

Faculdade de Ciências e Tecnologia  
Departamento de Matemática e Computação  
Bacharelado em Ciência da Computação

unesp

UNIVERSIDADE ESTADUAL PAULISTA  
"JÚLIO DE MESQUITA FILHO"

Engenharia de Software II

Aula 02

Prof. Dr. Rogério Eduardo Garcia  
(rogerio.garcia@unesp.br)

1

07/08/2025

unesp

Contextualizando...  
ISO 12207: Estrutura

Prof. Dr. Rogério Eduardo Garcia

Processos Fundamentais

Aquisição

Fornecimento

Desenvolvimento

Operação

Manutenção

Processos de Apoio

Documentação

Gerenciamento de Configuração

Garantia de Qualidade

Verificação

Validação

Revisão Conjunta

Auditoria

Resolução de Problemas

Processos Organizacionais

Gerência

Melhoria


Infra-estrutura

Treinamento

Adaptação

2

07/08/2025

 **Como atingir a qualidade?**


Prof. Dr. Rogério Eduardo Garcia

- Revisão
  - As revisões de software são como um “filtro” para a gestão de qualidade. Isso significa que as revisões são aplicadas em várias etapas durante o processo de engenharia de software e servem para revelar erros e defeitos que podem ser eliminados.
  - As revisões de software “purificam” o resultado do trabalho da engenharia de software, até mesmo os modelos de requisitos e de projeto, dados de teste e código

3

3

07/08/2025

 **Por que Revisar?**

Prof. Dr. Rogério Eduardo Garcia

- Erro é humano.
- Embora as pessoas sejam boas para “descobrir” alguns de seus próprios erros, vários tipos de erros escapam mais facilmente daquele que os cometeu do que de outras pessoas externas.

4

4

07/08/2025  
unesp

## Para que Revisar?

- Revisão é uma forma de usar a diversidade de um grupo de pessoas para:
  - 1. Apontar aperfeiçoamentos necessários no produto de uma única pessoa ou de uma equipe.
  - 2. Confirmar aquelas partes de um produto em que aperfeiçoamentos são indesejáveis ou desnecessários.
  - 3. Obter trabalho técnico de qualidade mais uniforme, ou pelo menos mais previsível; qualidade que possa ser alcançada sem revisões, de modo a tornar o trabalho técnico mais gerenciável.

5

5

07/08/2025  
unesp

## Terminologia



### Bugs, erros e defeitos

O objetivo do controle da qualidade de software e da gestão da qualidade em geral é, em sentido mais amplo, eliminar problemas de qualidade no software. Tais problemas são conhecidos por diversos nomes — bugs, falhas, erros ou defeitos, apenas para citar alguns. Esses termos são sinônimos ou existem diferenças sutis entre eles?

Neste livro é feita uma distinção clara entre erro (um problema de qualidade encontrado antes de o software ser liberado aos usuários finais) e defeito (um problema de qualidade encontrado apenas depois de o software ter sido liberado aos usuários finais<sup>1</sup>). Essa distinção é feita porque os erros e os defeitos podem acarretar impactos econômicos, comerciais, psicológicos e humanos muito diferentes. Os engenheiros de software têm a missão de encontrar e corrigir o maior número possível de erros antes dos clientes e/ou usuários finais. Devem-se evitar defeitos — pois (de modo justificável) criam uma imagem negativa do pessoal de software.

### INFORMAÇÕES


É importante notar, entretanto, que a distinção temporal entre erros e defeitos feita neste livro não é um pensamento dominante. O consenso geral na comunidade de engenharia de software é que defeitos e erros, falhas e bugs são sinônimos. Ou seja, o momento em que o problema foi encontrado não tem nenhuma influência no termo usado para descrevê-lo. Parte do argumento a favor desta visão é que, muitas vezes, fica difícil fazer uma distinção clara entre pré e pós-entrega (consideremos, por exemplo, um processo incremental usado no desenvolvimento ágil).

Independentemente da maneira escolhida para interpretar esses termos ou do momento em que um problema é descoberto, o que importa efetivamente é que os engenheiros de software devem se esforçar — *multissimmo* — para encontrar problemas antes que seus clientes e usuários finais os façam. Caso tenha maior interesse nessa questão, uma discussão razoavelmente completa sobre a terminologia envolvendo bugs pode ser encontrada em [www.softwaredevelopment.ca/bugs.shtml](http://www.softwaredevelopment.ca/bugs.shtml).

6

6

07/08/2025

 **Revisões**

Prof. Dr. Rogério Eduardo Garcia


- São atividades desenvolvidas com o objetivo de:
  - (1) descobrir erros na função ou lógica de qualquer produto de software;
  - (2) verificar que o produto de software satisfaz seus requisitos e os do cliente;
  - (3) garantir que tenha sido representado de acordo com os padrões;
  - (4) tornar o projeto mais administrável.”

