
Validação de Requisitos

Validação de Requisitos

- Checar que o produto **correto** estará sendo **construído**.
- Garantir que o software em desenvolvimento (ou em modificação) irá satisfazer as **reais** necessidades das partes interessadas.
- Checar a especificação dos requisitos de software em relação aos **objetivos** e **requisitos** das partes interessadas

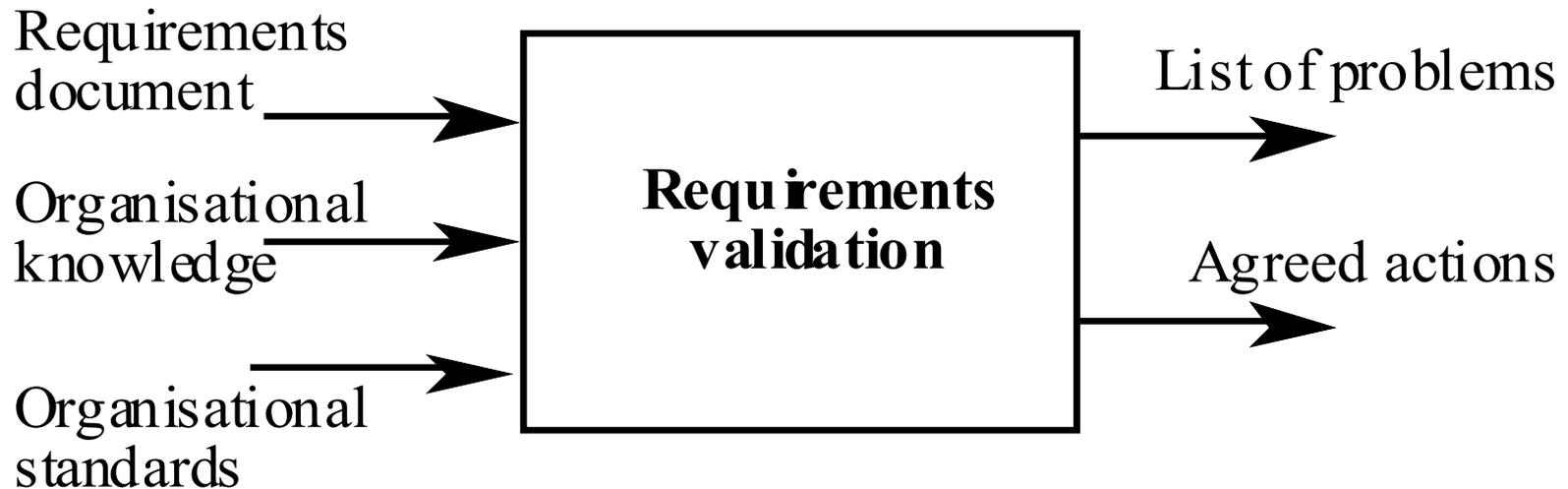
Objetivos da Validação

- Certificar que o documento de requisitos é uma descrição aceitável do sistema a ser implementado
- Confirmar/checar as seguintes propriedades do documento:
 - Completude e consistência
 - Se está de acordo com os padrões
 - Conflitos de requisitos
 - Erros técnicos
 - Requisitos ambíguos

Análise e validação

- Análise trabalha com os dados ‘crus’ que foram elicitados dos stakeholders do sistema
- Validação usa uma versão final do documento de requisitos, como os requisitos que foram negociados e concordados

Entradas e saídas da validação



Entradas da validação

- Documento de requisitos
 - Deve ser um versão completa do documento, não uma versão preliminar. Formatada e organizada de acordo com os padrões organizacionais.
- Conhecimento organizacional
 - Conhecimento, freqüentemente implícito, da organização que poderá ser usado para julgar o realismo dos requisitos
- Padrões organizacionais
 - Padrões locais, ex. para a organização do documento de requisitos

Saídas da validação

- Lista de problemas
 - Lista dos problemas descobertos no documento de requisitos
- Ações concordadas
 - Lista de ações que foram acertadas em resposta aos problemas dos requisitos. Alguns problemas podem ter várias ações corretivas; alguns problemas podem não ter ações associadas.

