



# Engenharia de Requisitos

Tecnologia de Software

## **Aula 3:** Atividades de ER e Visão

Fábio Levy Siqueira

[levy.siqueira@usp.br](mailto:levy.siqueira@usp.br)

# Visão

- Diferentes *stakeholders* têm diferentes objetivos de negócio
- Importante definir uma **visão comum**
  - Entender (e acordar) o problema a ser resolvido
  - Definir o escopo do sistema
  - Entender quais são os *stakeholders*
  - Capturar as metas e *requisitos básicos* dos *stakeholders*
    - **Features**
- Se for usado **Design Thinking** como processo
  - A visão comum já foi decidida?

# Visão

- Objetivo de negócio muda lentamente
  - O escopo muda durante o projeto
    - Afetado por prazo, orçamento, recursos e outras restrições
  - Importante alinhar o **escopo** ao objetivo de negócio
    - Fronteira do que está dentro ou fora do projeto
    - Software e o que interage com o software




# Visão

- Com quem deve-se conversar para decidir a visão comum?
  - Patrocinador do projeto
  - Cliente
  - Também envolve conversas com
    - Alta gerência da empresa
    - Visionário do produto
    - Usuários

# Atividades

- Identificar os *stakeholders*
- Definir o problema
- Descrever as *features* principais
- Definir o escopo
- Posicionamento do produto
- Identificar outras restrições



A ordem não é  
necessariamente  
sequencial

# Stakeholders

- Explicitar quais são os *stakeholders*

Pessoas ou organizações que tem um interesse no sistema a ser desenvolvido (POHL, 2011)

- Em especial, quais são os **usuários** do sistema
  - Descrever o perfil básico deles e o ambiente que os usuários trabalham
- Pode-se detalhar os perfis
    - Exige um maior entendimento ou até mesmo uma pesquisa / análise de mercado
    - Fundamental para produtos de “prateleira”

# Stakeholders

- Algumas perguntas para identifica-los (LEFFINGWELL; WIDRIG, 2003; BITTNER, SPENCE, 2003)
  - Quem são os usuários do sistema?
  - Quem é o cliente (quem está pagando)?
  - Quem será afetado pelo sucesso ou falha do sistema?
  - Quem será afetado pelas saídas do sistema?
  - Quem irá avaliar e aprovar o sistema quando ele for entregue e implantado?
  - Existem outros usuários internos/externos cujas necessidades precisam ser endereçadas?
  - Existem órgãos regulatórios ou padrões que o sistema precisa atender?
  - Quem irá desenvolver, instalar e manter o sistema?

# Stakeholders

- *Exemplo:* loja virtual
  - Compradores
  - Embaladores / estoque
  - Atendimento ao cliente
  - Contadores / departamento financeiro
  - Empresa de entrega
  - Gerência da loja
  - Fornecedores de produtos
  - Operadoras de cartão de crédito



# Definir o problema

- Acordar o problema a ser resolvido
  - Uma outra opção é descrever a **oportunidade de negócio**
  - Um formato útil (LEFFINGWELL; WIDRIG, 2003)

O problema de	<i>Descrever o problema</i>
afeta	<i>Identificar os stakeholders afetados pelo problema</i>
tem como impacto	<i>Descrever o impacto desse problema nos stakeholders e no negócio</i>
Uma solução de sucesso seria/teria	<i>Indicar alguns benefícios principais</i>

# Definir o problema

- *Exemplo: loja virtual*

O problema de	<i>Vender os produtos apenas nas lojas físicas</i>
afeta	<i>Clientes que não desejam ir à loja, clientes que tem dificuldades de ir às lojas, clientes que desejam comprar em horários diferenciados</i>
tem como impacto	<i>Menor venda de produtos e menor espaço no mercado</i>
Uma solução de sucesso seria	<i>Permitir a venda e entrega de todos os produtos disponíveis na loja física através da Internet, a qualquer hora</i>

- Objetividade é fundamental!

# Features

- Descreve em alto nível o produto em questão
  - Não é técnico: visão dos *stakeholders* e de **marketing**
    - Não precisam ter uma definição tão precisa
    - Benefícios e aspectos excitantes
      - O que distingue o sistema dos competidores
  - Podem ser funcionais ou não funcionais
    - Metas e requisitos dos stakeholders (de alto nível)
- Representam um conjunto de requisitos
- Podem ser usadas várias técnicas para obtenção
  - Pode ser o resultado do processo de DT

# Features

- *Exemplo: loja virtual*
  - Pagamento de compras usando cartão de crédito e boleto bancário
  - Controle de estoque
  - Compra de qualquer produto da loja em estoque
  - Compatível com qualquer navegador
  - ...
  
