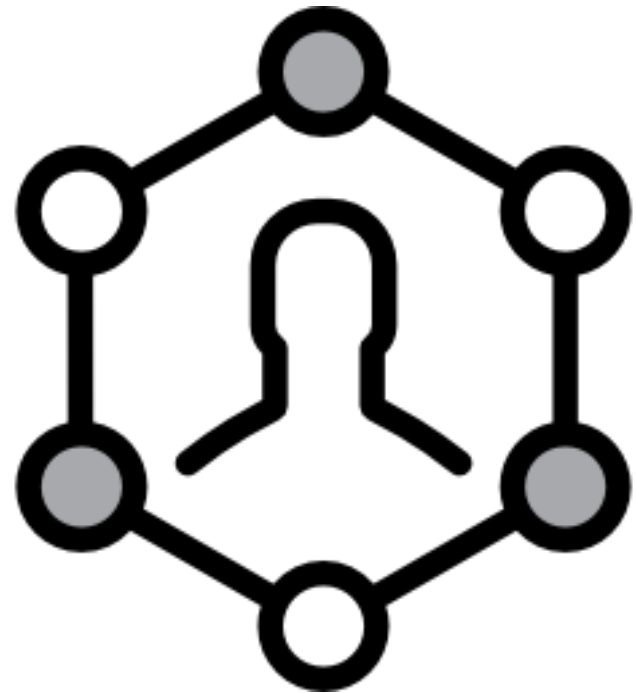


Curso TSW-003

# Interação Humano-Computador



**Profa. Lucia Filgueiras**

**Prof. Stefan Martins**

*2018*

# Após esta aula você aprenderá:

- Histórico de UCD
- ISO 13407
- Método de Mayhew
- Usabilidade e métodos ágeis
- Agile UX
- Lean
- Lean UX
- Tradicional, Lean e Agile?
- Melhor junto!

Processos centrados no  
usuário

**HISTÓRICO**

# USER CENTERED DESIGN

Engloba técnicas, processos e procedimentos para projeto de interfaces visando usabilidade, e principalmente a **filosofia** de se colocar o usuário no centro das atenções.

# Por quê?

- Criar um sistema adequado aos usuários exige tarefas determinadas, realizadas desde o **início do ciclo de vida**.
- A literatura sobre UCD é antiga, mas só recentemente começou a ser praticada na indústria.
- O investimento em atividades de usabilidade desde o início do ciclo melhora sensivelmente a **qualidade do produto final**.
- O aprendizado se dá a partir de exemplos concretos (Piaget).

# Projeto centrado em comportamento (Gould & Lewis, 1985)

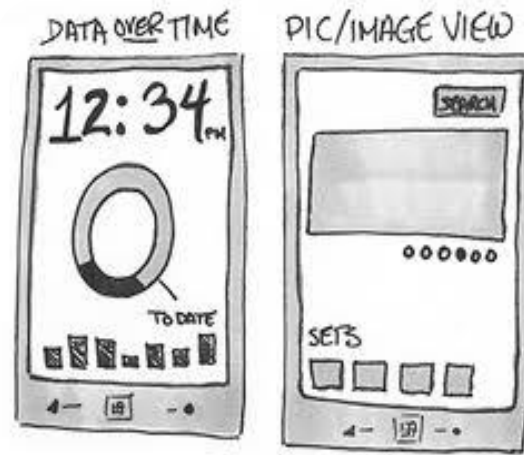
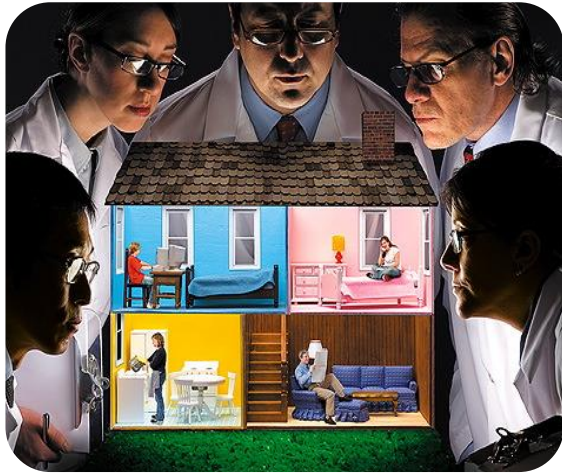
Sistema de correio de voz desenvolvido pela IBM para atender a atletas e suas famílias.

O projeto foi desenvolvido com o objetivo secundário de testar a metodologia focada no comportamento do usuário.

O resultado foi positivo e permitiu cumprir com qualidade as metas de prazo do projeto.

Gould&Lewis afirmam recomendaram esta abordagem desde os anos 70.





## Foco precoce nos usuários e tarefas

- Entender os usuários e suas tarefas, estudando suas características antropométricas, cognitivas, comportamentais e de atitude.
- Entender a natureza da atividade que se quer realizar.

## Medições empíricas

- Simulações e protótipos devem permitir a observação, registro e análise do comportamento do usuário e de suas reações.

## Projeto iterativo

- O processo de testar levanta problemas que precisam ser incorporados ao projeto; o que demanda novos testes.
- Ondas alternantes de análise e síntese.



# Foco precoce no usuário

- Os desenvolvedores devem **entender** os usuários-alvo o mais cedo possível: suas tarefas e necessidades.
- Os desenvolvedores devem ser colocados em **contato direto** com o usuário - vários aspectos reais do negócio são omitidos da versão “oficial”
  - Entrevistas
  - Discussões com usuários potenciais
  - Observações do ambiente real do usuário
- Usuários devem participar das atividades de projeto (projeto participativo)

*Involving the user*



By Frits Ahlefeldt

# Medidas empíricas

- Medidas reais de usabilidade em experimentos desde cedo no ciclo de vida
  - Testes de software não são obrigatoriamente experimentos de usabilidade.
  - **Testes baseados em cenários de uso com tarefas reais, permitindo análise do desempenho, pensamentos e atitudes.**
- Construção de protótipos com foco na avaliação do uso
  - No início do ciclo, protótipos de baixa fidelidade – como protótipos em papel ou “*mock-ups*” e *storyboards*.
  - À medida que o ciclo avança, protótipos mais refinados podem ser feitos.
- Sujeitos para o teste devem ser **usuários reais!**



Developer watching videotape of usability test.

# Processo iterativo

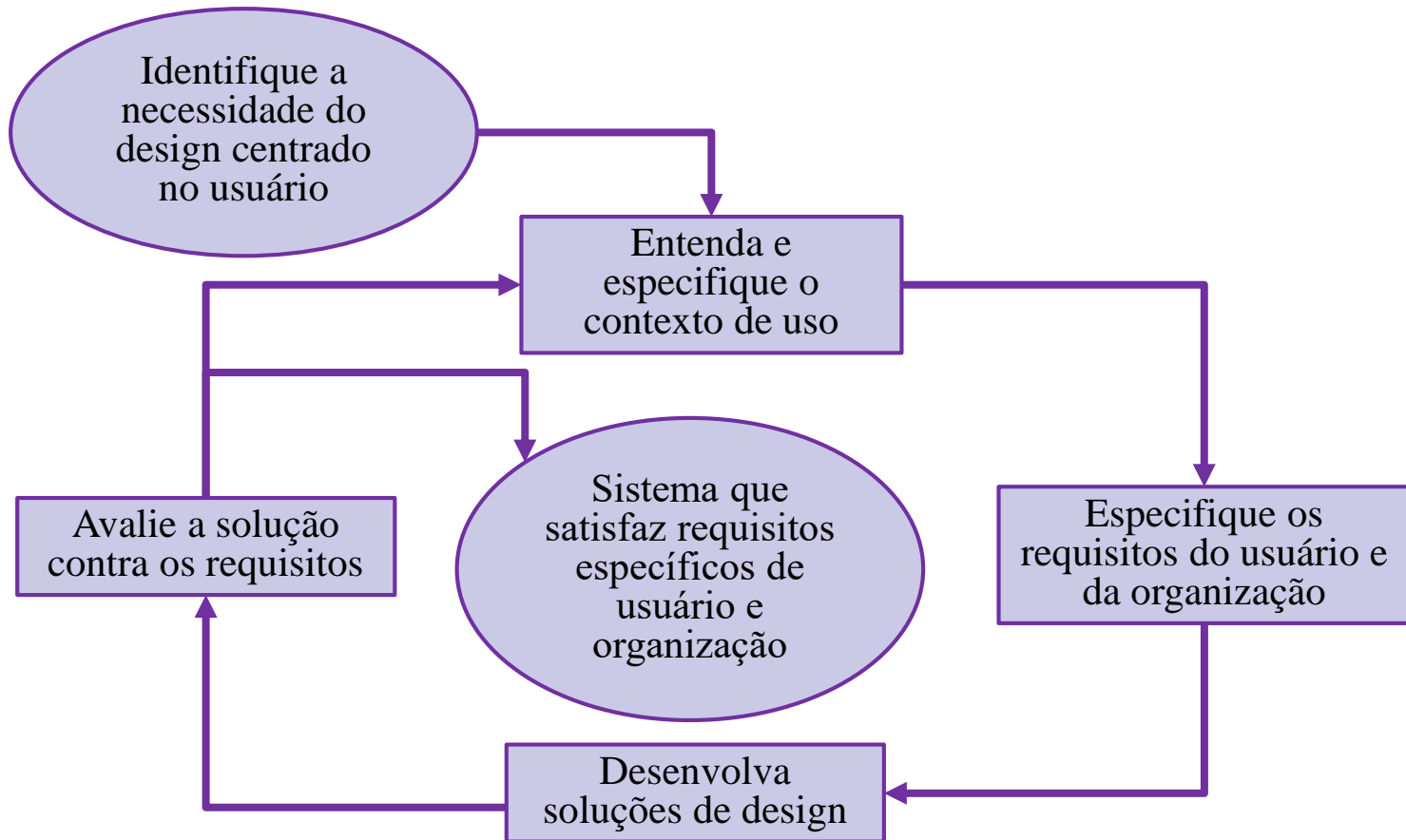
- Metas de usabilidade precisam ser estabelecidas para o projeto – são requisitos.
- São necessários ciclos de projeto com refinamento das soluções.
- Análises de custo-benefício precisam ser feitas para identificar as melhores alterações à interface, na busca das metas de usabilidade postuladas.