Exemplo 1

- O Gerenciador de Tarefas deve fornecer mensagens de status a intervalos regulares e não menos que a cada 60 segundos

Exemplo 1

- O Gerenciador de Tarefas deve fornecer mensagens de status a intervalos regulares e não menos que a cada 60 segundos

- Problemas:
 - Quais são as mensagens de status?
 - Quais as condições para fornecer a mensagem?
 - Como elas são fornecidas?
 - Por quanto tempo as mensagens são exibidas?
 - Intervalo de 80 segundos satisfaz o requisito. Intervalo de 80 dias também

Exemplo 2

- O Editor XML deve alternar entre exibir e esconder caracteres não-imprimíveis, instantaneamente

Exemplo 2

- O Editor XML deve alternar entre exibir e esconder caracteres não-imprimíveis, instantaneamente

- Problemas:
 - Instantâneo?
 - O que provoca a mudança? É temporizado, é acionado pelo usuário?
 - O que será alterado? O texto selecionado, o documento inteiro, a página atual, etc.
 - Quais são os caracteres não imprimíveis? Texto oculto, comentários, tags, etc.

Importância da Validação de Requisitos

1. Mariner Bugs Out (1962)

- **Cost:** \$18.5 million
- **Disaster:** The Mariner 1 rocket with a space probe headed for Venus diverted from its intended flight path shortly after launch. Mission Control destroyed the rocket 293 seconds after liftoff.
- **Cause:** A programmer incorrectly transcribed a handwritten formula into computer code, missing a single superscript bar. Without the smoothing function indicated by the bar, the software treated normal variations of velocity as if they were serious, causing faulty corrections that sent the rocket off course.

bug

Como realizar a validação?

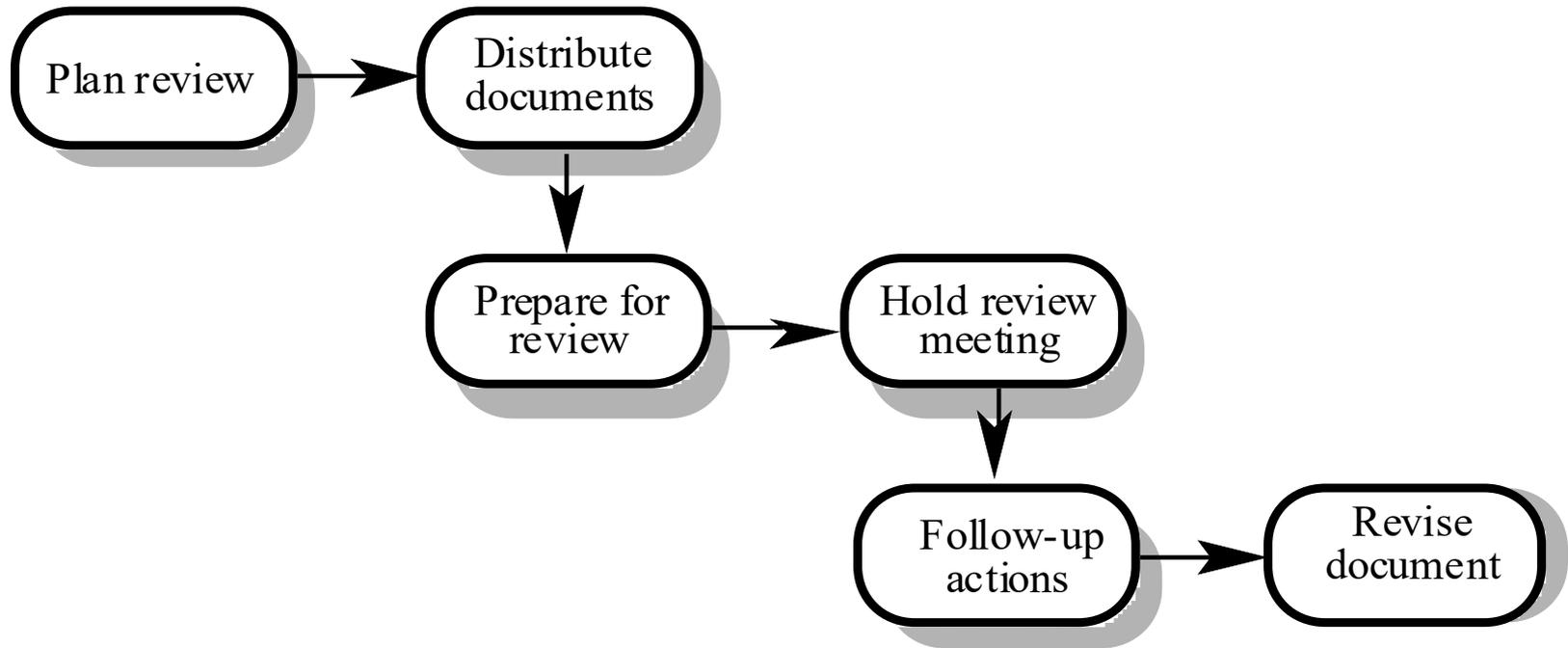
Como realizar a validação?

