

Curso de Especialização em Engenharia de Software

Disciplina:

Engenharia de Software

Prof. Dr. Rodolfo Miranda de Barros

Requisitos

Definir requisitos

- **Segundo Sommerville (2007):**
 - Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento;
 - Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações;
 - O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado engenharia de requisitos (RE, do inglês *requirements engineering*).

Definir requisitos

- **Requisito (IEEE):**
 - Uma condição ou capacidade necessitada por um usuário para resolver um problema ou alcançar um objetivo;
 - Uma condição ou capacidade que deve ser satisfeita por um sistema para satisfazer um contrato ou um padrão;
- **Requisito (Aurélio):**
 - Condição necessária para a obtenção de certo objetivo, ou para o preenchimento de certo fim.

Definir requisitos

- Sommerville (2007) apresenta dois níveis de requisitos:
 - Requisitos de usuário são declarações, em uma linguagem natural com diagramas, de quais serviços o sistema deverá fornecer a seus usuários e as restrições com as quais este deve operar.
 - Requisitos de sistema são descrições mais detalhadas das funções, serviços e restrições operacionais do sistema de software. O documento de requisitos do sistema (às vezes, chamado especificação funcional) deve definir exatamente o que deve ser implementado. Pode ser parte do contrato entre o comprador do sistema e os desenvolvedores de software.
- Diferentes níveis de requisitos são úteis, pois eles comunicam informações sobre o sistema para diferentes tipos de leitor.

Definir requisitos

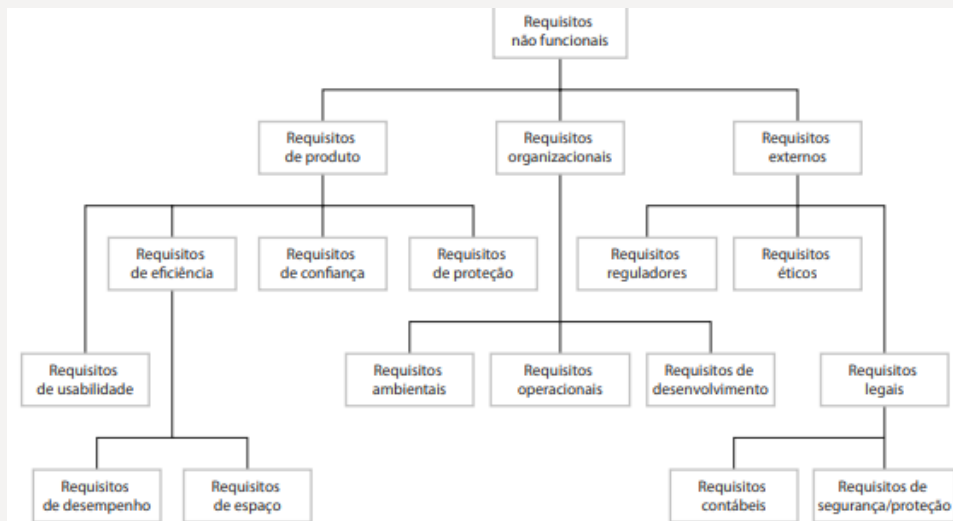
- Na definição de requisitos pode-se utilizar de uma narrativa ou de representações gráficas (linguagem natural, formal e rigorosa);
- Normalmente, as informações coletadas são fornecidas pelo usuário;
 - Entrevistas e Questionários;
 - Workshops de requisitos (brainstorming);
 - Cenários;
 - Prototipagem;
 - Observação (trabalho de Campo);
 - Análise de correlatos.
- Requisitos podem ser divididos em:
 - Requisitos Funcionais (RF);
 - Requisitos Não-Funcionais (RNF).

Requisitos Funcionais (RF)

- Os requisitos funcionais definem as funções do sistema, ou seja, o que se espera que o sistema faça;
- Eles relatam as diversas funções que deverão ser providas pelo software ao cliente ou usuário;
- Exemplos:
 - O software deve Monitorar sensores de temperatura;
 - O sistema deve Cancelar o débito na conta corrente caso a operação não seja completada;
 - O Sistema deve Avisar quando o estoque chegar no limite mínimo;
 - O aluno somente passará para a próxima pergunta se ele acertar;
 - O aluno deve tirar nota igual ou superior a 70 para ser aprovado;
 - O professor deve cadastrar um feedback para o aluno referente ao trabalho;

Requisitos Não-Funcionais (RNF)

- Requisitos não-funcionais definem as características de qualidade que o sistema deverá possuir;
- Exemplos:
 - O sistema deverá apresentar interface gráfica de acordo com a proposta da equipe de design de interface verificada e validada pelo cliente;
 - Depois que o aluno confirmar todas as respostas da prova online, o sistema deve gerar a nota e o feedback automático em, no máximo, 10 segundos;



Requisitos Funcionais (RF) - Métricas de Requisitos Não Funcionais

- **Definição de Confiabilidade:**

- É a habilidade que o software deve comportar-se consistentemente de maneira aceitável pelo usuário;

- **Requisitos de Confiabilidade:**

- Disponibilidade (xx.xx%);
- MTBF - média de tempo entre falhas (xx h);
- Máximo de erros por/KLOC (0-x) (Line Of Code);
- Erros por classe - crítico, importante, insignificante.