**ISO 13407**

**USER CENTERED DESIGN**

# ISO 13407 - Human-centred design processes for interactive systems



**A interdependência de atividades do design centrado no usuário**

# Princípios de HCD segundo ISO13407

Envolvimento ativo dos usuários e entendimento claro dos requisitos do usuário e de suas tarefas

Distribuição adequada de funções entre as pessoas e a tecnologia

Iteração na produção de soluções de projeto

Equipe multidisciplinar



# Papéis no time multidisciplinar

- Usuário final
- Comprador ou gerente do usuário final
- Especialista no domínio da aplicação, analista de negócio
- Analista de sistemas, engenheiro de sistemas, programador
- Profissional de marketing e de vendas
- Designer de interação
- Arquiteto de informação
- Analista de usabilidade
- *Designer* gráfico
- Redator técnico
- Instrutor
- Pessoal de suporte

# Plano do processo HCD

- Quais são as atividades do processo HCD adotadas na empresa?
- Quais os procedimentos para integrar estas atividades com outras atividades do desenvolvimento do sistema?
- Quem são os interessados (*stakeholders*) – indivíduos e organizações, suas capacidades e pontos de vista?
- Como obter retorno dos stakeholders?
- Quais os marcos de projeto para acompanhamento das atividades do processo HCD?

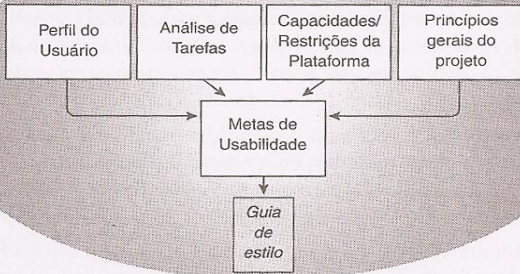
# **MÉTODO DE MAYHEW**

# Ciclo de vida da Engenharia de Usabilidade (Mayhew, 1999)

- Está relacionada com a metodologia de Jacobson (OOSE - Object Oriented Software Engineering)
- Fases:
  - Análise de requisitos
  - Desenvolvimento
  - Instalação

**ANÁLISE DE REQUISITOS**

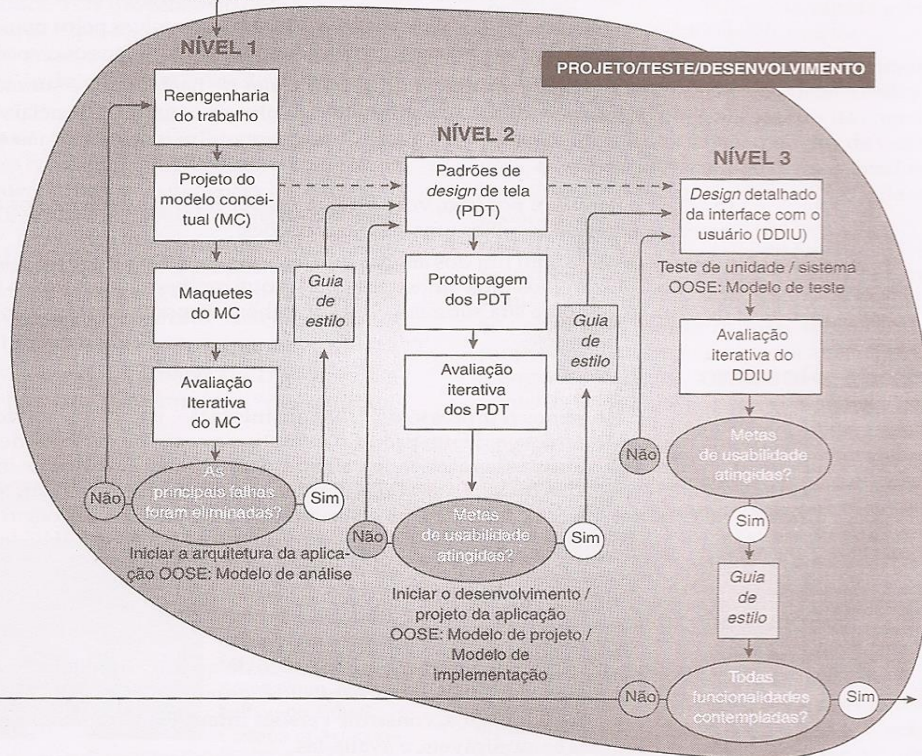
Funções/ Modelagem de dados  
OOSE: Modelo de requisitos



**O CICLO DE VIDA DA ENGENHARIA DE USABILIDADE**

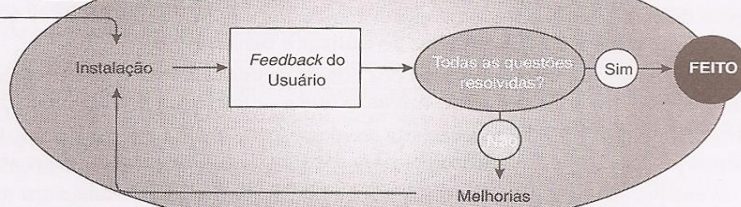
**NÍVEL 1**

**PROJETO/TESTE/DESENVOLVIMENTO**



**INSTALAÇÃO**

A



- Tarefa da Engenharia de Usabilidade
- ⊥ Tarefa de Desenvolvimento
- Ponto de decisão
- ▭ Documentação
- ↓ Aplicações complexas
- > Aplicações simples (ex.: websites)

# **UX E MÉTODOS ÁGEIS**

# O que é?



www.dilbert.com scottadams@aol.com



11-31-07 © 2007 Scott Adams, Inc./Dist. by UFS, Inc.



© Scott Adams, Inc./Dist. by UFS, Inc.

TRADUÇÃO LIVRE: TIRINHAS.COM

# Vamos começar novamente...

**Ágil** é uma nova forma de gestão e desenvolvimento de Software que usa uma abordagem de planejamento e execução iterativa e incremental voltado para processos empíricos (complexos, caóticos ou com muita incerteza, tem mudança ao longo do processo, não são repetitivos e são imprevisíveis) que divide o problema em produtos menores.

Seu principal objetivo é entregar o produto que o cliente realmente deseja e que será útil e com qualidade.



# Foco

- Nos métodos tradicionais, o foco é no processo, que deve assegurar a qualidade.
- Nos métodos ágeis, o foco é nas pessoas, que garantem o sucesso do projeto.
- Nos métodos tradicionais, o desenvolvimento é orientado por um planejamento – portanto, o processo é prescritivo – enquanto no método ágil, o processo é adaptativo.

# Adoção das metodologias

Apesar das metodologias existirem desde os anos 80, foi em 2001 que um grupo formado por 17 renomados desenvolvedores assinaram o **Manifesto para o Desenvolvimento Ágil de Software** e o grupo foi batizado de **Aliança dos Ágeis**.

"Estamos descobrindo 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."

Assinado por: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas © 2001.

Esta declaração pode ser copiada livremente em qualquer formato, mas somente integralmente através desta declaração.

[www.agilemanifesto.org](http://www.agilemanifesto.org)

# 12 princípios do agile (parte 1)

1. Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.
2. Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.
3. Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
4. Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.
5. Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.
6. O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face.

# 12 princípios do agile (parte 2)

7. Software funcionando é a medida primária de progresso.
8. Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
9. Contínua atenção à excelência técnica e bom design aumenta a agilidade.
10. Simplicidade--a arte de maximizar a quantidade de trabalho não realizado--é essencial.
11. As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis.
12. Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

# Quais são as metodologias ágeis?

“Desenvolvimento ágil” é o termo utilizado por diferentes metodologias e frameworks que desenvolvem software de forma iterativa e incremental.

Algumas são mais prescritivas ou menos mas as metodologias ágeis mais comuns são: Extreme Programming (XP), Scrum, Lean Development, Feature-Driven Development (FDD), Kanban, RUP e OpenUP.

# **EXTREME PROGRAMMING-XP**

# Valores do XP (Extreme Programming)

**Comunicação** para partilhar a visão do sistema entre os desenvolvedores e clientes.

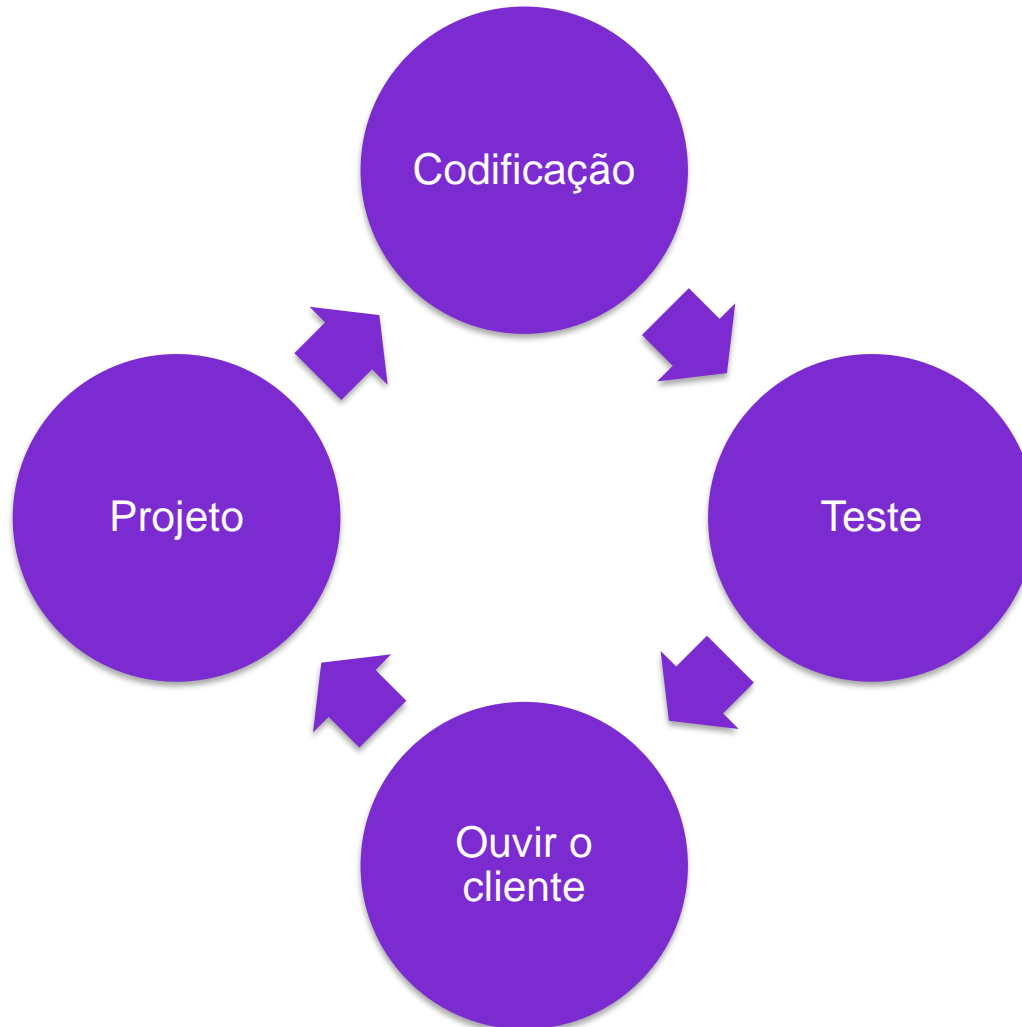
**Simplicidade** de começar com a versão mais simples possível e refabricar quando necessário.

