-lean-startup-frenzy

Princípios do Lean startup

- Aprendizado validado e contínuo
- Desenvolvimento do cliente (CustDev)
- Desenvolvimento ágil
- Tecnologia como *commodity*

Lean startup:
Aprendizado Validado

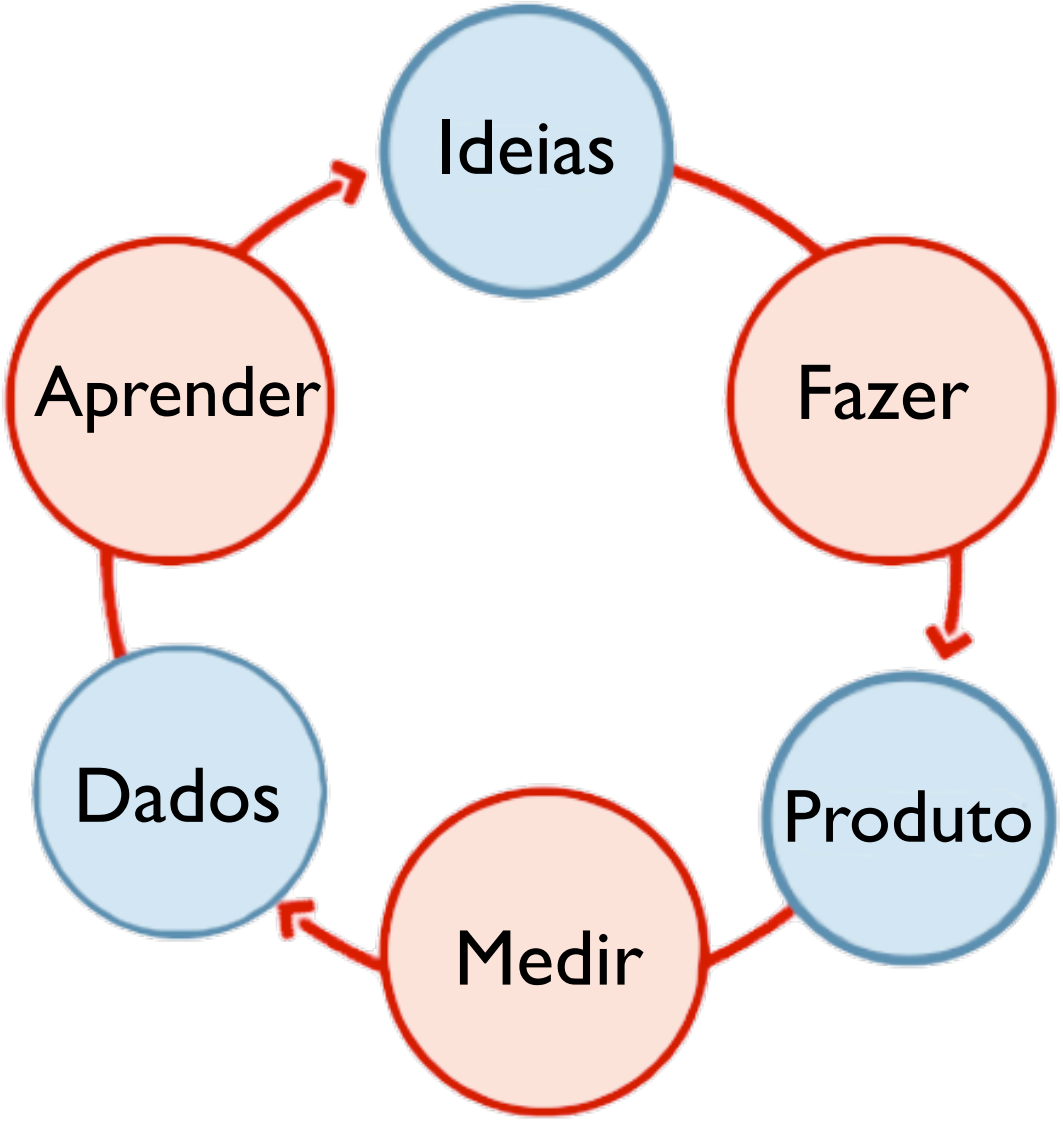
Aprendizado Validado

- Evita desperdícios com suposições
  - trabalhamos somente com métricas concretas
- Toda ação é validada através do quanto aprendemos sobre nossos clientes
  - trazendo ainda mais melhorias para nossos clientes

Loop fundamental de feedback

- A atividade fundamental de uma startup é transformar ideias em produtos, medir a resposta dos clientes e então aprender quando "pivotar" ou perseverar.
- Todos os processos de uma startup de sucesso devem ser orientados para acelerar o tempo de feedback.