- **Definição de Suportabilidade:**

- É a habilidade do software para ser facilmente modificado para acomodar melhorias e correções

- **Requisitos de Suportabilidade:**

- Linguagens, DBMS, ferramentas, etc.;
- Padrões de programação;
- Tratamento de erros e relatórios padrões;

- **Muita dificuldade para especificar:**

- Se não for mensurável ou observável, ele não é um requisito
- Ele é uma restrição do projeto?
- Ele é uma intenção ou meta?

Requisitos Funcionais (RF) - Métricas de Requisitos Não Funcionais

- **Definição de Performance:**

- É a medida da velocidade ou eficiência da execução de um sistema;

- **Requisitos de Performance:**

- Capacidade;
- Processamento;
- Tempo de Resposta;
- Memória;
- Modos de degradação;
- Uso eficiente de recursos escassos: processador, memória, disco, banda da rede.

- **Definição de Usabilidade:**

- É a facilidade que o software tem para entendimento e operação das intenções dos usuários.

- **Requisitos de Usabilidade:**

- Tempo de treinamento em requisitos, tarefa de medição de tempos
- Habilidades do usuário (iniciante/avançado)
- Comparação com outros sistema que os usuários conhecem e gostam
- Sistema de ajuda online, ferramentas de dicas, necessidades de documentação
- Conformidade com padrões: Windows, guias de estilo, padrões de GUI.

Engenharia de Requisitos

Engenharia de Requisitos

- O termo ENGENHARIA implica em dizer que um processo sistemático é usado para definição de um sistema de software;
- A Engenharia de Requisitos (ER) - *Requirements Engineering* - preocupa-se com o quê (compreensão) e não como fazer;
- Uma boa ER é um passo essencial para o desenvolvimento de um bom produto;
- Um dos problemas básicos da ER é o levantamento e documentação dos requisitos dos produtos de software;
- Quando este levantamento é bem feito, os requisitos implícitos (expectativas dos clientes e usuários que serão cobradas apesar de não estarem documentadas) são minimizados.



Engenharia de Requisitos

- Quando este levantamento é bem feito, os requisitos documentados têm maiores chances de serem corretamente entendidos pelos desenvolvedores;
- Cabe aos engenheiros de software insistirem sempre na elaboração de uma boa especificação de requisitos. Faz parte do seu trabalho convencer clientes e usuários de que:
 - Boas especificações de requisitos são indispensáveis;
 - Elas não representam custos supérfluos, mas investimentos necessários;
 - A participação dos usuários na ER é fundamental para que as suas necessidades sejam corretamente atendidas pelo produto;
 - Uma boa especificação de requisitos custa tempo e dinheiro;
 - A ausência de uma boa especificação de requisitos custa muito mais tempo e dinheiro.



Engenharia de Requisitos - Motivações

- Entre as várias motivações que nos levam a estudar a ER, que variam desde razões sociais à motivos econômicos, destacamos:
 - Estudos recentes mostram que existe um grande número de sistemas de informação que não são apropriados para as necessidades de seus usuários;
 - Do ponto de vista jurídico, desejamos que o documento de requisitos funcione como um acordo contratual entre os clientes e fornecedores de software;
 - Existem também fortes razões econômicas para se estudar a ER, pois embora a etapa de análise represente um pequeno custo no desenvolvimento e manutenção de software, é de vital importância que ela seja bem realizada;
 - Estudos realizados mostram que cerca de 40% dos erros cometidos ocorrem na fase de especificação do sistema. Como eles são os mais caros de consertar, a correção de problemas oriundos da fase de especificação atingem a um patamar de 60% do custo total de correção de erros.



