

# Levantamento, Análise e Gestão Requisitos

## Aula 06



ENGENHARIA  
DE REQUISITOS



Programadores {+ Inovadores ;}  
[www.x25.com.br](http://www.x25.com.br)

- Técnicas de Levantamento de Requisitos:
  - Entrevista
  - Workshop, Brainstorming, Storyboarding e Roleplaying
  - Prototipação
  - JAD – Joint Application Design



# Reuniões de Requisitos

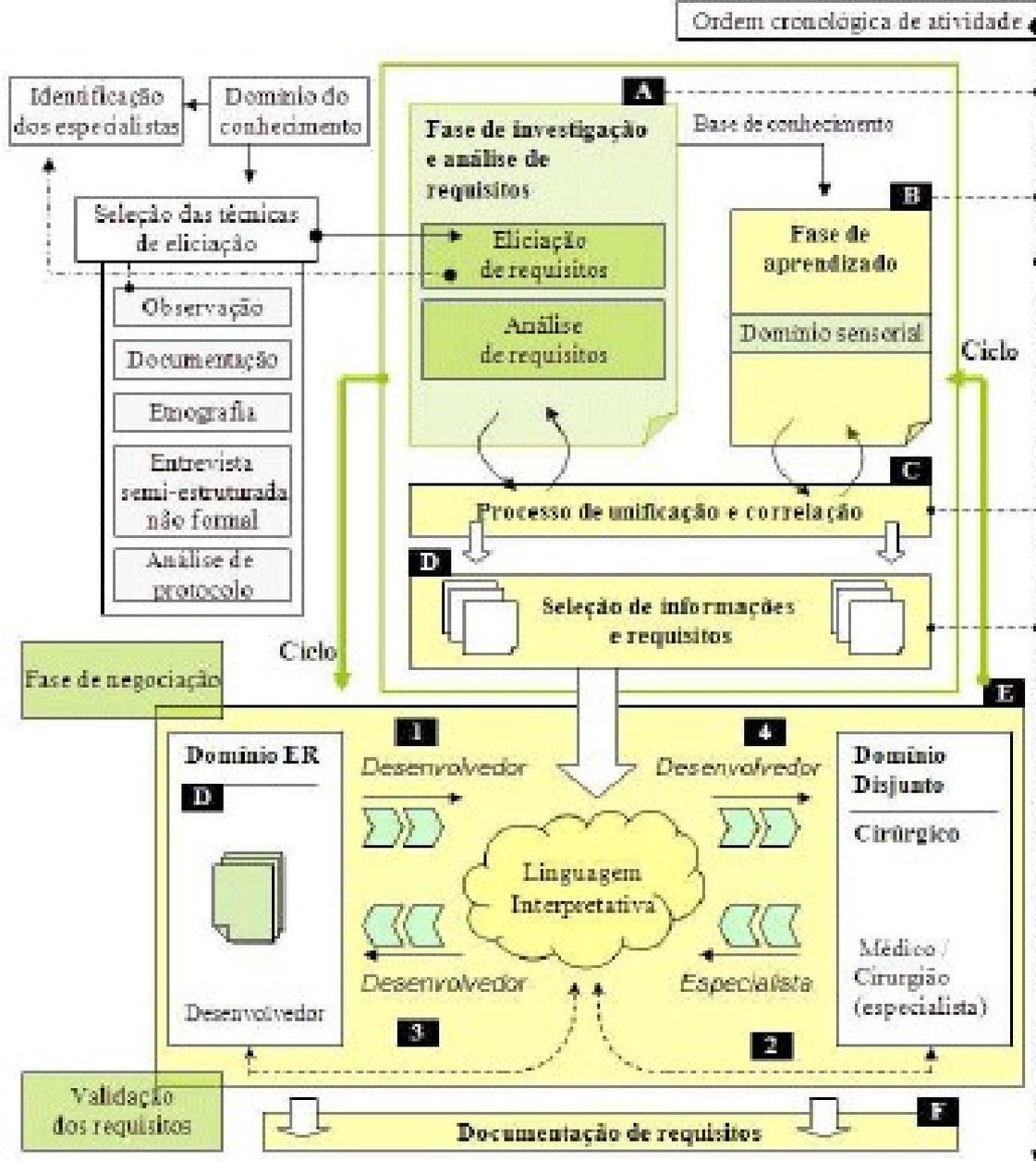


Programadores (+ Inovadores ;)