- Revisão de requisitos
- Prototipagem
- Validação de modelos
- Teste de requisitos

Revisão de requisitos

- Um grupo de pessoas lê e analisa os requisitos, procura problemas, se reúne, discute os problemas e concorda nas ações para tratar estes problemas

Processo de revisão de requisitos



Atividades de revisão

- Planejar a revisão
 - Selecionar time de revisão, hora e local para o encontro de revisão.
- Distribuir os documentos
 - O documento de requisitos é distribuído entre os membros do time de revisão
- Preparar para revisão
 - Cada revisor individualmente lê os requisitos e encontra conflitos, omissões, inconsistências e desvios dos padrões e outros problemas.

Atividades da revisão

- Realizar o encontro de revisão
 - Os problemas e comentários individuais são discutidos e um conjunto de ações para tratar dos problemas é concordado.
- Ações de acompanhamento
 - O chefe da revisão verifica se todas as ações acertadas foram executadas.
- Revisar o documento
 - O documento de requisitos é revisado para refletir as ações concordadas. Nestes estágio, pode ser aceito ou revisado novamente

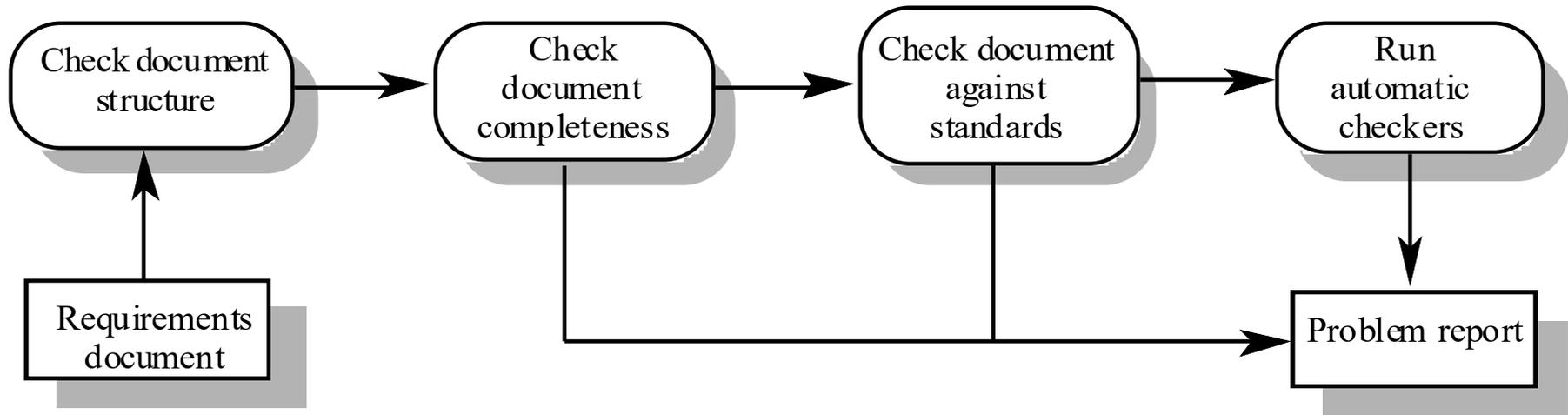
Ações para os problemas

- **Clarificação dos requisitos**
 - O requisito pode ter sido mal escrito ou pode ter acidentalmente omitido alguma informação que foi coletada durante a fase de requisitos.
- **Falta de informação**
 - Alguma informação está faltando no documento de requisitos. É responsabilidade do engenheiro de requisitos que está revisando o documento descobrir esta informação dos stakeholders do sistema.
- **Conflito de requisitos**
 - Existe um conflito significativo entre requisitos. Os stakeholders envolvidos devem negociar para resolver o conflito.
- **Requisito não realístico**
 - O requisito não poderá ser implementado com a tecnologia disponível ou dentro das limitações do sistema. Os stakeholders devem ser consultados para decidir como tornar o requisito mais realístico.

Cheque de pré-revisão

- Revisões são caras porque envolvem um número de pessoas que gastará tempo lendo e verificando o documento de requisitos
- Estes gastos podem ser reduzidos usando uma pré-revisão, onde uma pessoa verifica o documento e procura por problemas mais simples tais como: erros tipográficos, falta de aderência ao padrão, falta de algum requisito, etc.