(Pressman, 2011)

7

7

07/08/2025

 **Revisões**


Prof. Dr. Rogério Eduardo Garcia

- Em geral são classificadas em:
  - Revisões técnicas formais
    - Inspeção
    - *Walkthrough*
  - *Peer review*
  - Opiniões de especialistas
- Provê dados para
  - ações gerenciais
  - melhoria do produto

8

8

07/08/2025

 **Revisões**

Prof. Dr. Rogério Eduardo Garcia


- Diretrizes para uma RTF (cont.):
  - Revisar o produto, não o produtor
  - Estabelecer e seguir agenda elaborada
  - Focar na detecção ou prevenção de defeitos e não na solução dos mesmos
  - Limitar duração da sessão em 2h
  - Documentar resultados
  - Acompanhar plano de ação

(Pressman, 2011)

9

9

07/08/2025

 **Revisões**

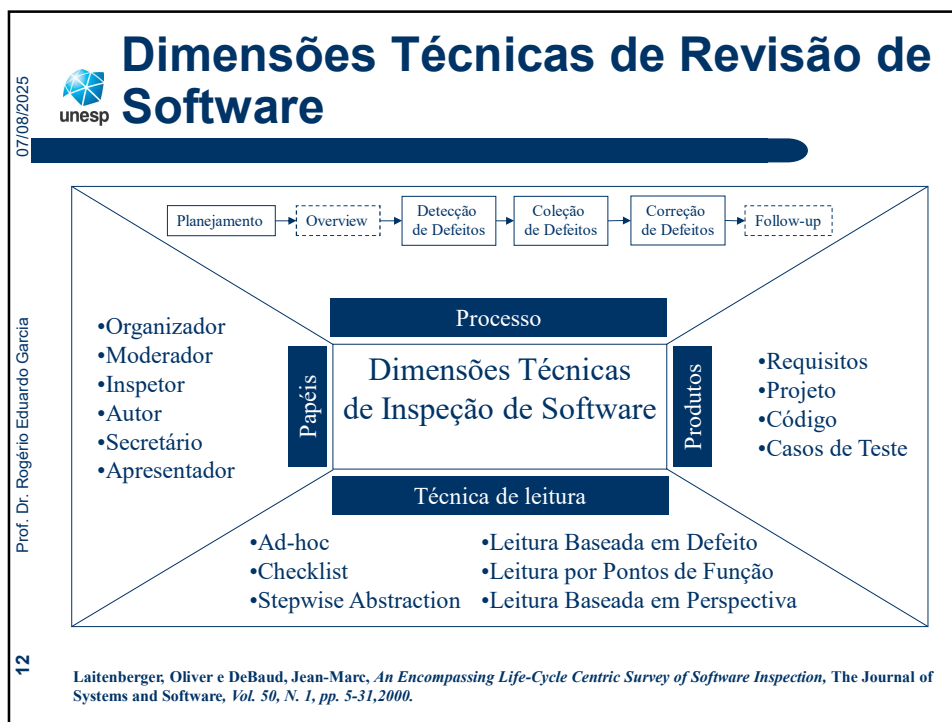
Prof. Dr. Rogério Eduardo Garcia

- Diretrizes para uma RTF:
  - Utilizar checklists de referência para revisão
  - Treinar os recursos humanos para desempenhar atividade
  - Avaliar periodicamente a efetividade dos resultados encontrados
  - Planejar as RTF como parte do cronograma do projeto
  - Limitar tamanho da equipe

(Pressman, 2011)

10

10



12

07/08/2025 unesp

## Inspeção de Software

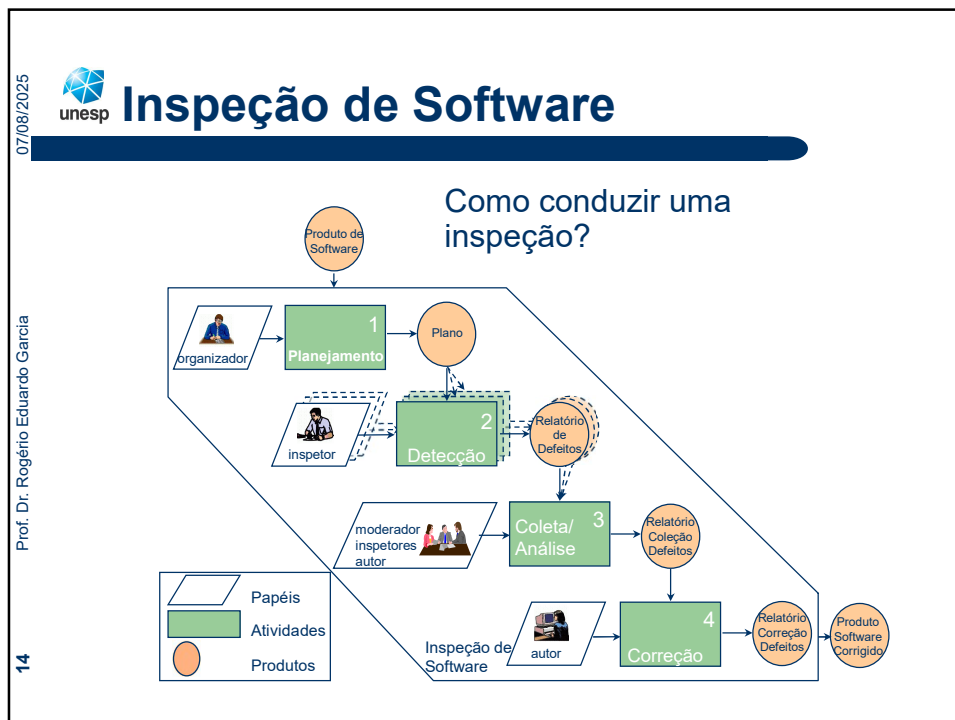
- Método de análise estática para verificar propriedades de qualidade de produtos de software.
- Características:
  - Processo estruturado e bem definido.
  - A equipe de inspeção consiste, geralmente, de pessoal técnico.
  - Os participantes possuem papel bem definido.
  - Os resultados da inspeção são registrados.

