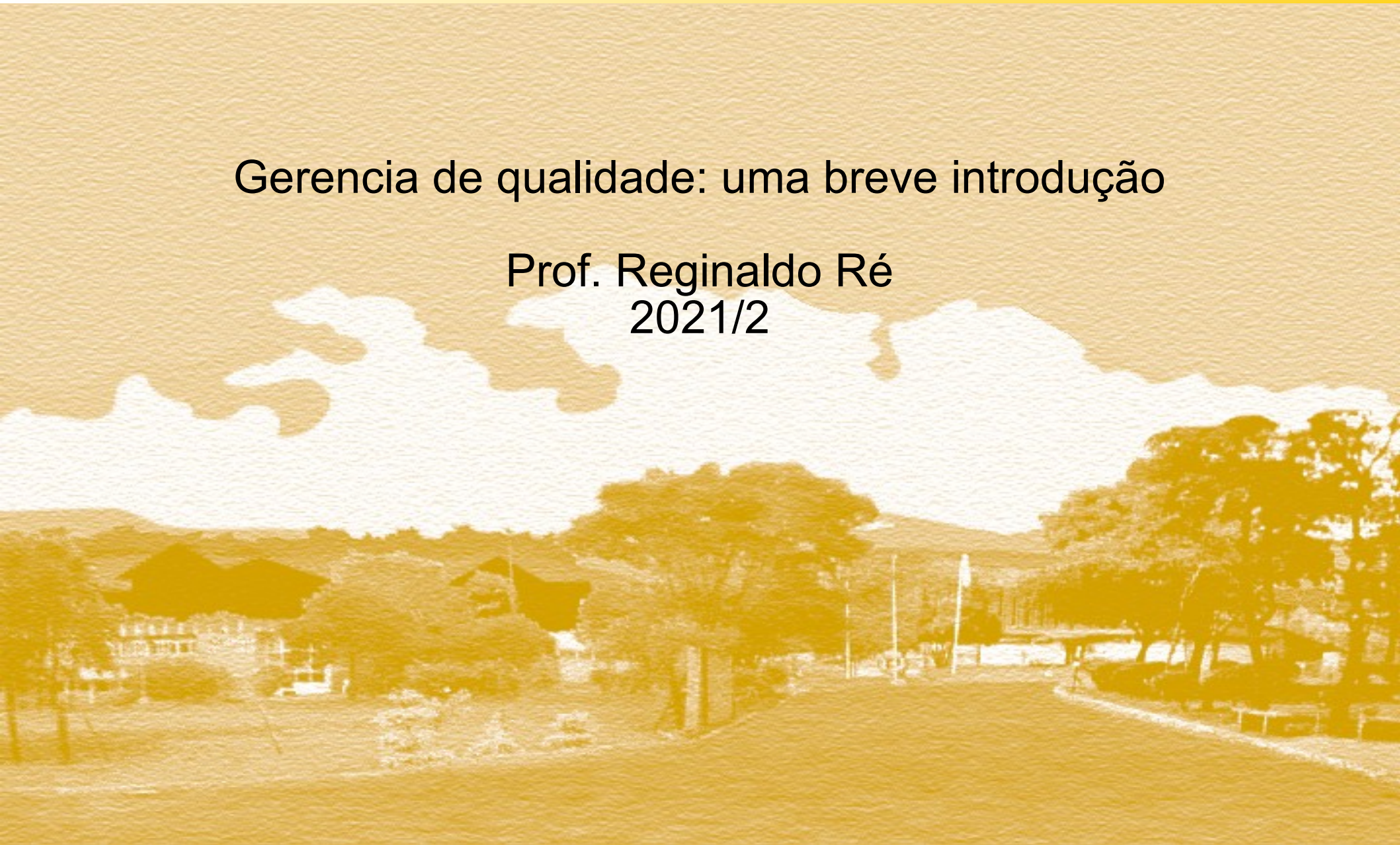


Gerencia de qualidade: uma breve introdução

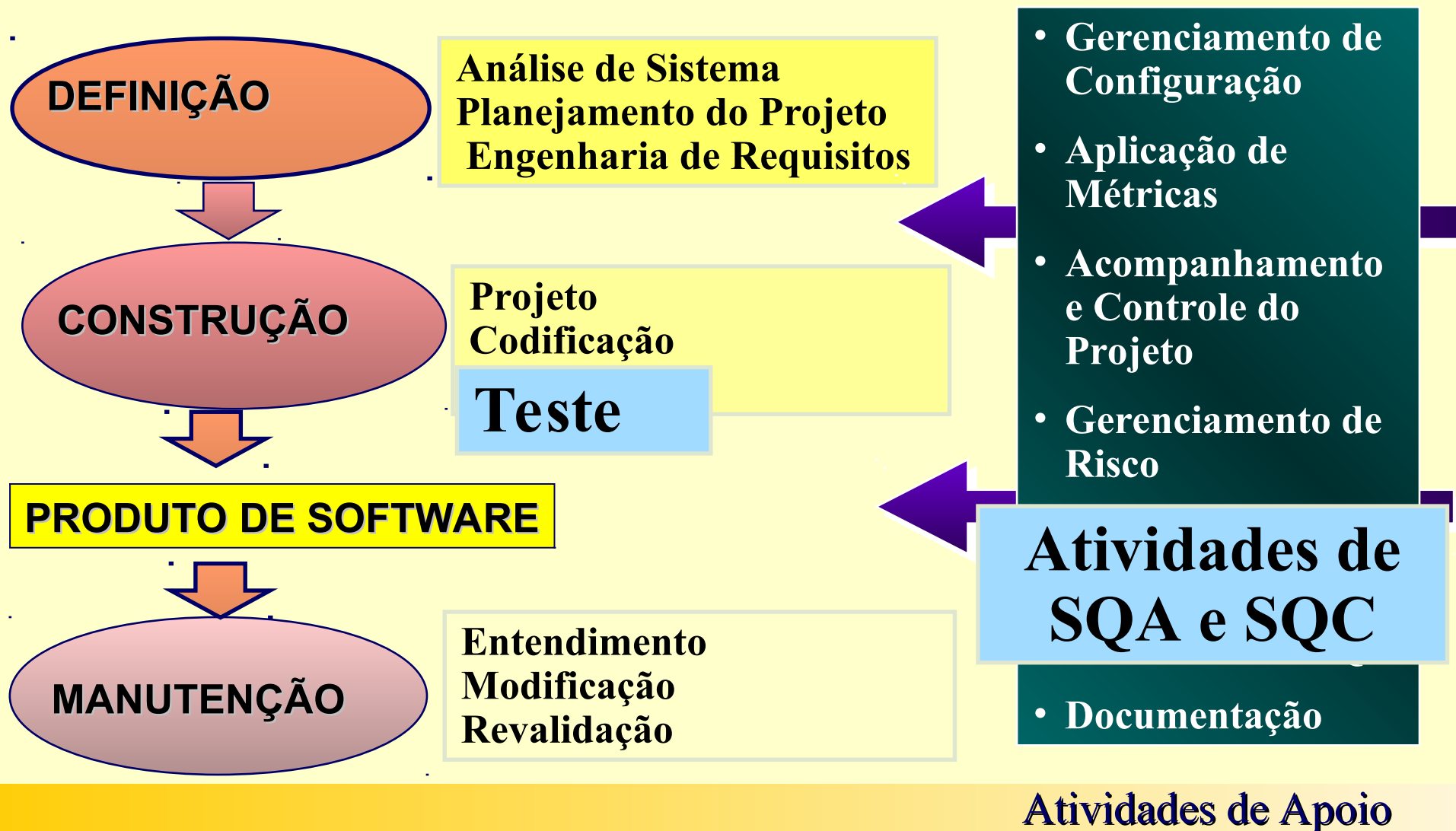
Prof. Reginaldo Ré  
2021/2



- Objetivos
- Gerencia de Qualidade
  - Processo de Software
  - SQA, SQP, SQC, QI
  - Verificação, Validação e Teste
- Objetivos do SQC
- Propriedades do SQC
- SQC Passo a Passo
- Laudo
- Técnicas de SQC
- Considerações Finais