- Detalhes
  - Importante numerar as *features*
  - Definir as *features* da primeira versão
    - Escopo / priorização

# Escopo do sistema

- Definir os atores envolvidos

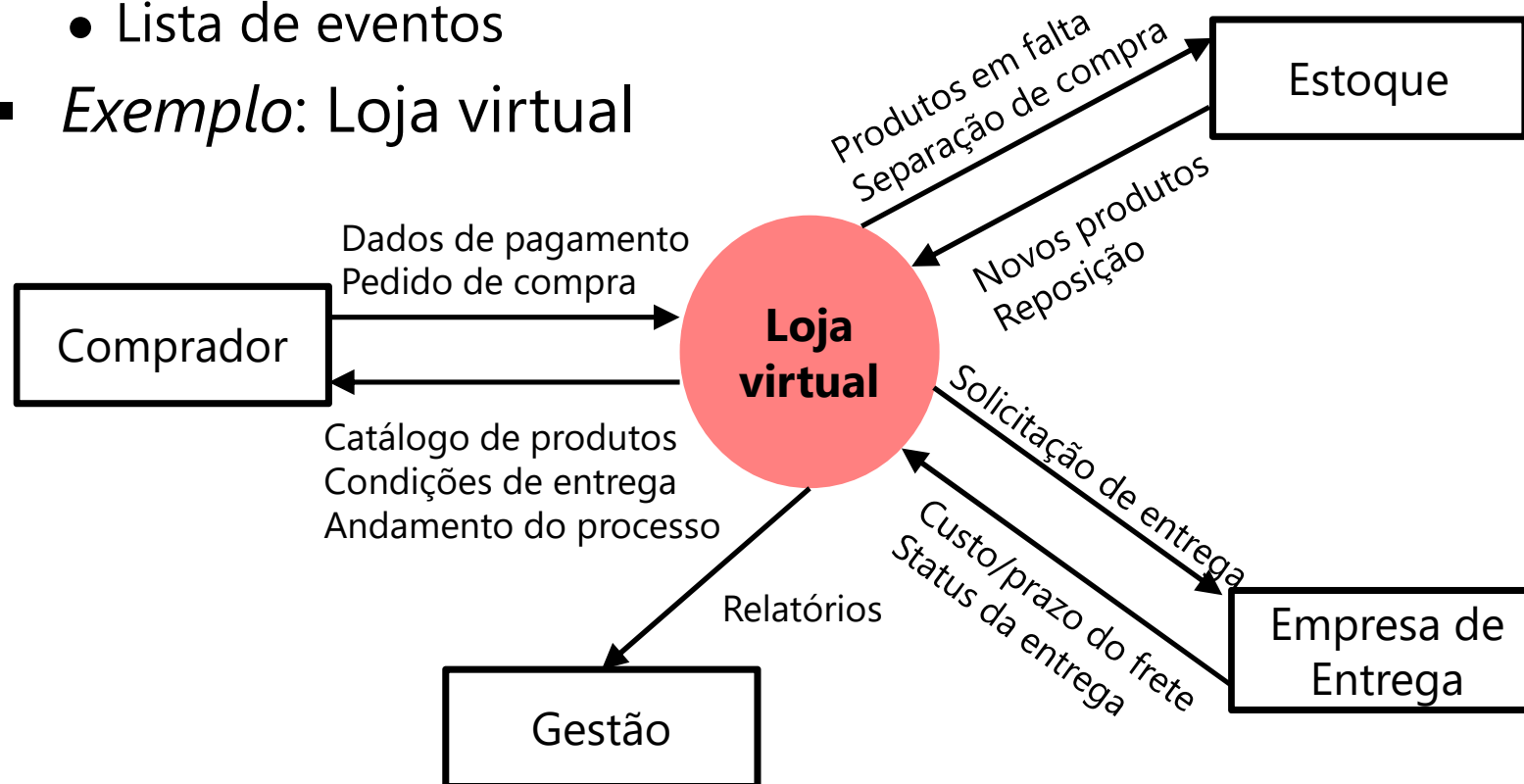
**Ator:** alguém ou algo **fora do software** que interage com o software

- Algumas perguntas

- Que irá fornecer, usar, ou remover informação do sistema?
- Quem irá operar o sistema
- Quem irá realizar a manutenção?
- Onde o Sistema será usado?
- De onde o sistema obtêm informação?
- Quais outros sistemas externos interagem com o Sistema?