[www.x25.com.br](http://www.x25.com.br)

# Modelo

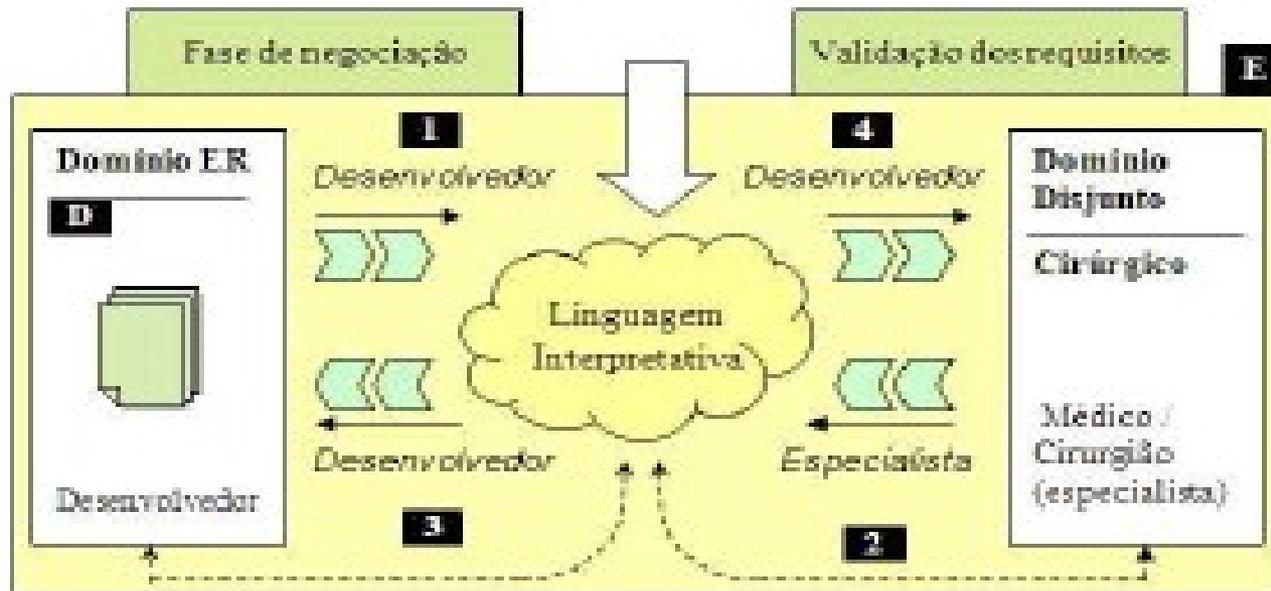
Todas as dificuldades mencionadas anteriormente se tornam ainda maiores quando o problema é especificar requisitos de sistemas automatizados (discretos) para aplicações cujo domínio está distante dos processos da engenharia



## Atividades Envolvidas na Linguagem Interpretativa

Ciclo de negociação e validação é a fase onde se trabalha diretamente com o especialista para apresentar e discutir os resultados do levantamento e análise de requisitos

Esta fase fecha o ciclo de eliciação e análise de requisitos, e sintetiza o que se chamará de especificação do sistema



## *Analisar o Problema do Projeto - Questões*

Por que leva tanto tempo para darmos um software como terminado?