O que é necessário fazer



O que os usuários querem



O que os usuários pedem

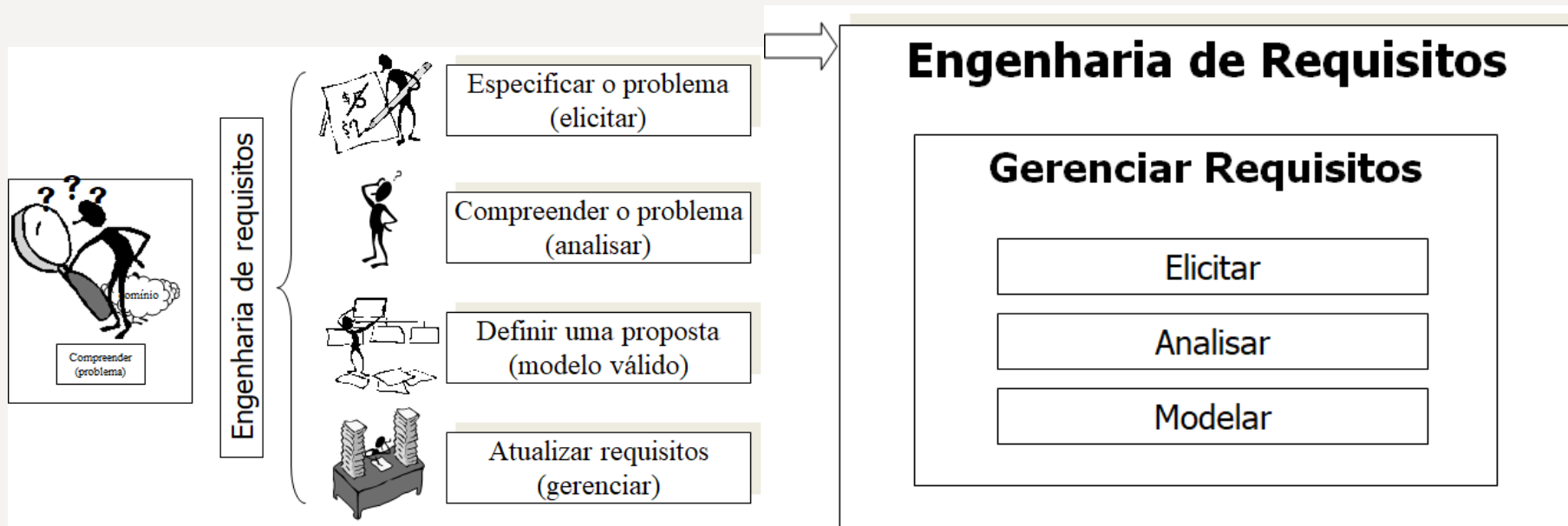


O que os desenvolvedores entendem

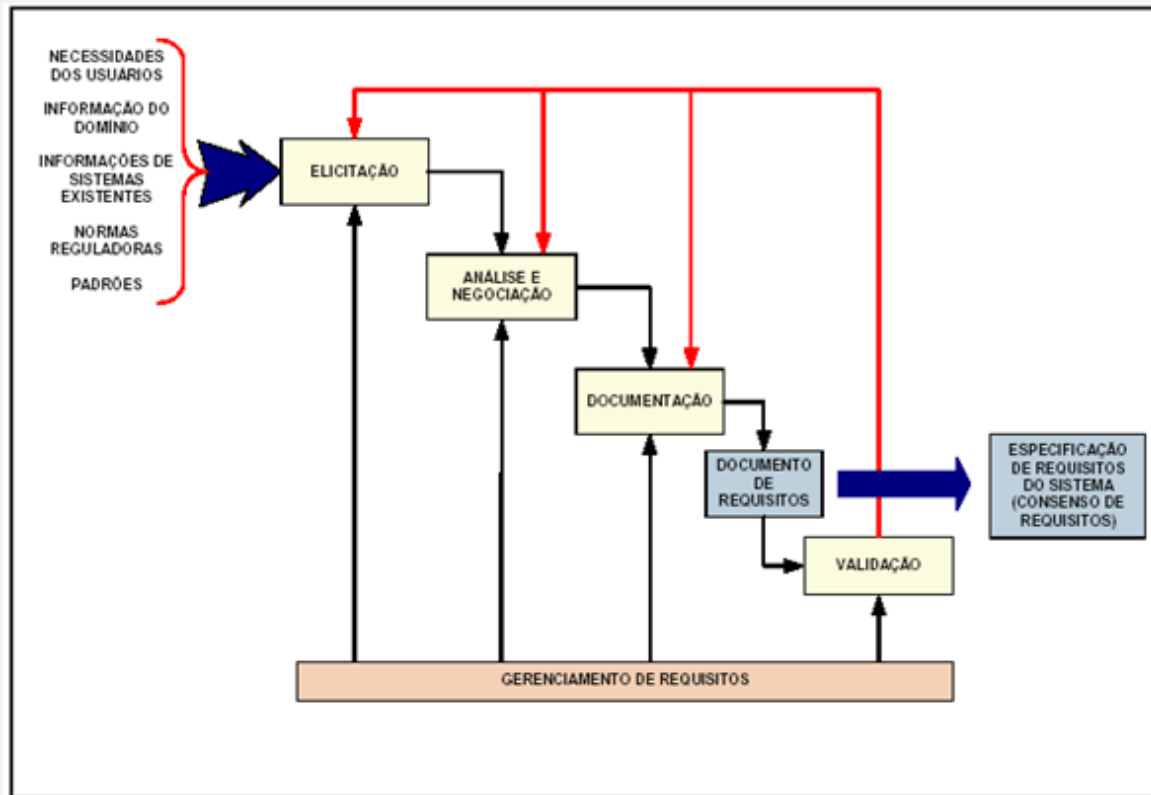


O que acaba sendo feito

Engenharia de Requisitos - Atividades



Engenharia de Requisitos - Atividades



Engenharia de Requisitos - Atividades - Elicitação

- **Elicitação:**

- Consiste na identificação dos requisitos a partir de consulta aos *stakeholders*, da análise de documentos, da análise de informações do domínio e/ou de estudos de mercado;