**Feedback** como aprendizado por meio do sistema, do cliente e do próprio time.

**Coragem** para refazer, para codificar hoje, para expor o código a todos os membros da equipe, para integrar o sistema várias vezes por dia.

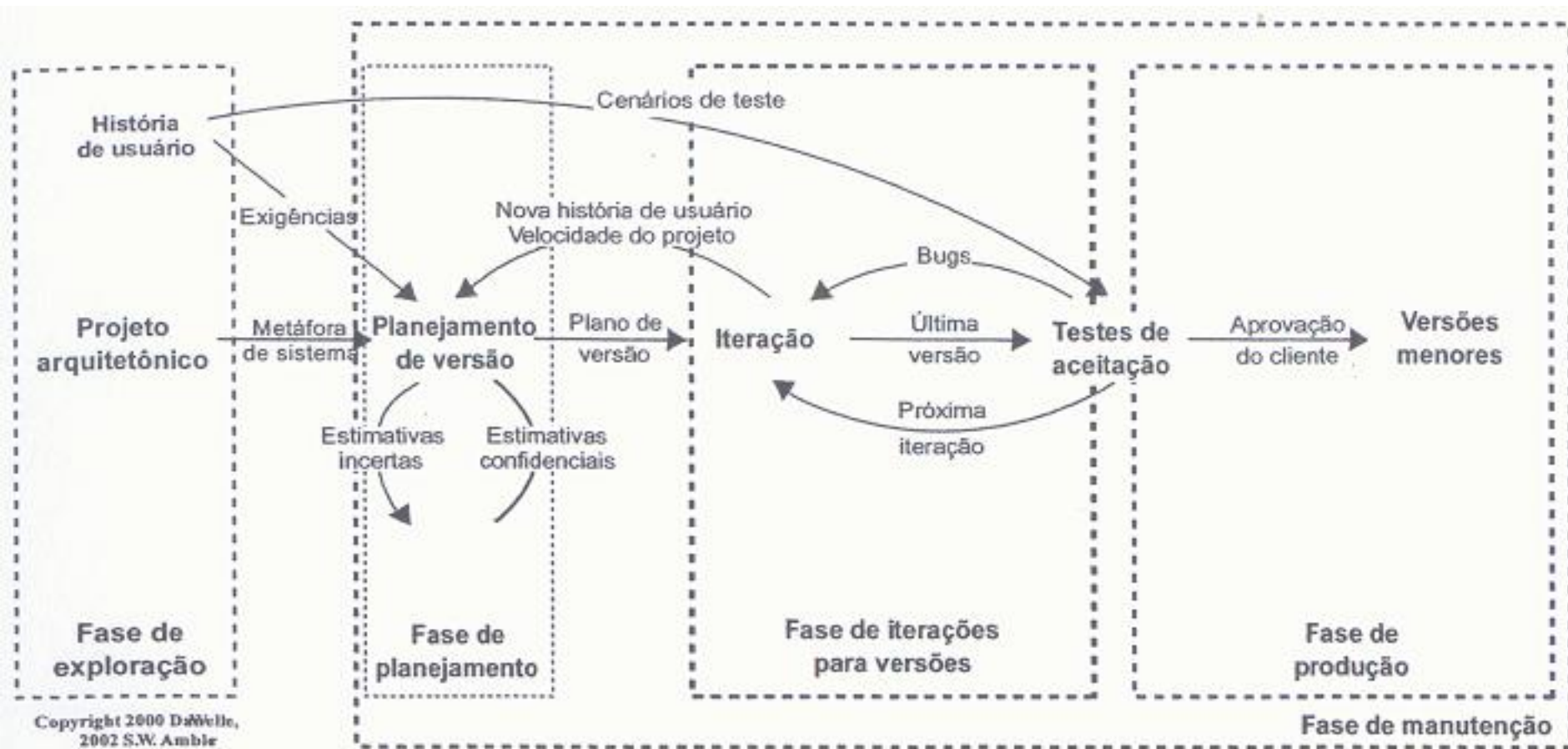


# Atividades do XP



# Práticas do XP

1. Jogo do planejamento
2. Entregas pequenas
3. Metáfora do sistema
4. Design simples
5. Desenvolvimento orientado a testes
6. Refactoring
7. Programação em pares
8. Posse coletiva do software
9. Integração contínua
10. Ritmo sustentável
11. Cliente no local
12. Padrões de codificação



**SCRUM**

# SCRUM

No Scrum, os projetos são divididos em ciclos chamados de **Sprints**. O Sprint representa um Time Box dentro do qual um conjunto de atividades deve ser executado.

As funcionalidades a serem implementadas em um projeto são mantidas em uma lista que é conhecida como **Product Backlog**.

No início de cada Sprint, faz-se um **Sprint Planning Meeting**, ou seja, uma reunião de planejamento na qual o **Product Owner** prioriza os itens do **Product Backlog** e a equipe seleciona as atividades que ela será capaz de implementar durante o Sprint que se inicia.

As tarefas alocadas em um Sprint são transferidas do Product Backlog para o **Sprint Backlog**.

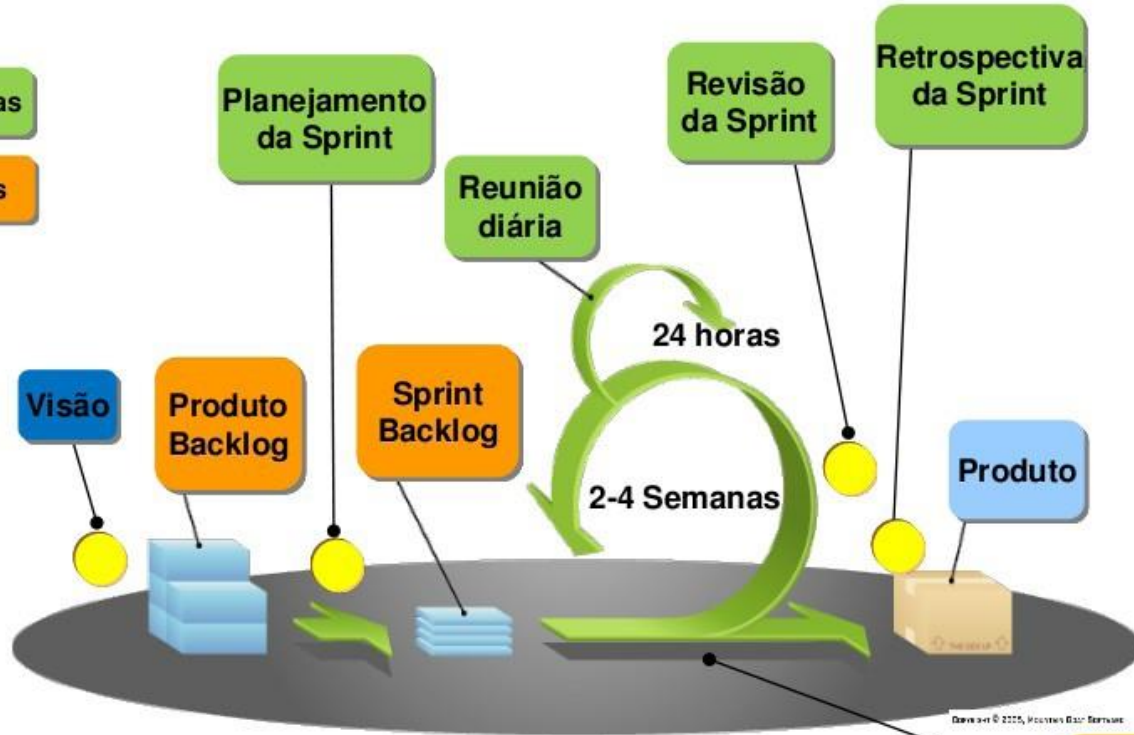
A cada dia de uma Sprint, a equipe faz uma breve reunião, chamada **Daily Scrum**. O objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia.

Ao final de um Sprint, a equipe apresenta as funcionalidades implementadas em uma **Sprint Review Meeting**. Finalmente, faz-se uma **Sprint Retrospective** e a equipe parte para o planejamento do próximo Sprint. Assim reinicia-se o ciclo.