- Discutir, de um ponto de vista macroscópico, as atividades e técnicas relacionadas com o garantia e controle da qualidade:
  - Especialmente teste
  - Uma visão abrangente das técnicas de controle da qualidade facilita a compreensão da interdependência entre elas e com os processos de desenvolvimento

# O Processo de Engenharia de Software relembrando...





- Definir procedimentos e padrões a serem utilizados durante o desenvolvimento de software.
- Verificar se os mesmos estão sendo seguidos por todos os engenheiros.
- Desenvolver uma **cultura de qualidade**.
  - Todos estão **comprometidos** a atingir um determinado **nível** de qualidade para o produto.

- Estruturado em quatro atividades principais:
  - Garantia de Qualidade de Software (SQA)
    - É um “Guia” de qualidade **organizacional** que:
      - Engloba padrões, normas e procedimentos para produzir, verificar, avaliar e aprovar “produtos” produzidos no processo de software
      - Incorpora boas práticas
      - Seleciona ferramentas para apoiar o gerenciamento de qualidade
  - Planejamento da Qualidade de Software (SQP)
    - Efetua planejamento no nível de projeto
      - Seleção de padrões, normas e procedimentos organizacionais específicos para cada projeto que conduzam a um software de qualidade
    - É definido pelo Plano de Qualidade do Projeto
      - Estabelece as qualidades desejadas para o produto.
      - Define como tais qualidades devem ser avaliadas.

- Estruturado em quatro atividades principais:
  - 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)
    - Procura melhorar continuamente processos **organizacionais**

- SQA vs. SQC

Critério	SQA	SQC
Definição	Conjunto de atividades que garante a qualidade dos processos de engenharia de software. As atividades estabelecem e avaliam processos que produzem produtos.	Conjunto de atividades que garante a qualidade dos produtos do processo de software. Busca encontrar defeitos nos produtos produzidos.
Foco	Processo	Produto
Orientação	Estratégias de prevenção (Orientado a prevenção de problemas)	Estratégias de detecção (Orientado a detecção de defeitos)
Alcance	Organizacional	Específico do projeto/produto
Escopo	Relacionado a todos os produtos que são criados por um processo	Relacionado a um produto específico
Atividades	Definição e implementação de processos Auditorias Treinamento	Verificações Validações Testes



- O objetivo do controle da qualidade é identificar discrepâncias com relação aos **interesses dos usuários e clientes**, e com relação aos requisitos **funcionais e não funcionais**
  - Quanto mais discrepâncias existentes forem identificadas melhor
  - Quanto mais cedo o forem, melhor ainda
  - Quanto menos recursos forem necessários, muito melhor
  - Quanto menos defeitos existirem antes mesmo de se realizar os primeiros testes, mais ainda nos aproximamos do ideal
    - ideal: correto por construção -> 0 defeitos

- Identificar (todas?) as discrepâncias com relação ao que é esperado pelos usuários e clientes
  - Serviços efetivamente prestados pelo artefato
  - Confronto com o desejado pelo usuário
    - Verificar se é a implementação correta e completa do problema correto
- Interface com o usuário
  - Facilidade de uso
  - Facilidade de aprendizado
  - Adequação à cultura do usuário
  - Segurança
  - Capacidade de tolerar e recuperar de erros do usuário (clemência)
  - Baixo custo de disponibilização

- Identificar discrepâncias com relação à especificação do artefato
  - Requisitos funcionais
    - Relacionados ao domínio da aplicação
  - Requisitos não funcionais
    - Em geral são os requisitos de qualidade
      - Segurança
      - Capacidade de processamento
      - Escalabilidade
      - Usabilidade
      - ...
  - Requisitos inversos
    - Coisas que não podem acontecer
    - Condições de aceitação
    - Condições contratuais
    - ...

