



Inspeção de Código Fonte

- Normalmente associa-se inspeção a código fonte, visando melhorar sua qualidade
- Utilizada para verificar:
 - Padrões de codificação
 - Estrutura dos algoritmos
- Para padrões de codificação, diversas ferramentas podem apoiar essa atividade:
 - Checkstyle, PMD, FindBugs, Dependency Finder, SQE
 - Existem plugins para algumas dessas ferramentas em diferentes IDEs

Inspeção de Código Fonte

Detecção de violação de código *Local Variable Names*

```
33 public String toString() {  
34     String Resultado = "(" + x + "," + y + ")";  
35     return  
36 }  
37
```

Local Variable Names: Nome 'Resultado' deve condizer com o padrão '^[a-z][a-zA-Z0-9]*\$'.
Press 'F2' for focus

Detecção de violação de código *Multiple Variable Declarations*

```
11 private double x, y;  
12  
13 public  
14 if ((x == outroPonto2D.x) && (this.y == outroPo
```

Multiple Variable Declaration: Each variable declaration must be in its own statement.
Press 'F2' for focus

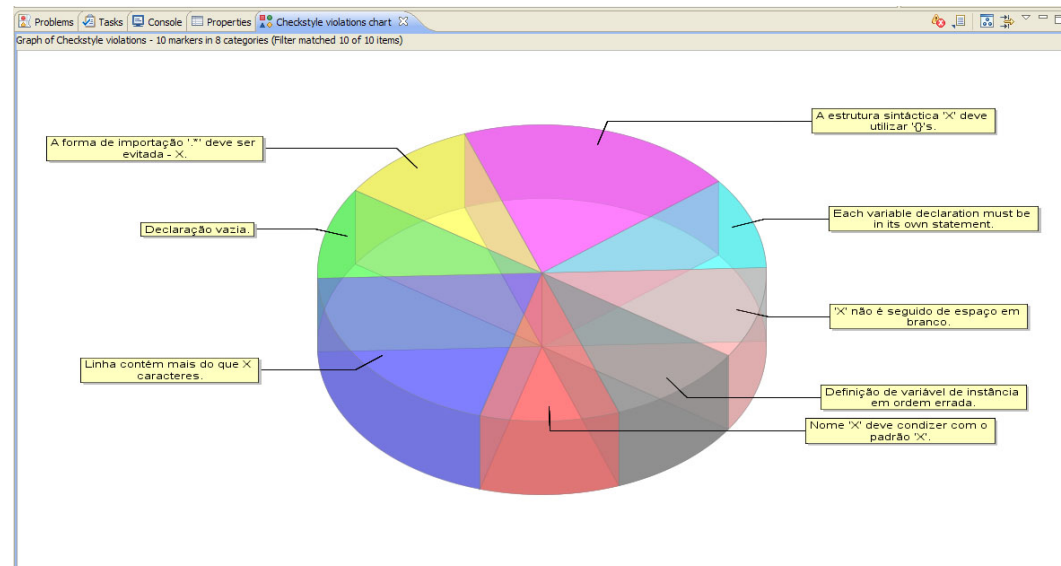
Detecção de violação de código *Maximum Line Length*

```
48 pt3 = pt1.clona();  
49 System.out.println("P3 está na mesma posição de P1?" + pt3.éIgual(pt1));  
50 pt4 = pt1.origem();  
51 System.out.println("P4 está na mesma posição de P1?" + pt4.éIgual(pt1));  
52 }  
53  
54
```

Maximum Line Length: Linha contém mais do que 80 caracteres.
Press 'F2' for focus

