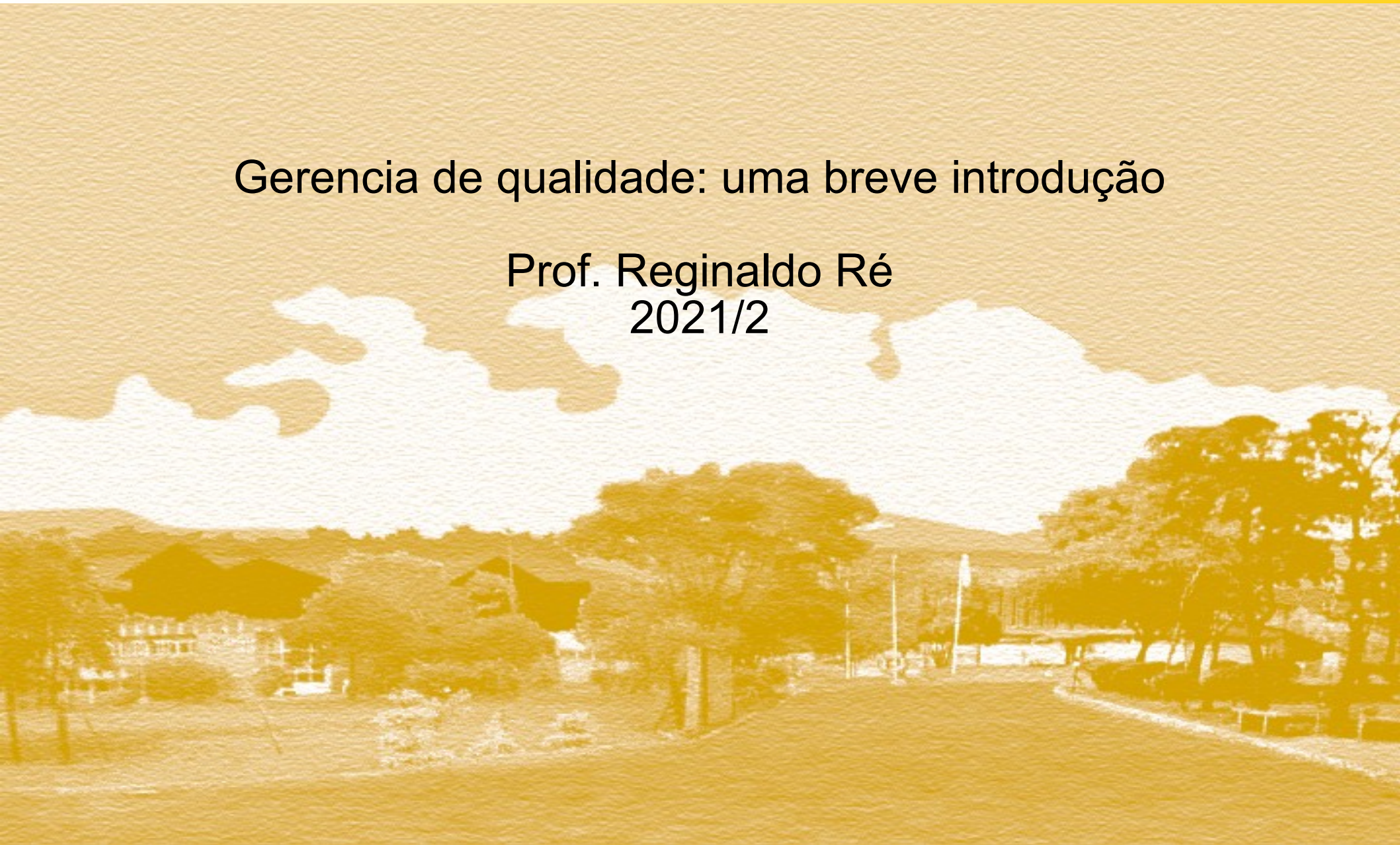


Gerencia de qualidade: uma breve introdução

Prof. Reginaldo Ré
2021/2



- Objetivos
- Recapitulação
- Terminologia IEEE 610.12-1990
- Revisões e inspeções
- Técnicas de Revisionar e Inspecionar
- Modos de Revisar
- Inspeções
 - Ferramentas de apoio
- Considerações finais