- É um filtro imperfeito
  - Identifica somente uma parte dos problemas existentes
    - Defeitos, deficiências e vulnerabilidades observadas
  - Os demais problemas permanecem e são desconhecidos
    - Defeitos, deficiências e vulnerabilidades remanescentes
  - Na indústria hoje software muito bom tem cerca de 1 defeito para cada 1.000 linhas de código
    - Mas pouco software é bom
    - O estado da prática precisa melhorar muito!
- **Problema da existência:** podemos procurar e encontrar defeitos, mas se não encontrarmos não podemos concluir que não existam

- Cada técnica de controle da qualidade possui seu domínio de eficácia
  - Classe de defeitos mais naturalmente encontrados ao aplicar a técnica
- Como reduzir a chance de defeitos existentes passarem despercebidos?
  - Utilize diversas técnicas de controle da qualidade
  - Procure usar técnicas rigorosas, ex. argumentação da correitude
    - Cuidado com excesso de zelo
    - Muitas técnicas rigorosas são caras e não trazem benefícios compatíveis

# Propriedades do SQC

- Controle da qualidade não assegura qualidade!
- Controle da qualidade não corrige!
  - Controle da qualidade somente verifica o quanto o artefato se aproxima da qualidade desejada
  - Procura encontrar e relatar defeitos ou falhas
  - O resultado do controle é um laudo
- Entretanto, saber como será controlado antes de desenvolver induz o desenvolvedor a se aproximar da qualidade requerida pelo controle



# Propriedades do SQC

- O resultado do controle da qualidade é um laudo
  - relaciona os problemas identificados
  - defeitos, vulnerabilidades, deficiências, falhas
- O laudo pode assumir diversas formas
  - relatório em formato livre relacionado com o artefato
  - anotações no próprio artefato
  - log gerado por ferramentas de controle da qualidade
  - caderno de registro de problemas
  - listas de pendências (*to do lists*, *backlog*)
  - conjunto de fichas de acompanhamento de problema
  - rascunhos, lembranças -> **péssimo**

# Propriedades do SQC

- Controle da qualidade não assegura qualidade!
- Porém, a solução dos problemas identificados no laudo pode levar à *melhoria* da qualidade
  - Cabe à equipe de desenvolvimento resolver os problemas indicados pela gerência
    - De forma completa
      - Diagnose para determinar a causa exata (defeito) da falha
      - Depuração para eliminar completamente a causa
    - Sem gerar novos defeitos !!!

- O controle da qualidade de cada artefato deve ser planejado (definido) junto com a especificação, arquitetura, projeto e codificação
  - Que padrões e normas devem ser obedecidos?
  - Como serão verificadas as especificações?
    - As especificações são verificáveis? São testáveis?
  - Que controles e quando devem ser aplicados?
  - Que ferramentas serão utilizadas?
  - Que instrumentação deve ser incluída?
  - Como será testado?
    - Plano de teste
  - Como será aceito?
    - Quais são os critérios de aceitação?
    - Quando sei que testei o suficiente?

- Idealmente todos os itens das especificações e dos padrões precisam ser verificáveis!
  - Inclusive os que tratam de qualidade
  - Sem dispor de uma especificação verificável como posso dizer racionalmente o que seria aceitável?
- Idealmente todos os itens das especificações funcionais e não funcionais deveriam ser testáveis
  - De preferência de forma automática
  - Teste automático é uma forma verificável de especificação através de exemplos

# Propriedades do SQC

- Após cada alteração é necessário repetir o controle da qualidade
  - Isso compromete a produtividade
  - Gera perdas devido ao retrabalho

- Sugestão: automatizar o controle da qualidade
  - como fazê-lo com relação a código?
    - ex. consistência entre tutoriais e *help* e a implementação como fazê-lo com relação a
      - Arquitetura
      - Projeto
      - Modelos
      - ...