<http://www.desenvolvimentoagil.com.br/scrum/>

# Como funciona o Scrum?

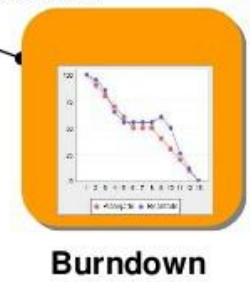
Legenda:



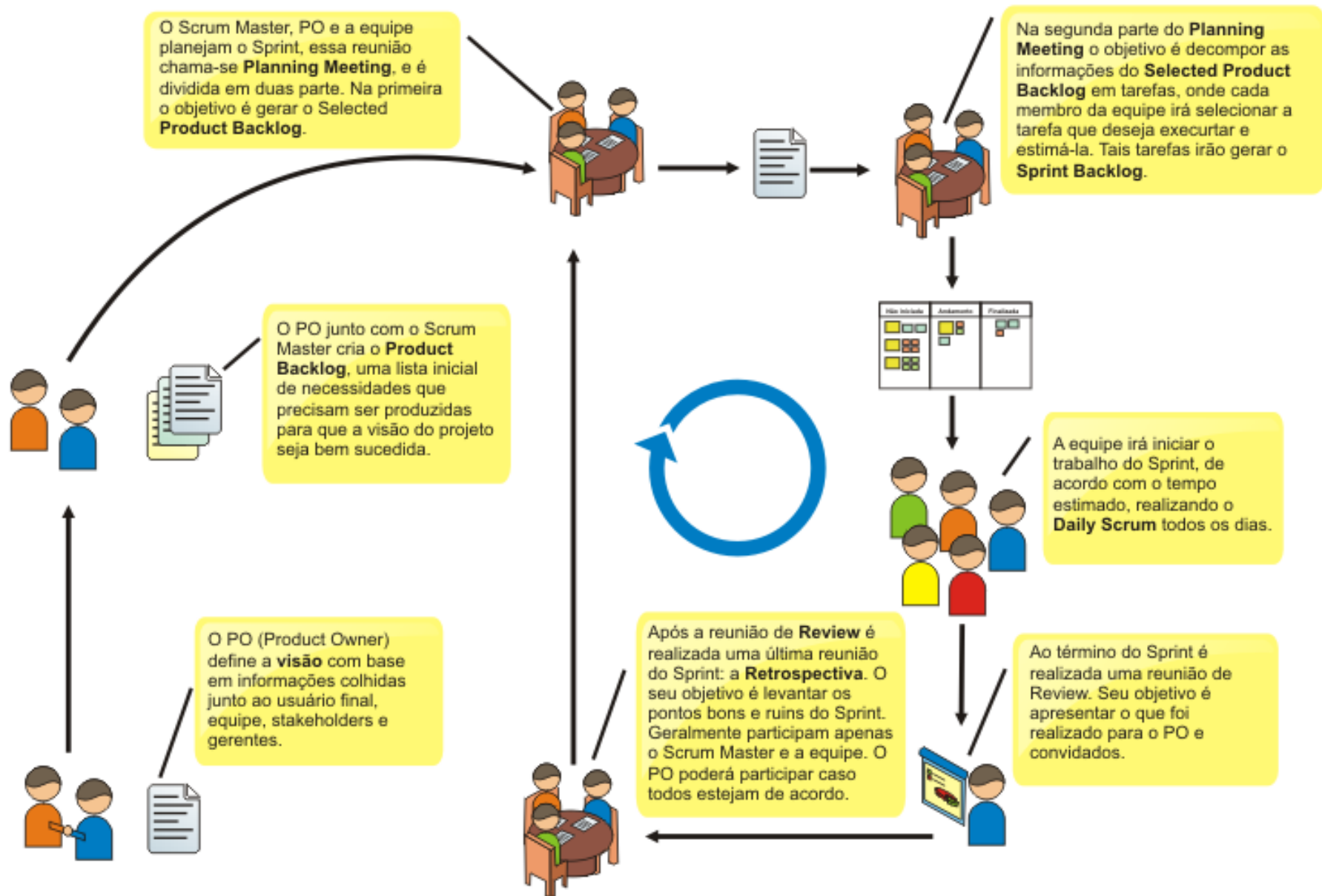
- Papéis**
- Product Owner (PO)
  - ScrumMaster (SM)
  - Equipe Scrum

- Cerimônias**
- Planejamento da Sprint
  - Reunião Diária
  - Revisão da Sprint
  - Retrospectiva da Sprint

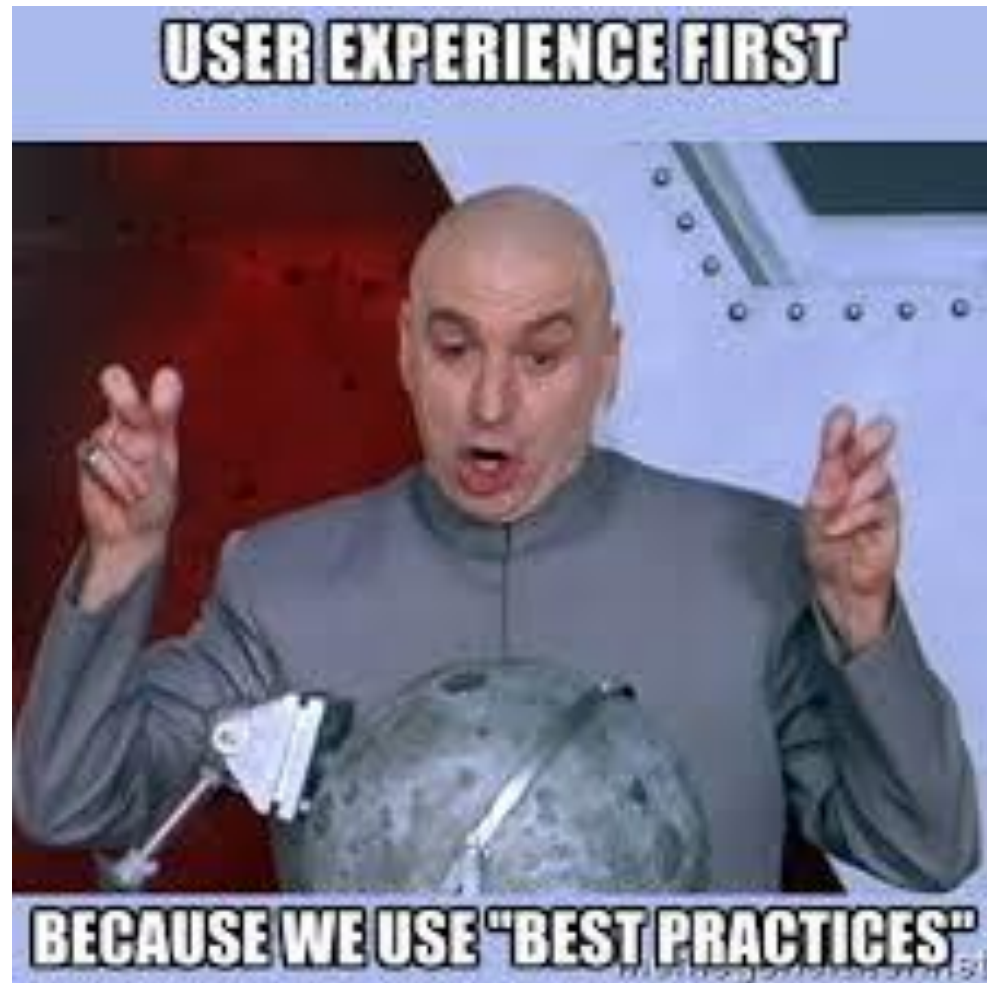
- Artefatos**
- Product Backlog
  - Sprint Backlog
  - Burndown (gráfico)



