



Universidade Federal do Ceará
Campus de Crateús
Sistemas de Informação & Ciência da Computação

ENGENHARIA DE REQUISITOS [CRT0393]
TRABALHO EM DUPLA #3

INCLUA ABAIXO DE CADA QUESTÃO A RESPECTIVA RESPOSTA
ENVIAR O DOCUMENTO EM FORMATO PDF

Prof. Allysson Allex Araújo
allysson.araujo@crateus.ufc.br
<http://crateus.ufc.br/allysson>

Antonia Stéfanne Sousa Alves - 415317 - Sistemas de Informação
Francisco Hartur Lopes de Alcântara - 403897 - Ciência da Computação

1) Por que é importante documentar o resultado da elicitação? Por que não partir direto para a Análise de Requisitos?

Porque nem toda sessão de elicitação gera informação para que possa fazer algo relevante de análise, às vezes são necessárias várias sessões de elicitação para ter massa crítica de informação para trabalhar na análise. Além disso, nem sempre as atividades de elicitação e análise estão sob a responsabilidade da mesma pessoa. E nem sempre uma única pessoa realiza todas as atividades de elicitação.

A razão para não partir direto para análise é que todo produto da análise está fundamentado nas informações levantadas na elicitação, se esta não estiver documentada, questões levantadas sobre a proposta de solução da análise podem ser difíceis de responder ou mesmo difíceis de verificar ou validar.

2) Com base na sua experiência com entrevistas passadas, liste os três maiores fatores na execução da entrevista que percebeu terem causado inibição no entrevistado. Se você ainda não realizou nenhuma entrevista, considere entrevistas realizadas por outros e que você tenha presenciado.

- O entrevistador não ouve o entrevistado, passa a impressão de desespero e nervosismo quanto a entrevista, não olha face a face com entrevistado e não anota o que o entrevistado fala ou expressões/gestos que transmite.
- O entrevistador leva uma solução pronta para o cliente, e não escuta de fato as suas necessidades.
- O entrevistado é tímido, pouco falante, ou não entende sobre as regras de negócio, então o entrevistador não consegue provocar o entrevistado, para que ele fale mais sobre o sistema/regras de negócio.
- O entrevistado traz uma solução pronta para seu problema, e o entrevistador tem que desconstruir a solução do entrevistado, para focar nos seus problemas e necessidades. (e muitas vezes a solução que o próprio cliente, não resolveria de fato seu problema)
- Uso de um gravador, o entrevistador solicitou uma prévia autorização, o entrevistado aprovou, porém houve uma certa inibição durante a entrevista.

3) Qual é o objetivo da Análise de Requisitos? E da Gerência de Requisitos? Qual a diferença entre essas fases?

Definir e elaborar uma representação do escopo do projeto, seja via especificação ou modelos. O produto principal da análise é a especificações de requisitos. Se o objetivo é definir um escopo de forma ampla, a especificação deve abordar os processos de negócio impactados e o ambiente no qual o software estará inserido. Se o objetivo é definir um escopo de forma aprofundada, a especificação deve

descrevê-lo em termos de cada tarefa realizada pelo software ao usuário com as respectivas regras de negócio aplicadas a elas.

O objetivo do gerenciamento de requisitos é assegurar que uma organização documente, verifique e atenda às necessidades e expectativas de seus clientes e partes interessadas internas e externas. O gerenciamento de requisitos começa com a análise e levantamento dos objetivos e restrições da organização. Ele inclui ainda o planejamento de apoio para os requisitos, requisitos de integração e a organização para trabalhar com eles (atributos para requisitos), bem como relacionamentos com outras entregas de informações sobre em relação aos requisitos, e mudanças para estas. O gerenciamento de requisitos envolve a comunicação entre os membros da equipe do projeto e as partes interessadas, e a adequação às mudanças de requisitos ao longo do curso do projeto.

A diferença dos dois é que o gerenciamento de requisitos é um processo de documentação, análise, rastreamento, priorização e concordância sobre os requisitos e então o controle de mudanças e comunicação às partes envolvidas (stakeholders) relevantes. É um processo contínuo durante um projeto.

Enquanto a análise de requisitos se concentra nas tarefas que determinam as necessidades ou condições para atender o produto ou projeto novo ou alterado, levando em consideração os requisitos possivelmente conflitantes das várias partes interessadas, analisando, documentando, validando e gerenciando software ou requisitos de sistema

4) Qual a diferença entre a verificação e a validação de requisitos? Exemplifique duas diferentes técnicas que podem ser utilizadas para apoio.