Inspeção de Código Fonte

Problems @ Javadoc Declaration				
0 errors, 10 warnings, 0 others				
Description	Resource	Path	Locat...	Type
Warnings (10 items)				
Avoid Nested Blocks: Avoid nested blocks.	Ponto2D.java	Qualidade/src	line 27	Checkstyle Problem
Avoid Star (Demand) Imports: A forma de importação '*' deve ser evitada - javax.swing.*.	Ponto2D.java	Qualidade/src	line 2	Checkstyle Problem
Declaration Order Check: Definição de variável de instância em ordem errada.	Ponto2D.java	Qualidade/src	line 11	Checkstyle Problem
Empty Statement: Declaração vazia.	Ponto2D.java	Qualidade/src	line 8	Checkstyle Problem
Maximum Line Length: Linha contém mais do que 80 caracteres.	Ponto2D.java	Qualidade/src	line 49	Checkstyle Problem
Maximum Line Length: Linha contém mais do que 80 caracteres.	Ponto2D.java	Qualidade/src	line 51	Checkstyle Problem
Multiple Variable Declaration: Each variable declaration must be in its own statement.	Ponto2D.java	Qualidade/src	line 11	Checkstyle Problem
Need Braces: A estrutura sintática 'else' deve utilizar '{}'.s.	Ponto2D.java	Qualidade/src	line 16	Checkstyle Problem
The import javax.swing is never used	Ponto2D.java	Qualidade/src	line 2	Java Problem
Whitespace After: ',' não é seguido de espaço em branco.	Ponto2D.java	Qualidade/src	line 6	Checkstyle Problem

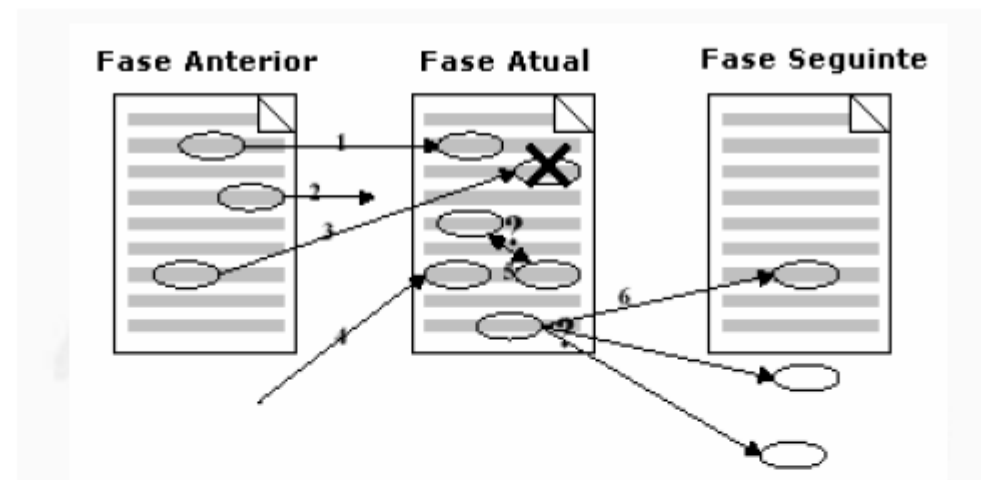




Inspeção de Requisitos

- Tem sido o tipo de revisão de software mais estudado e utilizado
- Objetivos
 - Identificar erros específicos em documentos e em sistemas
 - Identificar erros sistemáticos no processo de desenvolvimento
 - Identificar desvios em relação às especificações e padrões
- Benefícios
 - Provê ganhos significativos em relação a prazos e custos
 - Tende a achar mais defeitos que qualquer outro processo, e a um custo menor
- No início, era aplicado apenas a código fonte. Atualmente são amplamente utilizadas
- É um método eficaz para o controle da qualidade e aumento da produtividade

Defeitos no Processo de Desenvolvimento de Software



- 1. Informação transformada corretamente.
- 2. Informação perdida durante a transformação.
- 3. Informação transformada incorretamente.
- 4. Informação estranha introduzida.
- 5. Ocorrência de múltiplas transformações inconsistentes a partir de uma mesma fonte de informação.
- 6. Possibilidade de múltiplas transformações inconsistentes a partir de uma mesma fonte de informação.



Documento de Especificação de Requisitos de Software

- primeiro artefato tangível a ser produzido;
- descreve todas as características e as funções que o software a ser desenvolvido deve possuir;
- atua como um contrato entre o cliente e o desenvolvedor;
- base para todas as etapas subsequentes do desenvolvimento;
- normalmente escrito em linguagem natural.




