

# Engenharia de Software

Natália Schots

# Agenda

- Elicitação de Requisitos
- Técnicas para Elicitação
- Documento de Requisitos

# Elicitação de Requisitos

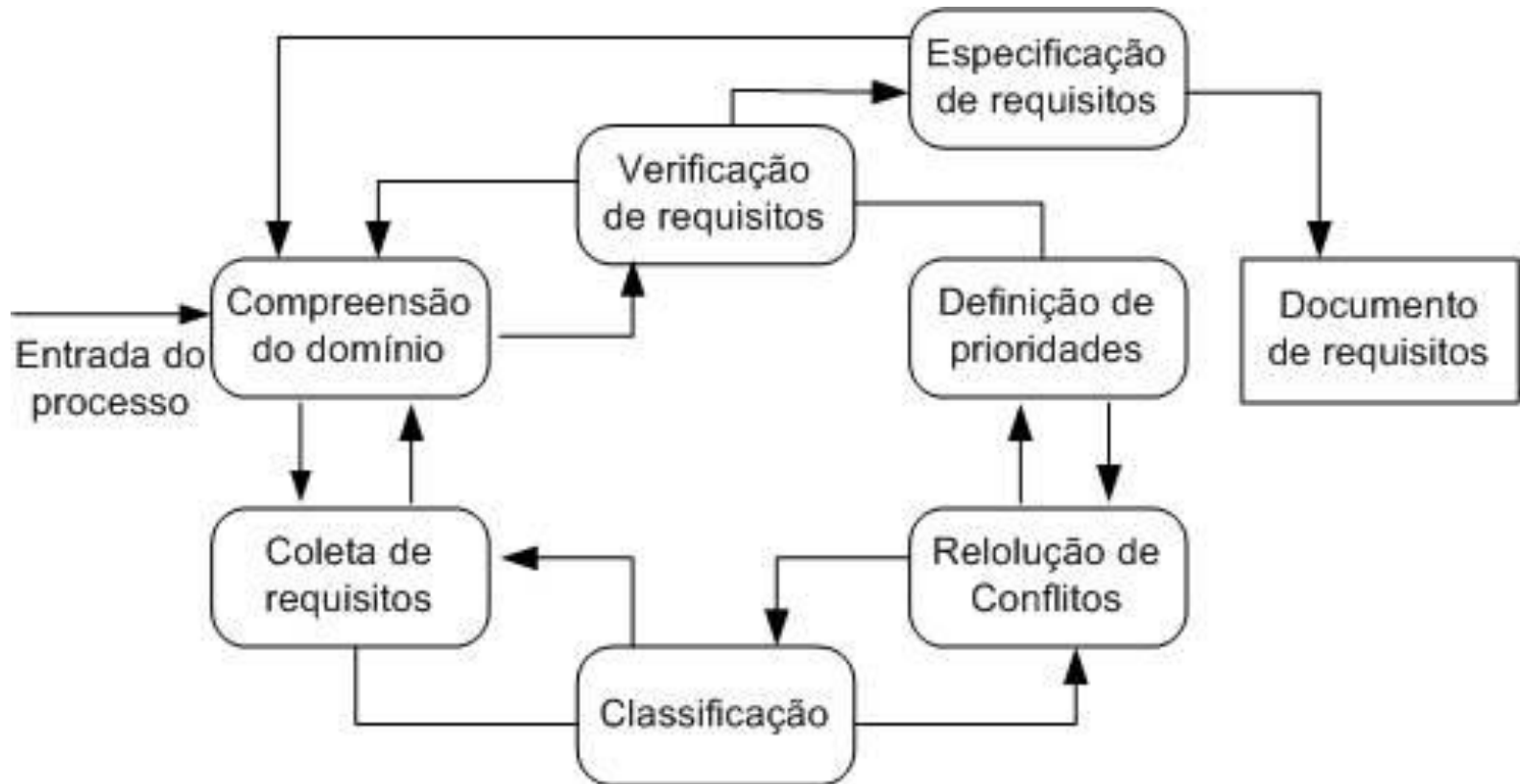
# O que é? (1/2)