Verificação (Fizemos o software corretamente?) - Envolve a análise do software para verificar se este está dentro dos padrões, e cumprindo os requisitos funcionais e não funcionais. A verificação prepara os requisitos para a validação, além disso a verificação também pode ser realizada para especificação de sistemas, para avaliar se os requisitos estão sendo documentados como deveriam e ainda prever falhas ou inconsistências entre requisitos.

Técnicas para verificação:

- **Controle de questões** - Se existem questões pendentes, logicamente o levantamento de requisitos não foi concluído, nem a análise de requisitos. Logo o software não tem como atender a todos os requisitos funcionais e não funcionais.
- **Matriz de rastreabilidade** - A matriz de rastreabilidade explicita a relação direta dos requisitos entre si ou com os outros componentes do projeto. Ajuda a verificar se existem requisitos de mais alto nível que ainda não foram

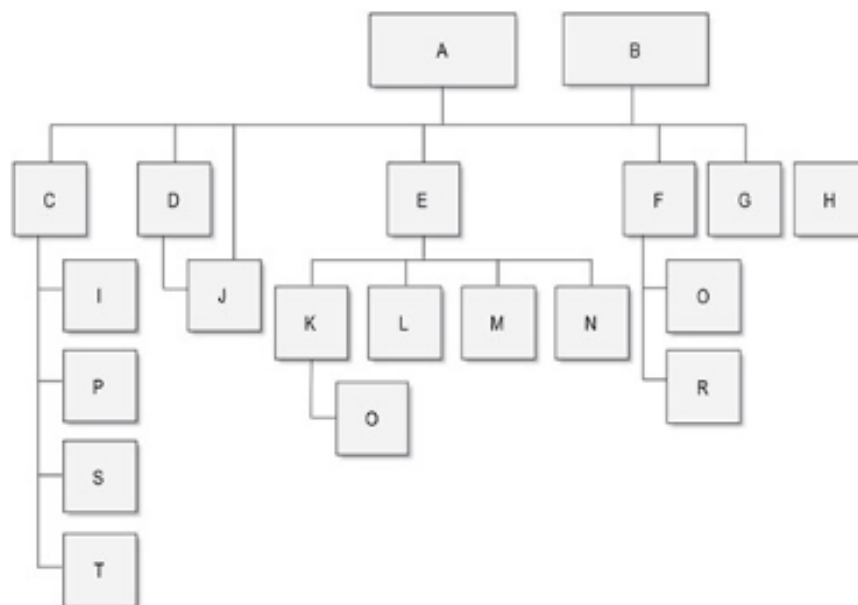
tratados, além de detectar requisitos que estão sobrando, ou seja, que não deveriam fazer parte do projeto. Ela é importante para garantir que, mesmo com alterações no escopo, o projeto consiga atender aos requisitos estabelecidos.

Validação (Fizemos o software correto?) - Analisa se o software está cumprindo as expectativas e necessidades do cliente, a implementação deve atender a regra de negócio do cliente, a validação exige a participação do cliente, já a verificação não.

Técnicas para validação

- **Protótipos** - Apresentar uma proposta (visual) concreta do que será entregue como produto, o cliente consegue avaliar a solução proposta. É onde provavelmente se obtém um feedback mais rico do quão correta é a solução proposta.
- **Checklist** - É um conjunto de perguntas e/ou afirmações sobre determinado cenário. Essas perguntas facilitam a identificação de problemas da especificação. Diferente da checklist de verificação que é respondida pela equipe do projeto, a checklist da validação é respondida pela parte interessada.

5) A figura a seguir representa a decomposição funcional de um sistema genérico. Verifique se a decomposição foi feita corretamente. Caso encontre problemas, liste-os.



É incorreto iniciar a decomposição com mais de uma função. O natural seria a partir de uma função, para que a partir daí suas funções sejam decompostas em partes menores. Assim, deveria haver apenas uma função no topo inicialmente.

A função “J” é subcomponente de “D”, por isso deveria estar ligada a apenas esta função. Não se deve conectar uma função a dois diferentes. A função “H” está desconectada das demais funções. Assim, não podemos dizer que ela faz parte da decomposição funcional.

6) Utilizando o estudo de caso do Anexo como referência, especifique duas histórias do usuário. Elabore-as de acordo com o formato sugerido na lição: “como (papel), eu quero (algo) para (me beneficiar)”.