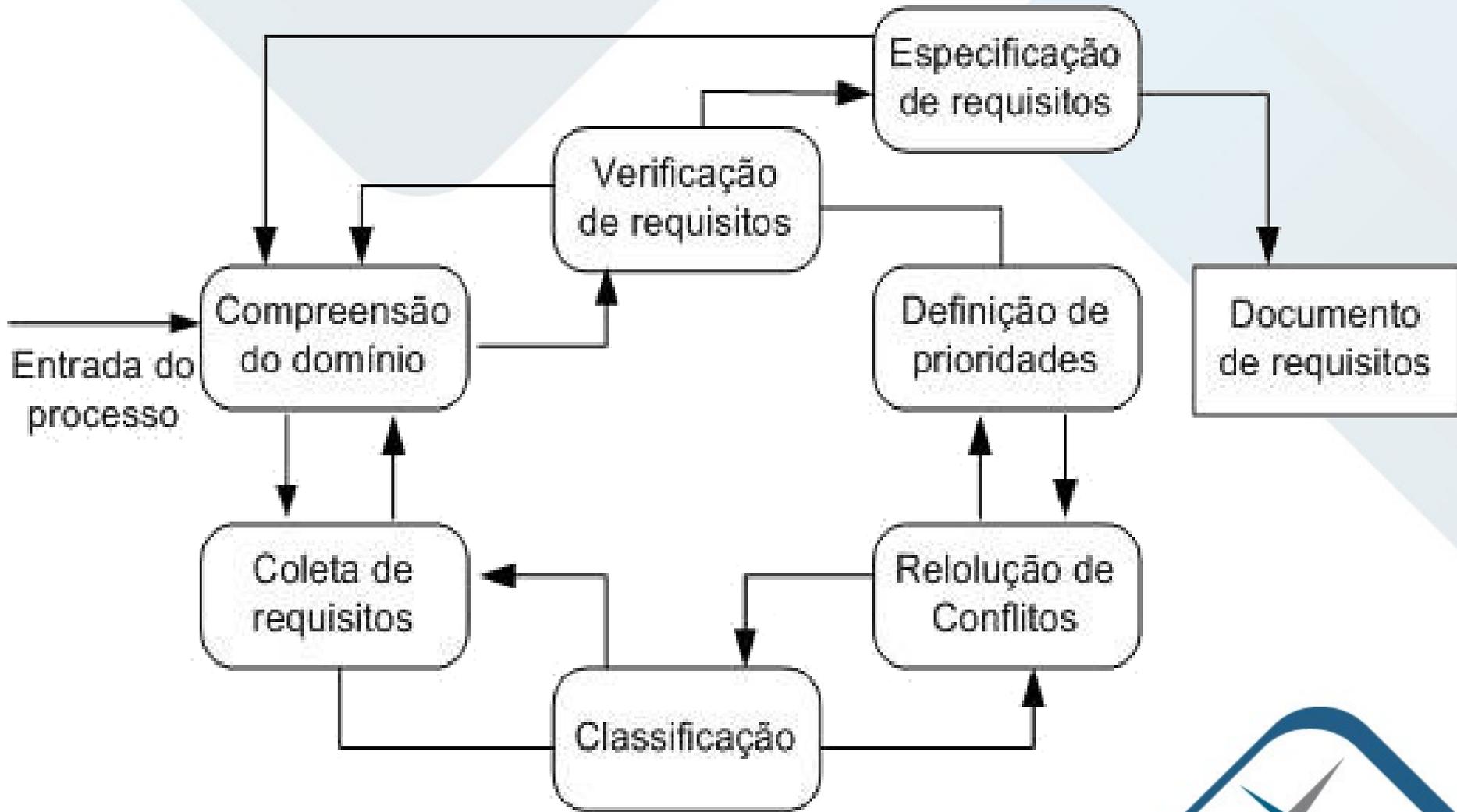
Por que os custos com desenvolvimento de software são tão altos?

Por que não conseguimos encontrar todos os erros antes de entregarmos o software para o cliente?

Por que ainda continuamos a ter dificuldades em medir o progresso de como um software está sendo desenvolvido?



# Levantamento de Requisitos de Software



# *Análise do Problema do Projeto*

## **Mito:**

Já temos um manual repleto de padrões e procedimentos para a construção de software. Isso não oferecerá ao meu pessoal tudo que eles precisam saber?

## **Realidade:**

O manual de padrões pode muito bem existir, mas será que ele é usado?

Os profissionais de software têm conhecimento de sua existência?

Ele reflete a moderna prática de desenvolvimento de software? É completo?

Em muitos casos, a resposta a todas estas perguntas é "não"



## *Análise do Problema do Projeto*

### **Mito:**

Se estamos atrasados nos prazos podemos adicionar mais programadores e tirar o atraso (*chamado conceito de horda dos mongóis*)

### **Realidade:**

O desenvolvimento de software não é um processo mecânico igual à manufatura

Brooks disse: "*...acrescentar pessoas em um projeto de software atrasado torna-o ainda mais atrasado*"

Quando novas pessoas são acrescentadas, as pessoas que estavam trabalhando devem gastar tempo educando os recém chegados, o que reduz o tempo despendido num esforço de desenvolvimento produtivo