Prof. Dr. Rogério Eduardo Garcia

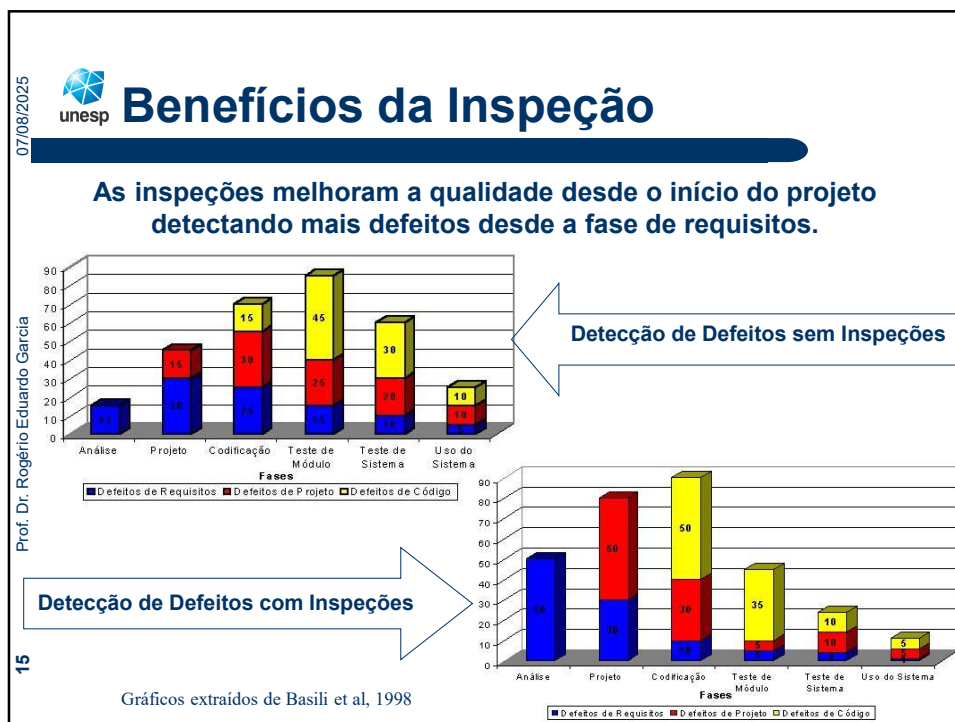
13

<http://www.cs.umd.edu/~mvz/mswe609/shull.pdf>

13



14



15

07/08/2025

Prof. Dr. Rogério Eduardo Garcia

16

## Benefícios da Inspeção

**Inspeções melhoram produtividade por acharem defeitos quando eles são mais baratos para corrigir.**

Fases	Requisitos	Projeto	Código
Análise	5	0	0
Projeto	10	5	0
Código	15	10	5
Teste de Módulo	20	15	10
Teste de Sistema	25	20	15
Uso do Sistema	30	25	20

Gráfico extraído de Basili et al, 1998

Quanto mais tarde um defeito é encontrado, mais caro torna-se localizá-lo e corrigi-lo!!

16

07/08/2025

Prof. Dr. Rogério Eduardo Garcia

17

## Benefícios Qualitativos da Inspeção

- Aprende-se pela experiência:
  - Participantes aprendem os padrões e o raciocínio utilizado na detecção de defeitos.
  - Participantes aprendem bons padrões de desenvolvimento.
- A longo prazo:
  - A inspeção convence os participantes a desenvolverem produtos mais compreensíveis e mais fáceis de manter.

**As inspeções ajudam a integrar o processo de prevenção de defeitos com o processo de detecção de defeitos.**

17



07/08/2025

Prof. Dr. Rogério Eduardo Garcia

18

## Defeitos de Software

**Situação ideal:**

1. A informação é transformada corretamente.

**Tipos de Erros:**

2. A informação é perdida durante a transformação.
3. A informação é transformada incorretamente.
4. Informação estranha é introduzida.
5. A mesma informação é transformada em diversas ocorrências inconsistentes.
6. A mesma informação possibilita diversas transformações inconsistentes.

18

07/08/2025

Prof. Dr. Rogério Eduardo Garcia

19

## Taxonomia de Defeitos

- **Definição:** são as classes de defeitos que serão usadas para classificar os defeitos encontrados.
- **Classes:**
  - **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):** quando a informação pode levar a múltiplas interpretações.
  - **Informação Estranha (IE):** qualquer informação que, embora relacionada ao domínio, não é necessária para o sistema em questão.

