

ACH2028- Qualidade de Software

AULA 06 – INSPEÇÃO DE REQUISITOS

Prof. Marcelo Medeiros Eler
marceloeler@usp.br

Conceitos básicos

- Requisito
 - Condição necessária para a obtenção de certo objetivo, ou para o preenchimento de certo objetivo.
- Especificação
 - Descrição minuciosa das características que um material, uma obra, ou um serviço deverão apresentar.
- Especificação de requisitos

Tipos de requisitos

- Requisito Funcional
 - Declarações de funções que o sistema deve fornecer e como o sistema reagirá a entradas específicas. Em alguns casos, os requisitos funcionais podem também declarar o que o sistema não faz.
- Requisito Não-Funcional
 - Restrições sobre as funções oferecidas pelo sistema. Exemplos: restrições de tempo, desempenho, sobre o processo de desenvolvimento, padrões, entre outros.
- Requisito de Domínio:
 - Tem origem e refletem características do domínio da aplicação. Podem ser dos dois tipos acima.

Exemplo de requisito funcional

- Requisito 1.3:
 - O Sistema deve permitir a inclusão, alteração, exclusão e consulta de contas correntes. As contas devem estar associadas a uma agência e a um cliente do banco.
 - Os atributos das contas corrente devem ser os seguintes: cliente, agência, número, saldo, limite, data de abertura, data de fechamento e status (aberta ou fechada).
 - Uma conta pode ser fechada e reaberta quantas vezes forem necessárias, mas será excluída se permanecer mais do que 6 meses fechada.

Exemplos de requisitos não funcionais

- Requisito 7.6.5:
 - O sistema não deverá revelar aos operadores nenhuma informação pessoal sobre os clientes, além de seus nomes e seu número de referência.
- Requisito 9.3:
 - O relatório de faturamento de cada setor de cada filial deve ser gerado em menos de 5s.
- Requisito 8.6:
 - Todas as transações do sistema devem ser registradas com as seguintes informações: usuário responsável, ação, data e hora.

Áreas problemáticas

- Comunicação: ruído (interpretação errônea, omissão)
- Obtenção de informações pertinentes
- Conflitos entre requisitos de diferentes pessoas
- Cuidado com a complexidade dos problemas
- Acomodação de mudanças que ocorrem durante e após a análise (requisitos voláteis e mudanças naturais do sistema [chuva de requisitos])

Técnicas de elicitação de requisitos

- Introspecção
- Entrevistas
- Questionários
- histórias de usuários
- Cenários
- Etnografia
- FAST (JAD – Joint Application Development)
- Brainstorming
- Recuperação (Engenharia Reversa)
- Análise de protocolos e leitura de documentos
- Prototipação

Especificação de Requisitos

- Consistem em descrever em detalhes os requisitos do sistema
- Em métodos tradicionais, um Documento de Especificação de Requisitos é gerado (ou apenas Documento de Requisitos).
- Em métodos ágeis, os requisitos podem ser expressos por meio de uma lista de histórias de usuários e cenários

Especificação de Requisitos

- Existem algumas formas de descrever os requisitos:
 - Linguagem Natural
 - Linguagem de descrição de projeto
 - Notações gráficas
 - Especificações matemáticas

Linguagem natural

- Problemas quando usada para especificação mais detalhada:
 - Ambiguidade
 - Flexibilidade (pode-se dizer a mesma coisa de modos diferentes)
 - Falta de padronização