Como **profissional de obras**, quero consultar a lista de obras pendentes para poder emitir registro de responsabilidade técnica.

Como **fiscal de obras**, quero consultar obras com possíveis irregularidades para tornar minha fiscalização mais analítica e menos operacional.

Como **gestor dos fiscais**, eu desejo consultar as estatísticas das ações realizadas para poder aumentar a eficiência do sistema.

Como **dono do projeto**, preciso que os profissionais declarem responsabilidade de obras, a fim de obter um aproveitamento melhor da minha equipe de fiscais.

Como **gestor responsável da SUPOPE**, quero poder definir parâmetros de metragem e data para definir quais obras são passíveis de registro de responsabilidade.

7) Por que é importante rastrear requisitos?

A rastreabilidade auxilia na gerência do projeto. Com ela podemos garantir se os requisitos estão sendo abordados no produto, se o requisito atende o que propõe e descobrir inconsistências entre os requisitos, assim garantido uma especificação de requisitos correta, completa e modificável.

A rastreabilidade facilita a análise de uma mudança, tornando-a mais rápida e fácil, e permitindo uma percepção mais profunda no âmbito e complexidade da mudança. Além disso, também ajuda a estimar variações em cronogramas e custos do projeto. Como requisitos com muitas relações apresentam mais riscos, a rastreabilidade também ajuda no aspecto de gestão de riscos.

8) Proponha um questionário para levantamento de requisitos com pelo menos 8 itens para cada um dos sistemas a seguir:

a) Sistema web (e-commerce) para venda de produtos de informática

1. Para realizar uma compra, o cliente precisaria estar logado no sistema?
2. O pagamento seria feito no próprio sistema, ou o cliente teria que entrar em contato com o estabelecimento para efetuar o pagamento?
3. Como os produtos iriam ser dispostos, através de departamentos?
4. Em uma página, como seria a organização dos produtos, em grade, lista, teriam botões de organização, como organizar por preço, destaques?
5. Teria filtros de busca no sistema?
6. Ao efetuar um pedido, o cliente poderia editar um pedido?
7. Quais seriam os status do pedido do cliente (pedido feito, aguardando pagamento, etc..)? Até que status o cliente poderia cancelar o pedido?
8. A venda é varejo, ou atacado? Tem um número máximo de pedidos de um mesmo produto em uma mesma compra?

b) Sistema de controle de horas complementares da UFC

1. Quem seriam os usuários do sistema? (Pode marcar mais de uma opção)
☐ Alunos
☐ Coordenadores
☐ Secretária
☐ Outros
2. O que são horas complementares, são atividades? precisam ser comprovadas? Quais as formas de comprová-las?
3. Cada atividade teria uma quantidade de horas complementares fixas, ou depende de cada atividade? quem determina a quantidade de horas?
4. O sistema precisa receber arquivos, tais como pdfs, imagens?
5. Caso o aluno fosse um usuário do sistema, este submeteria comprovantes de atividades, e receberia o retorno sobre a aceitação das horas complementares?

6. Quem faria a validação das horas complementares?
7. O sistema ofertaria alguma tela, onde o usuário poderia verificar seu desempenho em relação às horas complementares, como quantas horas já entregou, quantas ainda faltam entregar?
8. O usuário teria a opção de excluir ou editar horas complementares que já foram consolidadas?

c) Aplicativo de agendamento para salão de beleza/barbearia.

1. Quem seria os usuários do sistema, apenas os funcionários dos estabelecimentos poderiam marcar o agendamento, ou os clientes também?
2. Para realizar um agendamento, o usuário teria que realizar login no aplicativo?
3. O orçamento seria feito em contato direto com cliente ou funcionário (depende de quem está fazendo o agendamento), ou no aplicativo teria valores pré-determinados?
4. Caso o cliente possa utilizar o sistema, o aplicativo necessita de alguma função de comunicação direta com o estabelecimento, como um chat?
5. O usuário teria a possibilidade de desmarcar o agendamento pelo aplicativo, teria um tempo mínimo de antecedência para poder desmarcar o agendamento?
6. Após um usuário realizar um agendamento de um procedimento, ele poderia editar o procedimento a ser realizado, teria um horário mínimo de antecedência para editar o agendamento?
7. Se o estabelecimento tiver diferentes funcionários para um mesmo procedimento, o usuário tem a opção de escolher com qual funcionário vai realizar o procedimento?
8. Os horários disponíveis para agendamento, seriam dispostos em um calendário mensal/semanal ou seriam em forma de lista?