19

07/08/2025

Prof. Dr. Rogério Eduardo Garcia

20

## Exemplo: Omissão

- Omissão de Funcionalidade:
  - Informação que descreva algum comportamento desejado do sistema foi omitida do Documento de Requisitos (DR).
  - Ex: considere um sistema de biblioteca e os seguintes requisitos funcionais (RF):
    - ....
    - 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.
    - .....

Qual informação é necessária para possibilitar uma inserção incompleta?

20

07/08/2025

Prof. Dr. Rogério Eduardo Garcia

21


## Exemplo: Omissão

- 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.
  - Ex: considere o seguinte Requisito Não Funcional (RNF):
    - RNF1: o sistema deve fornecer os resultados tão rápido quanto possível.

?

21

07/08/2025



## Exemplo: Outros Tipos de Omissão


Prof. Dr. Rogério Eduardo Garcia

22

- 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.
- Omissão de Recursos do Ambiente:
  - Quando informação que descreva o hardware, software, base de dados ou detalhes do ambiente operacional no qual o sistema vai rodar for omitida.

22

07/08/2025



## Exemplo: Fato Incorreto

Prof. Dr. Rogério Eduardo Garcia

23

- Informação que consta do artefato mas que seja contraditória com o conhecimento que se tem do domínio da aplicação.
- Ex: considere um Sistema de Empréstimo numa Biblioteca e o seguinte RF:
  - ....
  - RF30: o sistema não deve aceitar devolução de livros se o usuário não tiver a carteirinha da biblioteca no momento.


**Para devolução de livros não é necessário apresentar a carteirinha pois todas as informações estão registradas no sistema !**

23

07/08/2025  
 unesp  
 Prof. Dr. Rogério Eduardo Garcia  
 24

## Exemplo: Inconsistência

- Informação que consta do artefato mais de uma vez e, em cada ocorrência, ela é descrita de forma diferente.
- Ex: considere um Sistema de Empréstimo numa Biblioteca e o seguinte RF:
  - ....
  - FR5: o sistema não deve permitir períodos de empréstimo maiores que 15 dias.
  - ....
  - FR9: professores podem retirar livros por um período de 3 semanas.
  - .....



24

07/08/2025  
 unesp  
 Prof. Dr. Rogério Eduardo Garcia  
 25

## Exemplo: Ambiguidade

- Quando a informação pode levar a múltiplas interpretações.
- Ex: considere um Sistema de Empréstimo numa Biblioteca e o seguinte RF:
  - ....
  - FR20: 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.

Qual a taxa a ser paga se o período for de uma semana?

No primeiro caso, a taxa deve ser calculada por dia?

25

07/08/2025

unesp

## Exemplo: Informação Estranha

- Qualquer informação que, embora relacionada ao domínio, não é necessária para o sistema em questão.
- Ex: considere um Sistema de Empréstimo numa Biblioteca e o seguinte RF:
  - ....
  - RF15: quando um novo livro é adicionado ao acervo, ele permanece em uma prateleira especial por um período de um mês.
  - ....

26

Essa informação não é necessária ao sistema !!

26

07/08/2025

unesp

## Inspeção de Software

RELATÓRIO DE DEFEITOS

Revisor ID: \_\_\_\_\_ Data: \_\_\_\_/\_\_\_\_/\_\_\_\_

Página 1 de 1

Nome do Documento: ABC Video System

Defeitos			
# Defeito	Página	Classe	Descrição
1	10	O	FR10: O que significa "atualizar" os arquivos diferentes não é especificado.
2	9	O	FR6: Não é especificado que o número da conta deve ser informado antes do número de uma fita.

27

27

07/08/2025

unesp

Técnicas de Leitura para Inspeção

Prof. Dr. Rogério Eduardo Garcia

28

- Questão: Como detectar defeitos?
  - Lendo o documento
  - Entendendo o que o documento descreve
  - Verificando as propriedades de qualidade requeridas