Engenharia de Requisitos - Identificar o Problema e a Oportunidade

Atividades - Elicitação

- Qual o problema a ser resolvido? (identificar a fronteira do problema);
- Onde está o problema? (compreender o domínio do problema e o contexto)
- De quem é o problema? (identificar os *stakeholders*)
- Porque necessita ser resolvido? (identificar os objetivos dos *stakeholders*)
- Como poderia um sistema de software ajudar? (definir cenários)
- Quando necessita ser resolvido? (identificar restrições ao desenvolvimento)
- O que nos pode impedir de o resolver? (identificar riscos)

- **Análise e Negociação:**

- Consiste na análise detalhada dos requisitos com a negociação entre os diferentes *stakeholders* visando decidir quais os requisitos que serão aceitos;
 - Classificação;
 - Resolução de conflitos;
 - Priorização;
 - Confirmação.

Engenharia de Requisitos - Atividades - Análise e Negociação

- **Classificação:**

- Agrupamento de requisitos em "módulos" para facilitar a visão global do funcionamento pretendido para o sistema;

- **Resolução de conflitos:**

- Dada a multiplicidade e diversidade de papéis das partes interessadas envolvidas na captura e análise de requisitos, é inevitável a existência de conflitos nos requisitos identificados; é importante resolver estes conflitos o mais breve possível;

- **Priorização:**

- Consiste na atribuição de uma "prioridade" a cada requisito (por exemplo elevada/média/baixa); obviamente, este pode ser um fator gerador de conflitos;

- **Confirmação:**

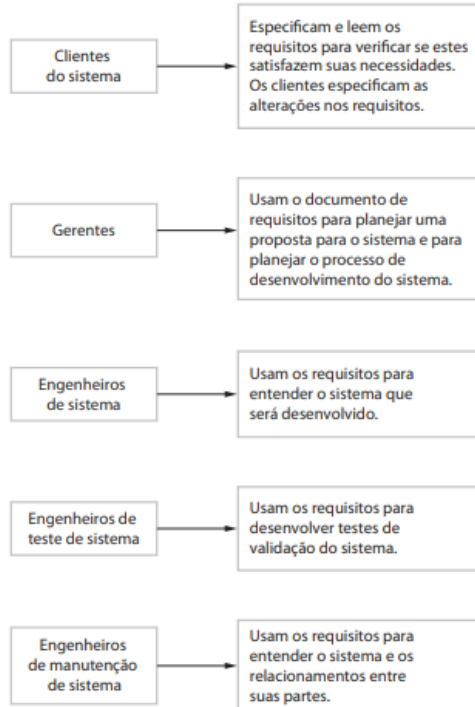
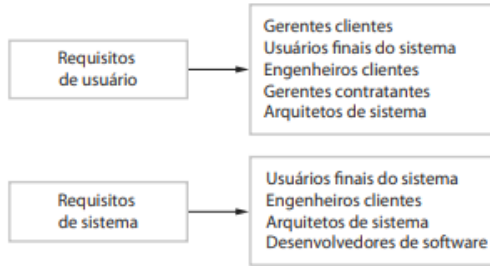
- É confirmada com as partes interessadas a completude dos requisitos, sua consistência e validade (de acordo com o que se pretende do sistema).

- **Documentação:**
 - Os requisitos aprovados na atividade de análise e negociação devem ser registrados em um documento apropriado, em nível de detalhamento adequado;

Documento de Requisitos de Software - Atividades - Documentação

- O documento de requisitos de software, às vezes chamado Especificação de Requisitos de Software é uma declaração oficial de o que os desenvolvedores do sistema devem implementar;
- Deve incluir tanto os requisitos de usuário para um sistema quanto uma especificação detalhada dos requisitos de sistema;
- Em alguns casos, os requisitos de usuário e de sistema são integrados em uma única descrição. Em outros, os requisitos de usuário são definidos em uma introdução à especificação de requisitos de sistema;
- Se houver um grande número de requisitos, os requisitos detalhados de sistema podem ser apresentados em um documento separado;
- Documentos de requisitos são essenciais quando um contratante externo está desenvolvendo o sistema de software;
- Os métodos ágeis de desenvolvimento coletam os requisitos de usuário de forma incremental e escrevem-nos em cartões como estórias de usuário. O usuário então prioriza os requisitos para implementação no próximo incremento do sistema.

Documento de Requisitos de Software - Atividades - Documentação



Definição de requisitos de usuário

1. O MHC-PMS deve gerar relatórios gerenciais mensais que mostrem o custo dos medicamentos prescritos por cada clínica durante aquele mês.