# Escopo do sistema

- Algumas representações (WIEGERS; BEATY, 2013)
  - **Diagrama de contexto**
  - Lista de eventos
- *Exemplo:* Loja virtual



# Posicionamento do produto

- Visão geral do produto
  - Descreve a razão para construir o sistema

Para	<i>Cliente / Usuário alvo</i>
Que	<i>Necessidade ou oportunidade</i>
O	<i>Nome do sistema</i>
É	<i>O tipo do sistema</i>
Que	<i>Benefício principal (motivo principal para comprar)</i>
Diferentemente de	<i>Concorrentes, alternativas, atual</i>
Nosso produto	<i>Diferença principal</i>

# Posicionamento do produto

- *Exemplo: loja virtual*

Para	<i>Clientes da loja X</i>
Que	<i>Desejam comprar os produtos sem sair de casa e em qualquer horário</i>
O	<i>lojax.com</i>
É	<i>Uma loja virtual</i>
Que	<i>Permite a compra de produtos pela web</i>
Diferentemente de	<i>Comprar na loja</i>
Nosso produto	<i>Permite a seleção de produtos, pagamento e entrega pela web</i>



# Outras restrições

- Restringem a liberdade em prover a solução
  - Recurso, custo máximo e prazo
  - Questões de negócio
  - Leis, padrões e regulamentos
  - Tecnologias
    - Proibições ou obrigações
  - Sistemas existentes
    - Comunicação entre sistemas ou compatibilidade
  - Política organizacional
  - ...
- Quanto menos restrito, melhor

# **Documento visão**

# Visão comum

- Como capturar / documentar a visão comum do ponto de vista do negócio?
  - **Documento visão**
  - *Project charter*
  - *Business case*
  - *Datasheet* do produto
  - *Press release* preliminar

# Documento visão

- Busca capturar a **essência do produto**
  - Alto nível
  - Representa o **consenso** sobre a necessidade do produto e sua utilidade
    - “Porquê” e o “o quê”
- Uso
  - Acordo sobre o projeto
  - Decisão sobre a viabilidade do projeto
  - Facilita o entendimento do projeto

# Documento visão

- Cobre análise do negócio e requisitos
  - Visão de negócio
    - Marketing e produto
    - (Não entra no plano de negócio)
  - Metas e requisitos principais
    - Não entra nos requisitos de sistema (normalmente)
- Documento conciso
  - Simples e objetivo
    - Normalmente de 5 a 10 páginas (Leffingwell, 2011)
  - O nível de detalhe do conteúdo varia com a necessidade do projeto

# Documento visão

- Conteúdo simplificado (WIEGERS; BEATY, 2013)

1. Requisitos de negócio

1. *Background*

2. Oportunidade de negócio

- Definição do problema

3. Objetivo de negócio

- Exemplo

- Capturar um market share de X% em Y meses
- Reduzir os custos de \$X para \$Y em Z meses

4. Métricas de sucesso

- Saber se o projeto irá atender os objetivos de negócio

5. Posicionamento do produto

6. Riscos de negócio

7. Fatos assumidos de negócio e dependências

# Documento visão

- Conteúdo simplificado (WIEGERS; BEATY, 2013)
  - 2. Escopo e limitações (*diagrama de contexto*)
    - 1. *Features* principais
    - 2. Escopo da primeira versão (*features*)
    - 3. Escopo das próximas releases
    - 4. Limitações e exclusões
  - 3. Contexto de negócio
    - 1. Perfil dos *stakeholders*
    - 2. Outras restrições
    - 3. Considerações de entrega

# Conclusão

- Documento visão é um documento inicial dos requisitos
  - Essencial para começar um projeto
  
- Existem diversos *templates*
  - Rational (IBM)
    - [https://www.ibm.com/support/knowledgecenter/en/SSCP65\\_6.0.4/com.ibm.rational.rrm.help.doc/topics/r\\_vision\\_doc.html](https://www.ibm.com/support/knowledgecenter/en/SSCP65_6.0.4/com.ibm.rational.rrm.help.doc/topics/r_vision_doc.html)
  - Wiegers e Beatty (2013)
  - Leffingwell e Widrig (2003)
  - Leffingwell (2011)



# Referências

- BITTNER, K.; SPENCE, I. **Use Case Modeling**. Addison Wesley, 2003.
- LEFFINGWELL, D. **Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and Enterprise**. Addison-Wesley, 2011.
- LEFFINGWELL, D.; WIDRIG, D. **Managing Software Requirements: A Use Case Approach**. 2ª edição. Addison-Wesley, 2003.
- POHL, K. **Requirements Engineering: Fundamentals, Principles, and Techniques**. Springer, 2010.
- WIEGERS, K.; BEATTY, J. **Software Requirements**. 3ª edição. Microsoft Press, 2013.