28

07/08/2025

unesp

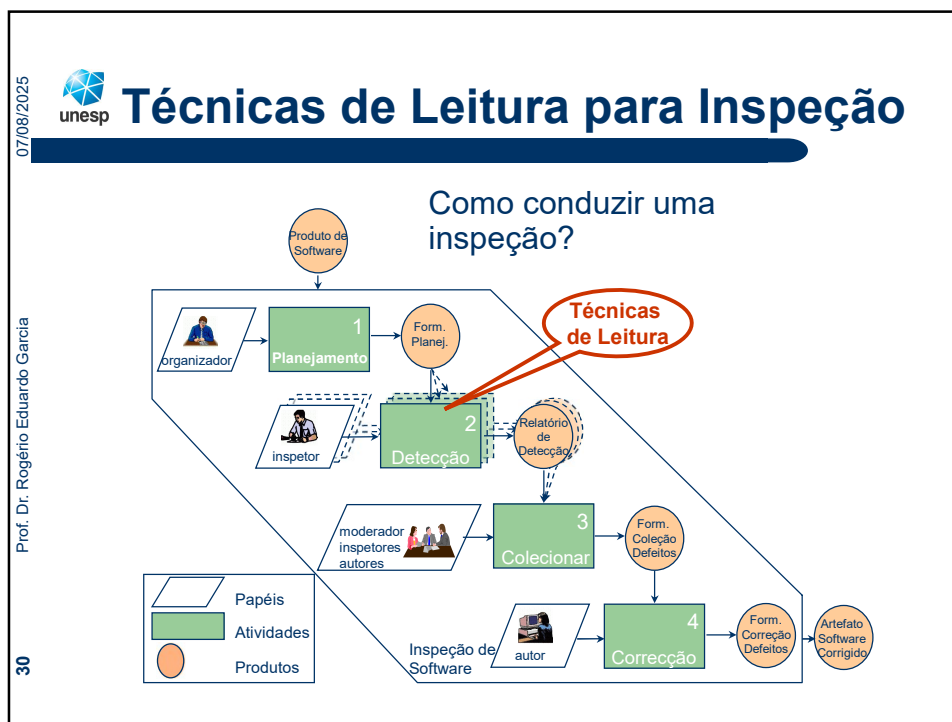
Técnicas de Leitura para Inspeção

Prof. Dr. Rogério Eduardo Garcia

29

- Problema:
  - em geral não se sabe como fazer a leitura de um documento !
- Razão:
  - em geral, os desenvolvedores aprendem a escrever documento de requisitos, código, projeto, mas não aprendem fazer uma leitura adequada dos mesmos.
- Solução:
  - fornecer técnicas de leitura bem definidas.
- Benefícios:
  - aumenta a relação custo/benefício das inspeções.
  - fornece modelos para escrever documentos com maior qualidade.
  - reduz a subjetividade nos resultados da inspeção.

29



30

07/08/2025 unesp

## Técnicas de Leitura para Inspeção


- O que é uma técnica de leitura?
  - é um conjunto de instruções fornecido ao revisor dizendo **como ler** e **o que procurar** no produto de software.
- Algumas técnicas de leitura para detecção de defeitos em Documentos de Requisitos:
  - Ad-hoc
  - Checklist
  - Leitura Baseada em Perspectiva

Prof. Dr. Rogério Eduardo Garcia

31

31

07/08/2025

 **Ad-hoc**


Prof. Dr. Rogério Eduardo Garcia

- 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.

32

32

07/08/2025

 **Checklist**

Prof. Dr. Rogério Eduardo Garcia

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.

33

33



07/08/2025

 **Perspective-Based Reading – PBR**

Prof. Dr. Rogério Eduardo Garcia

35


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.

<http://www.cs.umd.edu/~mvz/mswe609/shull.pdf>

35

07/08/2025

 **Perspective-Based Reading – PBR**

Prof. Dr. Rogério Eduardo Garcia

36

- 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.

36

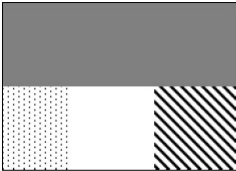
07/08/2025

Prof. Dr. Rogério Eduardo Garcia

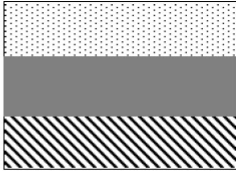
37

## Perspective-Based Reading – PBR

- Cada revisor vai ler o Documento de Requisitos com olhos diferentes
- Benefícios:
  - determina uma responsabilidade específica para cada revisor.
  - melhora a cobertura de defeitos.



