

Sumário

1 Desenvolvimento para a Web

1 Desenvolvimento para a Web

Desafios do desenvolvimento

- Há pouco tempo, o desenvolvimento era feito por diferentes atores:
 - Programador;
 - Designer;
 - DBA.
- Nova figura: **full stack developer**.
- Você é capaz de administrar quantos bancos de dados?
- Na nova Web, **tudo é serviço!**
- Importância da utilização de **padrões** e **frameworks**.

Engenharia de software

- Engenharia de software para aplicações de Internet [Andersson et al., 2006];
- Modelo de 4 passos para o desenvolvimento de aplicações para a Web:
 - ① Desenvolva um modelo de dados. Quais as informações que devem ser armazenadas e como representá-las?
 - ② Desenvolva uma coleção de transações para o modelo de dados (INSERT, UPDATE);
 - ③ Desenhe um fluxo de páginas. Como o usuário vai interagir com o sistema? Quais os passos necessários para chegar à execução das transações?
 - ④ Implemente individualmente cada uma das páginas:
 - Escreva as consultas que recuperem as informações para o modelo de dados;
 - Encapsule as informações em um template;
 - Retorne o resultado para o usuário de maneira intuitiva.
- Será que o modelo ainda se aplica?

Manifesto ágil

- A chegada do **manifesto ágil**¹ reverte um pouco a lógica da engenharia de software.
 - *Indivíduos e interações acima de processos e ferramentas;*
 - *Software funcional acima de documentação extensa;*
 - *Colaboração com o cliente acima de negociação de contratos;*
 - *Resposta a mudanças acima de planejamento detalhado.*

¹Fonte: <http://agilemanifesto.org/>

Desenvolvimento ágil

- Comece a programar imediatamente [Ruby et al., 2009]:
 - Não tente especificar todos os detalhes antes de começar;
 - Escreva uma especificação **mínima** e crie alguma funcionalidade **imediatamente**;
 - Implemente uma ideia;
 - Colete feedback;
 - Inicie outro ciclo de desenho mínimo e desenvolvimento.
- Não serve para todos os casos;
- Precisa de **integração** com o cliente;
- Figura do **PO – Product Owner**.

Processo ágil

- Fazendo um paralelo ao modelo de quatro passos:
 - 1 Identifique os **atores** com a ajuda do **cliente**;
 - 2 Desenhe um **fluxo de páginas** levando em consideração os principais **atores**;
 - O fluxo de páginas representa um **protótipo não funcional**;
 - Leve sempre em consideração o ponto de vista do **usuário**.
 - 3 Desenhe um modelo de dados **mínimo**;
 - A aplicação deve ser capaz de responder a **mudanças no modelo de dados**;
 - O versionamento do modelo deve fazer parte do projeto;
 - Princípio: o modelo de dados **nunca está completo**.
 - 4 Faça o planejamento dos **ciclos de desenvolvimento** ou **releases**;
 - 5 Comece a **programar**.