Linguagem natural estruturada

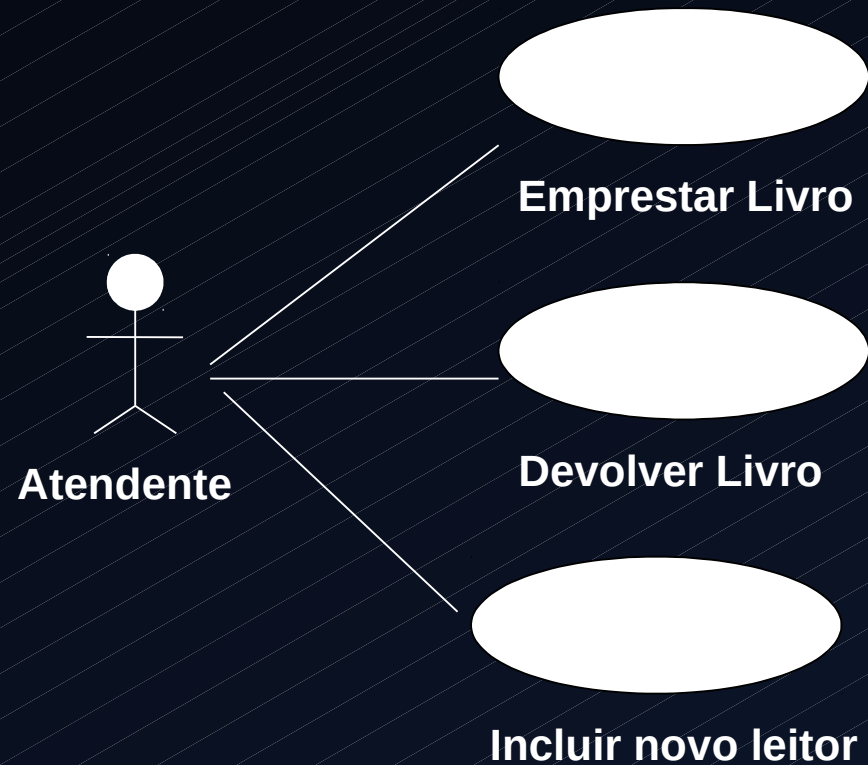
- A linguagem natural estruturada é uma forma restrita da linguagem natural, que se destina a escrever requisitos de sistema.
- Mantém a facilidade de expressão da linguagem natural, mas com um certo grau de uniformidade imposto à especificação.
- Limitação de terminologias e uso de templates.

Linguagem natural estruturada

- A especificação de requisitos deve ser composta por sentenças em linguagem natural, seguindo determinados padrões.
- Cada requisito deve ter um identificador único, por exemplo, um identificador numérico ou alfanumérico, para posterior referência.
 - Ex. 1, 1.1, 1.2, 2...
- Iniciar com “O sistema deve ...”.
 - Exemplo: “O sistema deve rodar em microcomputadores da linha IBM PC que possuam microprocessador Pentium IV ou superior.”

Notação gráfica

- Uma linguagem gráfica, complementada com anotações de texto, é utilizada para definir os requisitos funcionais do sistema.



Espe

- Not amb sob
- Exemp

Note that `checked_out` is a partial function, indicated by the stroke through the arrow \rightarrow . This means that some documents are checked out to people who have permission to change them.

A possible state of our system occurs when Doug has checked out the specification and design, and Phil has checked out

`checked_out = {(design, doug), (spec, doug), (code, phil)}`

We need two operations that change the state, `CheckOut` and `CheckIn`. Here is `CheckOut`. It has two input parameters, the

`CheckOut`

Δ Documents

$p?: \text{PERSON}$

$d?: \text{DOCUMENT}$

$d? \notin \text{dom } \text{checked_out}$

$(d?, p?) \in \text{permission}$

$\text{checked_out}' = \text{checked_out} \cup \{(d?, p?)\}$

`CheckOut` has two preconditions. They are the predicates that contain no primed "after" variables. First, document $d?$

Documento de requisitos

- Alguns processos de desenvolvimento (tradicionais) exigem que os requisitos do sistema sejam registrados e formalizados em um documento de requisitos
- Nestes processos, o documento de requisitos é o resultado do processo de engenharia de requisitos
- Este documento contém a especificação de todos os requisitos funcionais e não funcionais do software, incluindo as capacidades do produto, os recursos disponíveis, os benefícios e os critérios de aceitação.