*cobertura ad-hoc*



*cobertura baseada em perspectiva*

37

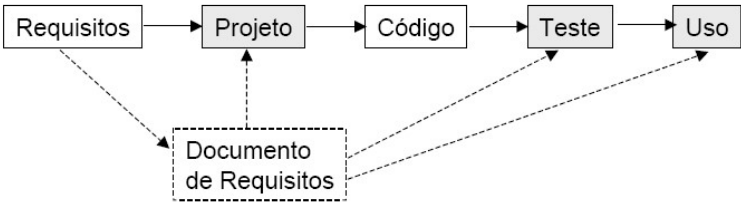
07/08/2025

Prof. Dr. Rogério Eduardo Garcia

38

## Perspective-Based Reading – PBR

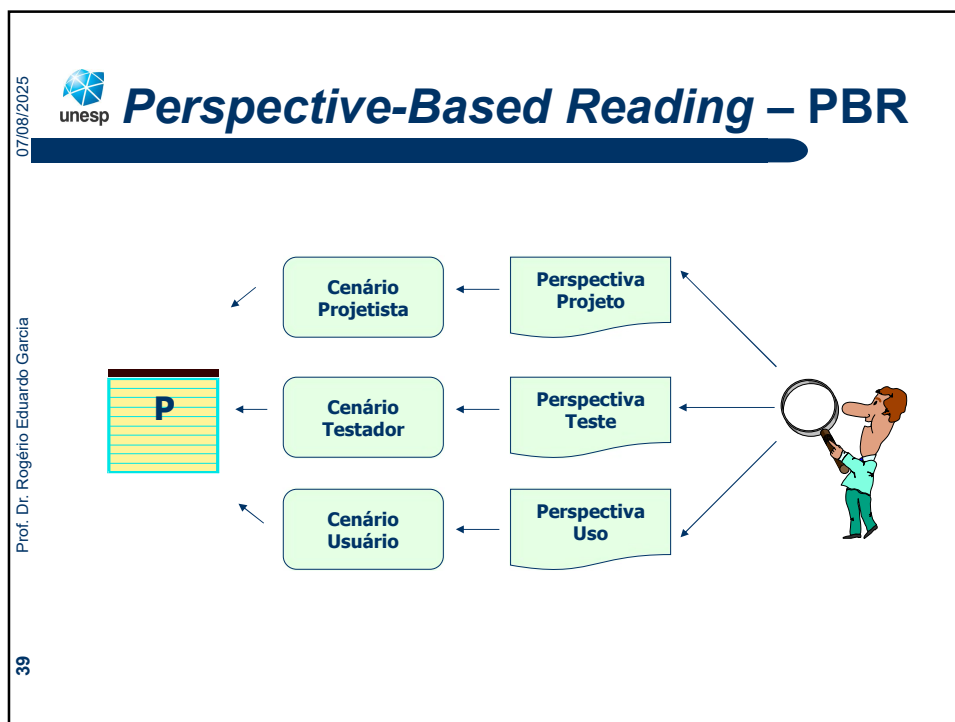
- Várias leituras podem ser feitas no Documento de Requisitos.
  - O projetista que usa o DR para gerar o projeto do sistema.
  - O testador que, com base no DR deve gerar casos de teste para testar o sistema quando este estiver implementado.
  - O usuário para verificar se o DR está capturando toda funcionalidade que ele deseja para o sistema.



```

graph LR
    R[Requisitos] --> P[Projeto]
    P --> C[Código]
    C --> T[Teste]
    T --> U[Uso]
    DR[Documento de Requisitos] -.-> R
    DR -.-> P
    DR -.-> T
    DR -.-> U
            
```

38



39

07/08/2025 unesp

## PBR – Visão do Projetista

Prof. Dr. Rogério Eduardo Garcia


- Gere o projeto de um sistema. Use sua abordagem “preferida”, aplicando a técnica que lhe é mais familiar, e inclua todos objetos de dados, estruturas de dados e funções necessários.

Sugestão: depende do paradigma utilizado. Pode-se usar Diagrama de Classe, por exemplo

40

40

07/08/2025

 **PBR – Visão do Projetista**


Prof. Dr. Rogério Eduardo Garcia

- Questões:
- Todos os elementos (dados, estruturas de dados e funções) necessários foram definidos?
- Todas as interfaces são especificadas e consistentes?
- Foi possível definir todos os tipos de dados? (ex: unidades e precisão)
- Todas informações necessárias para o projeto estão disponíveis? Está tudo especificado? (há alguma especificação funcional/requisito faltando?)
- Há algum ponto em que você não está certo sobre o que deve ser feito devido à falta de clareza na especificação do requisito/especificação funcional?
- A especificação funcional/requisito faz sentido considerando o que você sabe sobre o assunto ou considerando o que foi especificado nas descrições gerais/introdução?

41

41

07/08/2025

 **PBR – Visão do Usuário**

Prof. Dr. Rogério Eduardo Garcia


- 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 perguntas a seguir.

Sugestão: usar como modelo Caso de Uso

42

42

07/08/2025

 **PBR – Visão do Usuário**


Prof. Dr. Rogério Eduardo Garcia

43

- Questões:
  - todas as funções necessárias para escrever os cenários estão especificadas no documento de requisitos ou na especificação funcional?
  - as condições iniciais para definir (iniciar) 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?

43

07/08/2025

 **PBR – Visão do Testador**

Prof. Dr. Rogério Eduardo Garcia


44

- 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 e Análise do Valor Limite*

44

07/08/2025

 **PBR – Visão do Testador**


Prof. Dr. Rogério Eduardo Garcia

- 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 programador 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?

45

45

07/08/2025

 **Revisar em busca da Qualidade**


Prof. Dr. Rogério Eduardo Garcia

- PBR para buscar qualidade
- Como usar para gerenciar?

46

46

07/08/2025

 **Por que medir?**


Prof. Dr. Rogério Eduardo Garcia

- Se não é possível de medir (expressar em números), apenas uma análise qualitativa pode ser feita
  - Sujeito à subjetividade
- Medindo, tendências (boas ou não) podem ser detectadas
  - Melhorias reais podem ser alcançadas
- Números permitem análises, comparações e combinações
  - Impossível com outras informações (dados)

47

47

07/08/2025

 **Conceitos Básicos**


Prof. Dr. Rogério Eduardo Garcia

- Medida
- Medição
- Métrica
- Indicador

48

48

07/08/2025

 **Métrica e Medição**


Prof. Dr. Rogério Eduardo Garcia

- **Métrica:**
  - procura correlacionar medidas individuais expressar a entidade medida.
- **Medição:**
  - é o ato de medir, ou seja, obter uma medida

49

49

07/08/2025

 **Medida**

Prof. Dr. Rogério Eduardo Garcia


- **Medida:**
  - número ou categoria. Fornece uma indicação quantitativa da extensão, quantidade, dimensão, capacidade, tamanho de algum atributo de uma entidade
    - Quanto os dados de um quesito são coletados, tem-se uma medida. Ex: quantidade de erros descobertos em uma revisão.