9) Suponha que você foi contratado para estruturar a engenharia de requisitos de uma empresa desenvolvedora de software sob demanda para terceiros.

Como você estruturaria esse processo? Além da explicação textual em detalhes das atividades, inclua um diagrama de atividades demonstrando o processo e uma possível lista de verificação (checklist) para orientar a análise de requisitos.

Considerando que a empresa pode desenvolver softwares para terceiros de diferentes contextos, Então um processo genérico de levantamento e análise que contém as seguintes atividades:

- Compreensão do domínio: Os analistas devem desenvolver sua compreensão do domínio da aplicação;
- Coleta de requisitos: É o processo de interagir com os stakeholders do sistema para descobrir seus requisitos. A compreensão do domínio se desenvolve mais durante essa atividade;
- Classificação: Essa atividade considera o conjunto não estruturado dos requisitos e os organiza em grupos coerentes;
- Resolução de conflitos: Quando múltiplos stakeholders estão envolvidos, os requisitos apresentarão conflitos. Essa atividade tem por objetivo solucionar esses conflitos;
- Definição das prioridades: Em qualquer conjunto de requisitos, alguns serão mais importantes do que outros. Esse estágio 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 e se estão em concordância com o que os stakeholders desejam do sistema.

O levantamento e análise de requisitos é um processo iterativo, com uma contínua validação de uma atividade para outra, conforme exemplificado pela Figura 1.

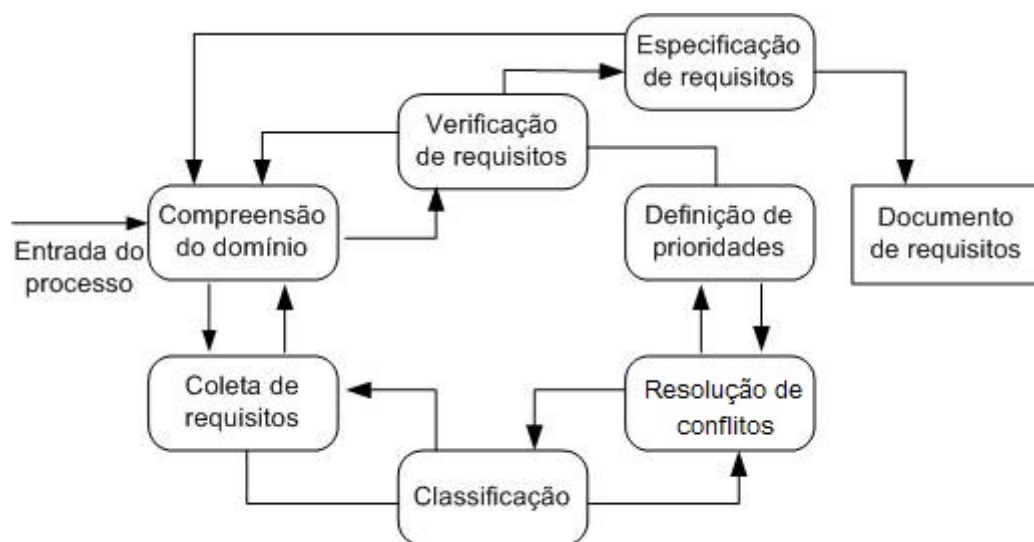


Figura 1. Processo de levantamento e análise de requisitos.

Um possível Modelo de Lista de Verificação (Checklist) de Requisitos seria:

ITEM	ITEM PARA VERIFICAÇÃO	SIM	NÃO
Não ambíguo: É não ambíguo se, e somente se, cada requisito declarado seja suscetível a apenas uma interpretação.			
1	Cada requisito está descrito com clareza, concisão e sem ambiguidade?		
Consistente: É consistente se, e somente se, nenhum dos requisitos do documento, tomado individualmente, está em conflito com qualquer outro requisito do mesmo documento.			
2	Existem requisitos conflitantes?		
Completo: É completo se, e somente se, conter toda e apenas a informação necessária para que o software correspondente seja produzido.			
3	Existem requisitos implícitos?		
4	Os requisitos exibem a distinção clara entre funções, dados e restrições?		
5	As restrições e dependências foram claramente descritas?		
6	Existem requisitos que contêm algum nível desnecessário de detalhe do projeto?		
7	Os requisitos definem todas as informações a serem apresentadas aos usuários?		
8	Os requisitos descrevem as respostas do sistema ao usuário devido às condições de erro?		
9	Existem situações não tratadas pelos requisitos que precisam ser consideradas?		
10	O documento contém realmente toda a informação prometida em sua introdução?		