- O documento poderá ser retornado para correção ou enviada a lista de problemas para os revisores

Cheque de pré-revisão



Participantes do time de revisão

- Os revisores devem incluir um número de stakeholders com backgrounds diferentes
 - Pessoas com backgrounds diferentes trazem seus conhecimentos e habilidades para a revisão
 - Os stakeholders se sentem envolvidos no processo e ER e desenvolvem um entendimento das necessidades dos outros stakeholders
- O time de revisão deve sempre incluir um especialista no domínio e um usuário final
- Time independente

Critérios de revisão

- Entendimento
 - Os leitores do documento podem entender o que o requisito significa?
- Redundância
 - A informação está desnecessariamente repetida no documento?
- Completude
 - O revisor conhece algum requisito que está faltando ou existe alguma informação que está faltando na descrição individual de um requisito?
- Ambigüidade
 - Os requisitos foram expressos usando termos que estão claramente definidos? É possível que leitores de backgrounds diferentes fazerem interpretações diferentes dos requisitos?

Critérios de revisão

□ Consistência

- As descrições dos diferentes requisitos incluem contradições? Existem contradições entre requisitos individuais e requisitos gerais do sistema?

□ Organização

- O documento está estruturado de uma forma apropriada? As descrições dos requisitos estão organizadas de forma que requisitos relacionados estejam agrupados?

□ Conformidade a padrões

- O documento de requisitos ou os requisitos individuais estão conforme o padrão definido? Os desvios do padrão estão justificados?

□ Rastreamento

- Os requisitos estão identificados de forma não ambígua, incluindo links a outros requisitos relacionados e às razões porque os requisitos foram incluídos?

Questões para o checklist

- ❑ Cada requisito está unicamente identificado?
- ❑ Os termos especializados estão definidos no glossário?
- ❑ O requisito sozinho faz sentido, ou precisamos examinar outros requisitos para entendê-lo?
- ❑ Os requisitos individuais usam os termos de forma consistente?
- ❑ O mesmo serviço é solicitado em requisitos diferentes? Existem contradições nestas solicitações?
- ❑ Se um requisitos faz referência a alguma outra facilidade, elas são descritas em outra parte do documento?
- ❑ Os requisitos que são relacionados estão agrupados? Se não, há um referência entre eles?