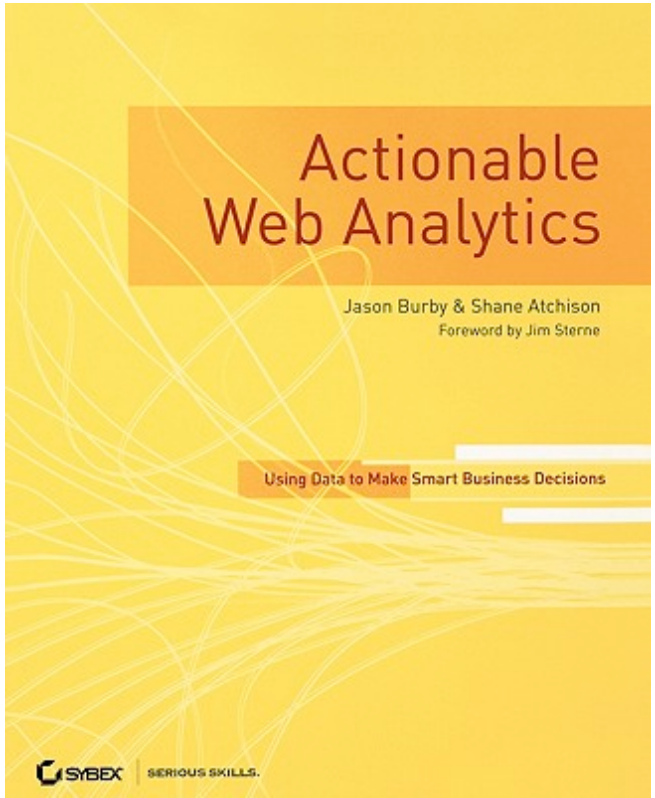
Looping básico de aprendizado



Métricas

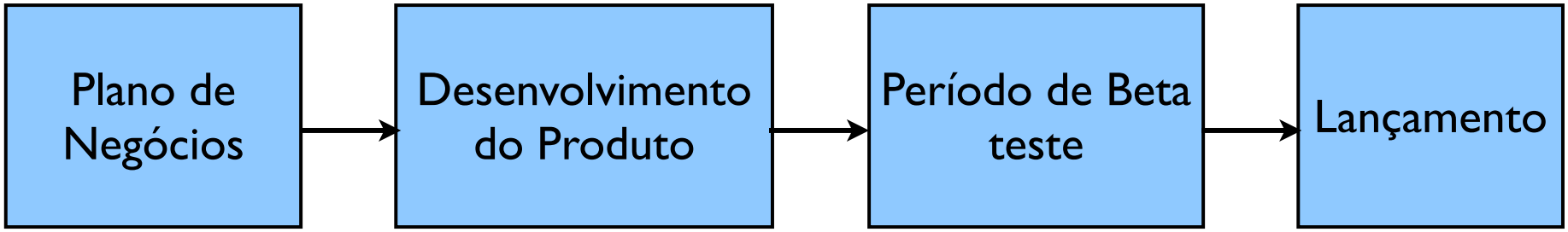
Métricas no aprendizado

- Prepare-se para ser direcionado por dados



Ok, posso começar a desenvolver meu produto?

Desenvolvimento do produto



Plano de negócios

"No plano de negócio ficam registrados o conceito do negócio, os riscos, os concorrentes, o perfil da clientela, as estratégias de marketing e o plano financeiro que viabilizará o novo empresa. Lembre-se que o plano de negócio não é um documento fechado em uma gaveta, mas um projeto vivo que você deve manter sempre atualizado" (Sebrae, 2011)