Especificação de requisitos de sistema

- 1.1 No último dia útil de cada mês deve ser gerado um resumo dos medicamentos prescritos, seus custos e as prescrições de cada clínica.
- 1.2 Após 17:30h do último dia útil do mês, o sistema deve gerar automaticamente o relatório para impressão.
- 1.3 Um relatório será criado para cada clínica, listando os nomes dos medicamentos, o número total de prescrições, o número de doses prescritas e o custo total dos medicamentos prescritos.
- 1.4 Se os medicamentos estão disponíveis em diferentes unidades de dosagem (por exemplo, 10 mg, 20 mg), devem ser criados relatórios separados para cada unidade.
- 1.5 O acesso aos relatórios de custos deve ser restrito a usuários autorizados por uma lista de controle de gerenciamento de acesso.

Estrutura de um Documento de Requisitos segundo Sommerville (2007)

Capítulo	Descrição
Prefácio	Deve definir os possíveis leitores do documento e descrever seu histórico de versões, incluindo uma justificativa para a criação de uma nova versão e um resumo das mudanças feitas em cada versão.
Introdução	Deve descrever a necessidade para o sistema. Deve descrever brevemente as funções do sistema e explicar como ele vai funcionar com outros sistemas. Também deve descrever como o sistema atende aos objetivos globais de negócio ou estratégicos da organização que encomendou o software.
Glossário	Deve definir os termos técnicos usados no documento. Você não deve fazer suposições sobre a experiência ou o conhecimento do leitor.
Definição de requisitos de usuário	Deve descrever os serviços fornecidos ao usuário. Os requisitos não funcionais de sistema também devem ser descritos nessa seção. Essa descrição pode usar a linguagem natural, diagramas ou outras notações compreensíveis para os clientes. Normas de produto e processos que devem ser seguidos devem ser especificados.
Arquitetura do sistema	Deve apresentar uma visão geral em alto nível da arquitetura do sistema previsto, mostrando a distribuição de funções entre os módulos do sistema. Componentes de arquitetura que são reusados devem ser destacados.
Especificação de requisitos do sistema	Deve descrever em detalhes os requisitos funcionais e não funcionais. Se necessário, também podem ser adicionados mais detalhes aos requisitos não funcionais. Interfaces com outros sistemas podem ser definidas.
Modelos do sistema	Pode incluir modelos gráficos do sistema que mostram os relacionamentos entre os componentes do sistema, o sistema e seu ambiente. Exemplos de possíveis modelos são modelos de objetos, modelos de fluxo de dados ou modelos semânticos de dados.
Evolução do sistema	Deve descrever os pressupostos fundamentais em que o sistema se baseia, bem como quaisquer mudanças previstas, em decorrência da evolução de hardware, de mudanças nas necessidades do usuário etc. Essa seção é útil para projetistas de sistema, pois pode ajudá-los a evitar decisões capazes de restringir possíveis mudanças futuras no sistema.
Apêndices	Deve fornecer informações detalhadas e específicas relacionadas à aplicação em desenvolvimento, além de descrições de hardware e banco de dados, por exemplo. Os requisitos de hardware definem as configurações mínimas ideais para o sistema. Requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e os relacionamentos entre esses dados.
Índice	Vários índices podem ser incluídos no documento. Pode haver, além de um índice alfabético normal, um índice de diagramas, de funções, entre outros pertinentes.

- **Validação:**

- Depois de documentados, os requisitos devem sofrer cuidadosa verificação de consistência e completitude.

- **Revisão técnica:**
 - É uma atividade de garantia de qualidade de software executada por profissionais da engenharia de software, e tem como objetivos:
 1. Descobrir erros de função, lógica ou implementação em qualquer representação do software;
 2. Verificar se o software que se encontra sob revisão atende a seus requisitos;
 3. Garantir que o software tenha sido representado de acordo com padrões predefinidos;
 4. Obter um software que seja desenvolvido uniformemente;
 5. Tornar os projetos mais administráveis.

Engenharia de Requisitos - Atividades - Validação

- **Diretrizes de uma Revisão Técnica:**

- Revise o produto, não o produtor;
- Fixe e mantenha uma agenda;
- Limite o debate e a refutação;
- Enuncie as áreas problemáticas, mas não tente resolver cada problema anotado;
- Faça anotações por escrito;
- Limite o número de participantes e insista numa preparação antecipada;
- Desenvolva uma lista de conferência para cada produto que provavelmente será revisto;
- Atribua recursos e uma programação de tempo para a revisão dentro do cronograma de desenvolvimento;
- Realize um treinamento significativo para todos os revisores;
- Reveja suas antigas revisões.

Engenharia de Requisitos - Atividades - Validação

- **A lista de conferência não pretende ser abrangente, mas, ao contrário, pretende oferecer um ponto de partida para a revisão:**
 1. As funções principais são definidas de uma forma delimitada e sem ambiguidades?
 2. As interfaces entre os elementos do sistema estão definidas?
 3. Foram definidos limites de desempenho para o sistema como um todo e para cada elemento?
 4. As restrições de projeto foram estabelecidas para cada elemento?
 5. As melhores alternativas foram selecionadas?
 6. A solução é tecnologicamente viável?
 7. Foi estabelecido um mecanismo de validação e verificação para o sistema?
 8. Há consistência entre todos os elementos do sistema?