Exemplo de checklist

*Karl E. Wiegers. Software Requirements.
2nd edition, Microsoft press, 2003*

**OBS: na prática, checklists não devem
ser tão grandes!**

Organização e completude

- Todas as referências a outros requisitos estão corretas?
- Todos os requisitos estão escritos em um nível de detalhamento apropriado e consistente?
- Os requisitos provêm uma base adequada para arquitetura/projeto?
- Cada requisito possui um valor de prioridade?
- Todo hardware e software externos, assim como interfaces de comunicação, estão definidos?
- Os algoritmos intrínsecos a requisitos funcionais estão definidos?
- A especificação inclui todas as necessidades do sistema e do cliente?
- Está faltando informação em algum requisito?
- O comportamento esperado para todas as condições de erros previstas está documentado?

Corretude

- Algum requisito está em conflito com outro requisito?
- Algum requisito está duplicado?
- Cada requisito está escrito com uma linguagem clara, concisa e não-ambígua?
- A satisfação de cada requisito pode ser verificada através de testes, demonstração, revisão ou análise?
- Cada requisito está dentro do escopo do projeto?
- Cada requisito está livre de erros de conteúdo e erros gramaticais?
- Todos os requisitos podem ser implementados, dentro das restrições estabelecidas?
- Todas as mensagens de erros são únicas e claras?

Outros

- Atributos de qualidade
 - Todos os objetivos de desempenho estão devidamente definidos?
 - Todas as considerações de segurança e proteção estão devidamente definidas?
 - Todos os outros objetivos de qualidade estão explicitamente documentados e quantificados, com os compromissos aceitáveis definidos?
- Rastreabilidade
 - Cada requisito está unicamente e corretamente identificado?
 - Cada requisito funcional de software está ligado um requisito de mais alto nível?
- Características especiais
 - Todos os requisitos são realmente requisitos? Ou seja, não são uma solução de projeto ou de implementação?
 - As funcionalidades nas quais o tempo é crítico estão identificadas, e o critério de tempo está bem definido?
 - Questões de internacionalização forem tratadas adequadamente?

Casos de Uso

□ Casos de Uso

- O caso de uso é uma tarefa auto-contida e individual?
- O objetivo do caso de uso está claro?
- Está claro qual(is) o(s) ator(es) que se beneficiam do caso de uso?
- O caso de uso está escrito com um detalhamento adequado, sem detalhes de projeto ou de implementação?
- Todos os fluxos alternativos previstos estão documentados?
- Todas as condições de exceção conhecidas estão documentadas?
- Há alguma sequência de ações comum, que pode ser dividida em casos de usos separados?
- Os passos de cada fluxo estão escritos de forma clara, completa e não-ambígua?
- Cada ator e passo no caso de uso é pertinente à realização da tarefa?
- Cada fluxo alternativo definido no caso de uso é viável e verificável?
- As pré-condições e pós-condições são apropriadas e suficientes para o caso de uso?