Figura 2 - Modelo de Lista de Verificação (Checklist)

Vale lembrar que o que muda na abordagem de checklist de verificação para o de validação é que provavelmente as listas de itens a verificar são distintas nos dois casos. Além do fato de que, na verificação, quem executa a lista é a equipe do projeto e na validação são as próprias partes interessadas chave.

10) Leia o artigo [“Supporting Requirements Engineering Research that Industry Needs”](#) e faça uma análise interpretativa de no mínimo 15 linhas (Arial, 12).

Depois de 40 anos a Engenharia de Requisitos tornou-se uma disciplina independente, respeitável e estável. Porém ainda há uma certa dificuldade em aplicá-la corretamente nas empresas, sempre há algum problema que causa o fracasso nos projetos, e parte do fracasso se dá na ER. Mas apesar da ER já ser uma disciplina independente, os esforços dos pesquisadores em melhorá-la não

param. E com isso, entra o desafio em realizar “pesquisas que a indústria precisa”, ou seja, pesquisas que ajudem a indústria a resolver problemas que os profissionais encontram. Mas qual a dificuldade? Por que ainda é difícil a luta contra a ER? Faltam contribuições científicas adequadas que visem resolver os problemas dos profissionais ou faltam conhecimentos adequados e transferência de tecnologia entre a investigação e a prática? Provavelmente ambos. A ER é uma desafio tanto na prática quanto na sua investigação. A aplicação da ER é dada principalmente por fatores humanos, como experiências e conhecimentos, e esses fatores afetam a escolha de métodos e abordagens. Então o que pode funcionar para uma empresa, pode não funcionar para outra. Tornando difícil a investigação da ER sobre o que de fato funciona. Um dos problemas é que os pesquisadores continuam a propor soluções (como uma fórmula mágica) para problemas (únicos) que não são entendíveis por eles, onde a empresa pode considerar a solução cegamente, sem o conhecimento dos seus benefícios e deficiências. É necessário que os pesquisadores entendam os problemas que os profissionais enfrentam em seus projetos, para poder traçar um caminho de sucesso. Com esses problemas, nasceu o NaPiRE, onde seu objetivo é entender as práticas e problemas da ER na indústria, para ajudar a orientar a pesquisa aos diferentes contextos da indústria. A pesquisa resumiu 21 problemas relatados pelos entrevistados e sua frequência. O problema mais frequente são os “requisitos incompletos”, seguidos por “falhas de comunicação entre a equipe do projeto e o cliente”, “requisitos sub específicos” e “alvos móveis”. Para propor soluções é necessário obter uma compreensão mais profunda sobre os problemas, capturando as causas e consequências do problema, então a pesquisa levantou algumas, tais como: recursos insuficientes, barreiras linguísticas, interpretações subjetivas, falta de experiência dos membros da equipe de ER, terminologia pouco clara, falta de tempo, falta de engajamento do cliente, entre outros. Cada um relacionado a uma área, como problemas de Entrada, de métodos, da organização, de pessoas ou de ferramentas. A maioria dos fatores são causados por problemas organizacionais ou sociais. Foi feita uma pesquisa entre pequenas empresas até grandes empresas, que utilizam metodologias ágeis, e todas relataram o problema de falhas de comunicação entre a equipe de projeto e o cliente, o que é curioso pois um dos pilares do manifesto ágil é colaboração com cliente, acima de negociação de contratos. Por fim, chegamos a conclusão de que não se pode associar soluções universais, para necessidades específicas, pois cada organização tem uma realidade distinta, e mais ainda a teoria da ER, não está em sintonia com a realidade industrial hoje.

11) Leia o Capítulo 1 do SWEBOK (Requirements Engineering) e faça um resumo de no mínimo duas laudas sobre o conteúdo. Existe uma versão traduzida neste [link](#), entre a página 31 e página 43.

Nota de consideração: Utilizamos a versão traduzida fornecida na questão, mas percebemos que a tradução não é totalmente fiel ao capítulo original.