- Objetivo da aula
 - Apresentar técnicas de revisão, inspeção e leitura
- Justificativa
 - O controle da qualidade deve ser realizado continuamente
 - junto com o desenvolvimento
 - ainda antes de se dispor do código
 - Algumas propriedades do código são difíceis ou muito caras de testar
 - torna-se necessário o uso de técnicas de controle da qualidade alternativas aos testes
 - Revisões, inspeções e leitura dirigida, se bem realizadas, têm mostrado muito bons resultados

- Estruturado em quatro atividades principais:
 - Garantia de Qualidade de Software (SQA)
 - Planejamento da Qualidade de Software (SQP)
 - Controle de Qualidade de Software (SQC)
 - Consiste em supervisionar o processo de desenvolvimento a fim de assegurar que as normas, procedimentos e padrões de qualidade sejam seguidos pela equipe
 - Supervisiona se o que está no SQA e no SQP está sendo efetivamente seguido
 - Envolve uma série de **verificações, validações e testes**
 - Melhoria de Qualidade (QI)

Atividades do SQC

Relembrando...

- *Verificação*: controla de forma isolada a qualidade de cada artefato
- *Validação*: controla a qualidade dos inter-relacionamentos entre artefatos
 - Se passou pela verificação e pela validação, o artefato estará correto com relação à sua especificação e a outros artefatos
- *Aprovação*: controla a qualidade do artefato com relação às atuais necessidades e expectativas dos usuários
 - Se passou pela verificação, pela validação e pela aprovação, o artefato será, em princípio, uma implementação correta do problema correto

- Técnicas de controle sem execução do artefato
 - Prova formal da corretude, Argumentação da corretude
 - Revisões, Inspeções
 - Desenvolvimento em pares, teste estático, análise estática, medição estática
- Técnicas de controle com execução indireta
- Técnicas de controle com execução direta
 - Testes, Medição dinâmica, Instrumentação, Aprovação a cada iteração

- Artefato
 - Artefato (work product) é qualquer resultado tangível do desenvolvimento e que teve a sua qualidade verificada de alguma forma
 - Artefato é um conceito recursivo
 - artefatos podem ser compostos por outros artefatos
 - Diferentes representações e linguagens
 - Rede de linguagens de representação
 - Interdependência de representações

Terminologia

Padrão IEEE 610.12-1990

- Defeito (*fault*): passo, processo ou definição de dados incorretos. Ex.: um comando incorreto.
- Engano (*mistake*): ação humana que produz um resultado incorreto. Ex.: ação incorreta tomada por um programador.
- Erro (*error*): diferença entre o valor obtido e o valor esperado. Qualquer estado intermediário incorreto ou resultado inesperado.
- Falha (*failure*): produção de uma saída incorreta com relação à especificação.

Defeitos Encontrados em Revisões e Inspeções

Defeito	Descrição	Aplicado a projeto
Omissão	falta informação necessária no artefato	um ou mais requisitos ou características não são abordados no artefato
Incorreção	alguma informação contida no artefato conflita com o domínio do problema, ou com o serviço a ser por ele prestado	o artefato de projeto contém uma reificação errônea de um requisito
Inconsistência	alguma informação contida no artefato contradiz o que se encontra em outros artefatos ou mesmo neste artefato	um conceito registrado no artefato de projeto está em desacordo com o que se encontra em outros artefatos

Defeitos Encontrados em Revisões e Inspeções

Defeito	Descrição	Aplicado a projeto
Ambiguidade	Informação contida no artefato admite uma variedade de interpretações	Um conceito contido no artefato de projeto favorece dúvidas quanto ao seu significado, possibilitando um erro de compreensão.
Desnecessário	Informação contida no artefato não é utilizada ou não é necessária	O artefato de projeto contém informação que, mesmo se relevante, não deveria se encontrar nesse artefato (ex. erro de nivelamento da abstração)
Duplicata	Um mesmo conceito é definido ou especificado em dois ou mais artefatos ou locais de um mesmo artefato	O artefato de projeto repete a especificação de um mesmo conceito já existente em um ou mais outros artefatos

Revisões e Inspeções de Artefatos

- Uma **revisão** é uma leitura **crítica** do artefato e da documentação complementar, anotando:
 - os defeitos encontrados
 - as dúvidas e dificuldades de compreensão
 - as possibilidades de melhorias
- Uma **inspeção** é uma leitura crítica **formalizada** do artefato e da documentação complementar
 - segundo um plano definido
 - obedecendo a regras de leitura definidas
 - envolvendo uma pequena equipe
 - anotando defeitos, dúvidas e melhorias
- Revisões e inspeções são complementares a testes
 - ou ao contrário?