50

50



07/08/2025  
Prof. Dr. Rogério Eduardo Garcia  
51




## Indicador

- **Indicador:**
  - informação relacionada uma medida, métrica ou combinação de métricas que pode ser utilizada para se ter uma compreensão da entidade medida.

51

07/08/2025  
Prof. Dr. Rogério Eduardo Garcia  
52




## Exemplo

- **Objetivo:** verificar se uma pessoa está com peso ideal ou não.
  - São necessárias duas medidas de duas dimensões: altura (H) e peso (P)
  - Ao medir essas dimensões, faz-se a medição
  - Métricas utilizadas: Kg, metros, IMC (correlaciona as duas)  
 $IMC = P / H^2$
- **Indicadores:**
  - Segundo OMS

Condição	IMC em adultos
abaixo do peso	abaixo de 18,5
no peso normal	entre 18,5 e 25
acima do peso	entre 25 e 30
obeso	acima de 30

52

07/08/2025

 **Na vida**

Prof. Dr. Rogério Eduardo Garcia


- Indicadores:
  - *National Health and Nutrition Examination Survey*

Condição	IMC em Mulheres	IMC em Homens
abaixo do peso	< 19,1	< 20,7
no peso normal	19,1 - 25,8	20,7 - 26,4
marginalmente acima do peso	25,8 - 27,3	26,4 - 27,8
acima do peso ideal	27,3 - 32,3	27,8 - 31,1
obeso	> 32,3	> 31,1

53

53

07/08/2025

 **Qualidade em Software**

Prof. Dr. Rogério Eduardo Garcia

- Produto
- Processo

54

54

07/08/2025

Prof. Dr. Rogério Eduardo Garcia

55

## Busca pela Qualidade

- **Exemplo em Software**
  - Entidade a ser avaliada:
    - Ferramenta de modelagem UML
  - Características:
    - Funcionalidade
    - Usabilidade
      - Facilidade para o aprendizado
      - Facilidade para a operação
      - Facilidade de compreensão
    - Eficiência
      - Tempo
      - Espaço (recursos)
  - Como medir?

55

07/08/2025

Prof. Dr. Rogério Eduardo Garcia

56

## Exemplo em Software


- **Funcionalidade**
  - **Adequabilidade** (quão adequada é a ferramenta para o meu uso)
    - Cobertura da Funcionalidade Implementada
      - Quão correta está a implementação funcional?
      - $CFI = 1 - (NFNI / NFE)$ 
        - NFNI: número de funções não implementadas ou implementadas incorretamente (detectado na avaliação)
        - NFE: número de funções descritas na especificação de requisitos

Quanto mais perto de um, melhor

$CFI < 0.9$ , descartar a possibilidade de adoção

56

07/08/2025

 **Qualidade em Software**


Prof. Dr. Rogério Eduardo Garcia

- Base importante para estimativas: dados históricos
- Boas estimativas são obtidas com base de dados históricos
  - Desde que coletados criteriosamente
- Dados de métricas são muito importantes para se obter estimativas significativas

57

57

07/08/2025

 **Medir e Estimar**


Prof. Dr. Rogério Eduardo Garcia

- Disponibilidade de dados é essencial para se obter “boas” estimativas
  - “Boas”?
  - Coleta criteriosa de registros
- Dados de projetos anteriores para estimar
  - Dados Históricos

58

58

07/08/2025

 **Medir e Acompanhar**


Prof. Dr. Rogério Eduardo Garcia

- Para acompanhar o andamento, é preciso medir o progresso e comparar com o estimado.
- Medidas coletadas dão visibilidade ao estado do projeto, fornecendo base para tomada de ações corretivas, quando necessário.
- Métricas têm um papel importante para a gerência, tanto para o andamento – tempo – quanto para a correção – rumo do projeto

59

59

07/08/2025

 **Medir e Melhorar a Qualidade**


Prof. Dr. Rogério Eduardo Garcia

- Para a qualidade, é fundamental medir!
  - Medir atributos específicos
  - Importante ter métricas significativas, e reflitam os estados dos atributos específicos
  - As medidas obtidas devem fornecer indicadores que norteiem o processo de melhoria

60

60

07/08/2025



## Medir e Melhorar a Qualidade


Prof. Dr. Rogério Eduardo Garcia

61

- Definir características de qualidade relevantes para a “entidade”
- Para cada característica selecionada, definir subcaracterísticas relevantes
  - Estabeleça como computar a característica a partir de suas subcaracterísticas
  - Para subcaracterísticas diretamente mensuráveis, selecionar métricas, coletar medidas, compor indicadores
  - Para subcaracterísticas não diretamente mensuráveis, fazer o caminho inverso (obter valores a partir de outros mensuráveis)
  - Comparar valores obtidos com padrões estabelecidos
  - Ações devem ser tomadas visando à melhoria