Engenharia de Requisitos - Atividades - Validação - Classificação dos Defeitos

Quanto à gravidade:

- Críticos: defeito que reflete a ausência de um requisito essencial ou impede a utilização do produto. Não pode ser contornado, exigindo ação corretiva imediata;
- Maiores: defeito que reflete a ausência de um requisito importante ou afeta, mas não impede, a utilização do produto. Exige ação corretiva tão logo possível;
- Menores: todos os outros problemas, como erros de documentação, erros de mensagens do produto, etc. Algumas correções, como a inclusão de facilidades menores solicitadas pelos usuários, podem ser deixadas para versões futuras. Em função desses aspectos, a determinação da prioridade de correção desses erros é de responsabilidade gerencial.

Quanto à natureza:

- Omissão: ausência de uma seção, requisito, funcionalidade ou algum outro item qualquer no documento;
- Violação: incoerência de um item do documento com um outro documento do projeto elaborado anteriormente, ou ainda alguma incorreção do documento levantada por um usuário participante da revisão técnica;
- Imprecisão: requisito, seção ou funcionalidade do documento especificada de forma ambígua ou incompleta;
- Estilo: erros de português, de formatação, etc.

Resultado Final:

- Aceito sem modificações;
- Aceito com pequenas modificações: não será necessária nova revisão técnica para o resultado do projeto em pauta, um revisão informal entre membros e gerente é o suficiente);
- Rejeitado para profundas modificações: haverá necessidade de nova revisão após serem feitas as modificações sugeridas;
- Rejeitado para reconstrução: será necessária nova confecção do material;
- A revisão não foi completada: foi necessário cancelar ou interromper a reunião, e uma nova será marcada.

Engenharia de Requisitos - Atividades - Gestão dos Requisitos

- Se fosse possível realizar de forma perfeita as atividades da ER, não deveria haver necessidade de alteração nos requisitos ao longo do projeto;
- Na prática, essas alterações podem ocorrer devido a fatores externos, como alterações tecnológicas, gerenciais, legais e políticas, e a fatores internos, como o melhor entendimento do problema por parte de usuários ou dos desenvolvedores;
- Todos esses fatores fazem com que seja necessário alterar os requisitos;
- Tais alterações precisam ser conduzidas de forma ordenada para que não se perca controle sobre o prazo e o custo do desenvolvimento;

Engenharia de Requisitos - Atividades - Gestão dos Requisitos

- Os benefícios desta atividade são percebidos no médio prazo sendo que são necessários investimentos no curto prazo;
- Assim, a atividade é, muitas vezes, tida como um fardo desnecessário à condução do processo;
- Contudo, sua não implementação faz com que as economias de curto prazo sejam logo suplantadas pelas despesas no longo prazo, verificadas com superação de custo e prazo nos projetos conduzidos;
- Considerando-se o forte impacto que as alterações de requisitos geralmente têm nos prazos e custos dos projetos, elas devem obedecer a um procedimento documentado, estabelecido de Gestão de Requisitos;

Engenharia de Requisitos - Atividades - Gestão dos Requisitos

- Alterações nos requisitos só podem ocorrer em marcos do projeto que sejam definidos e acordados entre clientes, usuários, a equipe do projeto e outros grupos afetados;
- O cadastro dos requisitos do software deve ser usado para localizar todas as alterações consequentes em outros artefatos do projeto;
- Os impactos nos compromissos devem ser avaliados pelas pessoas afetadas, pesquisando-se todas as associações pertinentes guardadas no Cadastro dos Requisitos;
- Devem-se comunicar de forma muito clara os impactos nas alterações nos requisitos ao cliente e usuários, requerendo-se a concordância destes em absorver esses impactos;

Engenharia de Requisitos - Atividades - Gestão dos Requisitos

- As alterações devem ser revistas pelos grupos envolvidos, e, em casos de alterações maiores, submetidas a revisões técnicas;
- As alterações devem ser revistas e aprovados pelos usuários chaves, pelo cliente, pelo gerente do projeto e pela gerência executiva;
- Devem ser produzidas novas versões da ER do Software e dos artefatos alterados em consequência, incorporando-se as alterações no respectivo texto;
- A nova versão da ER do Software, a nova versão do cadastro dos requisitos e demais artefatos devem ser incorporados em nova linha de base do projeto, ou em alteração da linha de base corrente.

Engenharia de Requisitos - Atividades - Gestão dos Requisitos

- Ao longo do projeto, deve ser mantido atualizado o status dos requisitos;
- O status reflete as modificações solicitadas e aprovadas: um requisito pode ser excluído, adicionado ou alterado;
- A alteração de um requisito pode se referir ao seu conteúdo (um detalhe de um caso de uso ou um valor de uma métrica de um requisito não-funcional);
- Também pode se referir a um de seus atributos (um requisito pode ser rebaixado de essencial para desejável);
- As alterações devem ser documentadas visando subsidiar futuras solicitações de alterações de requisitos nesse ou em outros projetos similares.