Revisões e Inspeções de Artefatos

- Revisões e inspeções podem ser utilizadas para controlar a qualidade de qualquer artefato em qualquer linguagem de representação destinada a humanos
 - Especificações,
 - Modelos,
 - Projetos,
 - Código, etc

Revisões e Inspeções de Artefatos

- Revisões e inspeções informam diretamente o defeito
 - em geral o custo é baixo para localizar completamente o defeito
 - podem ser realizadas com relação a artefatos não executáveis ou ainda incompletos
 - revisões **devem** ser praticadas a partir do primeiro momento do desenvolvimento
 - eliminam defeitos antes dos consequentes erros se propagarem para outros artefatos ou para o serviço
 - reduz significativamente o retrabalho inútil

Revisões e Inspeções São Eficazes

- Parcela significativa dos defeitos existentes em um artefato é encontrada através de revisões ou inspeções
 - uma parcela significativa dos defeitos de sistemas entregues têm origem em defeitos nas especificações
 - precisa-se reduzir esta classe de defeitos
 - segundo vários autores a inspeção, quando for praticada, encontra de 60% a 80% do total dos defeitos encontrados
 - mas isso não é de se esperar?
 - se o controle é feito antes dos testes, então sobram menos defeitos a serem encontrados através dos testes
 - também sobram menos defeitos remanescentes entregues ao usuário
 - alguns autores mencionam que se economiza perto de 40% do custo total de desenvolvimento quando se praticam inspeções
 - também é de se esperar, ou não? Afinal a inspeção reduz o custo do retrabalho inútil

Revisões e Inspeções vs. Testes

- Contrastando, testes informam o sintoma (falha) obrigando diagnosticar a causa (defeito)
 - custo alto para identificar a causa e localizar correta e completamente o defeito
 - somente podem ser realizados com relação a artefatos executáveis
 - ocorre tarde no desenvolvimento
 - depois de já se ter comprometido muitos recursos

Técnicas de Revisar ou Inspecionar

- Leitura simples
 - rever antes de prosseguir
 - habitue-se a fazer isso sempre
 - procure sempre ser rigoroso ao rever
 - ler para encontrar defeitos
- Leitura segundo critérios estabelecidos
 - padrões de uso das linguagens de representação
 - manual de critérios
 - pode envolver técnicas formais
 - argumentação, prova da correitude
- Leitura a partir de pontos de vista
 - segundo papéis desempenhados ou simulados pelo revisor

- Leitura baseada em pontos de vista encontram mais defeitos do que leitura não dirigida
 - leitura baseada em cenários
 - como se comporta o sistema nas condições A, B, C, ... ?
 - leitura usando a visão do papel desempenhado pelo observador
 - ponto de vista do testador
 - ponto de vista do mantenedor
 - ...
 - critérios explicitamente definidos a priori

- Do ponto de vista do testador
 - para cada requisito discriminado na especificação: quais seriam os casos de teste que você utilizaria para testar o requisito?
 - está claro como determinar o resultado esperado?
- Do ponto de vista do especificador
 - existem potenciais conflitos entre os requisitos?
 - faltam requisitos, o serviço está completamente descrito?
 - foram considerados requisitos funcionais e não funcionais?

- Do ponto de vista do implementador
 - quais seriam as estimativas de esforço para implementar cada requisito?
 - está claro o que é desejado?
- Do ponto de vista do usuário
 - preciso realmente deste requisito?
 - falta algum requisito (funcional ou não) de que necessito?

Check list do Ponto de Vista do Usuário

- Marque todos os elementos cuja especificação não deixe clara a sua intenção
- Marque todos os elementos com especificação ambígua
 - diferentes leitores podem entender coisas diferentes
- Marque todos os elementos com especificação imprecisa
 - ex. “deve ter bom desempenho”
- Marque todas as redundâncias de especificação
 - diferentes elementos especificam a mesma coisa
- Marque todos os elementos cuja abrangência possa ser reduzida sem comprometer o propósito do artefato
- Marque todos os elementos que possam ser excluídos sem
- comprometer as necessidades e expectativas do serviço
- ...