O capítulo 1 do livro SWEBOK trata sobre as áreas de conhecimento de Requisitos de software, como elicitação, análise, especificação e validação dos requisitos de software. A má condução dessas atividades tornam projetos de engenharia de software criticamente vulneráveis. Além disso, os requisitos de Software expressam as necessidades e restrições colocadas sobre um produto de software que contribui para a solução de um problema do mundo real. A área Requisitos de Software é fortemente relacionada a: Design do Software; Teste de Software; Manutenção do Software; Gerência de Configuração do Software; Gerência da Engenharia de Software; Processo de Engenharia de Software; Qualidade de Software.

- Definição de Requisito de Software - Propriedade que deve ser apresentada pelo software para resolver um problema do mundo real. Requisito de Software porque ele se ocupa com problemas endereçados pelo Software: Software é desenvolvido ou adaptado para resolver um problema em particular. Requisitos de software são tipicamente uma combinação de requisitos de diversas pessoas em diferentes níveis de organização e de ambientes no qual o software opera.
- Requisitos de Produto e Processo - Requisitos de Produto: Requisitos sobre o software a ser desenvolvido. Requisitos de Processo: são essencialmente restrições impostas ao desenvolvimento do software tais como linguagem de programação, processos e técnicas de desenvolvimento. Podem ser implícitos ou explícitos. Podem ser impostos pela organização desenvolvedora, pelo cliente ou por terceiros.
- Requisitos Funcionais e Não Funcionais - Requisitos Funcionais (capabilities): Descreve funções que o software deve executar como por exemplo “controlar fluxo de caixa”. Requisitos Não Funcionais (restrições ou requisitos de qualidade): São aqueles que agem para restringir uma solução.
- Propriedades Emergentes - São alguns requisitos que representam propriedades emergentes do software, isto é, propriedades que não são endereçadas por um componente mas cuja satisfação depende da inter-operação de diversos componentes do sistema. Propriedades emergentes são crucialmente dependentes da arquitetura do sistema.
- Requisitos Quantificáveis - Requisitos precisam ser definidos com clareza e sem ambiguidade, o quanto for possível, e, quando for o caso, quantitativamente; Evitar definições vagas ou que não possam ser verificadas.

Requisitos de Sistema e Requisitos de software -

- Sistema: Uma combinação de elementos interagindo para obter um determinado objetivo. Inclui software, hardware, pessoas, informações, técnicas, serviços e outros elementos de suporte.
- Requisitos de Sistema: Requisitos para o sistema como um todo.
- Requisitos de Software: Num sistema que possui componentes de software, requisitos de software são derivados dos requisitos do sistema e alocados ao software;
- Requisitos do Usuário: O guia define requisitos do usuário como sendo requisitos dos usuários do sistema ou usuários finais.

Processo de Requisitos

- Modelo de processo: Não é uma atividade discreta mas cobre desde o início até o final do projeto sendo refinado continuamente durante todo o ciclo de vida; Identifica os Requisitos de software como itens de configuração e os gerencia usando as práticas de GC. Customizado de acordo com o contexto da Organização e do Projeto Relaciona-se com atividades de Elicitação, Análise, Especificação e Validação. Inclui atividades de Pesquisa de Mercado e Factibilidade.
- Atores de Processo: Define os papéis dos participantes do processo; Interdisciplinar: Especialistas de requisitos fazem mediação entre os especialistas de domínio e demais interessados (stakeholders);

Elicitação de Requisitos - Preocupa-se com o “de onde” podem ser obtidos os requisitos de software e como os engenheiros de software podem coletá-los; Constitui o primeiro estágio para definir e entender o problema que o software pretende solucionar; É uma atividade onde os stakeholders são identificados e são estabelecidas relações entre equipe de desenvolvimento e o cliente; Pode ser conhecida por outros nomes: “captura de requisitos”, “descoberta de requisitos” ou “aquisição de requisitos”; Um princípio fundamental da boa Engenharia de software: boa comunicação entre usuários do software e os engenheiros de software. Deve-se esforçar para isso. Analistas devem mediar entre o domínio do usuário e o domínio técnico.

- Fontes de Requisitos - Principais fontes a serem utilizadas: Metas ou objetivos, Conhecimento do domínio, Stakeholders, Ambiente operacional e Ambiente organizacional.
- Técnicas de Elicitação - Identificadas as fontes de requisitos o engenheiro de software deve elicitar os requisitos, a elicitação não é uma atividade passiva, o engenheiro de software geralmente precisa trabalhar duro para elicitar as informações corretas. Principais técnicas de elicitação utilizadas: Entrevistas, Cenários, Protótipos, Reunião com facilitadores, Observação.