Engenharia de Requisitos - Atividades - Gestão dos Requisitos

No.	Nome do Requisito	Tipo	Importância	Complexidade	Estabilidade
1	Interface de usuário Tela de Abertura do Caixa	Interface	Essencial	Baixa	Média
2	Caso de Uso Abertura do Caixa	Caso de Uso	Essencial	Baixa	Média
3	Caso de Uso Pedido de Compra	Caso de Uso	Opcional	Baixa	Baixa
4	Caso de Uso Operação de Venda	Caso de Uso	Essencial	Alta	Média
5	Requisito desempenho tempo de resposta	Não-funcional	Desejável	Baixa	Alta

No.	Nome do Requisito	Tipo	Importância	Complexidade	Estabilidade	Status
1	Interface de usuário Tela de Abertura do Caixa	Interface	Essencial	Baixa	Média	Original
2	Caso de Uso Abertura do Caixa	Caso de Uso	Essencial	Baixa	Média	Original
3	Caso de Uso Pedido de Compra	Caso de Uso	Opcional	Baixa	Baixa	Alterado
4	Caso de Uso Operação de Venda	Caso de Uso	Essencial	Alta	Média	Original
5	Requisito desempenho tempo de resposta	Não-funcional	Desejável	Baixa	Alta	Original

Data	Item alterado	Tipo da Mudança	Motivo	Impacto
10/03/2010	Caso de Uso Pedido de Compra	Alteração no fluxo	Falha na Especificação de Requisitos	3 dias de atraso; mais 15 horas de esforço
12/03/2010	Interface de usuário Emissão de Notas Fiscais	Alteração no leiaute	Mudança na legislação	Mais 2 horas de esforço

Casos de Uso

- **Casos de Uso:**

- Técnica que visa auxiliar a documentar os requisitos do sistema de uma forma que seja, ao mesmo tempo, compreensível pelos *stakeholders* e descritos em um nível próximo a uma linguagem computacional.

Estrutura de um Casos de Uso

- Apesar dos casos de uso serem escritos em linguagem natural, português por exemplo, eles seguem uma sintaxe própria, como uma sequência de passos, o que os torna parecidos com um algoritmo;
- Uma sequência de passos lógicos que permite realizar uma ação.
- Um caso de uso especifica o comportamento de um sistema ou de parte de um sistema, e é uma descrição de um conjunto de sequências de ações;
- Os casos de uso são usados para captar o comportamento pretendido do sistema que está sendo desenvolvido, sem entrar em detalhes de implementação, podendo assim ser utilizado por clientes, programadores, arquitetos e engenheiros de *software*, e etc.;

Estrutura de um Casos de Uso

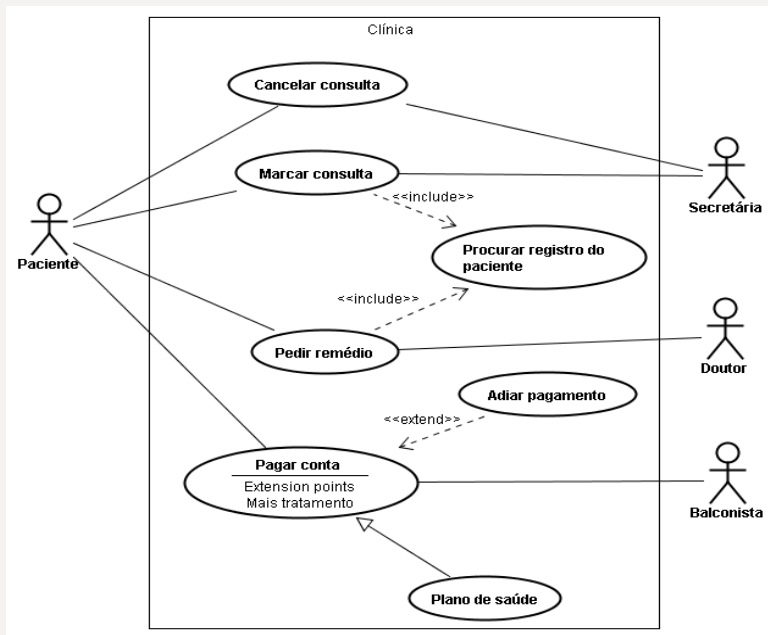


Diagrama de Casos de Uso

Caso de uso: Finalizar Compra

Referências:
RF_F2

Descrição Geral:

O caso de uso inicia-se quando cliente deseja efetuar compra dos produtos que estão inseridos no carrinho de compras.

Atores:
Cliente.

Pré-condições:

Cliente logado no sistema, produtos já inseridos no carrinho de compras.

Garantia de sucesso (Pós-condições):

Pedido fechado, compra efetuada, sistema aguardando confirmação de pagamento.