- É o processo de identificação e obtenção dos requisitos do software
- Também denominado “Levantamento de requisitos”

# O que é? (2/2)

- *“A parte mais árdua na construção de um sistema de software é decidir o que construir. Nenhuma outra parte do trabalho compromete mais o sistema se for feito de forma imprópria. Nenhuma outra parte é mais difícil de corrigir a posteriori”* (Brooks, IBM - 1987)

# Atividades básicas (1/3)



[Sommerville, 2003]

# Atividades básicas (2/3)

- Compreensão do domínio
  - Os analistas devem desenvolver sua compreensão do domínio da aplicação
- Coleta de requisitos
  - É a atividade de interagir com os *stakeholders* do sistema para descobrir seus requisitos
- Classificação
  - Considera o conjunto não estruturado dos requisitos e os organiza em grupos coerentes

# Atividades básicas (3/3)

- Resolução de conflitos
  - Visa solucionar possíveis conflitos de requisitos, quando múltiplos *stakeholders* estão envolvidos
- Definição das prioridades
  - Envolve interação com os *stakeholders* para a definição dos requisitos mais importantes
- Verificação de requisitos
  - Os requisitos são verificados para descobrir se estão completos e consistentes



# Técnicas de Elicitação de Requisitos

# Técnicas mais comuns

- Existem diferentes abordagens para a obtenção de requisitos de software
  - Entrevista
  - Questionário
  - Etnografia
  - *Brainstorming*
  - Rastreamento de processo
  - Prototipação
  - JAD (*Joint Application Development*)
- Não existe uma melhor técnica
  - Cada situação exige uma abordagem específica

# Entrevista (1/7)

- “Conversas” realizadas com o usuário (entrevistado) para obter os requisitos do sistema a ser desenvolvido
- Pode ser decomposta nas seguintes atividades:
  - Ler material de suporte
  - Estabelecer os objetivos da entrevista
  - Decidir quem entrevistar
  - Preparar o entrevistado
  - Decidir os tipos de questões e a sua estrutura

# Entrevista (2/7)

- Habilidade do entrevistador
  - Fator significativo na determinação do sucesso da entrevista e obtenção de conhecimento útil
- Uma entrevista pode ser:
  - Desestruturada
    - Deseja-se explorar um problema
    - Estágios iniciais de um tópico considerado
    - Entrevista mais informal
  - Estruturada:
    - Deseja-se obter informações específicas do conteúdo e do problema

# Entrevista (3/7)

- Pode ser estruturada de três diferentes formas:
  - Estrutura em pirâmide
    - Começam com perguntas mais específicas sobre o sistema e terminam com perguntas mais genéricas
    - Geralmente utilizadas com usuários mais relutantes
  - Estrutura em funil:
    - Começam com perguntas mais genéricas sobre o sistema e terminam com perguntas mais específicas
    - Geralmente utilizadas com usuários que tem uma relação mais afetiva com o assunto

# Entrevista (4/7)

- Pode ser estruturada de três diferentes formas (cont.):
  - Estrutura em diamante
    - Combina as duas estruturas anteriores
    - É utilizada para manter o entrevistado interessado no assunto e, para isto, utiliza-se de perguntas variadas

# Entrevista (5/7)

- Tipos de perguntas
  - Abertas
    - Tendem a não ser específicas
    - Não são seguidas por alternativas
    - Encorajam resposta livre
    - Indicadas quando deseja-se conhecer o escopo do entendimento do entrevistado
    - Podem consumir muito tempo e resultar em pouca informação útil
    - É necessário estar atento ao andamento das respostas

# Entrevista (6/7)

- Tipos de perguntas (cont.)
  - Fechadas
    - Impõem limites no tipo, nível e quantidade de informação fornecida pelo entrevistado
    - Fornecem escolha de alternativas ou níveis de resposta
    - Indicadas para avaliar características específicas do problema



# Entrevista (7/7)

- Alguns benefícios
  - Fácil de ser realizada
  - Quando bem planejada, coleta informações relevantes
- Algumas desvantagens
  - Depende do planejamento e habilidade do entrevistador
  - Depende da escolha correta do entrevistado