Ambiguidade em linguagem natural

“Navegar é preciso
Viver não é preciso”


Pompeu, 56 a.C.

citado por Fernando Pessoa



Falta de precisão em linguagem natural

De acordo com uma pesquisa de uma universidade inglesa, não importa em qual ordem as letras de uma palavra estão, a única coisa importante é que a primeira e última letras estejam no lugar certo. O resto pode ser uma bagunça total, que você ainda pode ler sem problema. Isso é porque nós não lemos cada letra isolada, mas a palavra como um todo.



Falta de precisão em linguagem natural

- Se os jurados condenam, eu não absolvo
 - Se os jurados condenam eu não, absolvo
-
- Deixo os bens para Ana, não para Pedro
 - Deixo os bens para Ana não, para Pedro



Falta de precisão na linguagem natural

- **Aquela velha senhora encontrou o garotinho em seu quarto.**
O garotinho estava no quarto dele ou da senhora?
- **Sentado na varanda, o menino avistou um senhor.**
Quem estava sentado na varanda: o menino ou o senhor?
- **A polícia cercou o ladrão do banco na rua Santos.**
O banco ficava na rua Santos, ou a polícia cercou o ladrão nessa rua?
- **Onde está a cachorra da sua amiga?**
O animal criado pela amiga ou a amiga que é uma cachorra?
- **O barco chegou ao cais vazio**
O barco chegou vazio ao cais ou o barco chegou aos cais que estava vazio?
- **Homem é preso acusado de terrorismo nos EUA**
Preso nos EUA, mas praticou terrorismo em qualquer outro lugar, ou praticou terrorismo nos EUA e foi preso em algum lugar?



Tipos de Defeitos encontrados em Requisitos

Tipo	Percentagem(%)
Fato Incorreto	40
Omissão	31
Inconsistência	13
Ambigüidade	5
Localização Incorreta	2



Consequências de Falta de Qualidade em Requisitos

- estimativas de esforço e prazo deixam de fazer sentido;
- desperdício de recursos;
- produto final não atende as necessidades do usuário;
- corrigir defeitos após a entrega do produto pode ser até cem vezes mais caro que corrigi-los nas primeiras fases do desenvolvimento (em projetos menores, 5:1);
- em projetos recentes de software, teríamos um esforço de retrabalho entre 40% e 50% do esforço total.



Inspeções de Software

- **Benefícios e Custo de Inspeções**

- inspeções vêm sendo utilizadas há mais de duas décadas;
- existe evidência experimental de sua usabilidade e adequabilidade;
- provêem um bom meio para o gerente do projeto monitorar a qualidade e progresso do projeto;
- podem amenizar atividades de manutenção, evitando que erros se propaguem pelo ciclo de vida;
- apresentam baixo custo devido ao fato do revisor não precisar investir muito tempo ou mesmo não demandar ferramentas sofisticadas para realizá-las.
 - entretanto uma alta taxa de atividades de inspeção ao longo do processo pode acrescer de 5% a 10% o custo final