# *Análise do Problema do Projeto*

## **Mito:**

Uma declaração geral dos objetivos é suficiente para começar a escrever programas – podemos preencher os detalhes mais tarde

## **Realidade:**

Uma definição inicial ruim é a principal causa de fracasso dos esforços de desenvolvimento de software

Uma descrição formal e detalhada do domínio da informação, função, desempenho, interfaces, restrições de projeto e critérios de validação é fundamental

Essas características podem ser determinadas somente depois de cuidados comunicação entre o cliente e o desenvolvedor



# Análise do Problema do Projeto

## Mito:

Os requisitos de projeto modificam-se continuamente, mas as mudanças podem ser facilmente acomodadas, porque o software é flexível

## Realidade:

É verdade que os requisitos de software se modificam, mas o impacto da mudança varia de acordo com o tempo em que ela é introduzida. Se uma séria atenção for dada à definição inicial, os primeiros pedidos de mudança podem ser acomodados facilmente:

- Definição . . . . . x
- Desenvolvimento . . . . . 1.5x a 1.6x
- Manutenção . . . . . 60x a 100x



# Análise do Problema do Projeto

## Mito:

Assim que escrevermos o programa e o colocarmos em funcionamento nosso trabalho estará completo

## Realidade:

*"Quanto mais cedo se começa a escrever o código, mais tempo demora para que se consiga terminá-lo"*

Os dados da indústria indicam que entre 50 e 70% de todo esforço gasto num programa serão despendidos depois que ele for entregue pela primeira vez ao cliente



# Análise do Problema do Projeto

## Mito:

A única coisa a ser entregue em um projeto bem sucedido é o programa funcionando

## Realidade:

Um programa funcionando é somente uma parte de uma *configuração de software* que inclui: plano, especificação de requisitos, projeto, estrutura de dados, listagem, especificação de teste, programa funcionando



# *Análise do Problema do Projeto*

## **Mito:**

Enquanto não tiver o programa "funcionando", eu não terei realmente nenhuma maneira de avaliar sua qualidade

## **Realidade:**

Um dos mecanismos mais efetivos de garantia da qualidade de software pode ser aplicado desde o começo de um projeto - a revisão técnica formal

As revisões de software são um "filtro da qualidade" que têm sido consideradas mais eficientes do que a realização de testes para a descoberta de certas classes de defeitos de software



# Análise do Problema

Problema de Negócio

*Identifique stakeholders para o problema.  
Análise da Causa-Raiz*

Idéia de Solução ou Oportunidade 

Problema de Negócio Definido

Problema Atual identificado e definido

*Entendimento dos Problemas no Contexto dos Objetivos de Negócio.*

Problema validado/ajustado

*Escolher as melhores soluções para alcançar os objetivos.*

*Reavaliar qual é a melhor idéia de solução.*

Melhor solução identificada

Elicitar Requisitos

*Expandir a lista de soluções do stakeholder*



## *Elicitação de Requisitos*

ELICITAR: descobrir, tornar explícito, obter o máximo de informações para o conhecimento do objeto em questão

Identificar os fatos que compõem os requisitos do sistema a fim de prover o mais correto e mais completo entendimento do que é demandado daquele software

Entender o processo como um todo

Envolver o ambiente de trabalho do cliente, ou seja, se misturar com os funcionários, observar, aprender, e questionar, de forma a superar a ignorância do domínio do problema

Entender a razão porque as coisas são feitas da forma que são



## *Dificuldades para Elicitação de Requisitos*

- Usuários podem:
  - Não ter uma ideia precisa do sistema por eles requerido
  - Ter dificuldades para descreverem seu conhecimento sobre o domínio do problema
  - Ter diferentes pontos de vista do problema (por terem formações diferentes)
  - Antipatizar com o novo sistema e se negar a participar da elicitação (ou mesmo fornecer informações errôneas)



## *Atividades para Elicitação de Requisitos*

- É uma atividade de negociação e comunicação entre especialistas e não especialistas
- A qualidade da comunicação entre as partes envolvidas depende:
  - de um bom entendimento
  - de uma análise rigorosa
- **Resultado:** Uma declaração em linguagem natural dos serviços que o sistema deverá prover e suas limitações operacionais