Requisitos em métodos ágeis

- Os métodos ágeis, diferente dos métodos tradicionais, não se preocupam em criar um documento de requisitos formal e com cada requisito detalhadamente especificado
- Os requisitos são continuamente refinados, alterados e aumentados conforme o software é desenvolvido e a equipe de desenvolvimento recebe o feedback dos clientes do sistema
- A documentação dos requisitos do sistema por meio de histórias de usuário tem sido amplamente adotada por diversos métodos ágeis

histórias de usuário

- As histórias de usuário descrevem a expectativa de uso da aplicação
- histórias de usuário são leves versões dos requisitos dos métodos tradicionais e por isso são mais adequadas aos métodos ágeis
- histórias de usuário ajudam os interessados a planejar e priorizar as funcionalidades de cada ciclo
- Em muitas situações as histórias de usuário substituem até mesmo os diagramas de análise e projeto dos métodos tradicionais (diagramas de caso de uso, diagramas de sequência, comunicação, etc)

histórias de usuário

- Informações básicas:
 - Quem? Para quem estamos desenvolvendo a funcionalidade.
 - O que? Uma descrição resumida da funcionalidade em si.
 - Por que? O motivo pelo qual o cliente precisa desta funcionalidade. Se possível citando o valor de negócio obtido.

histórias de usuário

- Um padrão:
 - COMO/SENDO [um ator]
 - EU QUERO/DESEJO [realizar uma tarefa]
 - PARA QUE [eu possa alcançar um objetivo]

histórias de usuário

- Exemplos:
 - Funcionalidade: adicionar um filme ao software MyMovies
 - COMO um fã de cinema
 - EU QUERO adicionar um filme à base de dados do MyMovies
 - PARA QUE eu possa compartilhar um filme com outros fãs de cinema

histórias de usuário

- Exemplos:
 - Funcionalidade: pagar um boleto bancário pela internet usando o número de código de barras
 - COMO um cliente da agência bancária
 - EU QUERO realizar o pagamento de um boleto bancário no site do banco usando a numeração do código de barras
 - PARA QUE eu possa pagar minhas contas pela internet

histórias de usuário

- SMART user stories
 - Specific
 - Measurable
 - Achievable
 - Relevant
 - Timeboxed

Dicas para escrever histórias de usuários efetivas

- Simplicidade (KIS ou KISS)
- Expresse o “o que?”, e não o “como?”
- Escreva histórias relevantes
- Evite ambiguidade e imprecisão
- Requisitos não funcionais mensuráveis

Simplicidade

- KIS = Keep It Simple
- KISS = Keep It Simple, Stupid
- Evite usar conjunções (E)
- Evite usar limitadores (exceto, mas, etc)

Simplicidade

- história com conjunções:
 - COMO um requerente, EU QUERO ter acesso à tela com as coberturas, inserir meus dados pessoais e dados do meu veículo, e submeter a proposta PARA obter cobertura de seguro para o meu automóvel
- histórias sem conjunções:
 - COMO um requerente, EU QUERO ter acesso à tela com as coberturas PARA selecionar o plano de cobertura que eu preciso
 - COMO um requerente, EU QUERO inserir meus dados pessoais e do meu veículo para comparar os valores dos planos
 - COMO um requerente, EU QUERO submeter a proposta para obter cobertura de seguro para o meu automóvel

Simplicidade

- história com limitadores:
 - COMO um subscritor, eu QUERO cancelar uma negação de cobertura PARA um possível cliente da seguradora para aumentar a base de clientes, a não ser que a negação de cobertura seja porque o cliente tem má reputação de crédito, caso em que eu QUERO confirmar a rejeição para proteger a base de clientes
- histórias sem limitadores:
 - COMO um subscritor, eu QUERO cancelar uma negação de cobertura de um cliente que tem bom crédito PARA aumentar a base de clientes
 - COMO um subscritor, eu QUERO confirmar a negação de cobertura de um cliente que tem má reputação de crédito PARA proteger a base de clientes

Expresse o “o que?”, e não o “como?”