# Ciclo de trabalho do Scrum

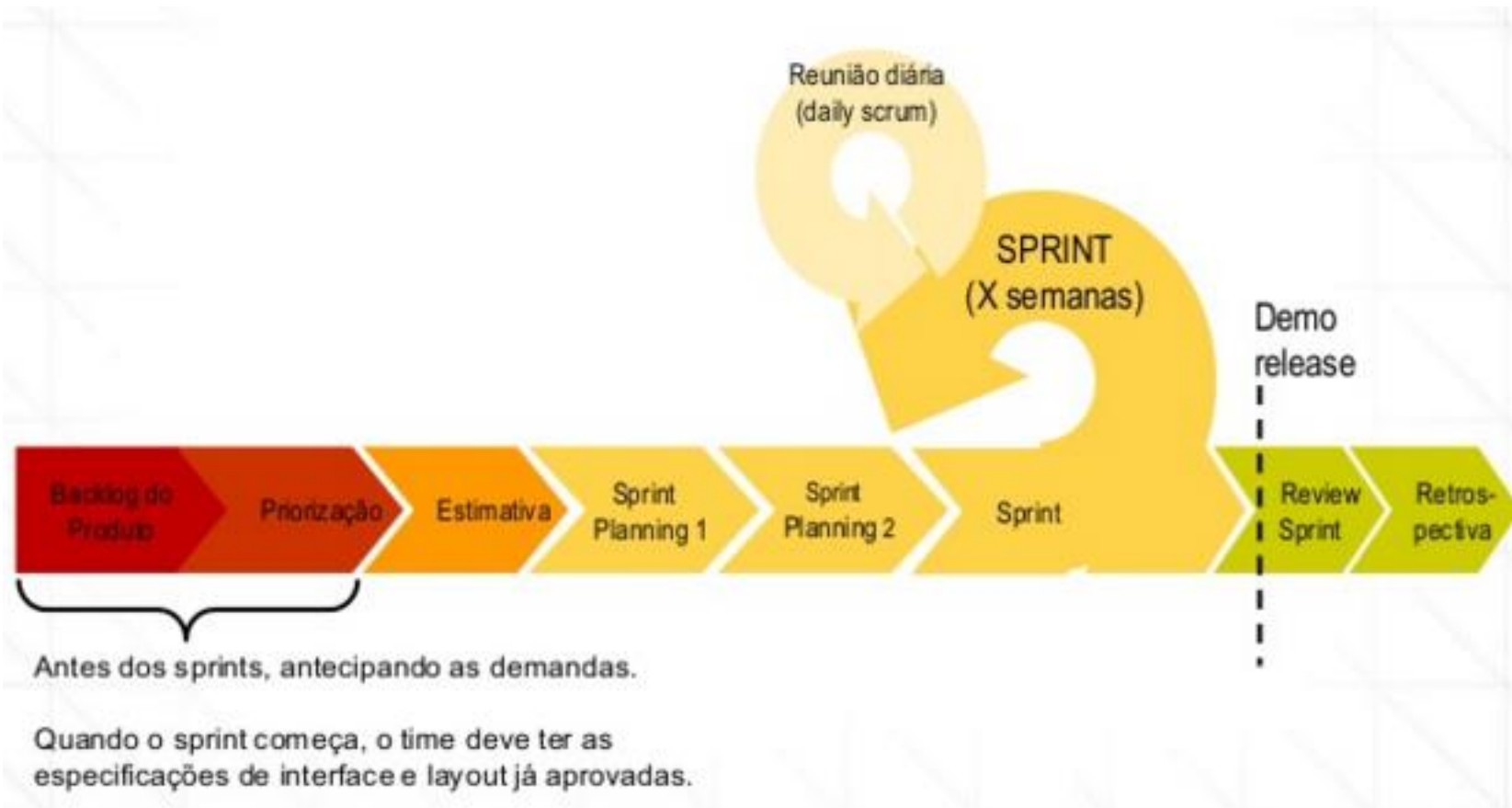


Scrum e UX = Agile UX

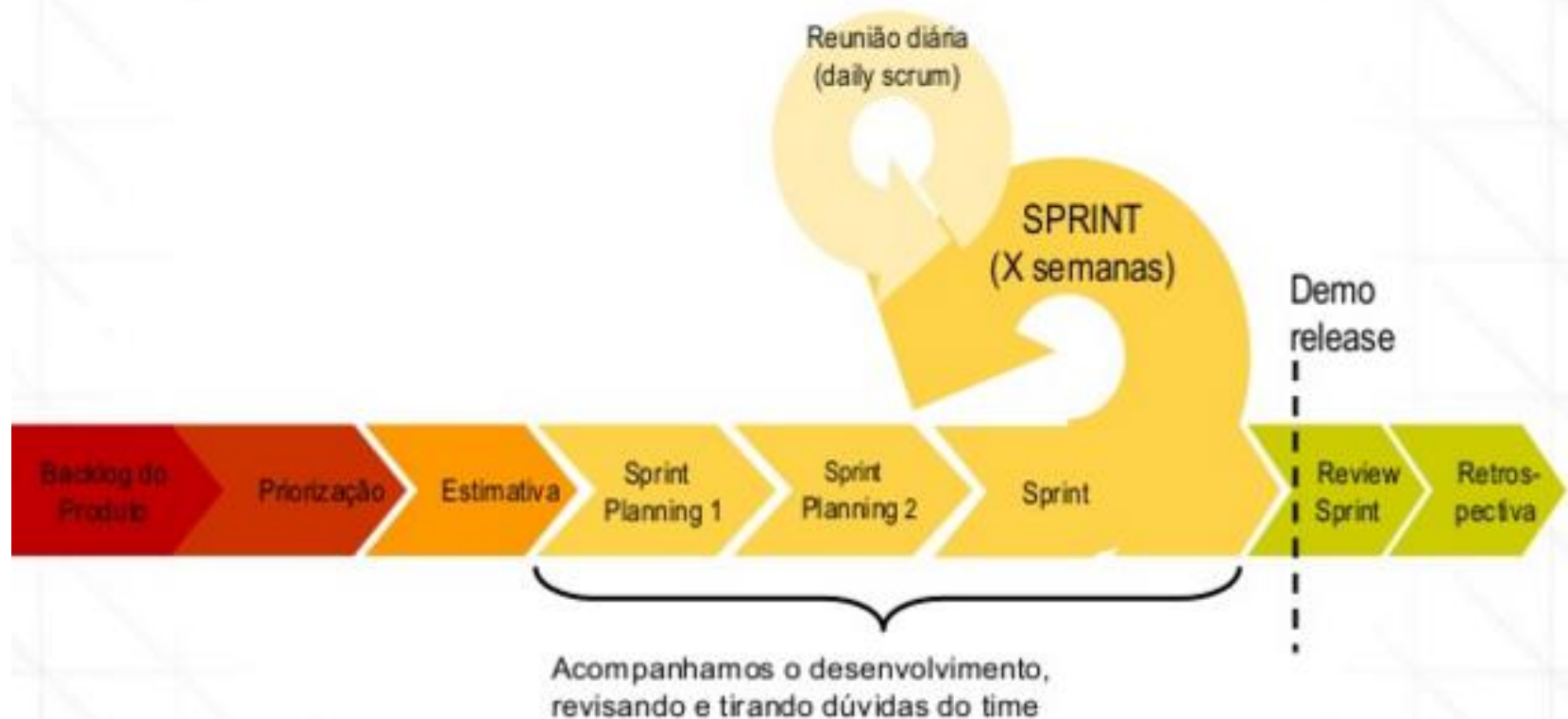




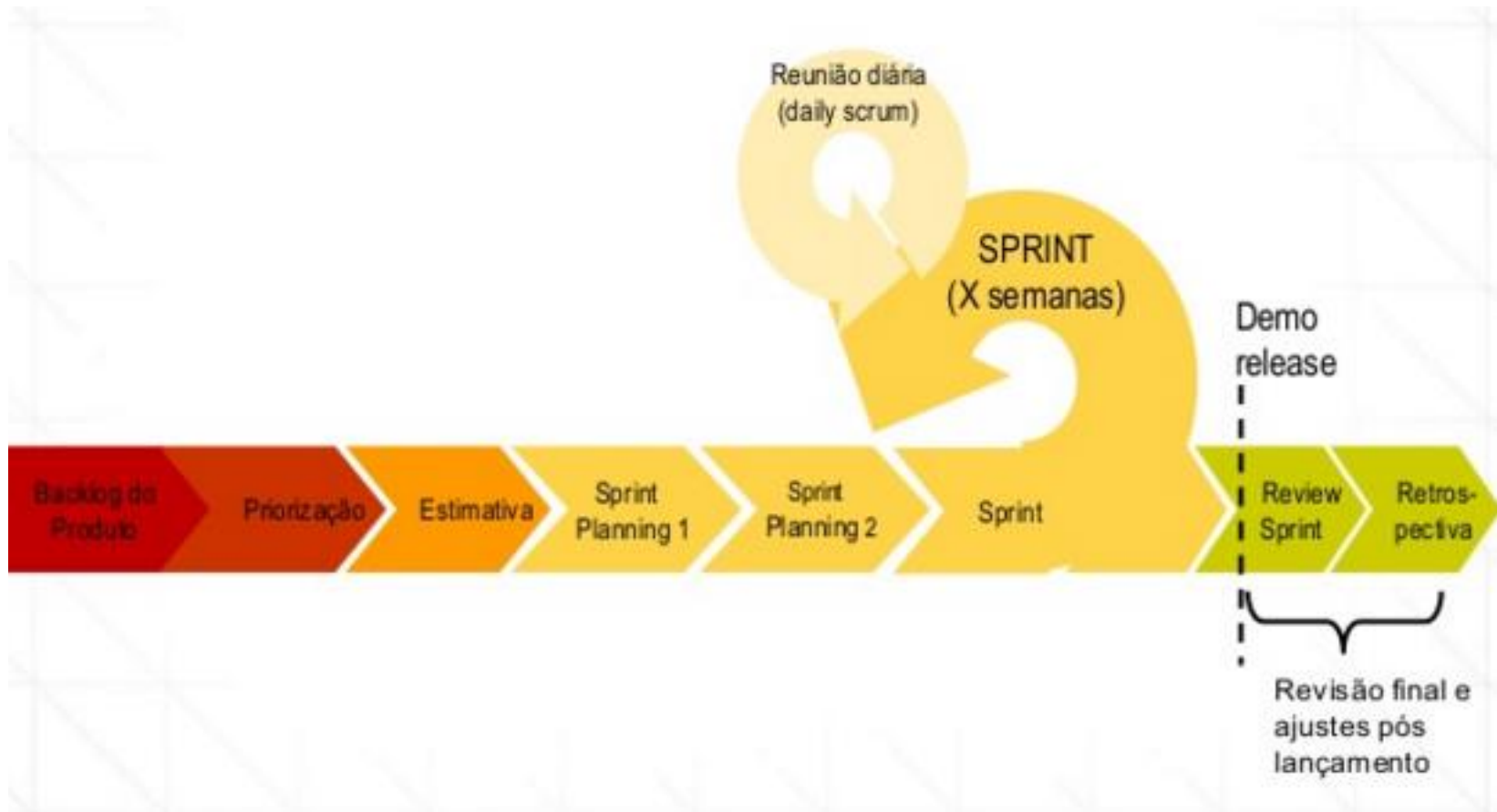
# UX e Scrum no UOL



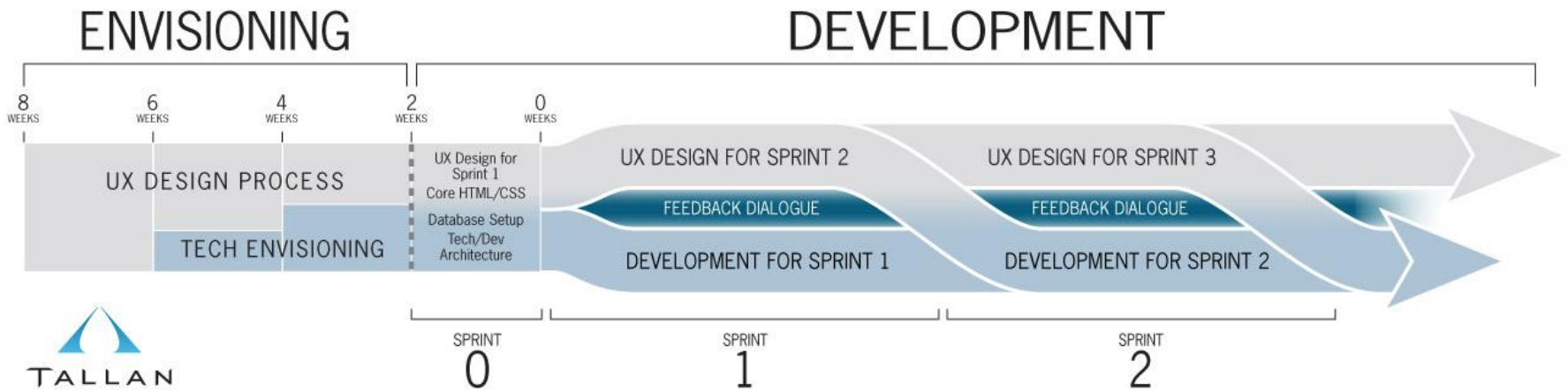
# UX e Scrum no UOL



# UX e Scrum no UOL



# Agile UX na TALLAN



<https://blog.tallan.com/2010/09/01/agile-user-experience-design/>

# Resumindo, na Agile UX

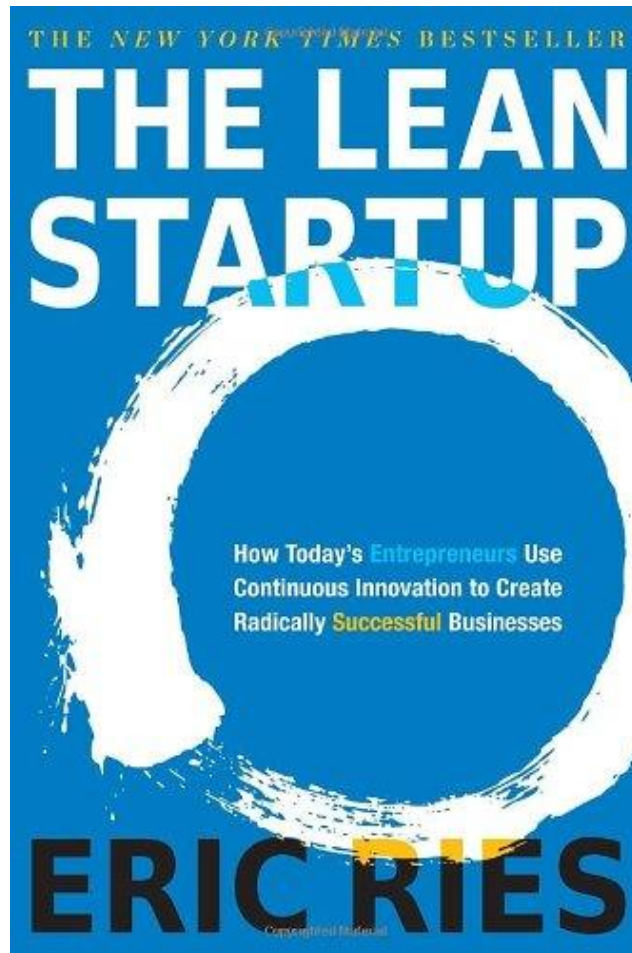
- Interações entre os membros do time são mais importantes do que os processos e ferramentas;
- Construir o software é mais importante do que produzir documentação super extensa;
- Envolver o consumidor no processo é extremamente importante para colher feedback o quanto antes;
- Saber adaptar a mudança ao invés de seguir um plano rígido.