Requisitos Especiais:

Fluxo Básico:

1. Cliente deseja finalizar compra, sistema solicita que informe a forma de pagamento e de entrega.
2. Cliente deseja efetuar pagamento em forma de cartão crédito/débito.
3. Sistema solicita informações do cartão do cliente.
4. Sistema faz validação das informações.
5. Sistema gera o número do pedido.
6. Compra finalizada com sucesso.

Fluxo Alternativo:

1. Cliente deseja efetuar pagamento através de boleto bancário.
 1. Sistema gera o boleto para o cliente. Retorna ao passo 5.

Descrição de um Caso de Uso

Histórias de Usuários

Escrevendo Histórias de Usuários

- As histórias de usuário são uma técnica de **elicitação de requisitos**, comumente utilizadas em métodos ágeis descrevendo as principais funcionalidades do ponto de vista de um usuário ou cliente de um sistema ou *software*;
- Histórias de usuário são escritas por ou para usuários ou “clientes de negócio” como uma maneira primária para influenciar a funcionalidade do sistema sendo desenvolvido;
- Histórias de usuário podem também ser escritas por desenvolvedores para expressar requisitos não-funcionais (segurança, desempenho, qualidade, etc.);

Escrevendo Histórias de Usuários

- Quando chega a hora de criar Histórias de usuário, um dos desenvolvedores se reúne com um representante do cliente ou a equipe multidisciplinar se reúne para criar as Histórias;
- Pode-se usar uma série de perguntas para obter a meta do “representante do cliente”, como perguntar sobre a conveniência de alguma funcionalidade em particular, mas deve tomar cuidado para não dominar o processo de criação da ideia.

Escrevendo Histórias de Usuários

- As Histórias de Usuário geralmente são levantadas em cartões(podem ser manuscritos);
- Cada História deve ter algum significado para o cliente;
- Descrições curtas de uma pequena parte da funcionalidade desejada, escritas no idioma do usuário.
- Histórias de Usuário geralmente são utilizadas em conjunto com métodos, como o *Scrum* e o XP, e são base para o planejamento das iterações;
- É comum que cada História de Usuário possua uma prioridade de implementação que pode ser definida por custo, importância para o negócio, riscos e relações de dependência entre Histórias;
- As Histórias de Usuário devem manter o máximo de independência de outras Histórias de Usuário;

Escrevendo Histórias de Usuários

- Exemplo:
 - “Eu como autor necessito saber a quantidade de livros de minha autoria que foram vendidos para que eu possa cobrar minha comissão junto à editora”.
- Na frase acima temos três coisas bem definidas:
 - Um papel ou ator: o autor;
 - Uma necessidade ou caso de uso: saber a quantidade de livros vendidos;
 - Um motivo: cobrar a comissão junto à editora.

Escrevendo Histórias de Usuários - Template

ID:	Título:	Prioridade:
Ação:		
Anotações:		
Aceitação:		

Escrevendo Histórias de Usuários - INVEST

- **Independente:** significa que essa história não possui dependência com nenhuma outra, podendo ser implementada individualmente;
- **Negociável:** a história deve ser escrita de maneira a permitir um nível de negociação, onde os envolvidos chegam a um consenso de quão “profunda” será sua implementação;
- **Valiosa** para o cliente ou usuário: uma história deve representar um item que agregue algum valor ao produto, e esse valor deve ser perceptível para o cliente ou usuário. É importante lembrar que o foco é desenvolver sempre o que possui mais valor para o cliente, pois afinal ele é quem é o principal interessado no produto (e quem vai pagar a conta);

Escrevendo Histórias de Usuários - INVEST

- **Estimável:** uma história deve ser escrita com um nível de detalhe tal que permita ao Time de Desenvolvimento realizar uma estimativa quanto ao tempo necessário para sua implementação. Na história de exemplo, de nada adiantaria que a história fosse escrita da seguinte maneira “Eu como autor necessito de um relatório para saber o valor de minha comissão.”. Perceba como existem detalhes ausentes, até mesmo em relação à base sobre a qual a comissão será calculada.
- **Pequena** (Small em inglês): uma história deve ser pequena ao ponto de poder ser realizada em uma sprint. Histórias grandes geralmente carregam um alto grau de incerteza. Geralmente o Time de Desenvolvimento ajuda o Product Owner a decompor uma história muito grande em histórias menores;
- **Testável:** deve ser claro para todos os envolvidos como aquela história deve ser testada, de forma a garantir o seu funcionamento. Os testes de aceitação, escritos no verso do cartão, por exemplo, costumam garantir essa propriedade das histórias.

REFERÊNCIAS

Pressman, Roger S. Engenharia de Software - Uma Abordagem Profissional, Amgh Editora, 8ª Ed., 2016.

SOMMERVILLE, I. Engenharia de Software, Addison-Wesley Brasil, 8ª Edição, 2007.

Laboratório GAIA – <http://www.gaia.uel.br>. Visitado em: 20/12/2017.

BOM ESTUDO!

Em caso de dúvidas, não deixe
de perguntar aos tutores.



UNIVERSIDADE
Estadual de Londrina