- Revisão pelo próprio autor (*desktop checking*)
 - o autor lê e anota todos os problemas encontrados para, depois, removê-los
 - inconvenientes intrínsecos:
 - o autor tende a “ler o que acha que está escrito e não o que está de fato escrito”
 - erros de entendimento por parte do autor não são observáveis
 - sujeito à síndrome da “idéia fixa”
 - O inconveniente pode ser atenuado utilizando técnicas formais (leves) para dirigir a leitura
 - a técnica formal induz uma forma alternativa (redundância útil) para observar o mesmo artefato

- Desenvolvimento em pares
 - XP
 - o desenvolvimento de um artefato é realizado sempre por duas pessoas trabalhando em conjunto à frente de um mesmo computador
 - um digita – escreve o código
 - outro monitora o que está sendo escrito
 - observa erros de digitação
 - observa erros de uso dos elementos do programa
 - observa desvios com relação à especificação ou ao projeto
 - observa não observância de boas práticas, padrões e normas
 - propõe soluções alternativas melhores
 - tem sido utilizado com sucesso ao desenvolver código
 - trabalho em grupo é também uma forma de desenvolvimento em “pares”
 - em geral utilizado para especificar e arquitetar sistemas

- Revisão por pares (peer review)
 - um dos parceiros (colega) do autor lê e anota todos os defeitos e demais problemas encontrados
 - ninguém deve ficar ofendido com as anotações, o que está em jogo é a qualidade do artefato!
 - ninguém deve deixar de anotar defeitos em virtude de algum receio de ofender o autor
 - isso não impede de ser bem educado
 - defeitos e outros problemas devem ser anotados em um documento (formulário)
 - pequenas melhorias podem ser anotadas no próprio artefato

- Revisões progressivas (round-robin)
 - seleciona-se um conjunto de parceiros que farão a revisão
 - cada um com uma determinada especialidade
 - cada um examinará a partir de um ponto de vista específico, exemplos de pontos de vista
 - cada parceiro lê e anota todos os problemas encontrados dentro de sua especialidade
 - a seguir passa a diante para o próximo parceiro da lista

- *Walk through*
 - realizado em uma reunião envolvendo um ou mais colegas
 - o autor percorre o artefato sendo revisado e narra o comportamento esperado
 - os parceiros
 - indicam erros
 - expõem dúvidas
 - propõem melhorias
 - propõem soluções alternativas a serem avaliadas
 - propõem possíveis soluções quando o autor estiver em dúvida ou estiver enganado
 - o autor, ou um parceiro designado para ser o relator, registra os problemas e soluções propostas

- Prós
 - Simplicidade
 - Eficácia
 - apesar de informais, revisões tendem a encontrar um número significativo de defeitos
 - se feitas por pessoas treinadas em aspectos formais (argumentação da corretude), são muito mais eficazes
 - Eficiência
 - em uma única revisão identifica-se uma quantidade grande de defeitos
 - Baixo custo
 - o custo da revisão é amplamente compensado pela redução do retrabalho inútil
 - muitas coisas podem ser automatizadas

- Contrás
 - a qualidade da revisão depende excessivamente da habilidade dos revisores
 - Proficiência
 - Cultura
 - a confiabilidade depende do rigor adotado pelo revisor
 - frequentemente não é repetível
 - revisor muda de opinião
 - diferentes revisores têm diferentes opiniões
 - a falta de treinamento dos revisores amplifica os problemas decorrentes da informalidade

- Uma inspeção é uma leitura crítica do artefato e da documentação complementar
 - segundo um plano definido
 - obedecendo a regras de leitura definidas
 - envolvendo uma pequena equipe
- Inspeções diferenciam-se das revisões por serem mais formalizadas e serem mais eficazes
 - são também (bem?) mais caras

- Papéis desempenhados
 - Autor
 - tem treinamento nos critérios a serem utilizados na avaliação
 - critérios visam aproximar-se do desenvolvimento correto por construção
 - desenvolve o artefato
 - idealmente segundo um processo definido e visando os critérios
 - escolhe a equipe de inspeção específica para o artefato e marca uma data para a reunião de inspeção
 - distribui com suficiente antecedência aos membros do comitê o material relativo ao artefato e as informações complementares
 - Inspetores, dois ou três
 - têm treinamento nas técnicas e critérios utilizados na avaliação
 - recebem com antecedência o material a ser inspecionado
 - lêem e anotam
 - defeitos encontrados, dúvidas e dificuldades de compreensão, possibilidades de melhorias