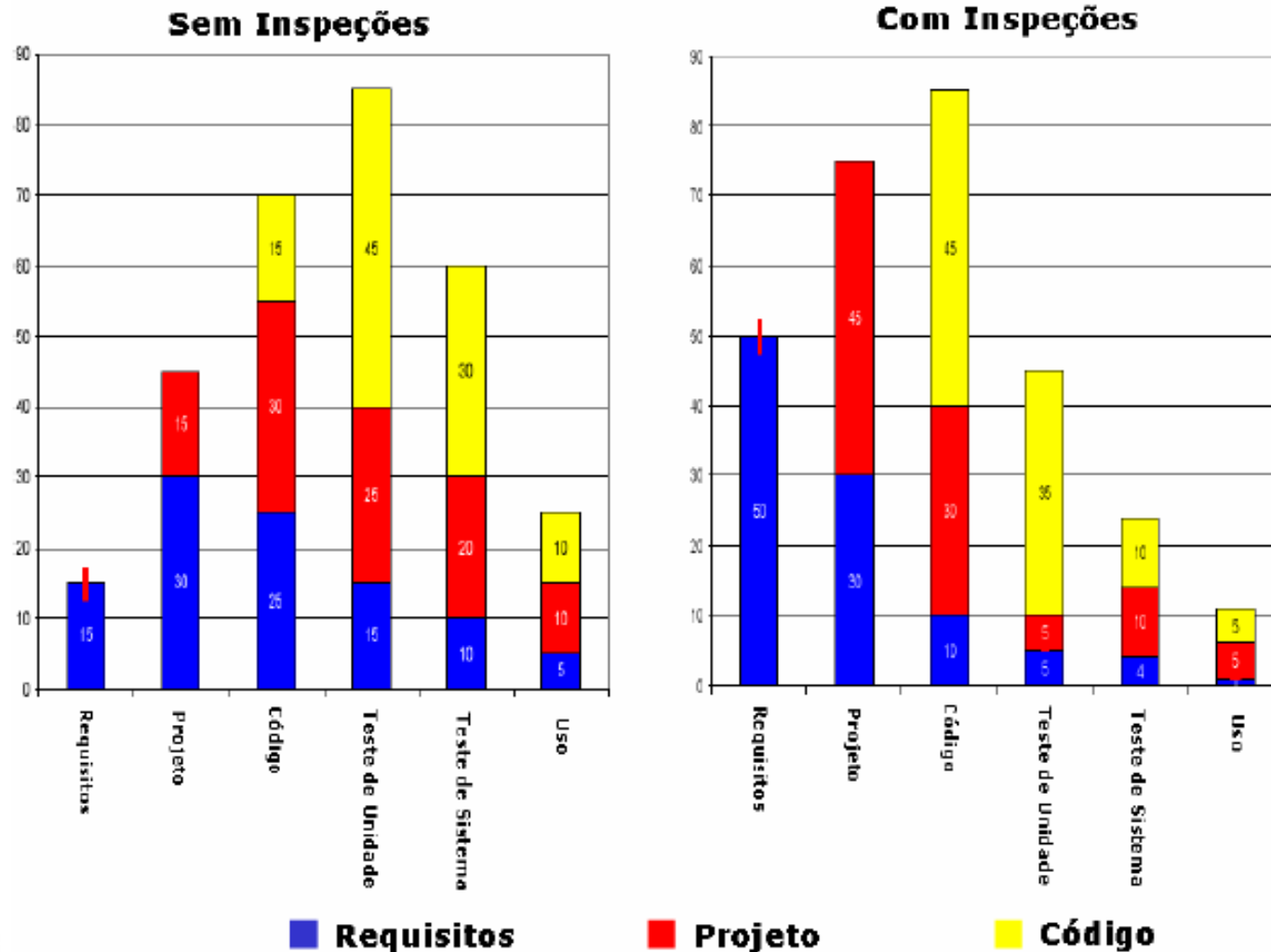


O que é uma inspeção

- Definição:
 - Método de análise estática para verificar as propriedades de qualidade de produtos de software.
- Dimensões:
 - Estruturada, processo bem definido
 - Pessoal técnico
 - Papéis bem definidos
 - Técnicas de leitura para identificação de defeitos
 - Resultados documentados