**LEAN**



**StartUps**

# Por onde começou?



Lean = Enxuto

- Reduza o desperdício;
- Fracasse depressa;
- Aprenda rápido!

MVP

Desenvolvimento contínuo

Métricas



# MVP (minimum viable product)

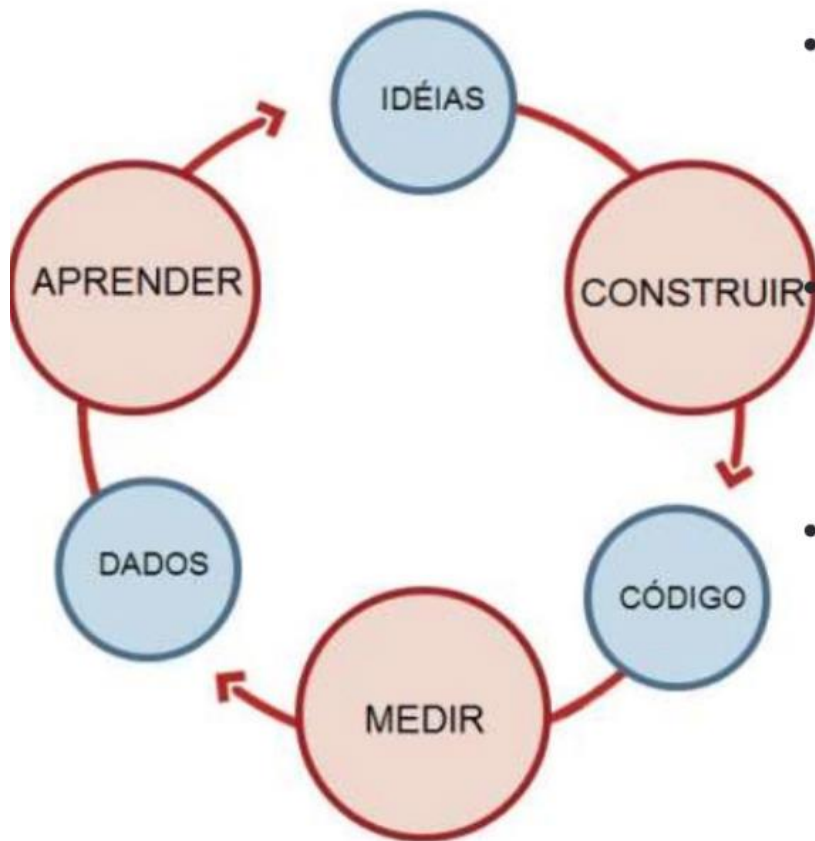
How **not to build** a minimum viable product



How **to build** a minimum viable product



# Ciclo do Lean Startup



- Construir rapidamente **experimentos** de validação das suposições do negócio

**Diminuir o tempo** de construção e geração de códigos fontes

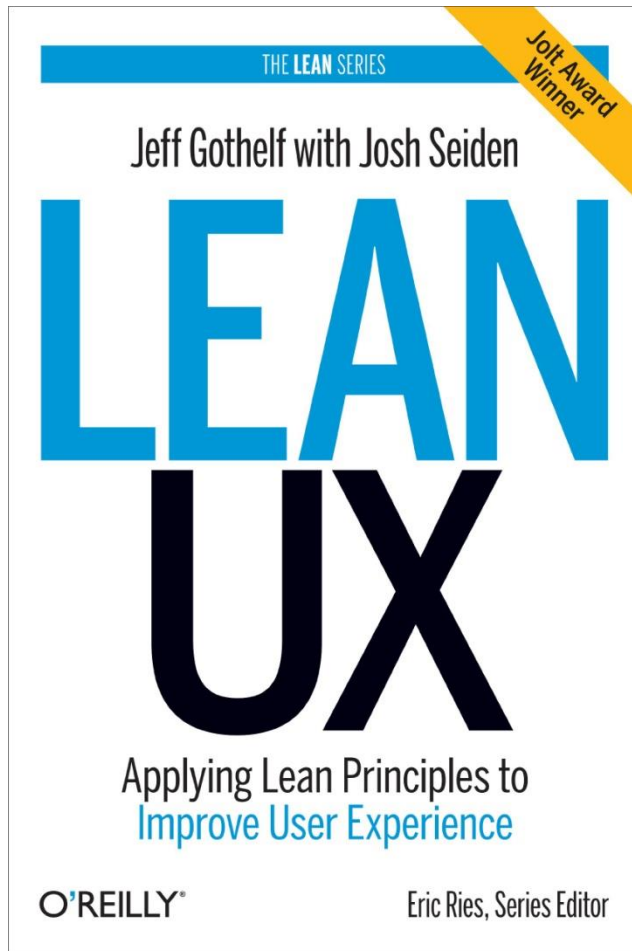
- Conseguir uma **aprendizagem validada** para tomada de decisão

E UX com isso? ˘\\_(ツ)\_/˘



**LEAN UX**

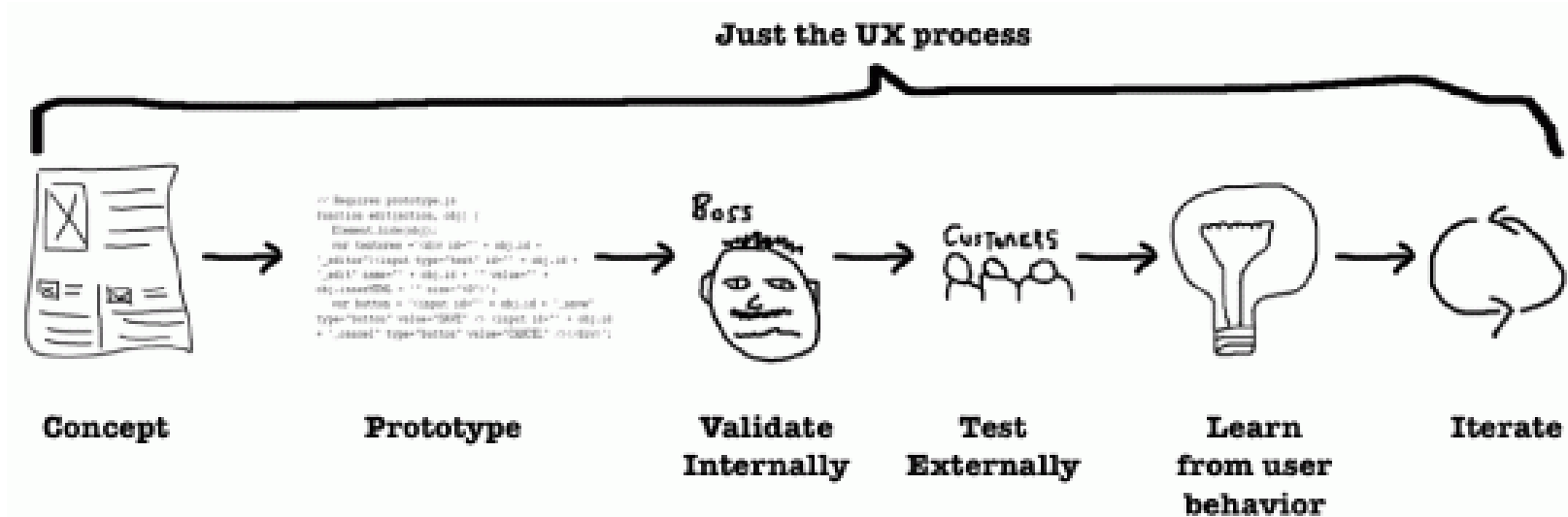
# LEAN + UX = LEAN UX



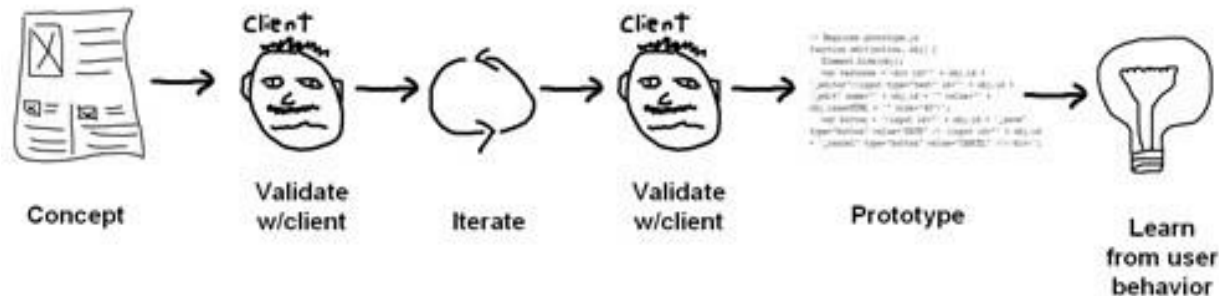
## Foco nos produtos e não nos entregáveis

- Menos documentos da área de UX
- Profissional de UX mais livre e focado nas necessidades do usuário