61

07/08/2025



## Medir e Melhorar a Qualidade


Prof. Dr. Rogério Eduardo Garcia

62

- Coleta de dados
  - Procedimento influencia a qualidade de dados
  - Obter dados confiáveis é fundamental, e é um risco à medição
  - Influências devem ser isoladas
    - De usuários
    - De hardware
  - Escolha de métricas

62

07/08/2025



## Busca pela Qualidade


Prof. Dr. Rogério Eduardo Garcia

63

- Métricas de Produto
- Métricas de Processo
- Métricas de Projeto
- Métricas de Qualidade
  - Relacionadas a defeitos
- Normalização de métricas
  - Medida de tamanho / função

63

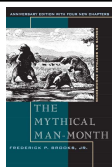
07/08/2025



## Métricas de Revisão

Prof. Dr. Rogério Eduardo Garcia

64




- Esforço de preparação (Ep): esforço (em homens/hora) exigido para revisar um produto resultante antes da reunião de revisão.
- Esforço de avaliação (Ea): esforço (em homens/hora) que é despendido durante a revisão em si.
- Esforço de Reformulação (Re): esforço (em homens/hora) dedicado à correção dos erros revelados durante a revisão.
- Tamanho do artefato de software (TPS): uma medida do tamanho do artefato de software que foi revisto (por exemplo, o número de modelos UML ou o número de páginas de documento ou então o número de linhas de código).

64

07/08/2025

Prof. Dr. Rogério Eduardo Garcia

65



## Métricas de Revisão (cont.)


- Erros secundários encontrados (Errsec): número de erros encontrados que podem ser classificados como secundários (exigindo menos para ser corrigidos do que algum esforço pré-especificado).
- Erros graves encontrados (Errgraves): o número de erros encontrados que podem ser classificados como graves (exigindo mais para ser corrigidos do que algum esforço pré-especificado).

65

07/08/2025

Prof. Dr. Rogério Eduardo Garcia

66



## Análise de Métricas

Erros

$$E_{\text{revisão}} = E_p + E_a + R_e$$

$$\text{Err}_{\text{tot}} = \text{Err}_{\text{sec}} + \text{Err}_{\text{graves}}$$

Densidade de Erros

$$\text{Densidade de erros} = \frac{\text{Err}_{\text{tot}}}{\text{TPS}}$$

Eficácia e Custos

$$\text{Esforço poupado por erro} = \frac{\text{Err}_{\text{testes}} - \text{Err}_{\text{revisões}}}{45 - 6} = 30 \text{ homens/hora/erro}$$

66





67

07/08/2025 unesp

## Para Refletir

Prof. Dr. Rogério Eduardo Garcia

- É melhor uma RTF fora de controle ou não ter revisão?
- O que fazer quando uma RTF não traz contribuições?
- Como argumentar com a alta gerência da organização sobre o valor agregado deste tipo de revisão?
- Como conduzir uma sessão na qual os participantes não se prepararam?
- Como gerenciar a sessão se o foco direcionar para aspectos de avaliação pessoa?

68

68

07/08/2025

unesp

Prof. Dr. Rogério Eduardo Garcia

69

## Plano de Projeto de Software

<p><b>I. Introdução</b></p> <ol style="list-style-type: none"> <li>1. Escopo e propósito do documento</li> <li>2. Objetivos do Projeto</li> </ol> <p><b>II. Estimativas de Projeto</b></p> <ol style="list-style-type: none"> <li>1. Dados históricos usados nas estimativas</li> <li>2. Técnicas de estimativa</li> <li>3. Estimativas</li> </ol> <p><b>III. Riscos do Projeto</b></p> <ol style="list-style-type: none"> <li>1. Análise dos riscos</li> <li>2. Administração dos riscos</li> </ol> <p><b>IV. Cronograma</b></p> <ol style="list-style-type: none"> <li>1. Divisão do trabalho (work breakdown)</li> <li>2. Rede de tarefas</li> <li>3. Gráfico de Gantt</li> <li>4. Tabela de recursos</li> </ol>	<p><b>V. Recursos do Projeto</b></p> <ol style="list-style-type: none"> <li>1. Pessoal</li> <li>2. Hardware e Software</li> <li>3. Recursos especiais</li> </ol> <p><b>VI. Organização do Pessoal</b></p> <ol style="list-style-type: none"> <li>1. Estrutura de Equipe</li> <li>2. Relatórios Administrativos</li> </ol> <p><b>VII. Mecanismos de Controle</b></p> <p><b>VIII. Apêndices</b></p>
---	---

69

07/08/2025

unesp

Prof. Dr. Rogério Eduardo Garcia

70

## Exercício

- Revisar o documento de requisitos (disponível na página) seguindo as diretrizes das perspectivas. Grupos de 3 alunos, cada um com uma perspectiva (diferente do projeto)
- Depois da revisão, cada revisor terá uma lista de defeitos do documento de requisitos que revisou. Os revisores devem se reunir e gerar uma lista única de defeitos.
- **Lembrem-se: TODA atividade deve ser registrada (dia, hora inicial, hora final, o que fez e o quanto fez). Esses dados serão usados mais adiante!!!**

70