Análise de Requisitos - Processo de analisar requisitos para: Detectar e resolver conflitos entre requisitos; Descobrir as fronteiras do software e como ele deve interagir com o seu ambiente; Elaborar requisitos do sistema para derivar requisitos do software.

- Classificação de Requisitos - A classificação é feita segundo diversos critérios: Funcional ou Não Funcional; Derivado de requisito de alto nível, propriedade emergente ou imposta diretamente sobre o software por um stakeholder ou qualquer outra fonte; Produto ou Processo; Prioridade; Escopo; Volatilidade/Estabilidade.
- Modelagem Conceitual - Seu propósito é auxiliar no entendimento do problema antes de iniciar o projeto da solução. Consequentemente, modelos conceituais compreendem modelos de entidades do domínio do problema configurado para refletir relacionamentos e dependências do mundo real.
- Projeto de Arquitetura e Alocação de Requisitos - Projeto da arquitetura é o ponto onde o processo de requisitos sobrepõe-se com o projeto do software ou do sistema, e ilustra como é impossível desacoplar com clareza essas duas tarefas. A alocação de requisitos é a atribuição a componentes, da responsabilidade por satisfazer os requisitos.
- Negociação de Requisitos - Resolver problemas decorrentes de conflitos entre requisitos; Dois stakeholders podem requerer features mutuamente incompatíveis; Conduzida por especialistas de requisitos; Evita-se decisão unilateral: stakeholders envolvidos são consultados em busca de consenso.

Especificação de Requisitos

- Documento de Definição do Sistema - Registra, num alto nível, os requisitos do sistema; • Destina-se aos clientes e usuários do sistema (ou seus representantes); • Define requisitos funcionais e não funcionais do sistema em alto nível, seus objetivos gerais, o ambiente alvo, suas restrições e pressupostos;
- Especificação de Requisitos do Sistema - Aplica-se a sistemas que possuem número considerável de componentes (software e não software) e frequentemente separa-se a descrição dos requisitos do sistema da descrição dos requisitos de software. Assim os requisitos de sistema são especificados e os requisitos de software são derivados deles e então os requisitos de componentes do software são especificados;

- Especificação de Requisitos de Software - Estabelece base para acordo entre clientes, contratadores ou fornecedores; • Define o que o produto de software deve ser e o que ele não deve ser; • Para leitores não técnicos, o documento de especificação de requisitos de software é frequentemente acompanhado por um documento de definição dos requisitos de software;

Validação de Requisitos - Documentos devem ser revisados por stakeholders diferentes, inclusive por representantes do cliente e do desenvolvedor; É normal explicitar um ou mais pontos no processo de requisitos onde a validação é realizada. Isso ajuda prevenir alguns problemas antes que recursos sejam alocados; Diz respeito ao processo de examinar os documentos de requisitos para ter certeza que eles definem o software correto, ou seja, o software que faz o que o usuário espera que ele faça.

- Revisões de Requisitos - Revisões devem ser constituídas quando for completado o Documento de Definição do Sistema, Documento de Especificação do Sistema e Documento de Especificação de Requisitos de Software, ou em qualquer outro ponto do processo; Pode ser útil o fornecimento de checklists para auxiliar.
- Prototipagem - Meio de validar a interpretação que o engenheiro de software faz do requisito; Meio para eliciar novos requisitos.
- Validação do Modelo - É necessário validar a qualidade dos modelos desenvolvidos durante a análise; Se for utilizada especificação formal, é possível utilizar raciocínio lógico para provar propriedades das especificações.
- Testes de Aceitação - Propriedade importante dos requisitos. O produto deve ser validado nos seus requisitos:
 - Requisitos que não possam ser validados são meros “desejos”; Para serem validados, eles precisam primeiro ser analisados até o ponto de serem expressos quantitativamente; • Pode ser difícil projetar Testes de Aceitação para os Requisitos Não Funcionais;

Considerações Práticas - Visão Simplista: A decomposição de subáreas apresentada é uma simplificação do processo, na verdade ele se expande por todo o ciclo de vida do software.