# *Atividades para Elicitação de Requisitos*

## **Entendimento do domínio da aplicação**

- O conhecimento do domínio da aplicação é o conhecimento geral onde o sistema será aplicado

## **Entendimento do problema**

- Os detalhes específicos do problema do cliente onde o sistema será aplicado deve ser entendido

## **Entendimento do negócio**

- Você deve entender como os sistemas interagem e contribuem de forma geral com os objetivos do negócio

## **Entendimento das necessidades e limitações dos *stakeholder* sistema**



# Produtos para Elicitação de Requisitos

- Lista dos usuários do sistema
- Entendimento do fluxo de trabalho dos usuários
- Papel de cada departamento/setor que irá utilizar o sistema
- O que atrapalha o andamento do processo do usuário e o que pode ser feito para melhorar
- Lista dos requisitos



# *Técnicas para Elicitação de Requisitos*

- Técnicas especiais que podem ser usadas para coletar conhecimento sobre os requisitos dos usuários

1. Entrevista
2. Workshop
3. Brainstorming
4. Storyboarding
5. Roleplaying
6. Prototipação
7. JAD (*Joint Application Design*)



## *Escolhendo a Técnica*

- Selecionar a(s) técnica(s) a ser(em) utilizada(s)
- Caso várias técnicas sejam selecionadas, estabelecer a maneira como serão integradas
- Escolher dessa(s) técnica(s) seu esquema de integração depende do problema e da equipe participante
- Conhecer e identificar onde uma técnica é aplicada melhor



## Técnica 1: Entrevista

- O Engenheiro de requisitos ou analista discute o sistema com diferentes *stakeholders* para obter um entendimento dos requisitos
- Vantagens: contato direto com o usuário e validação imediata
- Desvantagens: conhecimento tácito e diferenças de cultura



## Técnica 1: Entrevista – Tipos

- **Fechadas:** buscar respostas em um conjunto de questões pré-definidas
- **Abertas:** discutir de forma aberta o que o *stakeholder* quer do sistema
- **Tutorial:** o cliente ministra uma aula explicando o seu trabalho



## ***Técnica 1: Entrevista – Dicas***

- Identificar candidatos
- Preparar a entrevista: agendar e preparar um questionário
- Não ter noções pré concebidas
- Explicar o relacionamento entre o que está em discussão e as demais partes do sistema
- Descrever o ponto de vista de outros usuários em relação ao item que esteja sendo discutido
- Descrever informalmente a narrativa do item em que o analista deseja obter informações
- Perguntar ao usuário se o item em discussão depende para a sua existência de alguma outra coisa



## *Técnica 1: Entrevista versus Questionários*

- Muitas vezes os desenvolvedores se questionam em relação à real necessidade de fazer as entrevistas, quando, supostamente, um questionário com os mesmos itens poderia ser enviado a 100 pessoas no mesmo período de tempo em que uma entrevista é feita
- Não há substituto possível para o contato pessoal, nem para a maneira informal em que o desenvolvedor pode se comportar durante alguns momentos na entrevista
- Embora a aplicação de um questionário possa permitir tratamento estatístico dos resultados quantitativos, o questionário não substitui a entrevista em termos do seu potencial de contribuição



# *Técnica 1: Entrevista versus Questionários*

- Problemas dos Questionários:
  - Adição de perguntas determinadas ao longo da atividade (entrevista) não pode ser feita de maneira adiantada, já que não há como prever as respostas do entrevistado
  - Os pressupostos subjacentes às perguntas acabam comprometendo as respostas
  - É difícil explorar novos domínios e não há interação para explorar os assuntos a serem explorados
  - É difícil dar continuidade a respostas vagas ou ambíguas



## Técnica 2: Workshop

*Workshop* ou Oficina de Requisitos constituem uma das técnicas mais poderosas para a extração de requisitos

Reunir todos os *stakeholders* durante um período curto mas intensivo e focado

Ser concebido com o intuito de encorajar o consenso sobre os requisitos da aplicação

Conseguir concordância rápida entre as partes



Programadores {+ Inovadores ;}

[www.x25.com.br](http://www.x25.com.br)