Desenvolvimento do produto

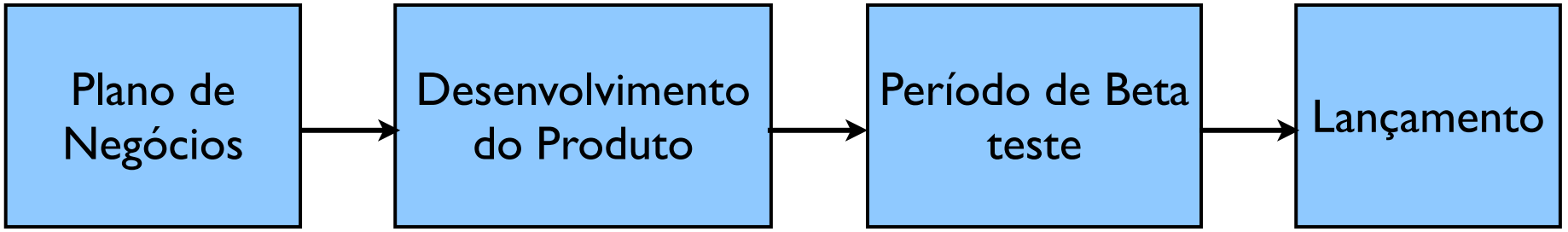
- Conversar com o cliente o máximo possível para adicionar *features* interessantes
- Diversos métodos de desenvolvimento disponíveis

Beta teste

- Testar o funcionamento do produto com alguns clientes
- Acesso *free* em troca de *feedback*

Lançamento: a hora de verdade

Desenvolvimento do produto



Qual o problema desse modelo?

Plano de negócios: Críticas

- Documento quase nunca lido
- Hipóteses que não serão realidade

Beta teste: **Riscos**

- Não se testa preço
- Baixo foco em achar bugs

Lançamento: a hora de verdade

waiting...

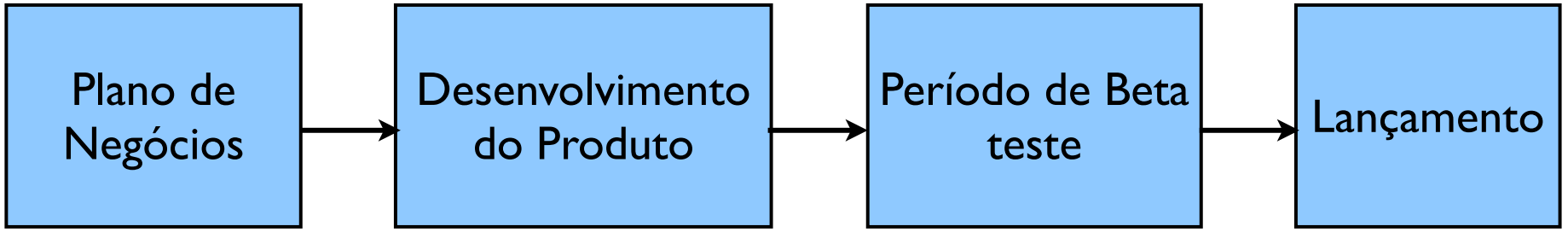
#fail

Problema geral do modelo

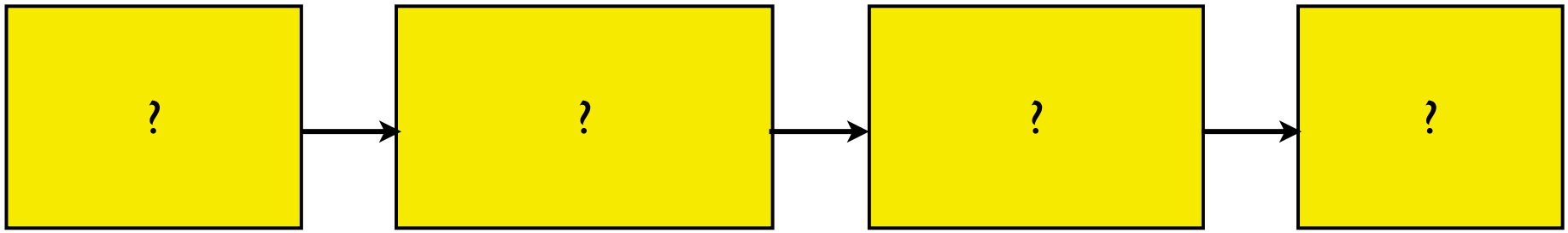
- Se a startup falha por falta de cliente (e não por falha de desenvolvimento)
- Então:
  - Por que temos processo de desenvolvimento de software e
  - Não temos um processo de desenvolvimento de cliente??