# Como funciona?



**Independente dos entregáveis, o importante é validar o mais rápido possível (internamente e com usuários)**



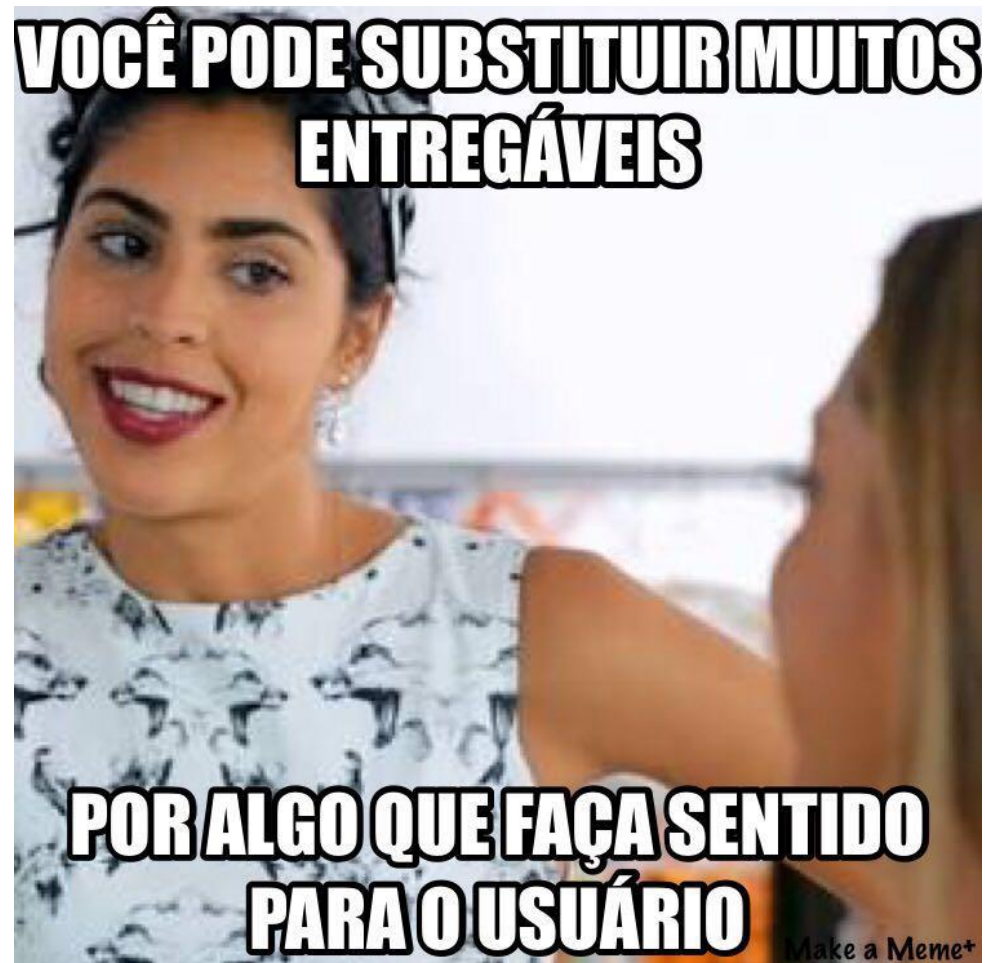
**AWWW**



**QUE LINDO!!!**

www.giphy.com

E na prática?





E o mais importante



## THE DESIGN PROCESS

IDEAL

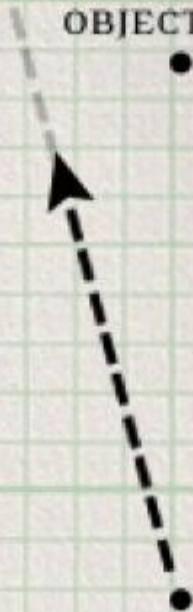
OBJECTIVES



BLANK PAGE

REALISTIC

OBJECTIVES



BLANK PAGE

SOLUTION

OBJECTIVES



BLANK PAGE

FEEDBACK

# Resumindo, na Lean UX:

- Construir os MVPs (Minimum Viable Products) para resolver os problemas do usuário;
- Prototipar rapidamente e colaborativamente;
- Validar hipóteses com os usuários o quanto antes;
- Usar métricas e outros insights para melhorar as ideias e refiná-las com o passar do tempo.

**LEAN UX E AGILE UX**

# Diferenças

Agile UX está mais preocupada no “**como**” o produto será desenhado, enquanto Lean UX foca muito mais no “**porquê**”.

Enquanto Agile é uma metodologia de projeto, Lean é uma forma de pensar o produto como um todo.

# THE DIFFERENCE BETWEEN Lean UX and Agile UX

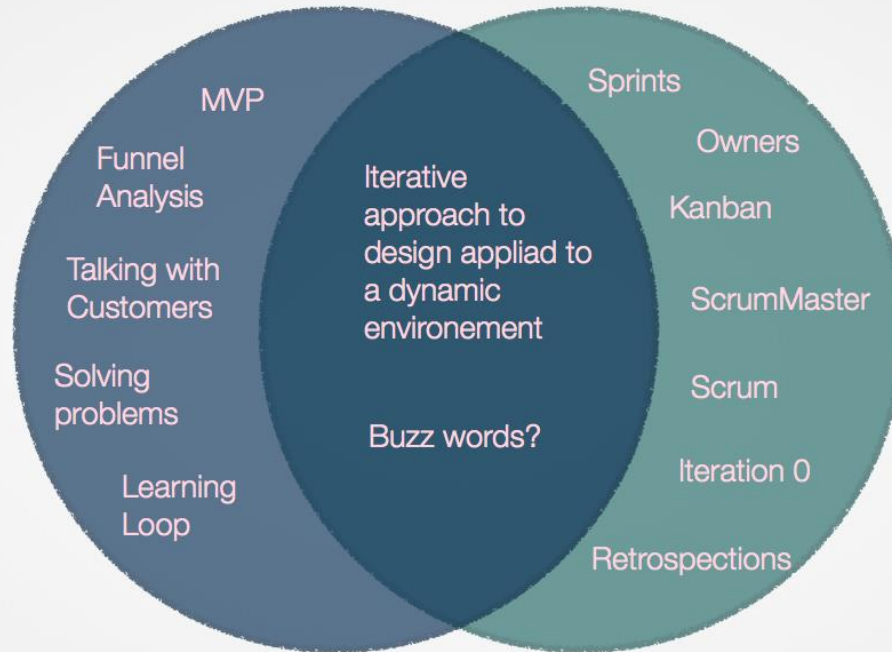
## Lean UX

Application of User Experience Design methods into product development, tailored to fit Build-Measure-Learn cycles.

Design united with business and development.

### Methods

Inspired by startup development and The Lean Startup Methodology.



## Agile UX

An attempt to integrate User Experience Design and Agile Software Development Methodology

UX Design team & Dev team working together.

### Methodology

Inspired by Agile Manifesto. Forced to blossom by growing popularity of Agile Software Development Methodology.

Try Lean UX & Agile UX in **UXPin - The UX Design App** (<http://uxpin.com>)

# Agile Sprints

*Velocity  
Points  
Iterations  
Continuous Deployment*



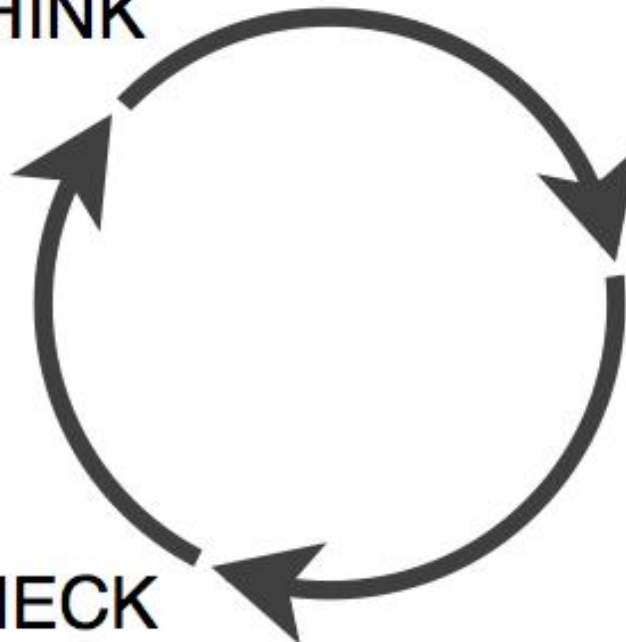
# Lean Cycles

*Generative Research  
Ideation  
Mental models  
Behavior Models  
Test Results  
Competitive Analysis*

**THINK**

*A/B Testing  
Site Analytics  
Usability Testing  
Funnel  
Sign-ups*

**CHECK**



**MAKE**

*Prototypes  
Wireframes  
Value Prop  
Landing Page  
Hypotheses  
Comps  
Deployed Code*



**TRADICIONAL, LEAN E AGILE**



## Traditional UX

Design, Usability

*O que estamos fazendo?*

Startup  
✓

## Lean UX

Measuring, validating  
product/market fit.