# Questionário (1/4)

- Usado quando muitas pessoas conhecem as informações necessárias para o desenvolvimento do sistema e/ou quando os usuários estão em diversas localidades
- Deve ser preparado antecipadamente com questões objetivas (múltipla escolha) ou questões abertas

# Questionário (2/4)

- Durante a preparação do questionário
  - Identificar o tipo de informação que se deseja obter
  - Escolher um formato adequado para o questionário
  - Enviar carta acompanhando o questionário, enfatizando a importância de seu preenchimento
- As questões devem ser montadas de forma simples e concisa.
  - Cuidado com as ambiguidades!
- Caso adote questões descritivas, deixar espaço suficiente para as respostas

# Questionário (3/4)

- Recomendações:
  - Elaborar instruções detalhadas de como realizar o preenchimento correto
  - Estabelecer prazo para devolução dos formulários
  - Analisar e consolidar as informações fornecidas pelos respondentes por meio dos questionários devolvidos
  - Documentar as principais descobertas
  - Enviar uma cópia do relatório com as principais descobertas para todos os respondentes

# Questionário (4/4)

- Algumas vantagens:
  - Abrangência maior dos *stakeholders*
  - São mais eficazes para consultar *stakeholders* que se expressam melhor por escrito do que oralmente
- Algumas desvantagens:
  - Comunicação restrita com o usuário
    - Dificuldade em interpretar e abstrair corretamente as respostas
  - A preparação do questionário exige tempo e atenção
    - Perguntas mal feitas podem levar a resultados não desejados

# Etnografia (1/4)

- É uma técnica de observação que pode ser utilizada para compreender os requisitos sociais e organizacionais
  - Entender a política organizacional bem como a cultura de trabalho com objetivo de familiarizar-se com o sistema e sua história
- Utilizada como processamento e confirmação de outros resultados (entrevista e questionário)
- Deve ter aprovação antecipada do cliente

# Etnografia (2/4)

- O analista se insere no ambiente de trabalho em que o sistema será utilizado.
  - O trabalho diário é observado e são anotadas as tarefas reais em que o sistema será utilizado.
- Principal objetivo:
  - Descobrir requisitos de sistema implícitos, que refletem os processos reais (em vez de os processos formais) onde as pessoas estão envolvidas.

# Etnografia (3/4)

- É particularmente eficaz na descoberta de dois tipos de requisitos:
  - Os requisitos derivados da maneira como as pessoas realmente trabalham
    - Em vez da maneira pelas quais as definições de processo dizem como elas deveriam trabalhar
  - Os requisitos derivados da cooperação e conscientização das atividades de outras pessoas



# Etnografia (4/4)

- Pode ser vantajosa quando:
  - Os usuários não possuem disponibilidade para serem entrevistados
  - Os usuário não querem colaborar com a identificação dos requisitos
    - Podem sonegar ou distorcer informações necessárias
- Possíveis desvantagens:
  - Pode consumir muito tempo
  - O analista pode ser induzido a erros em suas observações

# *Brainstorming (1/4)*

- Técnica utilizada para encorajar a criatividade em grupo
  - Promove a geração de ideias
- Útil para obter rapidamente informações sobre a atual situação do problema
- Ajuda a um grupo a gerar tantas ideias quanto forem possível em um curto espaço de tempo
- Bastante útil principalmente para a sessões iniciais de obtenção de requisitos

## *Brainstorming (2/4)*

- Reúne pessoas com diferentes níveis de informação e conhecimento sobre o sistema
- A discussão é conduzida por um moderador
- Segue um conjunto de regras

# *Brainstorming* (3/4)

- Regras de uma sessão de *brainstorming*
  - Qualquer um pode apresentar espontaneamente uma ideia
  - As ideias devem ser relacionadas ao tópico em discussão
  - Um participante não deve expressar discordância com a ideia do outro, nem criticar a ideia ou comentar sobre a importância da mesma
  - É aceitável que um participante expanda a ideia sugerida por outro com detalhes adicionais ou ideias relacionadas

