



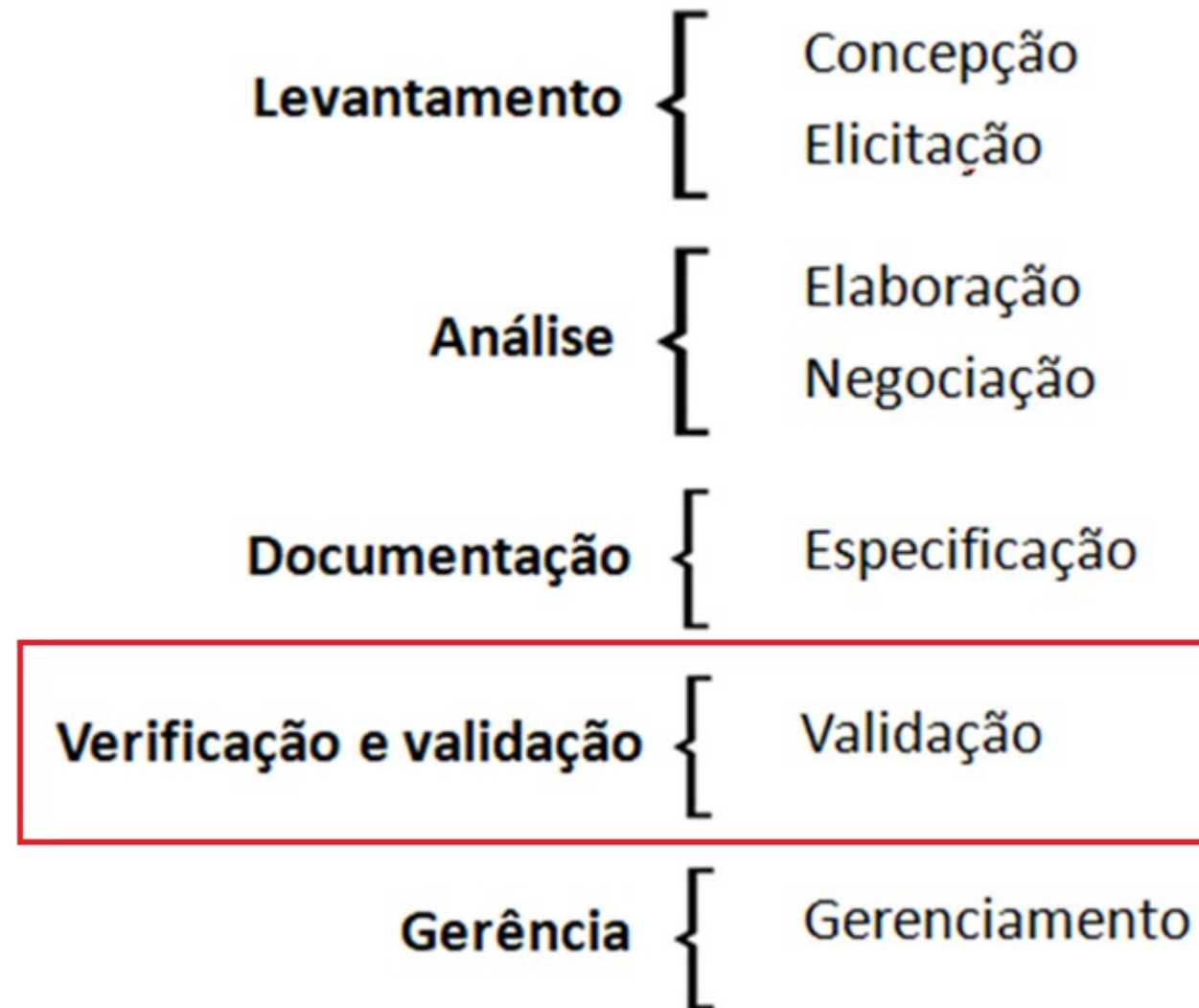
Engenharia de requisitos

Maria Eduarda Lavina

Verificação e validação de requisitos



Processos da engenharia de requisitos





Verificação e validação

A verificação e validação de requisitos são processos que buscam verificar se um sistema atende aos requisitos especificados e se estes atendem à necessidade do cliente.

A etapa de verificação e validação se torna importante pois erros em requisitos podem gerar altos custos de retrabalho quando descobertos durante o desenvolvimento ou após o sistema já estar em serviço.

Verificação de requisitos: O produto está sendo construído corretamente? O software está de acordo com sua especificação?

Validação de requisitos: O produto construído é o correto? Os requisitos especificados atendem à necessidade do cliente?