## Técnica 2: Workshop – Agenda

período	item	descrição
8:00 às 8:30	introdução	Revisar agenda, facilidades e regras
8:30 às 10:00	contexto	Apresentar o estado do projeto, as necessidades, os resultados das entrevistas de usuários e demais
10:00 às 12:00	<i>brainstorming</i>	Explicar aspectos de <i>brainstorming</i> para o projeto
12:00 às 13:00	almoço	
13:00 às 14:00	<i>brainstorimng</i>	Resgatar e continuar com as contribuições abertas
14:00 às 15:00	definição de aspectos	Escrever definições de 2 ou 3 sentenças de cada aspecto
15:00 às 16:00	priorização	Priorizar aspectos
16:00 às 17:00	fecho	Resumir e atribuir ações, registrar aspectos futuros



## Técnica 2: Workshop – Preparação

Construir um time propriamente dito, comprometido com um objetivo do projeto

Todos os *stakeholders* devem ter o seu espaço de contribuição

A técnica leva à concordância entre os *stakeholders* e os desenvolvedores em relação ao que a aplicação deve fazer

Expor e resolver questões políticas que possam interferir no sucesso do projeto

O resultado, uma definição preliminar do sistema no nível dos aspectos, fica pronta imediatamente



## Técnica 3: Brainstorming



- Tipo específico de *Workshop*
- Consiste de uma ou várias reuniões que permitem que as pessoas sugiram e explorem várias ideias
- Ideias são encorajadas, pois elas frequentemente estimulam os participantes
- Pode levar a soluções criativas para o problema



## ***Técnica 3: Brainstorming – Principais Etapas***

**Seleção dos participantes:** Os participantes devem ser selecionados em função das contribuições diretas que possam dar durante a sessão.

**Explicar a técnica e as regras a serem seguidas:** O líder da sessão explica os conceitos básicos de *brainstorming* e as regras a serem seguidas durante a sessão

**Produzir uma boa quantidade de ideias:** Os participantes geram tantas ideias quantas forem exigidas pelos tópicos que estão sendo o objeto do *brainstorming*



## Técnica 3: Brainstorming – Dicas

- Encorajar a participação de todos os presentes
- Permitir o aproveitem das ideias dos demais
- Ter uma alta amplitude, muitas ideias em um curto período de tempo
- Indicar inúmeras soluções alternativas para o problema inicialmente colocado para discussão
- Estimula o raciocínio aberto, sem amarras, sem as restrições comuns



Programadores {+ Inovadores ;}

[www.x25.com.br](http://www.x25.com.br)

# Técnica 4: Storyboarding

STORYBOARD

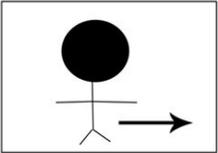
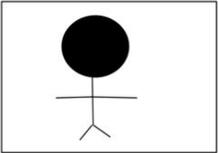
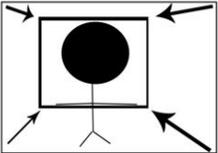
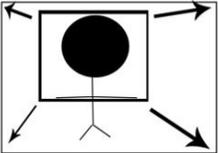
---

Title \_\_\_\_\_ Date \_\_\_\_\_

---

Producer(s) \_\_\_\_\_ Page \_\_\_\_\_ of \_\_\_\_\_

---

Video	Time	Audio
	_____	When story Boarding Arrows in the frame denote movement of a person or object.
	_____	An arrow outside the frame shows Movement of the Camera
	_____	A frame inside of a frame Can show a push or zoom in
	_____	Like wise it can show a push or zoom out.

Após a determinação dos atores, explicar o que acontece com eles e descrever a forma como isso acontece