Fluxo de páginas 1

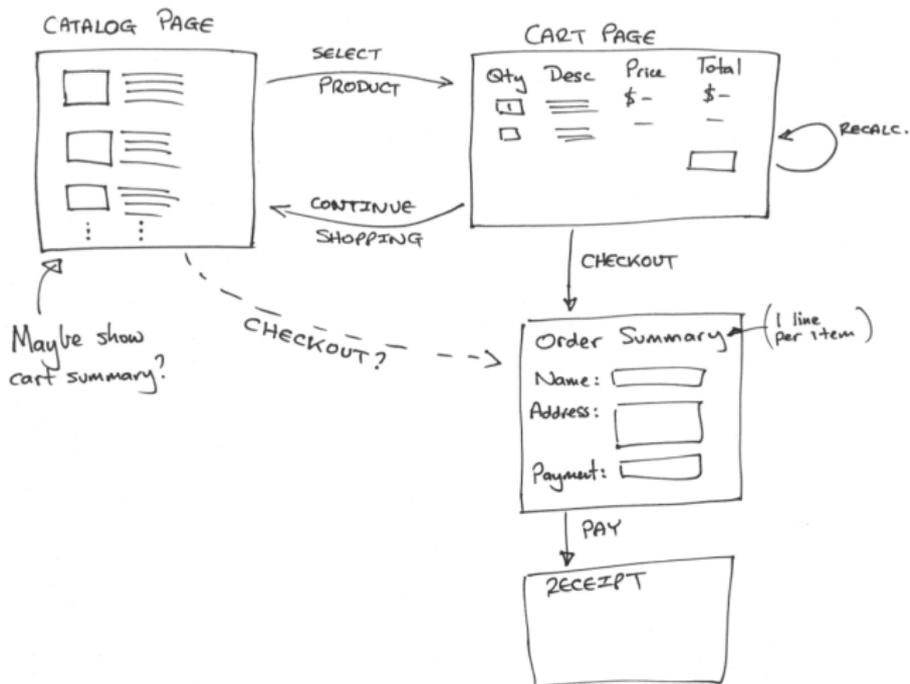


Figura 1.1: Exemplo de página de um ator comprador [Ruby et al., 2009]

Fluxo de páginas 2

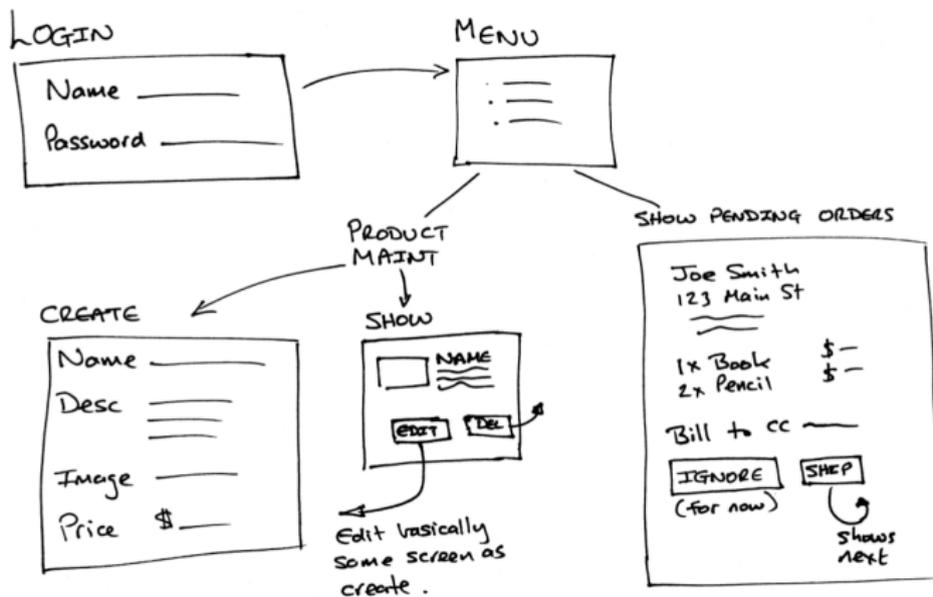


Figura 1.2: Exemplo de página de um ator vendedor [Ruby et al., 2009]