Verificação e validação

A verificação de requisitos busca assegurar que o software está sendo especificado da forma correta, seguindo padrões e critérios de qualidade definidos, enquanto que a validação busca assegurar que o software especificado é o correto, ou seja, se este atende às necessidades dos clientes.

Existem vários padrões internacionais para os processos e documentação de V&V em software, podendo citar o Padrão IEEE 1012-2012 - Padrão IEEE para Verificação de Software e Validação - , que fornece algumas diretrizes para ajudar a preparar planos de V&V, sendo que este modelo pode ser personalizado para a maioria das situações.

Verificação e validação

Verificação

Esta etapa busca garantir que os requisitos especificados e seus modelos atendam ao padrão de qualidade, para serem utilizados de forma eficaz nas etapas seguintes do projeto.

Na verificação de requisitos é comparado o produto modelado e especificado com modelos de referência e são adotadas ações para apontar os itens que não estão em conformidade, a fim de justificar ou corrigir estes itens.

A verificação de requisitos é uma atividade interna à equipe de projeto e não envolve o cliente. Para realizar a verificação de requisitos recomenda-se que uma pessoa diferente do analista que especificou os requisitos.



Verificação e validação

São efetuados diversos tipos de verificação dos requisitos no documento de requisitos:

Verificações de validade: identifica se o requisito proposto atende às necessidades de todos os usuários do sistema proposto.

Verificações de consistência: não deve haver conflitos entre os requisitos do documento, não podem existir contradições ou descrições diferentes da mesma função do sistema.

Verificações de completude: o documento de requisitos deve incluir requisitos que definam todas as funções e as restrições pretendidas pelo usuário do sistema.

Verificação e validação

Verificações de realismo: os requisitos devem ser verificados quanto à viabilidade de implementação, podendo considerar o orçamento , equipe, cronograma, tecnologias, entre outros pontos.

Verificabilidade: os requisitos do sistema devem ser passíveis de verificação para poder reduzir conflitos entre cliente e contratante, ou seja, deve ser possível escrever testes que demonstrem que o sistema atende a cada requisito especificado.

Adequação da informação: verificar se as necessidades de informação para o modelo ou requisito em análise estão atendidas de maneira completa. Por exemplo, ao analisar um diagrama de atividades, verificar se o mesmo apresenta os componentes corretamente, de acordo com a linguagem UML.

Verificação e validação

Elementos omissos em modelos: cada modelo deve ser comparado com outros modelos existentes no projeto, buscando por elementos que são mencionados em um modelo e que estão omissos em modelos que apresentam o dado sob outras perspectivas. Por exemplo, em um diagrama de atividades existe a etapa de aprovação de agendamentos, porém não existe um requisito funcional para esta atividade.

Terminologia: Deve-se verificar se a terminologia utilizada para um mesmo conceito de negócio está consistente em todos os modelos e em todo o documento. A terminologia usada deve ser compreensível para todas as partes interessadas. O glossário é uma ótima ferramenta para esse fim.



Verificação e validação

Cenários alternativos: para os requisitos funcionais, deve ser verificado se todas as alternativas de fluxos foram devidamente especificadas.

Compreensão do requisito: o requisito deve estar expressado de forma que o leitor consiga compreender a necessidade especificada.

Requisitos de software

Os requisitos devem atender aos atributos de qualidade:

Atributo	Descrição
Completo	Todas as funcionalidades solicitadas foram documentadas
Consistente	Os requisitos não devem possuir conflitos e contradições
Correto	O requisito é correto quando satisfaz uma determinada necessidade
Sem ambiguidade	O requisito deve possuir apenas uma interpretação
Verificável	Devem ser descritos de forma que seja possível verificar a sua implementação
Compreensível	Deve ser escrito levando em consideração vários tipos de leitores
Modificável	Deve permitir mudanças sem um grande impacto
Rastreável	A origem de um requisito deve estar claramente identificada



Verificação e validação

Validação

A validação de requisitos busca assegurar que todos os requisitos especificados estejam alinhados com os requisitos de negócio e que todas as necessidades dos clientes sejam satisfeitas.

Tem como finalidade garantir que a especificação defina o produto certo a ser desenvolvido pelo projeto.

Verificação e validação

Técnicas de V&V

Controle de questões: se ainda existem questões em aberto para serem discutidas, é válido supor que a especificação de requisitos pode não estar completa.

Geração de casos de teste: De modo que os requisitos devem ser testáveis, os casos de teste podem revelar problemas em requisitos.

Testes baseados nos requisitos funcionais ou derivados dos requisitos do usuário ajudam a validar os comportamentos esperados do sistema. A própria atividade de projetar testes pode revelar problemas com os requisitos.