Benefícios de inspeções

- Identificação mais cedo de defeitos



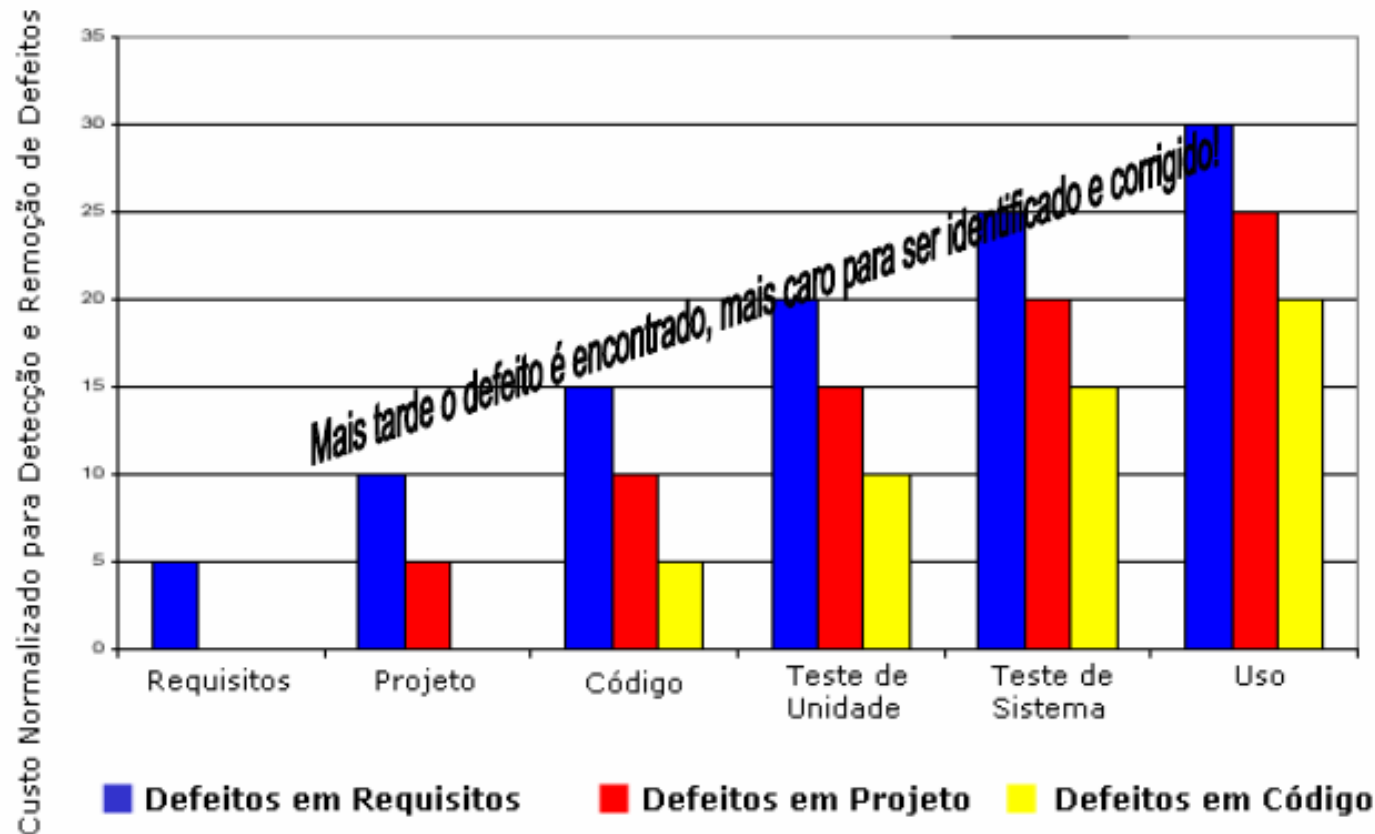
Benefícios de inspeções

- Inspeções em requisitos e projeto (Miller, 1990);

	Defeitos Req.	Defeitos Proj.	Outros	Total
Fase Requisitos	520			520
Fase Projeto	22	513		535
Fase Teste Unid.			409	409
Fase Teste Sist.			4	4
Total	542	513	413	1468

Benefícios de inspeções

- Teoria: Inspeções melhoram a produtividade encontrando defeitos nos momentos em que sua correção é mais barata.
 - Inspeções melhoram a produtividade e reduzem o custo



Defeitos em Requisitos

- Exemplo: Sistema de Controle para Postos de Gasolina

Requisito Funcional 5

- Se o pagamento for feito em dinheiro, o caixa é responsável por receber o pagamento e, caso necessário, devolver o troco. Quando o pagamento for feito o caixa fornece essa informação ao sistema. O sistema e a bomba de combustível retornam então para seu estado inicial.