- história com o “como?”
 - COMO um requerente, eu QUERO selecionar meu estado de uma caixa do tipo combobox com as abreviações PARA evitar submeter um estado inválido
- história apenas com o “o que?”
 - COMO um requerente, eu QUERO submeter a abreviação válida do meu estado PARA garantir a precisão da cobertura do meu seguro

Escreva histórias relevantes

- Tenha certeza de que a história de usuário a ser escrita será realmente utilizada por algum usuário
- Tenha certeza de que a história de usuário a ser escrita está dentro do escopo do software

Escreva histórias relevantes

- Tenha certeza de que a história de usuário a ser escrita será realmente utilizada por algum usuário
- Tenha certeza de que a história de usuário a ser escrita está dentro do escopo do software

Evite ambiguidade e imprecisão

- história imprecisa:
 - COMO um gerente de almoxarifado, EU QUERO pedir a quantidade correta de produto que vamos vender PARA evitar muito estoque
 - COMO um operador de telefone, EU QUERO completar pelo menos 12 reservas por hora durante horário de pico PARA reduzir o tempo de espera dos clientes
- história mais precisa:
 - COMO um operador de reservas, eu QUERO completar pelo menos 12 reservas durante os horários de picos em dias comuns PARA reduzir o número de clientes que desligam o telefone ao esperar mais do que 5 minutos.

Requisitos não funcionais mensuráveis

- histórias mensuráveis devem expressar valores em números, necessariamente
- Exemplos:
 - Frequência
 - Urgência
 - Volume
 - Precisão numérica
 - Escalabilidade
 - Confiança

Inspeção

- Seu principal objetivo é assegurar a qualidade do software que está sendo desenvolvido.
- É considerada como um dos melhores métodos para detectar defeitos nos diferentes subprodutos do processo de software
- É um método estruturado para verificar documentos do software, tais como: especificação de requisitos, documentos do projeto, código-fonte, etc...

Inspeção

- É um método cujo propósito é descobrir os defeitos nos artefatos gerados em cada atividade do processo de desenvolvimento antes que ele seja utilizado na etapa seguinte

Dinâmica

- A inspeção de um documento de software inicia-se com a leitura individual do documento pelos avaliadores do processo de inspeção.
- Após a leitura há uma reunião em que os defeitos são discutidos e listados
- O próximo passo é remover os defeitos e então iniciar o ciclo novamente para obter a qualidade desejada no documento do software

Tipos de inspeção

- As técnicas de inspeção diferem em
 - Estratégia
 - Sistemática
 - Não-sistemática
 - Defeitos procurados
 - Lista específica e fechada
 - Defeitos em geral
 - Política de responsabilidades dos avaliadores
 - Técnicas distintas
 - Técnicas idênticas

Processo de Inspeção

- Planejamento
- Visão Geral
- Preparação
- Realização da Inspeção
- Retrabalho
- Revisão

Equipe de Inspeção

- Moderador
- Autor
- Relator
- Inspetor

Vantagens da Inspeção

- Aumento da qualidade e da confiança nos produtos intermediários
- Detecção precoce de defeitos
- Aperfeiçoamento dos inspetores nas tarefas do processo de desenvolvimento
- Melhora na comunicação da equipe de desenvolvimento
- Diminuição de retrabalho
- Feedback rápido
- Melhoria de processo de desenvolvimento

Dificuldades da Inspeção

- Falta de treinamento
- Falta de comprometimento
- Tempo insuficiente
- Objetivos vagos
- Falta de apoio das classes executivas
- Conflitos entre autor e equipe de inspeção

Outras técnicas com objetivos parecidos

- Além da inspeção, existem outras técnicas usadas para garantir a qualidade de subprodutos e processos:
 - A revisão técnica (Technical review) tem o objetivo de avaliar artefatos específicos para verificar se eles apresentam os respectivos padrões e especificações e se eventuais modificações no artefato foram efetuadas de maneira correta. As revisões técnicas são aplicadas a documentos, como Especificação de Requisitos e Descrição de Testes, com o objetivo principal de verificar a conformidade com padrões do processo, e seus atributos de qualidade; tais como completude, corretude, consistência, verificabilidade, modificabilidade, rastreabilidade.

Outras técnicas com objetivos parecidos

- Além da inspeção, existem outras técnicas usadas para garantir a qualidade de subprodutos e processos :
 - *walkthrough* é uma revisão na qual o autor apresenta o material em ordem lógica, sem limite de tempo, a um grupo que verifica o material, conforme ele vai apresentando. Este tipo de revisão não exige muita preparação prévia, e pode ser feito com um número maior de participantes pelo fato dos participantes terem um papel mais passivo.