Modelo de dados

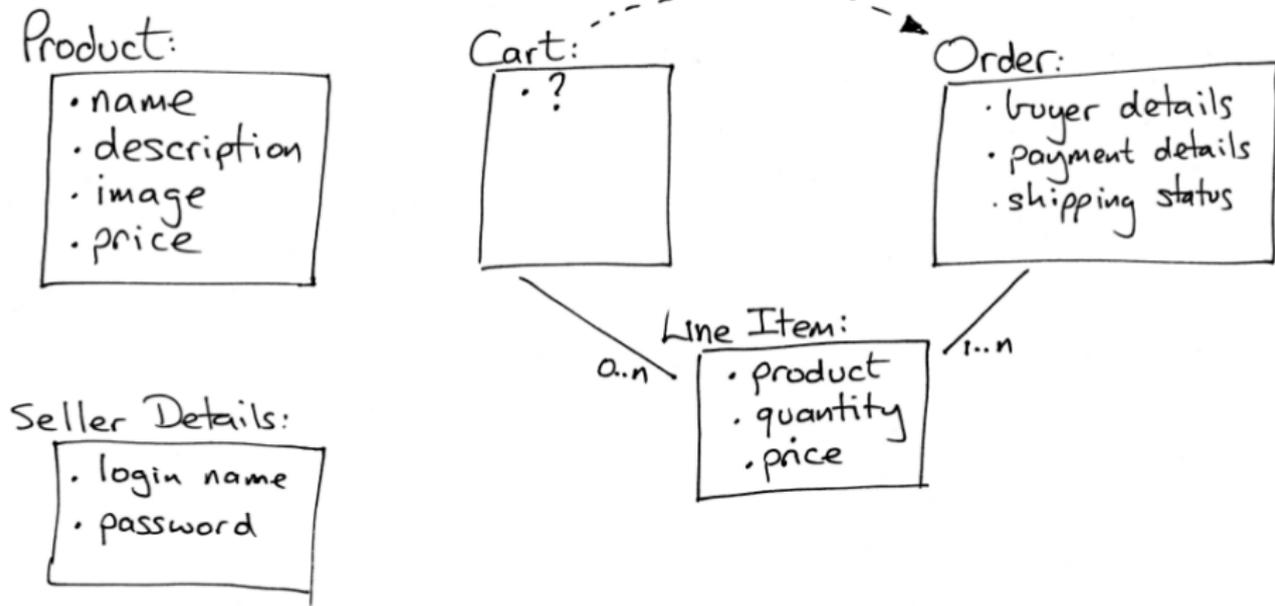


Figura 1.3: Modelo de dados mínimo [Ruby et al., 2009]

Planejamento

- Conceito de **histórias de usuário**: o que as pessoas querem que esteja no produto, em ordem de prioridade;
- A primeira etapa do planejamento é construir uma lista de histórias de usuário (**backlog list**);
- Cada história representa uma **funcionalidade** do ponto de vista do **usuário**;
- A funcionalidade pode ser dividida em uma ou mais **tarefas**:
 - A tarefa é a menor **unidade de implementação**;
 - Conceito de gerência de projetos: as tarefas devem durar entre **4 e 40 horas**;
 - A dificuldade/duração da tarefa é relativa ao programador.
- Toda funcionalidade deve ser **testável** com base nos **critérios de aceitação**;
- As histórias são organizadas em **ciclos de desenvolvimento** ou **sprints**.

Scrum board



Figura 1.4: Exemplo do quadro de tarefas do scrum

Histórias de usuário 1

- Uma boa história de usuário funciona como uma **conversa** com o cliente;
- Pensamentos importantes para a história de usuário:
 - COMO** Definição do ator envolvido (QUEM)?
 - EU QUERO** Como você vai interagir com o sistema?
 - PARA QUE** Qual o impacto da funcionalidade do ponto de vista do usuário?

Histórias de usuário 2

- Para saber se a história foi concluída escreva **critérios de aceitação**:
 - Exemplo em relação à funcionalidade:
 - DADO QUE** Quais os pré-requisitos para que a história seja executada?
 - QUANDO** Qual é a ação que deve ser executada?
 - ENTÃO** O que o sistema deve apresentar/realizar?
 - Um outro exemplo, levando em consideração possíveis consultas:
 - ENQUANTO** Qual o ator?
 - QUERO** Qual a ação?
 - PELOS SEGUINTE ATRIBUTOS** Em que condição?
- Uma boa história de usuário tem entre **3 e 9** critérios de aceitação.

Sprints

- Os **sprints** são as **iterações** de programação ao final das quais se obtém **feedback** sobre o produto entregável;
- Cada sprint deve possuir impacto suficiente para produzir uma **mudança visível** no programa;
- Como regra os sprints duram entre **2-4 semanas**;
- Organização do trabalho:
 - Reunião de abertura: define-se quais as histórias que serão implementadas no sprint;
 - Reuniões diárias (*stand-up meeting*): feedback diário sobre a evolução em relação ao dia anterior. Diz-se *stand-up* porque deve ser realizada **de pé** para ter curta duração (15 min);
 - Reunião de fechamento: avalia-se o produto do sprint para planejar o andamento do próximo.

Ciclos de desenvolvimento

- Conceito de **release**: menor unidade implementável que possua **impacto para o usuário**;
- O release é como uma nova versão do software, mas do ponto de vista do usuário;
- No ágil, o software **nunca está pronto**. Por isso, cada release é **muito importante**.

Observações gerais

- Durante o desenvolvimento, é muito importante o ciclo de feedback;
- O software nunca está pronto;
- A produtividade é uma medida tangível, mas subjetiva;
- Controle total e absoluto do tempo.

OBRIGADO!!!
PERGUNTAS???

-  Andersson, E. A., Greenspun, P., Grumet, A., Eve Andersson, P. G., et al. (2006).
Software engineering for Internet applications.
Number Sirsi) i9780262511919.
-  Ruby, S., Thomas, D., and Hansson, D. (2009).
Agile web development with rails . pragmatic bookshelf.
Technical report, ISBN 1-934356-16-6.