Verificação e validação

Critérios de aceite:

Os critérios de aceite determinam as circunstâncias em que o cliente irá aceitar o resultado final do projeto. São critérios com o que se pode medir e comprovar a finalização de um trabalho.

Critérios de aceite definem as condições mínimas para que uma funcionalidade seja considerada pronta, conforme os objetivos de negócio, e podem estipular itens como condições legais, regulatórias ou contratuais específicas que devem ser atendidas pelo sistema.

Definir critérios de aceite com o cliente fornece uma maneira de validar tanto os requisitos quanto a própria solução.



Verificação e validação

São características dos critérios de aceite:

- Devem ser testáveis, e os resultados dos testes não devem deixar margem para interpretação. O teste de aceitação deve produzir uma resposta clara para aprovação ou reprovação do resultado.
- Os critérios devem ser claros e concisos.
- Todos os envolvidos no projetos devem compreender os critérios de aceite propostos.
- Devem fornecer uma perspectiva do usuário, sendo que este deve ser capaz de testá-los e aprová-los/reprová-los.

São definidos no SRS e podem também ser considerados como parte de acordos contratuais com os clientes.



Verificação e validação

Revisões técnicas/Inspeções:

Os requisitos são analisados sistematicamente por uma equipe de revisores que verifica erros, inconsistências e se a especificação consegue transmitir a mensagem pretendida pelo autor.

Uma inspeção é composta pelas fases de planejamento, visão geral, detecção de erros, correção de erros, acompanhamento e reflexão. No processo de validação de requisitos, as fases de planejamento, visão geral, detecção de erros e coleta de erros são as mais relevantes.

Planejamento: são definidos o objetivo da inspeção, os resultados esperados do trabalho de revisão e os papéis e participantes da inspeção.

Verificação e validação

Visão Geral: nesta fase, o autor explica aos membros da equipe os requisitos a serem revisados, para que haja um entendimento comum sobre a exigência entre todos os revisores.

Detecção de erros: os revisores buscam por erros nos requisitos. A detecção de erros pode ser realizada individualmente ou por uma equipe, de forma colaborativa.

Coleta de erros: todos os erros identificados são coletados, consolidados e documentados. Durante a consolidação, são identificados os erros que foram encontrados várias vezes ou erros que não são realmente erros. Os erros identificados e as medidas de correção são documentados em uma lista de erros.

Verificação e validação

Checklists (listas de verificação)

Definem uma lista padrão de itens que são verificados para identificar problemas na especificação.

É uma técnica usada tanto na verificação quanto na validação pelo cliente. O que difere o *checklist* de verificação para o de validação é a lista de itens a serem verificados, sendo que o *checklist* de validação pode ser enviado ao cliente para que este responda aos itens. Podem ser utilizados em processos de revisão.

Uma lista de verificação e validação de requisitos contém perguntas ou assertivas encadeadas sequencialmente que facilitam a identificação de erros, pois os requisitos são avaliados com base em cada item presente no *checklist*.



Verificação e validação

Deve-se atentar para a forma como as questões são formuladas, pois questões muito genéricas ou abstratas podem dificultar o uso da lista de verificação.

Por exemplo: **“O requisito está formulado adequadamente?”**

Esta questão pode levar a diversas respostas diferentes, dependendo do que a pessoa que está respondendo o *checklist* considera um requisito devidamente especificado.



Verificação e validação

Protótipos:

Nesta técnica, um modelo do sistema em questão é demonstrado para os usuários finais e clientes. Estes podem experimentar ou analisar o modelo para verificar se ele atende a suas reais necessidades. Com o protótipo, a parte interessada pode avaliar com mais atenção a solução proposta e proporcionar um *feedback* mais rico do quão correta é a solução proposta.

Podem ser criados protótipos descartáveis, que não são mantidos depois de utilizados, ou protótipos evolutivos, que serão melhorados em etapas posteriores.

Verificação e validação

Para a implementação de um protótipo, os requisitos a ser representados devem ser selecionados, sendo que o conjunto de requisitos a ser validado pelo protótipo é limitado por recursos de desenvolvimento, como tempo, dinheiro, equipe, etc. Pode-se utilizar como critério de seleção a criticidade de um requisito.

Protótipos de baixa fidelidade: são protótipos simples, baratos e rápidos de desenvolver e representa visualmente as funcionalidades. Podem ser utilizados para validar conceitos, informações e ideias. Tem como objetivo definir de modo simples como seria a interação do usuário com uma funcionalidade, sem se preocupar com o design.