Outras técnicas com objetivos parecidos

- Além da inspeção, existem outras técnicas usadas para garantir a qualidade de subprodutos e processos :
 - A auditoria (audits) tem o objetivo de verificar a conformidade de produtos e projetos com padrões e processos. O padrão prevê as auditorias da qualidade, que verificam aspectos de conformidade com o processo e com o Plano de Qualidade do projeto.

Inspeção de Documento de Requisitos

- Classes de defeitos:
 - Omissão (O): qualquer informação necessária que tenha sido omitida.
 - Fato Incorreto (FI): informação que consta do artefato mas que seja contraditória com o conhecimento que se tem do domínio de aplicação.
 - Inconsistência (I): informação que consta do artefato mais de uma vez e em cada ocorrência ela é descrita de forma diferente.
 - Ambiguidade (A): informação que pode levar a múltiplas interpretações.
 - Informação Estranha (IE): informação que, embora relacionada ao domínio, não é necessária para o sistema em questão.
 - Diversos (D): qualquer outro tipo de defeito que não se encaixe nas outras categorias. Ex: declarações em seções erradas.

Inspeção de Documento de Requisitos

- Omissão de Funcionalidade:
 - Informação que descreva algum comportamento desejado do sistema foi omitida do Documento de Requisitos (DR).
- Exemplo:
 - RF2: o sistema deve solicitar a informação necessária para inserir um item bibliográfico: título, autor, data, lugar, assunto, resumo, número, editor, periódico, congresso.
 - RF3: o sistema deve dar uma mensagem de alerta quando o usuário tentar inserir um **item incompleto**. Essa mensagem deve questionar o usuário se ele deseja cancelar a operação, completar a informação ou concluir a inserção como está.
 -

Inspeção de Documento de Requisitos

- Omissão de Desempenho:
 - Informação que descreva um desempenho desejado para o sistema foi omitida ou descrita de uma forma não apropriada para que possa ser verificada posteriormente no teste de aceitação.
- Exemplo:
 - RNF1: o sistema deve fornecer os resultados **tão rápido quanto possível**.

Inspeção de Documento de Requisitos

- Omissão de Interface:
 - Quando informação que descreva como sistema proposto vai fazer interface e se comunicar com outros objetos fora de seu escopo for omitida do DR.
- Omissão de Recursos do Ambiente:
 - Quando informação que descreve o hardware, software, base de dados ou detalhes do ambiente operacional no qual o sistema vai rodar for omitida do DR.

Inspeção de Documento de Requisitos

- Fato incorreto
 - Informação que consta do artefato mas que seja contraditória com o conhecimento que se tem do domínio da aplicação.
- Exemplo
 - RF30: o sistema não deve aceitar devolução de livros se o usuário não tiver a carteirinha da biblioteca no momento.

Inspeção de Documento de Requisitos

- Inconsistência
 - Informação que consta do artefato mais de uma vez e, em cada ocorrência, ela é descrita de forma diferente.
- Exemplo:
 - RF5: o sistema não deve permitir períodos de empréstimo maiores que 15 dias.
 -
 - RF9: professores podem emprestar livros por um período de 3 semanas.

Inspeção de Documento de Requisitos

- Ambiguidade
 - quando a informação pode levar a múltiplas interpretações ou deixar em aberto para interpretação pessoal
- Exemplo:
 - RF20: se o número de dias que o usuário está em atraso é menor que uma semana, ele deve pagar uma taxa de R\$1,00; se o número é maior que uma semana, a taxa é de R\$0,50 por dia.

Inspeção de Documento de Requisitos

- Informação estranha
 - qualquer informação que, embora relacionada ao domínio, não é necessária para o sistema em questão.
- Exemplo:
 - RF15: quando um novo livro é adicionado ao acervo, ele permanece em uma prateleira especial por um período de um mês.