Construa

Desenvolvimento do produto



Desenvolvimento do **cliente**



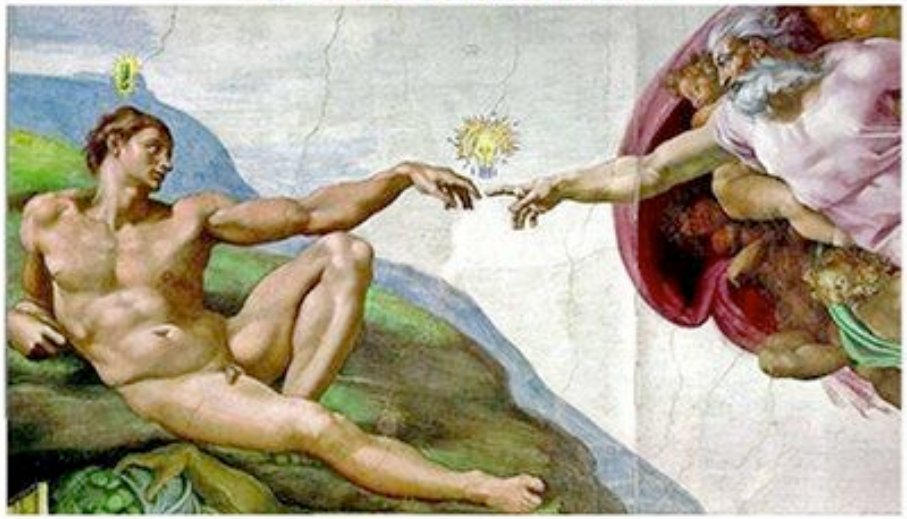
Princípios do Lean startup

- Aprendizado validado e contínuo
- Desenvolvimento do cliente (CustDev)
- Desenvolvimento ágil
- Tecnologia como *commodity*

Desenvolvimento do cliente

The Four Steps to the Epiphany

Successful Strategies for Products that Win



Steven Gary Blank

Desenvolvimento do cliente

- Um caminho para encontrar o encaixe produto-mercado
- Um dos pilares de Lean startups

#1

genchi gembutsu

vá e veja você mesmo

#2

- Tipos de mercado
  - Criando um novo mercado?
  - Trazendo um novo produto para mercado existente?
  - Re-segmentando um mercado existente?

#3

- Mínimo de features para conquistar o máximo de clientes CEDO

#4

- Fases do produto & crescimento da empresa

#5

- Aprendendo e iterando x Execução linear

MVP

That version of a new product that will allow a team to collect the maximum amount of validated learning about customers with the least effort.

by Eric Ries

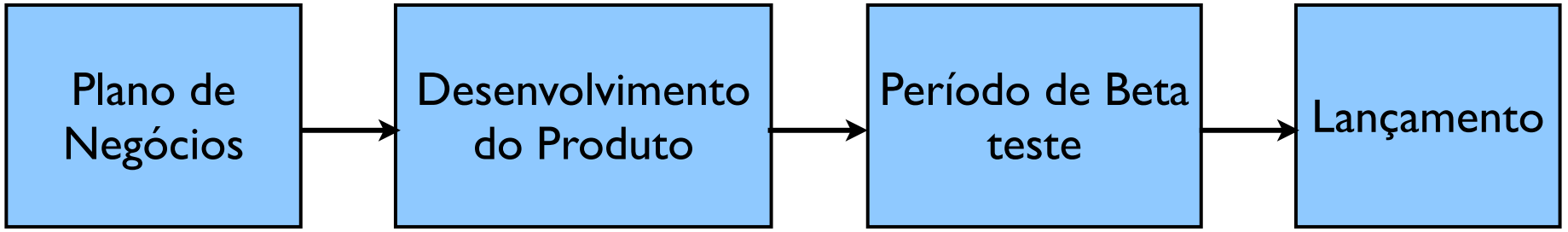
Uma versão do produto que permite uma equipe obter o maior aprendizado validado sobre seus clientes com o menor esforço