Verificação e validação

A hand-drawn sketch of a sign-in form on a notepad. The form is titled "Sign In" in a box at the top left. Below the title, there are two input fields: "Sign in name" and "Password". The "Password" field has a link "(I forgot my password)" next to it. Below the password field, there is a checkbox labeled "Next time auto sign in" which is checked with an 'X'. At the bottom of the form, there is a "Sign In" button.

Sign In

Sign in name

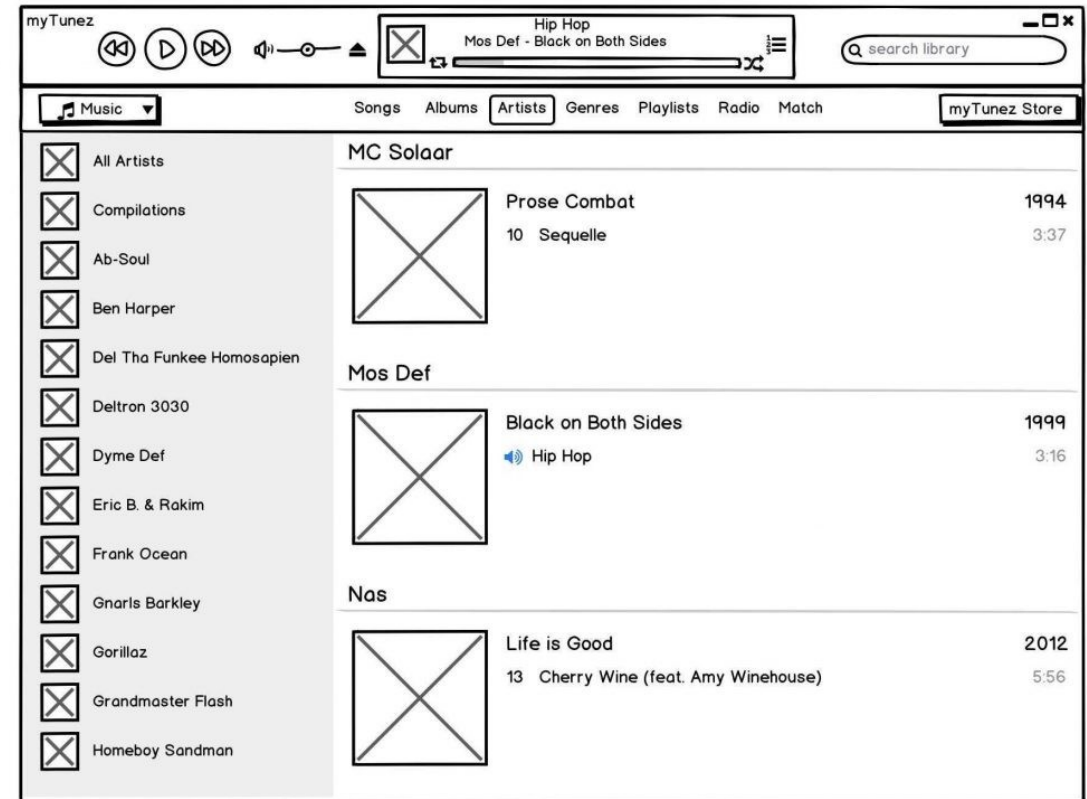
Password (I forgot my password)

☒ Next time auto sign in

Verificação e validação

Protótipos de média fidelidade: são protótipos com layout e navegação simplificados, que sejam suficientes para testar alguns aspectos da interação do usuário com a ferramenta. É utilizado quando o foco é validar a interatividade com os elementos da interface. Busca definir a estrutura de conteúdo da interface e criar um layout básico da funcionalidade.

Ferramentas: Draw.io, Mockflow



Verificação e validação

Protótipos de alta fidelidade: representam de forma bastante real o produto final mas, em geral têm apenas uma ou poucas funções realmente funcionando. É possível simular o fluxo das funcionalidades e já fica bem mais próximo do layout final da funcionalidade.



Sign in name

Password (I forgot my password)

☐ Next time auto sign in



Referências

LAPLANTE, Phillip A.. **Requirements Engineering for Software and Systems**. 3. ed. Boca Raton: Crc Press, 2018.

POHL, Klaus; RUPP, Chris. **Requirements Engineering Fundamentals**. 2. ed. Santa Barbara: Rocky Nook, 2015.

PRESSMAN, Roger S.; MAXIM, Bruce R.. **Engenharia de software**: uma abordagem profissional. 8. ed. Porto Alegre: Amgh, 2016.

SOMMERVILE, Ian. **Engenharia de software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme. **Engenharia de requisitos**: Software orientado ao negócio. Rio de Janeiro: Brasport, 2016.

WIEGERS, Karl; BEATTY, Joy. **Software Requirements**. 3. ed. Redmond: Microsoft Press, 2013.