Qual o valor da compra?

- Para poder completar a transação, o caixa precisa saber o valor da compra.
- Informação foi perdida durante a criação dos requisitos.
- Esta informação foi omitida da descrição da funcionalidade, logo temos um defeito.



Defeitos em Requisitos

- Requisito Funcional 3
 - Se o cliente optar pelo pagamento à vista, ele poderá pagar em dinheiro ou com cartão de crédito. Se escolher pagar em dinheiro, a bomba de combustível deve enviar mensagem instruindo o cliente a procurar o caixa. Se escolher pagar com cartão de crédito, a bomba deve instruir o cliente a inserir o cartão no leitor. Se o cliente não fizer uma opção ou fizer uma opção inválida, o sistema assume o pagamento com cartão de crédito como *default*.

A informação não foi traduzida corretamente.

O conhecimento do domínio indica que o pagamento com cartão de crédito não deve ser a opção *default*, logo temos um defeito.

Defeitos em Requisitos

- Requisito Funcional 2
 - Após o abastecimento, a bomba de combustível informa o valor ao sistema. O valor máximo é de R\$ 999,99.
- Requisito Funcional 4
 - Se o pagamento for feito com cartão de crédito, o leitor de cartão envia o número do cartão para o sistema. Se o sistema recebe um número de cartão inválido, então uma mensagem é enviada a bomba para o cliente introduzir o cartão novamente no leitor. Ao receber um número de cartão válido, o número do cartão e o valor da compra são enviados para o sistema da operadora do cartão, e o sistema e a bomba de combustível retornam para o seu estado inicial. O valor máximo de uma compra é de R\$1.000,00.

A informação foi descrita de forma inconsistente
Como o conhecimento de domínio não nos permite saber qual
a informação está correta, temos um defeito.

Defeitos em Requisitos

- Requisito Funcional 6.1
 - O cliente deve informar o número de sua conta ao caixa, que cadastra a informação no sistema. Se o número de conta for válido, então o número da conta, o valor da compra e uma breve descrição do tipo de transação são registrados (*log*). Se um número inválido for registrado, uma mensagem de erro é mostrada ao caixa, que deve informar novamente o número. O caixa deve ter também a opção de cancelar a operação. Neste caso, o valor da compra é mostrado novamente juntamente com as opções de pagamento (à vista ou a prazo).

A informação pode ser traduzida incorretamente.

Que informação é armazenada?

O que significa 'tipo de transação'? Combustível/Manutenção?

Cartão/Dinheiro?