Princípios do Lean startup

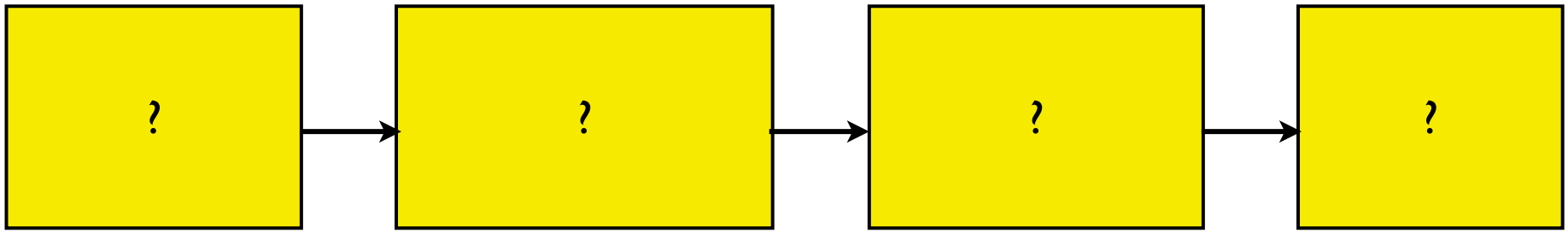
- Aprendizado validado e contínuo
- Desenvolvimento do cliente (CustDev)
- Desenvolvimento ágil
- Tecnologia como *commodity*

Lembra?