- Papéis desempenhados, cont.
 - Moderador
 - verifica se estão satisfeitas as condições para realizar a reunião
 - conduz a reunião
 - procura manter o foco da discussão
 - produz um laudo com as observações (se não tiver secretário)
 - deve ser um desenvolvedor sênior, mas não gerente
 - opcionalmente um secretário
 - produz um laudo (relatório) com as observações
 - opcionalmente um controlador da qualidade
 - verifica se os quesitos de qualidade foram devidamente abordados

- Observações
 - O objetivo de uma inspeção é examinar a qualidade do artefato
 - Inspeções não se destinam a participantes mostrarem que sabem muito
 - Inspeções jamais devem ser usadas para avaliar os autores dos artefatos inspecionados
 - gerentes não deveriam participar das reuniões

- Plano de inspeção, etapas
 - Produzir sinopse
 - apresentação concisa do artefato, redigida pelo autor
 - Organizar a inspeção
 - o autor seleciona a equipe de inspeção
 - o autor marca a data e horário
 - todos que concordaram devem comparecer e ser pontuais
 - Realizar a leitura individual
 - o autor distribui aos revisores o material a inspecionar e o de apoio
 - os revisores (pares) lêem e anotam os problemas observados
 - Coletar e filtrar as observações
 - todos: reunião de inspeção
 - resulta no laudo da inspeção
 - Corrigir de acordo com o laudo
 - o autor faz as correções segundo o laudo
 - Acompanhar o progresso
 - são registrados os eventos: distribuição, reunião, conclusão
 - é acompanhada a resolução dos problemas

Inspeções, execução

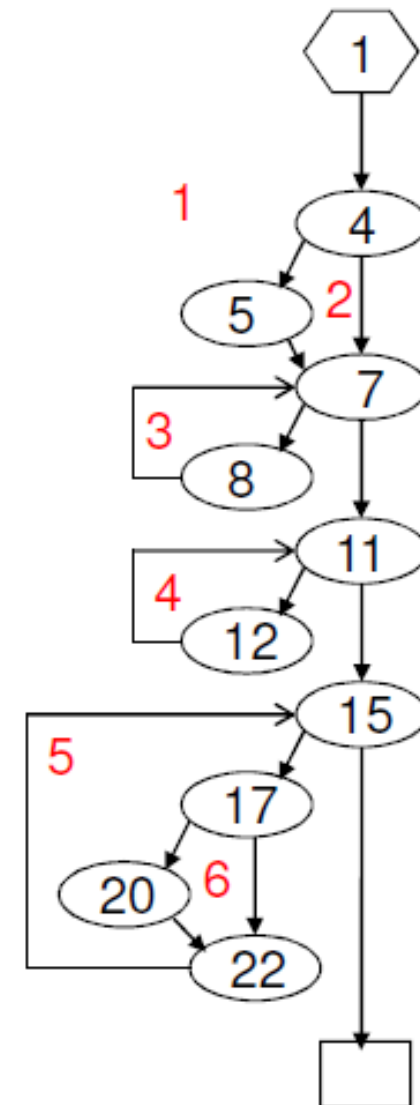
- Durante a reunião
 - o autor narra o comportamento do artefato
 - os parceiros que já leram a documentação fornecida
 - solicitam explicações
 - indicam a existência de problemas
 - conforme as anotações deles
 - os identificados durante a apresentação
 - sugerem melhorias
 - é produzido um laudo com todos os problemas e sugestões observados
- Após a reunião
 - autores fazem as correções
 - possivelmente modificam coisas que não haviam sido anotadas
 - dependendo da gravidade, repete-se a inspeção