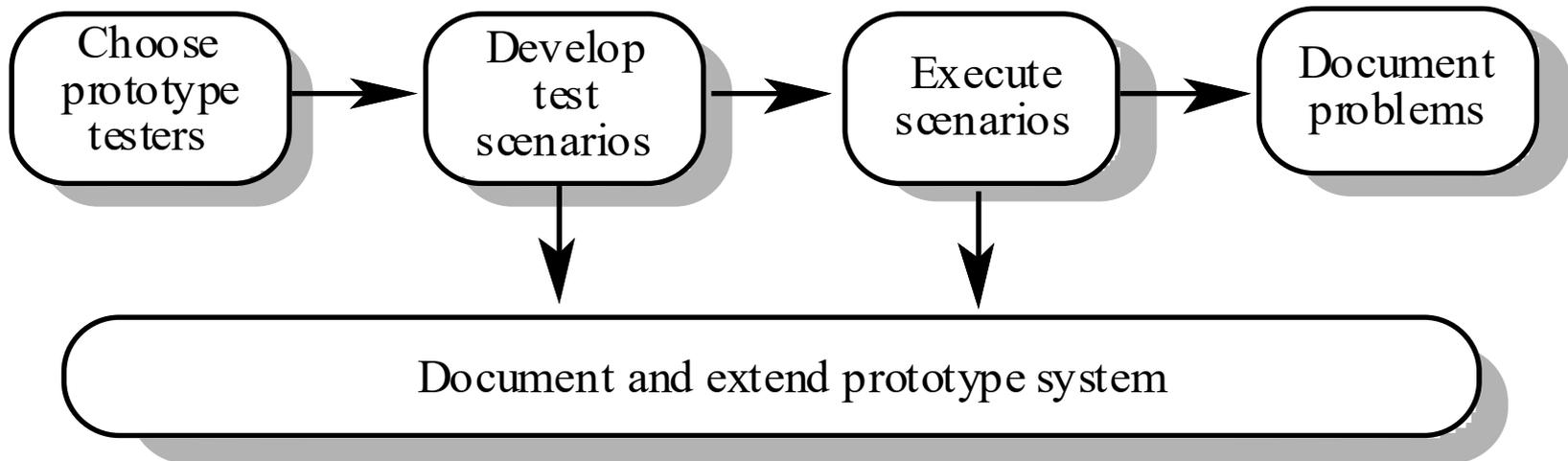
Modelos de objetivos

- Há algum ator sem link com outro ator?
- Há algum ator dentro de outro ator?
- Há algum elemento interno com um link para o próprio ator?
- Há alguma ligação entre atores que não seja uma ligação de dependência?
- Há algum softgoal que deveria ser goal?
- Há algum goal que deveria ser softgoal?
- Há algum goal ligado a outro goal?
- Há algum elemento com uma dependência para outro elemento dentro de um mesmo ator?
- Há alguma ligação de dependência sem dependum?
- Há algum goal recebendo links de contribuição?
- Há algum link de contribuição entre um elemento de um ator e um outro elemento fora desse ator?
- Algum goal é um meio em uma ligação meio-fim?
- Há alguma ligação de meio-fim fora de ou ultrapassando a fronteira de um ator?
- Há alguma ligação de decomposição fora de ou ultrapassando a fronteira de um ator?

Prototipagem

- Protótipos para validação de requisitos demonstram os requisitos e ajudam aos stakeholders a descobrirem problemas
- Protótipos para validação devem ser completos, razoavelmente eficientes e robustos. Deverá ser possível usá-los da mesma forma que o sistema requerido
- Documentos do usuário e treinamento devem ser providenciados

Prototipagem para validação



Atividade de Prototipagem

- Escolha os testadores do protótipo
 - Os melhores testadores são os usuários bem experientes e que tenham cabeça aberta sobre o uso do novo sistema. Usuários finais que têm funções diferentes devem estar envolvidos para que diferentes áreas da funcionalidade do sistema possam ser cobertas.
- Desenvolva os cenários de teste
 - É necessário um planejamento detalhado para preparar um conjunto de cenários de teste amplo, que faça cobertura de uma grande quantidade de requisitos. Os usuários finais não devem apenas brincar com o sistema, pois isto poderá não exercitar aspectos críticos do sistema.
- Execute cenários
 - Os usuários do sistema, geralmente sozinhos, passa a testar o sistema através da execução do cenário planejado.
- Documente problemas
 - É melhor definir algum formulário de problemas (eletrônico ou em papel) para que os usuários possam preencher ao encontrar um problema.