# *Brainstorming (4/4)*

- Principal vantagem:
  - Permite que os *stakeholders* cheguem a conclusões que, individualmente, não chegariam
- Principal desvantagem:
  - Depende muito da capacidade do moderador em conduzir a técnica e as pessoas, sem permitir que o foco seja perdido

# Rastreamento de processo (1/2)

- Conjunto de técnicas que permite a determinação do “processo de pensar” do indivíduo enquanto ele realiza uma tarefa ou chega a um conclusão
- O analista de requisitos registra ou grava todo o procedimento adotado para resolver o problema
  - Posteriormente, revisa os resultados junto com o especialista

# Rastreamento de processo (2/2)

- Pode ser realizado de duas maneiras:
  - Verbalização corrente
    - O especialista “pensa alto” enquanto resolve o problema
    - Enquanto realiza uma tarefa, vai relatando ao analista todos os passos realizados e o que o levou a cada etapa
  - Verbalização retrospectiva
    - O especialista verbaliza o seu processo de raciocínio logo após realizar uma determinada tarefa

# Prototipação (1/3)

- Visa explorar aspectos críticos dos requisitos de um produto, implementando de forma rápida um pequeno subconjunto de funcionalidades deste produto
- Permite aos utilizadores identificar os pontos fortes e fracos do sistema, por ser algo concreto que pode ser criticado



# Prototipação (2/3)

- Há dois tipos de protótipos:
  - Protótipos “*Throw-away*”: ajudam a esclarecer requisitos do usuário, especialmente em sistema com interface complexa
    - São descartáveis
  - Protótipos Evolutivos: ajudam o desenvolvimento rápido de uma versão inicial do sistema e suportam os requisitos bem definidos e conhecidos
    - O sistema é construído gradualmente

# Prototipação (3/3)

- Alguns benefícios:
  - Redução dos riscos no desenvolvimento do sistema
- Algumas desvantagens:
  - Custos de desenvolvimento
  - Usuário/cliente pode entender o propósito do protótipo de forma incorreta

# *JAD (Joint Application Development)*

## *(1/4)*

- É uma técnica para promover cooperação, entendimento e trabalho em grupo entre os usuários desenvolvedores
- Permite a interação entre pessoas que necessitam tomar decisões que afetam múltiplas áreas de uma organização

# *JAD (Joint Application Development)*

## *(2/4)*

- Deve ser utilizada nos casos onde existe a necessidade de consenso entre diversos usuários, pois possibilita a todos os envolvidos ter uma visão global do sistema
  - Isso ajuda a consolidar interesses de diversos usuários quanto ao sistema a ser desenvolvido
- Objetivos:
  - Aumentar o comprometimento e participação do usuário
  - Obter subsídios para elaborar o documento de Especificação de Requisitos com consenso de todos
    - Permite uma validação formal dos requisitos do sistema

# JAD (*Joint Application Development*)

## (3/4)

- Envolve três fases:
  - Adaptação: preparação para as reuniões
    - Organização da equipe, adaptação do processo JAD para ao produto a ser construído, preparação do material
  - Sessão: sessões de workshop com os participantes, envolvendo desenvolvedores e usuário, para identificar e documentar os requisitos
    - Uso de facilidades visuais, como *flipcharts* e quadro negro
  - Finalização: conversão da informação da fase de sessão em sua forma final
    - Documento de especificação de requisitos é gerado

# *JAD (Joint Application Development)*

## *(4/4)*

- Algumas vantagens:
  - Comprometimento dos usuários com os requisitos
  - Elimina requisitos de valor questionável
  - Apresenta de forma mais rápida no processo, possíveis problemas políticos
- Algumas desvantagens:
  - Pessoas chaves podem não participar
  - Depende da preparação dos participantes

# Desafios da Elicitação de Requisitos

# Dificuldades (1/3)

- Falta de conhecimento do usuário das suas reais necessidades
  - Usuário com vaga noção do que precisa e do que um produto de software pode oferecer
- Falta de conhecimento do desenvolvedor do domínio do problema
  - Desenvolvedor sem conhecimento adequado do domínio, o que leva a decisões erradas