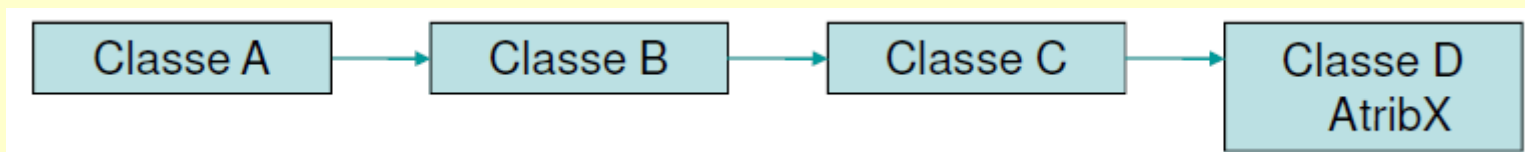
- Medições Estáticas
 - medem propriedades e indicam os fragmentos que têm mais chance de gerar problemas
 - produzem indicadores de possíveis problemas (*bad smells*), ex
 - estilo de programação que tende a ser defeituoso (*fault proneness*)
 - estilo de programação difícil de manter
 - estilo de programação difícil de testar
 - existem alguns (poucos) indicadores para projetos e especificações
 - propriedades cujo risco foi avaliado através de experimentos
 - número ciclomático (McCabe)
 - Acoplamento
 - Coesão
 - Encapsulamento
 - *Fan in* e *Fan out*

```

2  pValor = ( char * ) pValorParm ;
3  numBytes  = TamValor - Offset ;
4  if ( numBytes > DIM_LINHA ) {
5      numBytes = DIM_LINHA ;
6  } /* if */
7  for( i = 0 ; i < numBytes ; i ++ ) {
8      ValChar = *( pValor + Offset + i ) ;
9      fprintf( pArqLog , " %02X" , ValChar ) ;
10 } /* for */
11 for( ; i < DIM_LINHA ; i ++ ) {
12     fprintf( pArqLog , "    " ) ;
13 } /* for */
14 fprintf( pArqLog , "    " ) ;
15 for( i = 0 ; i < numBytes ; i ++ ) {
16     Ch = *( pValor + Offset + i ) ;
17     if ( ( Ch < 32 )
18         || ( Ch == 127 )
19         || ( Ch == 255 ) ) {
20         Ch = '.' ;
21     } /* if */
22     fprintf( pArqLog , "%c" , Ch ) ;
23 } /* for */

```





- a métrica *número de classes servidoras requeridas* deve ser pequena (Lei de Demeter)
 - o que é pequeno?
 - deve ser evitado:
 - `ObjA.metodoX(){ return ObjB->ObjC->ObjD->atribX ; }`
- reorganização (refatoração) proposta:
 - procure realizar a operação na classe que define o atributo
 - substitua atributos públicos por métodos (*getters*)

- Análise estática
 - examinam propriedades a partir da análise do código
 - a partir do grafo de chamadas envolvendo o programa
 - que função ou método chama que outra função ou método?
 - que **catch** trata de um dado **throw**?
 - existem pontos de corte que permitam empacotar componentes?
 - algumas ferramentas:
 - a primeira delas: *lint* para UNIX
 - ESC/Java, findbugs, **PolySpace**, **Coverity Prevent**, **Klocwork**,
 - Cppcheck

Ferramentas de apoio a inspeções

- **Análise dinâmica** (incluído aqui para confronto com análise estática)
 - na realidade não apoia inspeções
 - verifica propriedades relevantes durante a execução
 - Exemplos
 - satisfação de condições
 - redundâncias explicitamente inseridas
 - assertivas executáveis, verificadores
 - controle dinâmico de tipos
 - vazamento de recursos
 - violação de espaços e ponteiros
 - avaliação do desempenho
 - algumas ferramentas
 - **valgrind** para C++ / UNIX; **Insure++**
 - instrumentação contida no código

- Estude e use padrões e técnicas de revisão ou inspeção
 - melhora a eficácia das revisões e inspeções reduz a frequência de enganos -> reduz a inserção de defeitos
 - mais qualidade sem perda de produtividade ou de aumento de custos em virtude da redução de retrabalho inútil!
 - treinamento deve ser adequado às características da organização ou projeto
 - os critérios usados devem amoldar-se à natureza do problema, à cultura da organização e à qualidade requerida
- O treinamento em padrões de projeto e de programação genéricos é requerido como parte da formação de bons programadores

- Utilize técnicas formais leves
 - também faz parte da formação de bons programadores
 - assertivas e argumentação da correteza
 - especificar, projetar e programar utilizando técnicas e modelos suficientemente formais
 - operar com ferramentas
 - verificadores de modelos (*model checkers*)
 - verificadores de código, analisadores estáticos, medidores estáticos
 - transformadores que convertem modelos em outros de nível de abstração mais baixo ou redigidos em outra linguagem de representação, possivelmente esqueletos de código
 - geradores que convertem modelos em código compilável
 - grande maioria converte para esqueletos de código que precisam ser preenchidos pelo desenvolvedor
 - a geração de esqueletos frequentemente dificulta a manutenção

- Verificação, validação e teste
- Análise dinâmica e análise estática
- Revisões e Inspeções
 - Uma **revisão** é uma leitura **crítica**
 - **Inspeção** é uma leitura crítica **formalizada**
 - podem ser utilizadas para controlar a qualidade de qualquer artefato em qualquer linguagem de representação destinada a humanos