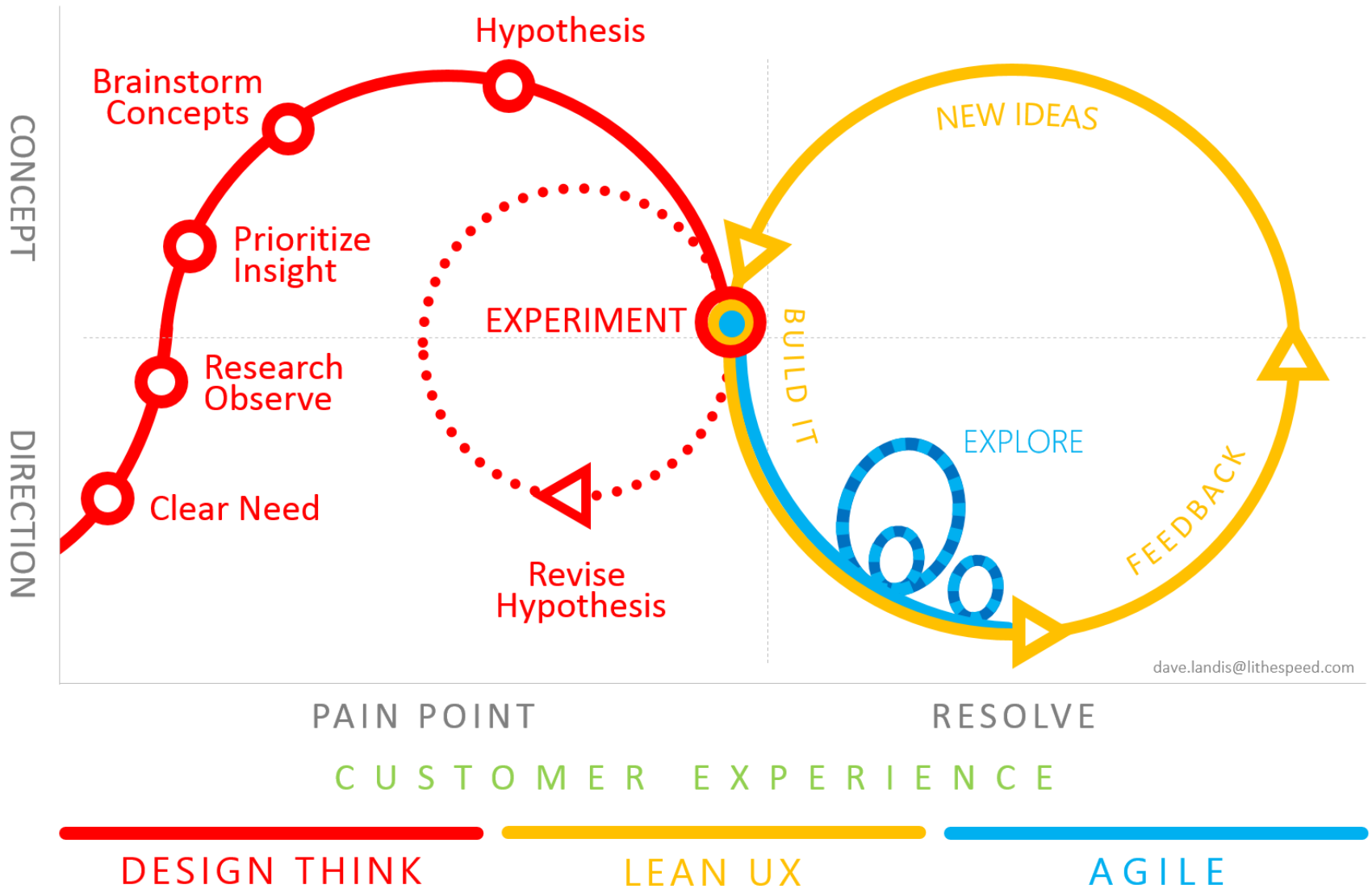
*Estamos fazendo  
certo?*

## Agile UX

Collaboration,  
Delivery

*Como fazemos  
isso?*

# BETTER TOGETHER



dave.landis@lithespeed.com

PAIN POINT

RESOLVE

CUSTOMER EXPERIENCE

DESIGN THINK

LEAN UX

AGILE

# Práticas de usabilidade com métodos ágeis

Referência para estudo:  
dissertação de mestrado de  
Ana Paula Oliveira dos Santos,  
orientada pelo Prof. Dr. Fabio  
Kon, IME-USP

Aplicação de práticas de  
usabilidade ágil em software livre

Ana Paula Oliveira dos Santos

DISSERTAÇÃO APRESENTADA  
AO  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
DA  
UNIVERSIDADE DE SÃO PAULO  
PARA  
OBTENÇÃO DO TÍTULO  
DE  
MESTRE EM CIÊNCIAS

Programa: Ciência da Computação

Orientador: Prof. Dr. Fabio Kon

Durante o desenvolvimento deste trabalho a autora recebeu auxílio financeiro do projeto Qualipso financiado pela Comissão Europeia.

São Paulo, fevereiro de 2012

# Identificar necessidades de UCD

## 1. Um Sprint na frente (*One sprint ahead*)

*A equipe de UX trabalha no mínimo uma iteração à frente da equipe de desenvolvimento.*

## 2. Sprint Zero

*Curto Sprint antes da implementação do código para definir visão do produto, metas gerais, planejar superficialmente os próximos sprints, definir princípios de design.*

## 3. Especialistas em UX como Donos do Produto (*UX experts like Product Owners*)

*Praticantes de UX fazem parte da equipe de Donos do Produto Product Owners, integrando requisitos de negócio com requisitos de usabilidade.*

# Identificar necessidades de UCD (cont.)

## 4. Tempo do usuário é valioso (*User's time is valuable*)

*Cada encontro com usuários típicos é utilizado para várias atividades como testes de usabilidade do design desenvolvido na iteração anterior e pesquisa de usuário para definição de requisitos para próxima iteração.*

## 5. Caminhos paralelos (*Parallel tracks*)

*Equipe de UX ou usabilidade trabalha em caminho paralelo sincronizado com o ciclo/sprint de desenvolvimento.*

## 6. 100% de compromisso UX (*UX 100 percent commitment*)

*Praticantes são membros em tempo integral de uma única equipe ágil e não atribuídos a múltiplos produtos. Também usa-se design pareado como programação pareada.*

# Especificar contexto de uso

## 1. Pouco design antecipado (*Little Design Up Front*)

*Realização de pesquisa com menor granularidade de detalhes, enfoque apenas sobre um conjunto de funcionalidades essencial para a próxima iteração, ou seja, quebra-se o trabalho de design em pedaços menores.*

## 2. Plano de contato de usuários (*User contact plan*)

*Criar um canal de comunicação adiantado com usuários finais (e responsáveis) para fornecer suprimento estável de feedback.*

# Especificar requisitos

## 1. Histórias de usuários (*User stories*)

*Inclusão de requisitos de usabilidade em histórias de usuários, definindo critérios de aceitação para cada história.*

## 2. Mais colaboração, menos documentos (*More collaboration, fewer documents*)

*Atualizações frequentes em seus designs e no processo de design; comunicação contínua durante implementação e teste.*

## 3. Tratar protótipo como especificação

*Utilizar protótipos para especificar requisitos de interface de usuário.*

# Criar soluções de design

## 1. Prototipação (*Prototyping*)

*Realizar prototipação de baixa fidelidade nas fases iniciais do sistema. O próprio é avaliado como protótipo de alta fidelidade que vai evoluindo.*

## 2. Estúdio de Design (*Design Studio*)

*Participantes geram diversos designs alternativos para uma interface durante um tempo pré-determinado e, ao terminar, abre-se para discussão e seleção de um design.*

## 3. Design Colaborativo (*Collaborative Design*) e Participativo (*Participatory Design*)

*Sessões para criação de designs de forma colaborativa ou participativa, através do envolvimento de usuários reais.*



# Avaliar designs

## 1. Testes com usuários (*User testing*)

*Aplicação de testes de usabilidade utilizando o protocolo “Pense em voz alta”.*

## 2. Avaliação por Inspeção (*Inspection evaluation*)

*Inspecionar interfaces de usuário dos protótipos por meio de cenários, regras ou heurísticas de usabilidade.*

## 3. Método RITE (*Rapid Iterative Testing and Evaluation*)

*Identificar e resolver quantos problemas forem possíveis e verificar a eficácia das soluções no menor tempo possível.*

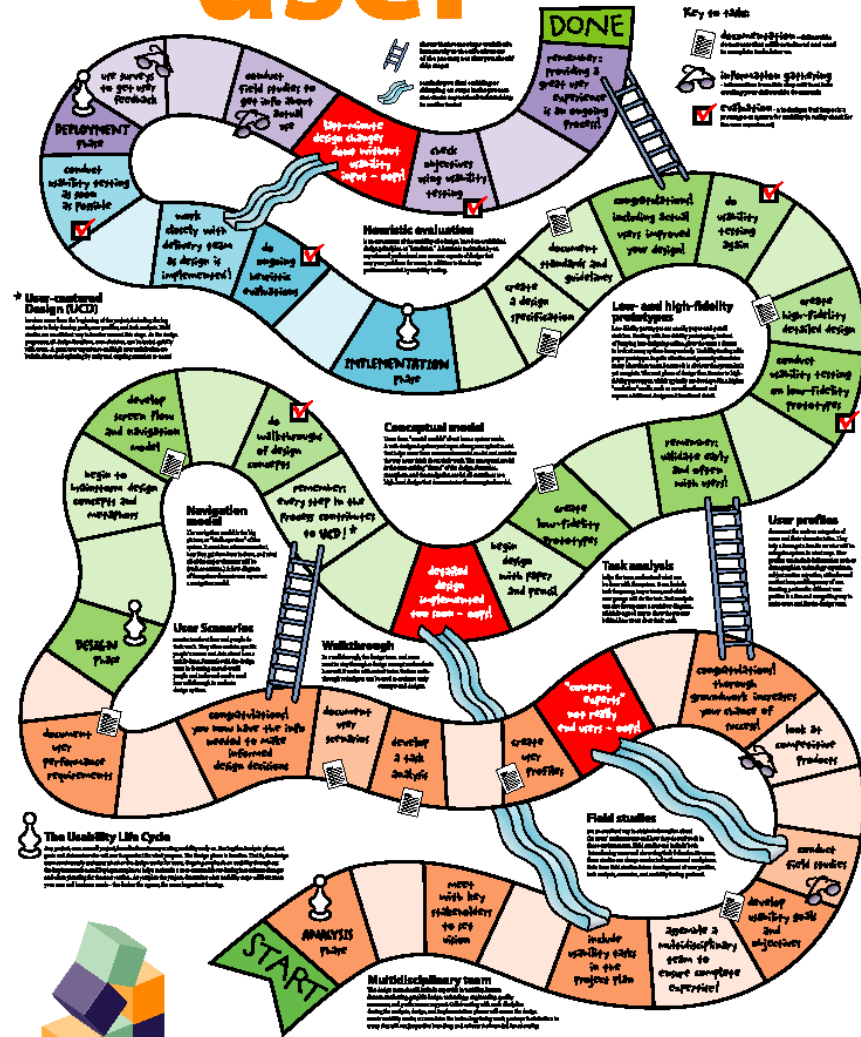
## 4. Testes de aceitação (*Acceptance tests*)

*Criação de testes de aceitação automáticos, aproveitando a prática já existente, em métodos ágeis, de TDD. Os testes são criados em linguagem natural.*

# **OUTROS PROCESSOS COMERCIAIS**

# designing the user experience

Additional rights to host copies of this the people distribution  
 permission is granted to anyone who wishes to reproduce this work  
 for non-commercial purposes. The copyright for this work remains  
 with the author. For more information, please contact the author  
 at [design@upa.org](mailto:design@upa.org)



**upa** usability professionals' association  
[www.upa.org](http://www.upa.org)

acknowledgements  
 Dan Salsbery  
 Jeff Todd  
 David G. Galletta  
 Dan Salsbery  
 David G. Galletta  
 Dan Salsbery  
 David G. Galletta

# Ciclo da peça de mídia interativa



Apesar de tudo, não podemos esquecer....