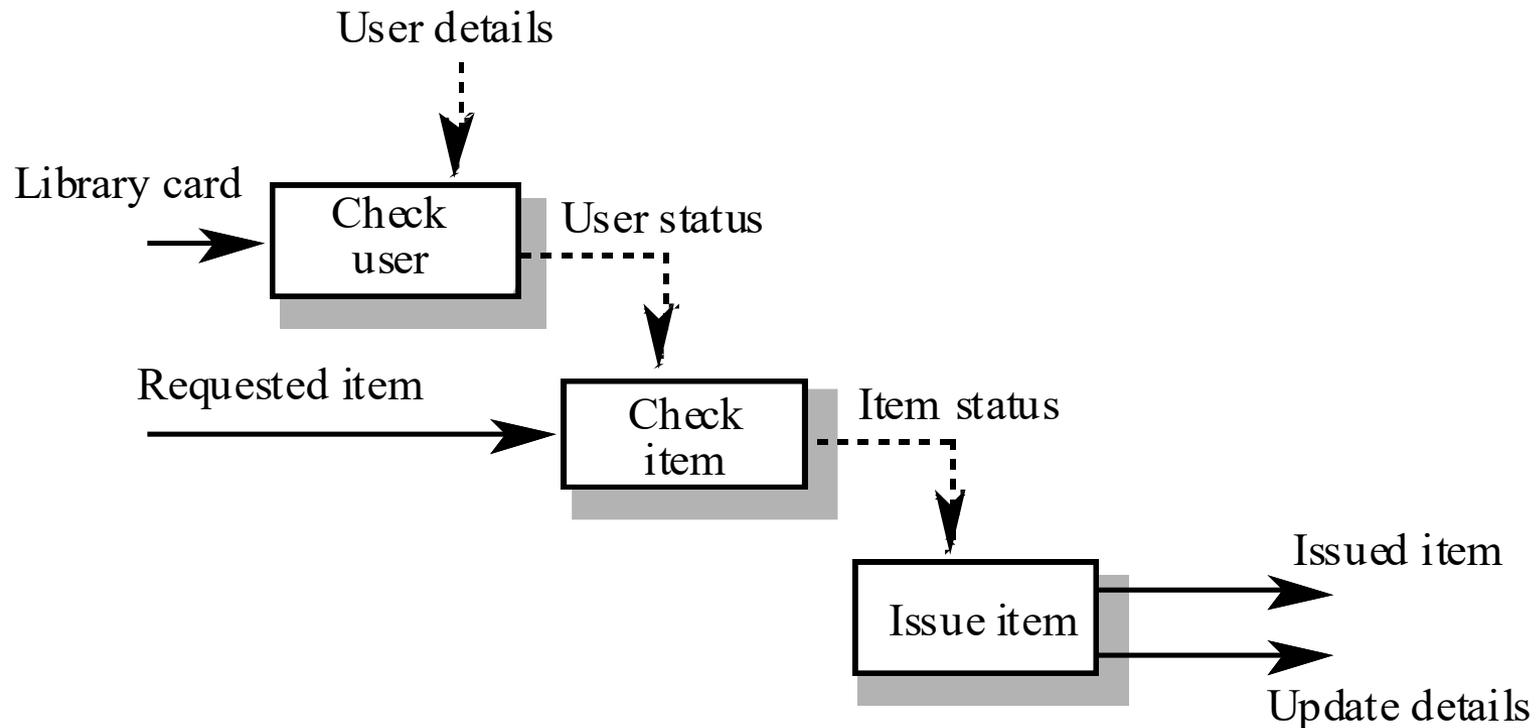
Desenvolvimento do manual de usuário

- A escrita de um manual de usuário a partir de requisitos força uma análise detalhada dos requisitos e assim poderá revelar problemas com os requisitos
- Informação do manual de usuários
 - Descrição da funcionalidade e como ela é implementada
 - Que partes do sistema não foi implementada
 - Como resolver problemas
 - Como instalar e começar o sistema

Validação do Modelo

- Validação dos modelos do sistema é uma parte essencial do processo de validação
- Objetivos da validação dos modelos
 - Demonstrar que cada modelo é auto-consistente
 - Se existem vários modelos do sistema, demonstrar que eles são internamente e externamente consistentes
 - Demonstrar que os modelos refletem de forma precisa os reais requisitos dos stakeholders do sistema
- Alguma verificação é possível com ferramentas automáticas
- Parafrasear o modelo é uma forma efetiva de checagem

Diagrama de fluxo de dados para emissão (Issue)



Descrição parafraseada

Check user

Inputs and sources	User's library card from end-user
Transformation function	Checks that the user is a valid library user
Transformation outputs	The user's status
Control information	User details from the database

Check item

Inputs and sources	The user's status from Check user
Transformation function	Checks if an item is available for issue
Transformation outputs	The item's status
Control information	The availability of the item