Desenvolvimento do produto

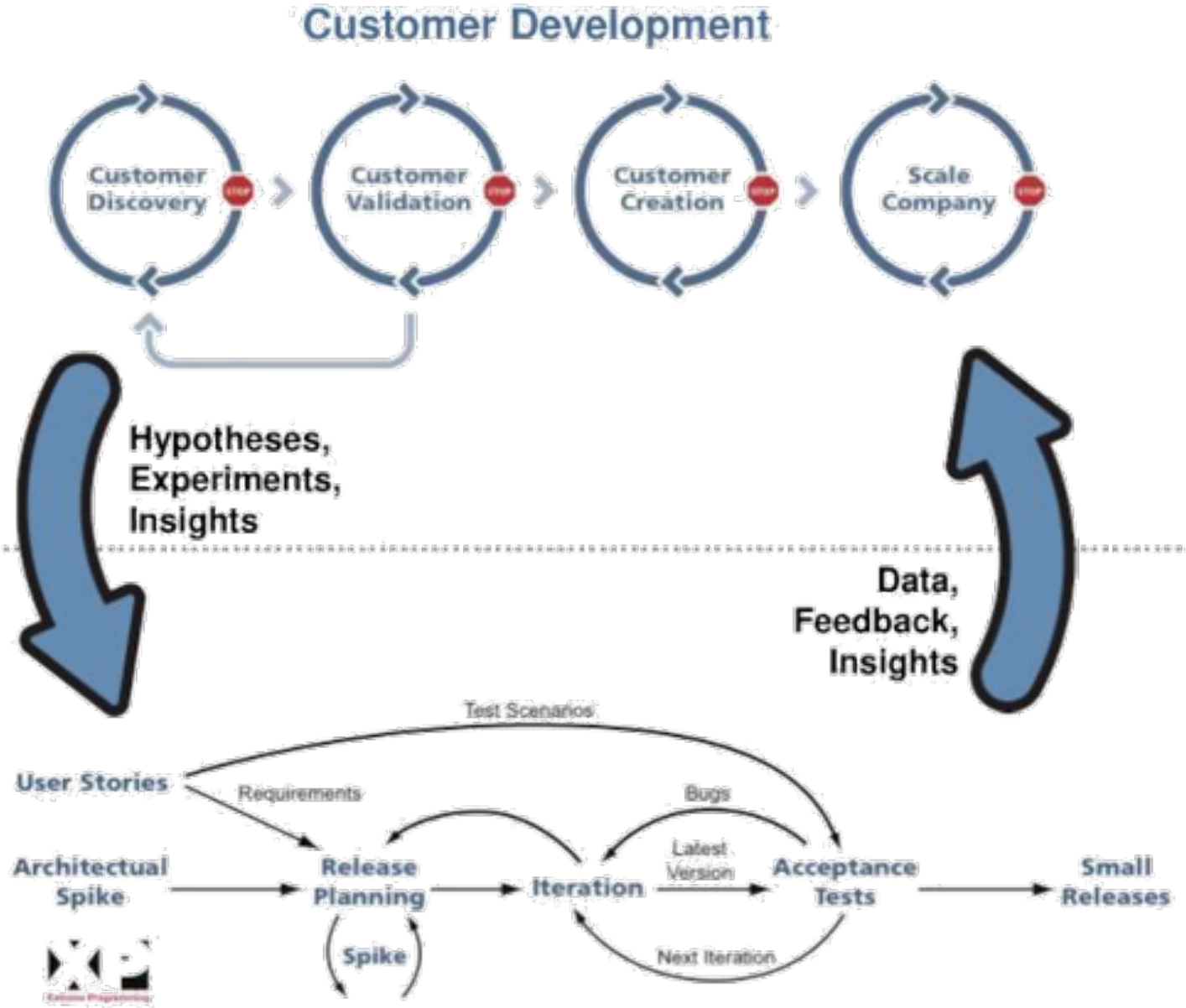


Desenvolvimento do **cliente**



Problem: unknown

Solution: unknown



E para finalizar

5 mitos de Lean startups

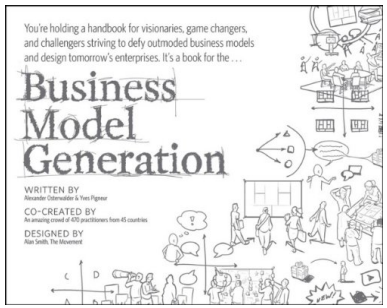
5 Mitos sobre Lean startups

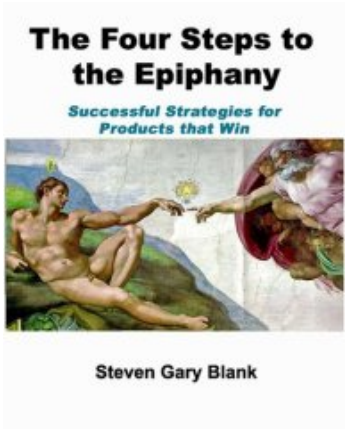
- Myth 1: Lean means cheap. Lean startups try to spend as little money as possible.
- Myth 2: The Lean Startup methodology is only for Web 2.0, Internet and consumer software companies
- Myth 3: Lean Startups are bootstrapped startups
- Myth 4: Lean Startups are very small companies
- Myth 5: Lean Startups replace vision with data or customer feedback

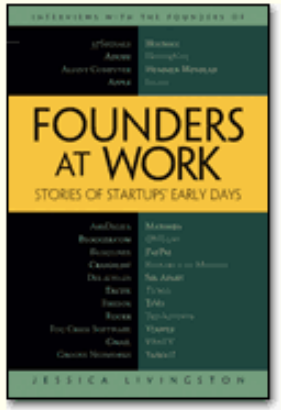
http://blogs.reuters.com/small-business/2010/04/27/top-5-myths-about-the-lean-startup/

Leituras

- Business Model Generation
- Four Steps to the Epiphany
- Founders at Work







Extras

Exercício

- Uma empresa simples:
  - 2 produtos: camisas de Mulher e de Homem
  - 2 recursos: um que corta o tecido e outro que costura

Corte	Costura
M: 2 min/peça	M: 15 min/peça
H: 10 min/peça	H: 10 min/peça

Exercício

Camisas	Mulheres	Homens
Demanda máxima semanal	120	120
Preço de venda	R$105	R$100
Custo Matéria Prima	R$45	R$50
Tempo de Corte(min)	2	10
Tempo de Costura(min	15	10
Tempo total(min)	17	20

- Cada máquina tem um operador e cada um trabalha 2400 minutos por semana. O salário semanal de cada operador é de R$ 200. Os operadores somente trabalham em sua estação
- As outras despesas semanais são de R$ 10.100
- Não há estoque em processo
- Qual o maior lucro que essa empresa consegue gerar por semana?

Exercício

- Corte
  - 120 * 10m(h) + 120*2m(m)= 1440m
- Costura
  - 120 * 10m(h) + 120 * 15m(m) = 3000m
- C. Homem
  - Margem = 100 - 50 = R$ 50
  - T(Costura) = 10m
- C. Mulher
  - Margem = 105 - 45= R$ 60
  - T(Costura) = 15m
- Precisamos levar em conta: valor agregado e uso da restrição
- Logo Margem / Tempo restr.

Exercício

- Logo: vamos maximizar a produção de C. Homem e fazer o que der de C.Mulher:
  - 120 C.Homem = 1200min (120 * 10), sobram 1200 min para:
  - 80 C.Mulher = 1200/15min/C.Mulher
  - Lucro/semana = R$ 300

O que aconteceu?

- Se concentrar em medidas locais (margem ou "lucro" de um produto) nos leva a enganos
- Conclusão:
  - Otimizar as partes não otimiza o todo
- Como enfrentar essa complexidade?