Inspeção de Documento de Requisitos

- Técnicas de Leitura para inspeção de documentos
 - Ad-hoc
 - Checklist
 - Baseada em perspectiva

Inspeção de Documento de Requisitos

- Ad-hoc
 - Os revisores não utilizam nenhuma técnica sistemática de leitura.
 - Cada revisor adota sua maneira de “ler” o Documento de Requisitos
 - Desvantagens:
 - depende da experiência do revisor
 - não é repetível
 - não é passível de melhoria pois não existe um procedimento a ser seguido.

Inspeção de Documento de Requisitos

- Checklist
 - É uma técnica que fornece diretrizes para ajudar o revisor alcançar os objetivos de uma atividade de revisão formal
 - É similar ao ad-hoc, mas cada revisor recebe um checklist.
 - Os itens do checklist capturam lições importantes que foram aprendidas em inspeções anteriores no ambiente de desenvolvimento.
 - Itens do checklist podem explorar defeitos característicos, priorizar defeitos diferentes e estabelecer questões que ajudam o revisor a encontrar defeitos.
 - Pode ser desenvolvido para documentos de requisitos, análise, projeto, código e mesmo documentos de teste.

Checklist

- Questões Gerais:
 - Os objetivos do sistema foram definidos?
 - Os requisitos estão claros e não ambíguos?
 - Foi fornecida uma visão geral da funcionalidade do sistema?
 - Foi fornecida uma visão geral das formas de operação do sistema?
 - O software e o hardware necessários foram especificados?
 - Se existe alguma suposição que afete a implementação ela foi declarada?
 - Para cada função, os requisitos foram especificados em termos de entrada, processamento e saída?
 - Todas as funções, dispositivos e restrições estão relacionadas aos objetivos do sistema e vice-versa?

Checklist

- Omissão de Funcionalidade
 - As funções descritas são suficientes para alcançar os objetivos do sistema?
 - As entradas declaradas para as funções são suficientes para que elas sejam executadas?
 - Foram considerados os eventos indesejáveis e as respostas a eles foram especificadas?
 - Foram considerados o estado inicial e os estados especiais (por ex. inicialização do sistema, término anormal)?
- Omissão de Desempenho
 - O sistema pode ser testado, analisado ou inspecionado para mostrar que ele satisfaz seus requisitos?
 - Os tipos de dados, unidades, limites e resolução foram especificados?
 - A frequência e volume de entrada e saída foram especificados para cada função?

Checklist

- Omissão de Interface
 - As entradas e saídas para todas as interfaces são suficientes?
 - Foram especificados os requisitos de interface entre hardware, software, pessoas e procedimentos?
- Omissão de Recursos do Ambiente
 - Foram especificadas de forma apropriada as funcionalidades de interação entre hardware, software com o sistema?
- Informação Estranha
 - Todas as funções especificadas são necessárias para alcançar os objetivos do sistema?
 - As entradas das funções são necessárias para executá-las?
 - As entradas e saídas das interfaces são necessárias?
 - As saídas produzidas por uma função são usadas por outra função ou transferidas para a interface externa?

Checklist

- Ambiguidade
 - Cada requisito foi especificado de forma discreta, não ambígua e testável?
 - Todas as transições do sistema foram especificadas de forma determinística?
- Inconsistência
 - Os requisitos estão consistentes entre si?
- Fato Incorreto
 - As funções especificadas são coerentes com o sistema e com os objetivos a serem alcançados?

Inspeção de Documento de Requisitos

- Baseada em Perspectiva
 - É um conjunto de técnicas de leitura que focam em determinados pontos de vista.
 - Fazem com que cada revisor se torne responsável por uma perspectiva em particular.
 - Possibilita que o revisor melhore sua experiência em diferentes aspectos do documento de requisitos.
 - Assegura que perspectivas importantes sejam contempladas.