# Dificuldades (2/3)

- Domínio do processo de obtenção de requisitos pelos desenvolvedores
  - Desenvolvedor não ouve o que os usuários têm a dizer e força suas próprias visões e interpretações
- Comunicação inadequada entre os desenvolvedores e usuários
  - Usuários incapazes de expressar suas necessidades apropriadamente
  - Significados diferentes a termos comuns

# Dificuldades (3/3)

- Dificuldade do usuário em tomar decisões
  - Falta de entendimento sobre as consequências das decisões ou sobre as alternativas possíveis
- Problemas de comportamento
  - A obtenção de requisitos é um processo social
  - Conflitos e ambiguidades nos papéis que os usuários e desenvolvedores desempenham
- Questões técnicas
  - Complexidade crescente dos sistemas atuais

# Documento de Requisitos

# Para que serve? (1/3)

- É uma declaração oficial do que é exigido dos desenvolvedores do sistema
- Utilizado pelos diferentes *stakeholders* com diferentes propósitos
  - Clientes (é isso que eu quero?)
  - Gerentes de projeto (atividades?, pessoas?, riscos?)
  - Engenheiros de sistema (o que vamos desenvolver?)
  - Engenheiros de teste (como validar estes requisitos?)
  - Engenheiros de manutenção (como manter o sistema sem alterar os requisitos?)

# Para que serve? (2/3)

- Importantíssimo para contratar o desenvolvimento de um sistema de software
- O contratante pode exigir que este documento siga determinada norma, exemplo IEEE 830-1998

# Para que serve? (3/3)

- Um erro neste documento pode ocasionar em grandes prejuízos
  - Perda da sonda Mars Polar Orbiter da NASA, custo de mais de \$100 milhões
  - Ao investigar o motivo da perda, descobriu-se um erro de comunicação dos requisitos
    - “Os dados de impulso... Foram transmitidos em libras por segundo, em vez da unidade esperada e especificada, em Newtons por segundos”

# Template IEEE 830-1998 (1/2)

- 1. Introdução
  - 1.1. Propósito do documento de requisitos
  - 1.2. Escopo do produto
  - 1.3. Definições, acrônimos e abreviações
  - 1.4. Referências
  - 1.5. Visão geral do restante do documento
- 2. Descrição Geral
  - 2.1. Perspectiva do produto
  - 2.2. Funções do produto

# Template IEEE 830-1998 (2/2)

- 2.3. Características do usuário
- 2.4. Restrições gerais
- 2.5. Suposições e dependências
- 3. Requisitos Específicos
- Apêndices
- Índice



# Boas práticas de escrita (1/2)

- Alocar tempo
  - Não pensar que a etapa de engenharia de requisitos é perda de tempo. O tempo usado na escrita é economizado futuramente
- Consistência
  - Os requisitos devem ser lidos constantemente e por várias pessoas diferentes. Todos devem interpretar as funcionalidades de forma única
- Concisão
  - Evitar o uso de adjetivos (rápido, fácil, amigável)
  - Usar frases curtas, com uma informação por frase

# Boas práticas de escrita (2/2)

- Formato e a linguagem de descrição padrão
  - Ex. emprego dos termos “deve”, “requer”, “deverá”, “deveria” e “poderá” para especificar o grau de obrigatoriedade dos requisitos
- Evite o uso de jargões
  - Ex. “o sistema deve ser tunado para maximizar a agregação de valor ao produto e assim favorecer sua relação custo/benefício”

# Referências

- Slides do Prof. Ricardo Ajax, “Engenharia de Requisitos”
- Slides do Prof. Marcelo Schots, “Engenharia de Requisitos de Software”
- Pressman, R.S., Engenharia de Software, 6ª edição, Ed. McGraw-Hill, 2006
- Sommerville I., Engenharia de software, 6º ed, São Paulo: Ed Addison-Wesley, 2003
- DevMedia, Técnicas para levantamento de Requisitos, disponível em: <http://www.devmedia.com.br/engenharia-de-software-2-tecnicas-para-levantamento-de-requisitos/>
- Koscianski, A; Soares, M. S., Qualidade de Software, Novatec Editora

Obrigada!