Defeitos em Requisitos

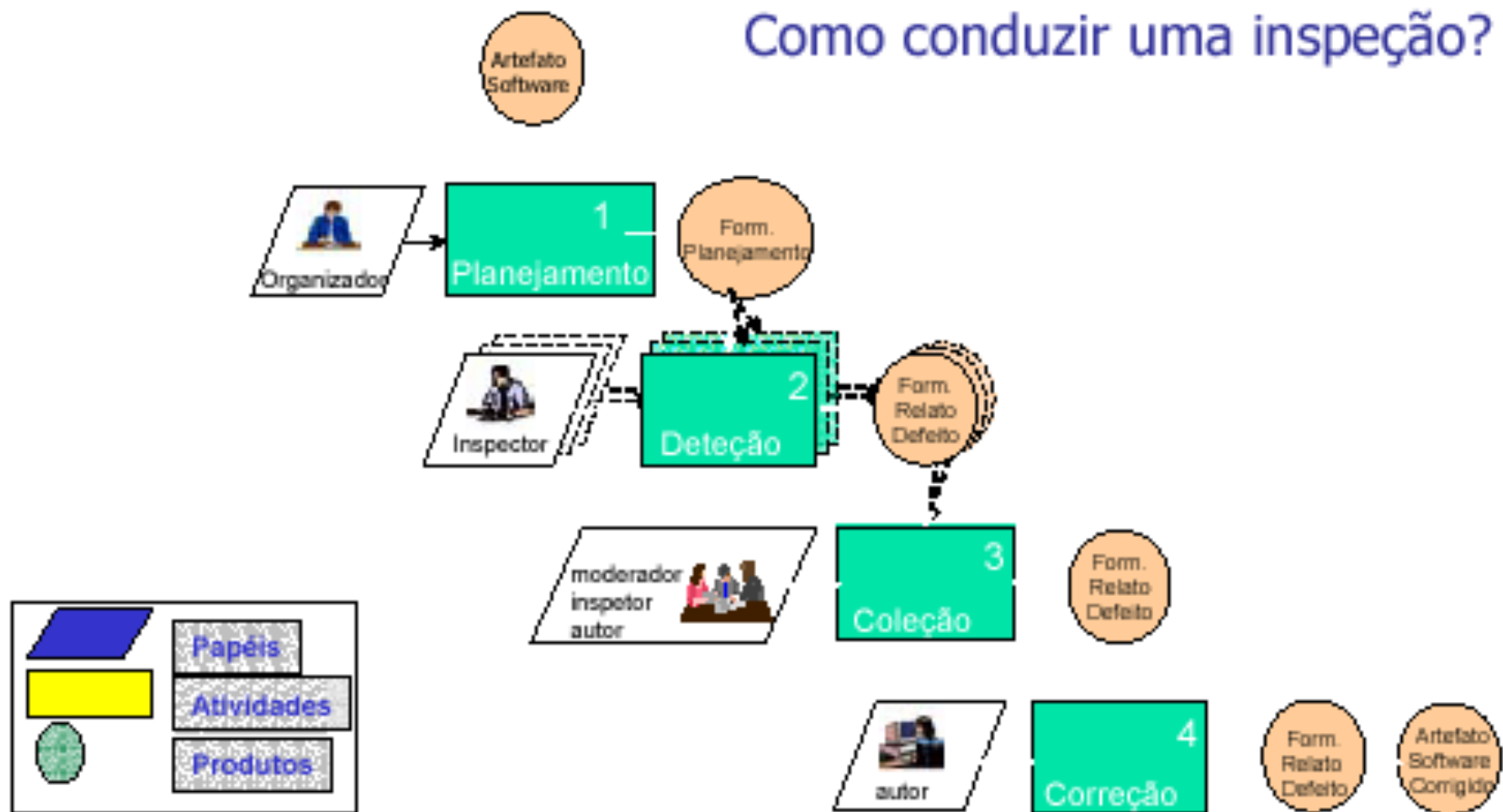
- Requisito Funcional 7
 - Para pagar uma conta mensal, o cliente deve enviar o pagamento junto com o número de sua conta mensal. O caixa seleciona este tipo de pagamento e o sistema envia uma mensagem ao caixa para que o número da conta mensal, o tipo de pagamento e o valor do pagamento sejam informados. Se alguma dessas informações forem inválidas, o sistema retorna para a tela anterior. Se o tipo de pagamento for cartão de crédito, o número do cartão também deve ser informado, e então o recibo do cartão deve ser foto-copiado e armazenado com os outros recibos.

Uma informação estranha foi introduzida.

Uma informação que, apesar de poder estar correta, pode causar problemas já que não faz parte do sistema.

Como isto pode ser testado?

O Processo de Inspeção de Requisitos





Inspeção *Ad-Hoc*

- Inspetor lê o documento de acordo com sua perspectiva
- Experiência individual afeta o resultado final:
 - Foco reside na especialidade do inspetor
 - Produtividade individual
 - Difícil garantir que inspetor leu o documento de forma adequada, pois cada um aplica “sua própria técnica”
- Não existe garantia de cobertura do documento como um todo
- Custo/Eficiência (# defeitos/tempo) pode ser adequado quando inspetores possuem experiência alta (> custo)