- Natureza Iterativa do Processo de Requisito
 - Mercado atual pressiona indústria para ciclos curtos de desenvolvimento;

- Alguns projetos são desenvolvidos adaptando projetos anteriores ou aproveitando partes deles;
- É quase sempre impraticável implementar requisitos como um processo linear e determinístico;
- Processo de requisitos não é tarefa front-end mas expande-se por todo o ciclo de vida do software.
- Gerência de Mudanças
 - A gerência de mudanças é central no gerenciamento de requisitos, ela descreve o seu papel, os procedimentos que precisam ser efetuados e a análise que deve ser aplicada às propostas de mudanças e que está intimamente ligada à Gerência de Configuração de software.
- Atributos de Requisitos
 - Requisitos não são constituídos apenas pela especificação do que é requerido, mas também por um conjunto de informações adicionais que auxiliam a interpretar e gerenciar os requisitos.
- Rastreamento de Requisitos
 - Tem a ver com as fontes dos requisitos e com os efeitos sobre os seus descendentes. Ele é fundamental para fazer análise de impacto quando o requisito é modificado.
- Medição de Requisitos
 - É útil para avaliar o “tamanho” de uma mudança de requisitos, na estimativa de custo de desenvolvimento, tarefa de manutenção ou simplesmente para ter um denominador comum de comparação com outras medições.

12) Questão cuja pontuação será referente às práticas do projeto de sistema para To Do para apoio ao gerenciamento de atividades de alunos.

Práticas realizadas:

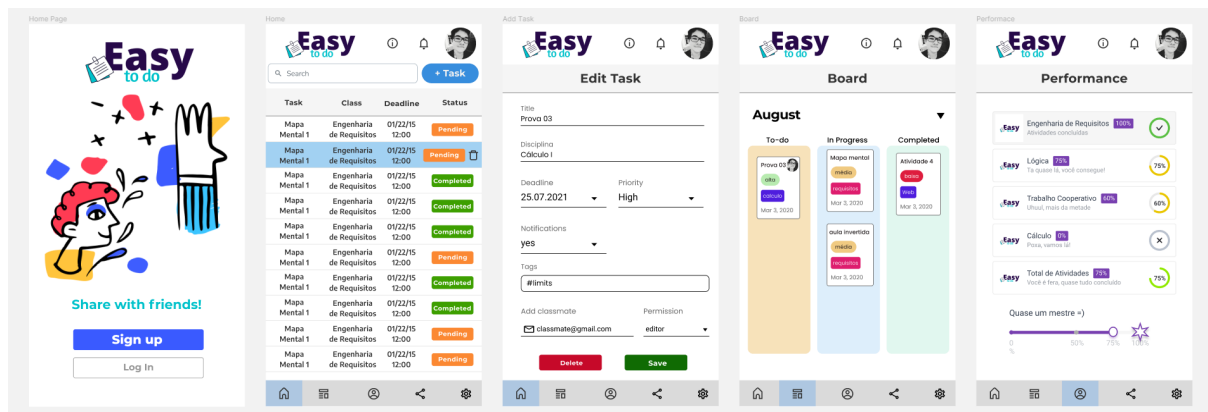
- 1. Entrevista.
- 2. Documento de Visão.
- 3. Documento de Requisitos de Software (DRS).
- 4. História de usuários.
- 5. Protótipo de Baixa Fidelidade.

Trabalho feito com a equipe: Stéfanne Sousa, Hartur Alcântara e Matheus Sampaio.

- 1. Entrevista.
- 2. Documento de Visão.
- 3. Documento de Requisitos de Software (DRS).

Trabalho feito com a equipe: Robéria, Hartur Alcântara, Stéfanne, Antônio Francisco, Lucas Lima.

- 4. História de usuários.
- 5. Protótipo de Baixa Fidelidade.



Referências

https://en.wikipedia.org/wiki/Requirements_engineering

https://en.wikipedia.org/wiki/Requirements_analysis

https://en.wikipedia.org/wiki/Requirements_management

<https://imasters.com.br/devsecops/utilizacao-de-checklist-para-validacao-de-requisitos-de-software>

https://www.academia.edu/31449888/TMP08_engenharia_de_software_9_edicao

<https://blog.vinco.com.br/levantamento-de-requisitos-de-software/>

<https://www.phase2technology.com/blog/successful-requirements-gathering>

<https://opencadd.com.br/engenharia-de-requisitos-conheca-todas-as-etapas-do-processo/>

<https://www.devmedia.com.br/tecnicas-para-levantamento-de-requisitos/9151>