Inspeção de Documento de Requisitos

- Baseada em Perspectiva
 - Cada revisor possui um “cenário” para guiar seu trabalho de revisão.
 - Todo “cenário” consiste de duas partes:
 - Construir um modelo do documento que está sob revisão a fim de aumentar o entendimento sobre o mesmo.
 - Responder questões sobre o modelo, tendo como foco itens e problemas de interesse da organização.
 - Cada revisor deve ler o Documento de Requisitos com olhos diferentes
 - Benefícios:
 - determina uma responsabilidade específica para cada revisor.
 - melhora a cobertura de defeitos.

Inspeção de Documento de Requisitos

- Perspectivas principais
 - o projetista que usa o DR para gerar o projeto do sistema.
 - o testador que deve gerar casos de teste para testar o sistema quando este estiver implementado
 - o usuário para verificar se o DR está capturando toda funcionalidade que ele deseja para o sistema

Leitura Baseada em Perspectiva

- Perspectiva do usuário
 - definir um conjunto de funções que o usuário esteja apto a executar.
 - definir o conjunto de entradas necessárias para executar cada função e o conjunto de saídas que são geradas por cada função.
 - isso pode ser feito escrevendo todos os cenários operacionais que o sistema deve executar.
 - iniciar com os cenários mais óbvios até chegar nos menos comuns ou condições especiais.
 - ao fazer isso, faça a você mesmo as seguintes perguntas:
 - sugestão: usar como modelo subjacente Casos de Uso

Leitura Baseada em Perspectiva

- Perspectiva do usuário (questões)
 - todas as funções necessárias para escrever os cenários estão especificadas no documento de requisitos ou na especificação funcional?
 - as condições iniciais para inicializar os cenários estão claras e corretas?
 - as interfaces entre as funções estão bem definidas e compatíveis (por ex., as entradas de uma função têm ligação com as saídas da função anterior?)
 - você consegue chegar num estado do sistema que deve ser evitado (por ex., por razões de segurança)?
 - os cenários podem fornecer diferentes respostas dependendo de como a especificação é interpretada?
 - a especificação funcional faz sentido de acordo com o que você conhece sobre essa aplicação ou sobre o que foi especificado em uma descrição geral?

Leitura Baseada em Perspectiva

- Perspectiva do testador
 - para cada especificação funcional ou requisito gere um ou um conjunto de casos de teste que faça com que você se assegure de que a implementação do sistema satisfaz a especificação funcional ou o requisito .
 - use a sua abordagem de teste normal e adicione critérios de teste.
 - ao fazer isso, faça a você mesmo as seguintes perguntas para cada teste:
 - sugestão: usar como critérios de teste Particionamento de Equivalência, Análise do Valor Limite

Leitura Baseada em Perspectiva

- Perspectiva do testador (questões)
 - você tem toda informação necessária para identificar o item a ser testado e o critério de teste? Você pode gerar um bom caso de teste para cada item, baseando-se no critério?
 - você tem certeza de que os teste gerados fornecerão os valores corretos nas unidades corretas?
 - existe uma outra interpretação dos requisitos de forma que o implementador possa estar se baseando nela?
 - existe um outro requisito para o qual você poderia gerar um caso de teste similar, mas que poderia levar a um resultado contraditório?
 - a especificação funcional ou de requisitos faz sentido de acordo com aquilo que você conhece sobre a aplicação ou a partir daquilo que está descrito na especificação geral?

Inspeção de Documento de Requisitos

- Relatório final: lista de defeitos

Nro Sequencial	Local no Doc. Requisitos	Tipo do Defeito	Descrição
1	RF5	O	Não discriminadas as informações necessárias para que seja feito o cadastro da pessoa.
2	RF12	A	Não fica claro qual a taxa que deve ser paga, no caso de atraso de livro

Trabalho – Inspeção de Requisitos

- Ver especificação no edisciplinas
- Entrega: 29 de Setembro de 2022

Referências

- Koscianski, André; Soares, Michel dos Santos. “Qualidade de Software”. Editora Novatec, 2007, 2ª edição.
 - Capítulo 9 – Influência dos Requisitos na Qualidade
- Livros de Engenharia de Software – Requisitos de Software

ACH2028- Qualidade de Software

AULA 06 – INSPEÇÃO DE REQUISITOS

Prof. Marcelo Medeiros Eler
marceloeler@usp.br