Inspeção com *Checklists*

- Inspetor segue uma lista de itens com características a serem revisadas, mas ainda aplica leitura *Ad-hoc* para identificar os defeitos
- Resultado final mais direcionado:
 - Características de qualidade definidas à priori
 - Produtividade individual
 - Difícil garantir que inspetor leu o documento de forma adequada, mesmo tendo sido definidas as características de qualidade a se procurar
- Cobertura do documento relacionada aos itens do *Checklist*
- Custo/Eficiência depende do *Checklist* e dos inspetores
- *Checklist* pode ser adaptado ou construído para capturar uma determinada especificidade



Lista de Verificação de Inspeção de Requisitos

- Os requisitos exibem uma clara distinção entre funções e dados?
- Os requisitos definem todas as informações a serem exibidas aos usuários?
- Os requisitos definem as respostas ao sistema e ao usuário às condições de erro?
- É cada requisito indicado claramente, conciso, e não ambíguo?
- É cada requisito testável?
- Há requisitos ambíguos ou inconsistentes?
- Há conflitos de requisitos?
- Existem áreas não identificadas na especificação de requisitos de software que necessitam ser?
- Os requisitos de desempenho (tais como o tempo de resposta, requisitos de armazenamento de dados) são indicados?
- Se os requisitos envolverem complexas cadeias de decisão, estão expressos em um formato que facilite sua compreensão (isto é, tabelas de decisão, árvores da decisão, etc..)?
- Os requisitos para executar melhorias do software foram especificados?
- Há requisitos que contém um nível desnecessário de detalhe do projeto?
- As restrições de tempo real foram especificadas em detalhe suficiente?
- A precisão e a exatidão dos cálculos foram especificadas?
- É possível desenvolver um conjunto completo de testes baseados nas informações contidas nas especificações de requisito do software? Se não, que informação falta?
- As suposições e as dependências foram indicadas claramente?
- O documento contém toda a informação identificada no escopo para a especificação de requisitos do software?

Taxonomia de Defeitos



Defeito	Descrição Geral
<i>Omissão</i>	Informação necessária sobre o sistema foi omitida do artefato de software
<i>Fato Incorreto</i>	Alguma informação no artefato de software contradiz informação do documento de requisitos ou o conhecimento geral do domínio.
<i>Inconsistência</i>	Informação contida em uma parte do artefato de software está inconsistente com outra informação no artefato de software
<i>Ambigüidade</i>	Informação contida no artefato de software é ambigua, isto é, várias interpretações podem ser derivadas da definição levando o desenvolvedor a implementação correta
<i>Informação Estranha</i>	Informação que é fornecida não é necessária ou nunca utilizada

Relato de Defeitos em Inspeção de Requisitos

Formulário para Relato de Defeito

Nome: J.J. XPT

Checklist utilizado: 01

Documento Revisado: Doc. De Requisitos da Ferramenta de construção de Casos de Uso para utilização na Técnica de Leitura PBR.

Tempo utilizado: 2 hs

Defeito No.	Página No.	Req. No.	Tipo Defeito	Descrição
1	2	RF 8	Omissão	Falta uma forma de consulta aos elementos do modelo presentes na ferramenta. Exemplo: estrutura em árvore ou diretórios
2			Omissão	Os requisitos não abordam tratamento de erros
3	3	RF 11/12	Ambigüidade	Não está muito clara a diferença entre os requisitos 11 e 12
4	2	RF 5	Ambigüidade	Usa-se no documento os termos participante e ator para descrever a mesma coisa.
5			Omissão	Falta descrição de como será a interface com o usuário e sua navegação.



Dicas para realizar uma boa inspeção

- Independente da abordagem utilizada, lembre-se:
 - Revise o produto, nunca o desenvolvedor
 - Defina uma agenda e cumpra-a rigorosamente
 - Limite o debate e a discussão
 - Mostre onde estão os problemas, mas não tente resolvê-los durante a inspeção
 - Não tente se lembrar, faça anotações sempre



Dicas para realizar uma boa inspeção

- Limite o número de participantes e motive a preparação antecipada da inspeção
- Desenvolva uma lista de tópicos para cada produto que deverá ser revisado
- Aloque recursos e tempo para a inspeção
- Forneça treinamento adequado a todos os revisores
- Aprimore-se: Revise suas revisões anteriores