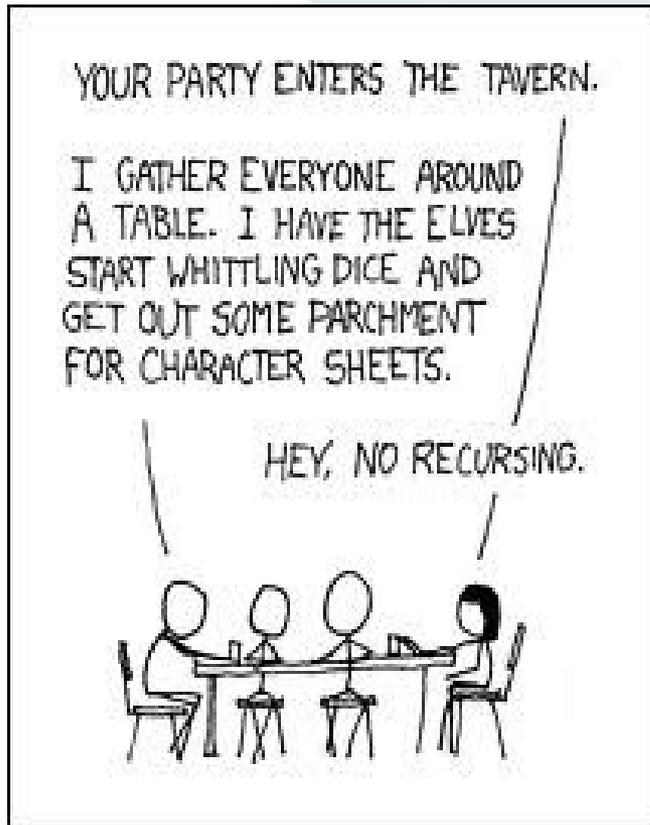
Suscitar as reações dos *stakeholders* o quanto antes em relação ao sistema proposto

Elaborar um modelo esquemático e fácil de modificar

Realizar o mais cedo possível no processo de desenvolvimento de software com conteúdo novo e inovador



## Técnica 5: Roleplaying



Determina os atores, explica o que acontece com eles e descreve a forma como isso acontece

Suscitar as reações dos *stakeholders* o quanto antes em relação ao sistema proposto

Elaborar um modelo esquemático e fácil de modificar

Realizar o mais cedo possível no processo de desenvolvimento de software com conteúdo novo e inovador



# Técnica 6: Prototipação



## *Técnica 6: Prototipação*

Explorar aspectos críticos dos requisitos de um produto, implementando de forma rápida um pequeno subconjunto de funcionalidades deste produto

Indicado para estudar as alternativas de interface do usuário; problemas de comunicação com outros produtos; e a viabilidade de atendimento dos requisitos de desempenho

Técnicas utilizadas na elaboração do protótipo são várias: interface de usuário, relatórios textuais, relatórios gráficos, entre outras



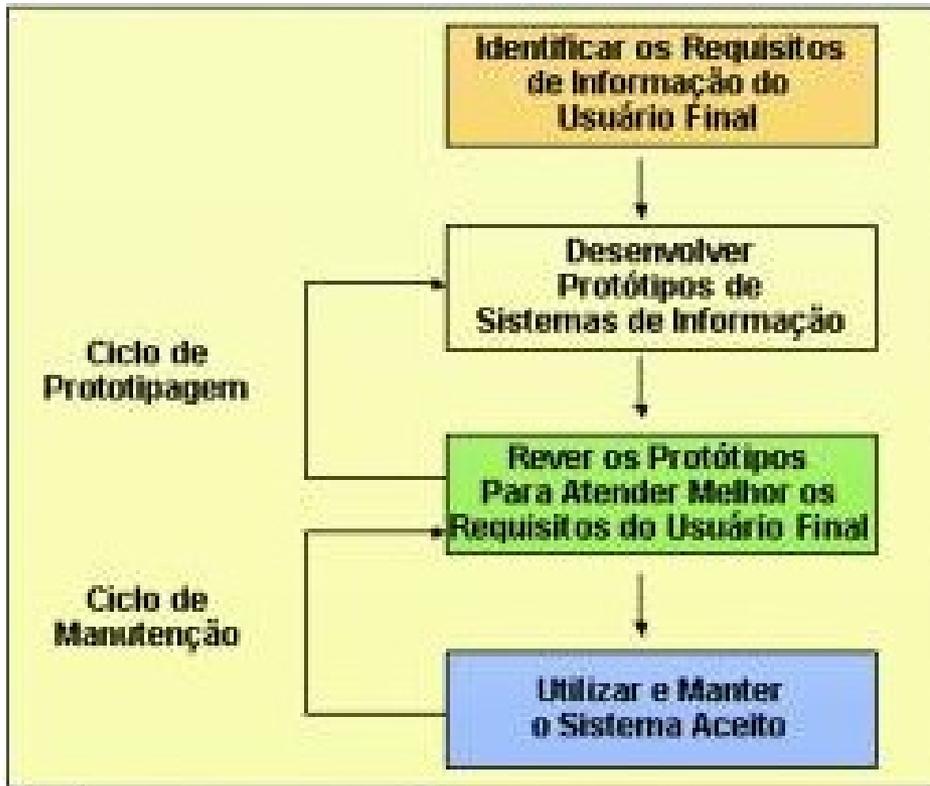
Programadores {+ Inovadores ;}

[www.x25.com.br](http://www.x25.com.br)