Issue item

Inputs and sources	<i>None</i>
Transformation function	Issues an item to the library user. Items are stamped with a return date.
Transformation outputs	The item issued to the end user Database update details
Control information	Item status - items only issued if available

Teste dos requisitos

- Cada requisito deve ser testável, isto é deverá ser possível definir um teste para verificar se o requisito foi ou não alcançado.
- A invenção de testes de requisitos é uma técnica efetiva de validação, pois informação ambígua ou incompleta dificulta a formulação dos testes
- Cada requisito funcional deve ter um teste associado

Definição de caso de teste

- ❑ Qual cenário de uso poderá ser usado para testar um requisito?
- ❑ O requisito, sozinho, inclui informação suficiente para a definição de um teste?
- ❑ É possível testar o requisito usando um único teste ou são necessários múltiplos testes?
- ❑ O requisito poderá ser re-escrito para tornar os casos de teste mais óbvios?

Formulário de teste de requisito

- O identificador do requisito
 - Deve haver pelo menos um para cada requisito.
- Requisitos relacionados
 - Devem ser referenciados, pois o teste poderá ser relevante também a estes requisitos.
- Descrição do teste
 - Uma breve descrição do teste. Deve incluir as entradas do sistema e as saídas correspondentes.
- Problemas do requisito
 - Uma descrição dos problemas que tornaram difícil ou impossível a definição do teste.
- Comentários e recomendações
 - São conselhos de como resolver os problemas dos requisitos que foram descobertos.

Exemplo de teste de requisitos

- 10.(iv) Quando os usuários acessarem o EDDIS, eles serão apresentados a páginas web com todos os serviços disponíveis para eles.

Formulário de teste de requisitos

Requirements tested: 10.(iv)

Related requirements: 10.(i), 10.(ii), 10.(iii), 10.(vi), 10.(vii)

Test applied: For each class of user, prepare a login script and identify the services expected for that class of user.

The results of the login should be a web page with a menu of available services.

Requirements problems: We don't know the different classes of EDDIS user and the services which are available to each user class. Apart from the administrator, are all other EDDIS users in the same class?

Recommendations: Explicitly list all user classes and the services which they can access.

Requisitos difíceis de testar

□ Requisitos do sistema

- Requisitos que se aplicam ao sistema como um todo. Em geral, estes são os requisitos mais difíceis de validar independentemente do método usado, pois podem ser influenciados por quaisquer dos requisitos funcionais. Testes que não são executáveis, não podem testar características gerais não-funcionais do sistema, tais como usabilidade.

□ Requisitos exclusão

- Existem requisitos que excluem comportamentos específicos. Por exemplo, um requisito poderia dizer que falhas do sistema nunca devem corromper o banco de dados. Não será possível testar este requisito exaustivamente.

□ Alguns requisitos não-funcionais

- Alguns requisitos não-funcionais, tais como requisitos de confiabilidade só podem ser testados com um grande conjunto de teste. O projeto destes testes não ajuda a validação dos requisitos.

Pontos principais

- A validação de requisitos deve focar em verificar se a versão final do documento de requisitos apresenta conflitos, omissões ou desvios dos padrões.
- As entradas do processo de validação são os documentos de requisitos, padrões organizacionais, e conhecimento implícito da organização. As saídas são uma lista de problemas dos requisitos e as ações concordadas para tratar estes problemas.
- Revisões envolvem um grupo de pessoas fazendo análise detalhada dos requisitos.
- Os custos de revisão podem ser reduzidos se forem verificados, antes da revisão, desvios do padrão organizacional.

Pontos principais

- Checklists sobre o que procurar podem ser usadas para guiar o processo de revisão de requisitos.
- Prototipagem é efetivo para validação de requisitos se um protótipo for desenvolvido durante o estágio de elicitação de requisitos.
- Os modelos do sistema são validados através do seu parafraseamento. Isto significa que eles são sistematicamente traduzidos em uma descrição em linguagem natural.
- Projetando testes para requisitos pode revelar problemas com os requisitos. Se um requisito não estiver claro, poderá ser impossível definir uma teste para ele.