## Técnica 6: Prototipação

**Benefícios do protótipo:** Reduções dos riscos na construção do sistema, pois o usuário chave pode verificar se o analista captou nos requisitos do produto

### Processo de Prototipagem



Para ter sucesso na elaboração dos protótipos é necessária a escolha do ambiente de prototipagem, o entendimento dos objetivos do protótipo por parte de todos os interessados no projeto, a focalização em áreas menos compreendidas e a rapidez na construção



## Técnica 7: JAD (Joint Application Design)

- Desenvolvedores ajudam os usuários a formular problemas e explorar soluções
- Usuários ganham um sentimento de envolvimento, posse e responsabilidade com o sucesso do produto



*"People, the facts are inescapable. Any ideas on how we can ignore them?"*

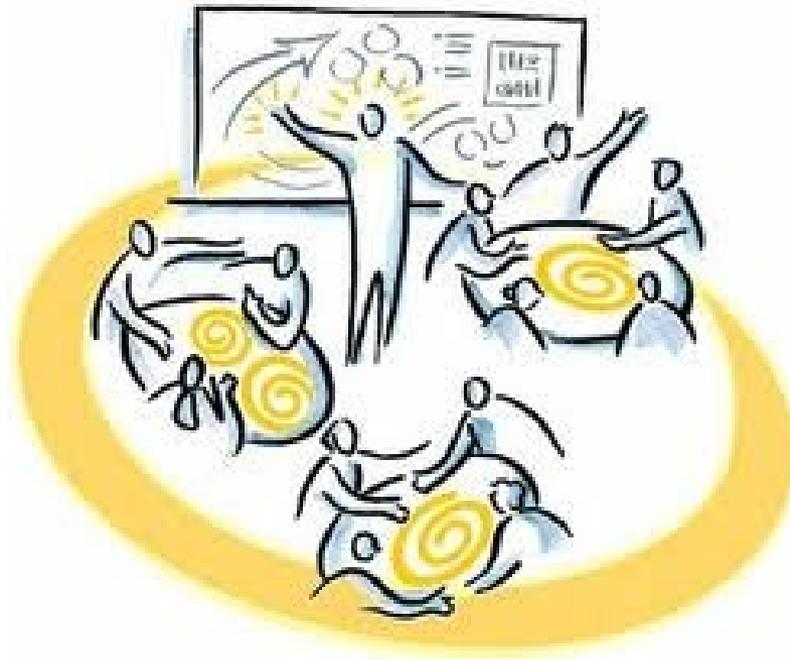


Programadores (+ Inovadores ;)

[www.x25.com.br](http://www.x25.com.br)

## *Técnica 7: JAD (Joint Application Design)*

- Promover cooperação, entendimento e trabalho em grupo entre os usuários desenvolvedores
- Facilitar criação de uma visão compartilhada do que o produto de software deve ser



## ***Técnica 7: JAD (Joint Application Design)***

- 1. Dinâmica de grupo:** realizar reuniões com um líder experiente, analista, usuários e gerentes, para despertar a força e criatividade dos participantes. O resultado final será a determinação dos objetivos e requisitos do sistema
- 2. Uso de técnicas visuais:** aumentar a comunicação e o entendimento
- 3. Manutenção do processo organizado e racional:** empregar análise *top down* e atividades bem definidas. Possibilitar a garantia de uma análise completa reduzindo as chances de falhas ou lacunas no projeto e cada nível de detalhe recebe a devida atenção
- 4. Utilização de documentação padrão:** preencher e assinar por todos os participantes. Este documento garante a qualidade esperada do projeto e promove a confiança dos participantes



# Dúvidas? Agradecimentos

**Home Page**

<http://fernandoans.site50.net>

**Blog**

<http://fernandoanselmo.blogspot.com>

**X25 Home Page**

<http://www.x25.com.br>



**Fernando Anselmo**

**[fernando.anselmo@x25.com.br](mailto:fernando.anselmo@x25.com.br)**



Programadores {+ Inovadores ;}

[www.x25.com.br](http://www.x25.com.br)