- O controle da qualidade deve envolver todos os tipos de artefatos, ex.
  - Programas
  - Documentos para o usuário
    - Inclusive os de marketing
  - Sistemas de instalação
  - Sistema de auxílio
  - Tutoriais
  - Bases de dados inicializadas
  - Arquivos de dados persistentes
  - ...
- Deve assegurar a coerência entre todos os artefatos

- **Verificação:** controla de forma *isolada* a qualidade de cada artefato
  - Examina se o artefato está em conformidade exata com a sua especificação
  - Examina a conformidade com os padrões exigidos
  - Examina se o conteúdo do artefato forma um todo coerente
- Examina se a intenção do redator é correta e completamente apreendida ao ler as representações que constituem o artefato
  - Não contém sub-entendidos
  - Não está fora de foco
  - Nada falta
  - Não contém excesso
  - Não contém ambigüidades
- Examina o correto uso das linguagens de representação
- Examina se está em um nível uniforme de abstração
  - ex. especificação não deve conter código

- **Validação:** controla a qualidade dos *inter-relacionamentos* entre artefatos
  - Examina se não existem conflitos com outros artefatos
    - Em especial com relação às especificações
  - Examina se o conjunto de artefatos forma um todo coerente
  - Examina se todas as interfaces entre artefatos são respeitadas
- Se passou pela verificação e pela validação, o artefato estará correto com relação à sua especificação e a outros artefatos
  - Implementação correta do problema especificado
  - Infelizmente se a especificação não estiver correta: pode levar à Implementação correta do problema errado
    - Para o usuário estará errado, tanto faz a causa

- **Aprovação**: controla a qualidade do artefato com relação às **atuais** necessidades e expectativas dos usuários
  - para **especificações**
    - examina se a solução proposta *poderá vir a* atender às **atuais** necessidades e expectativas do usuário
  - para **implementações**
    - examina se o artefato *efetivamente atende* às **atuais** necessidades, expectativas do usuário ou potenciais usuários
- Se passou pela verificação, pela validação e pela aprovação, o artefato será, em princípio, uma implementação correta do problema correto

- Artefato
  - nome,
  - versão,
  - data,
  - quem,
  - como:
    - revisão, inspeção,
    - caso de teste,
    - uso,
    - outros, ex. desenvolvimento de outro artefato
- Tipo do problema reportado
  - código,
  - consulta,
  - documentação,
  - especificação (*design*),
  - sugestão

- Severidade
  - Possíveis danos provocados pelo problema
  - É possível continuar a usar?
    - não, provoca danos sérios
    - não, é impossível utilizar os resultados
    - sim, se evitar a região problemática
    - sim, usando outra sequência de trabalho
    - sim, se desprezar alguns resultados
    - sim, pois somente incomoda
- É reprodutível
  - Identificação do problema
  - descrição do problema e como reproduzi-lo
- Sugestão de solução
  - isso nem sempre é desejável



- Solução
  - estado da solução
    - datas de mudança de estado
    - responsáveis pelo trabalho nos estados de execução
  - descrição da solução
  - artefatos criados, alterados, eliminados
    - versões resultantes
  - possíveis causas das faltas identificadas

- O laudo é muito formal?
  - Não utilizado na prática?
  - <https://github.com/JabRef/jabref/issues/7882>
  - <https://github.com/JabRef/jabref/issues/7860>

- Técnicas de controle sem execução do artefato
  - Prova formal da corretude
    - demonstração matemática da *correspondência exata* entre o artefato e a sua especificação formal
  - Argumentação da corretude
    - verificação baseada em matemática da correspondência entre o artefato e a sua especificação *suficientemente formal*
    - utiliza os princípios de prova formal da corretude, mas sem o mesmo rigor
  - Um programa argumentado correto *pode conter* defeitos
    - infelizmente a prática mostra o mesmo para programas provados corretos, embora com frequência menor

- Técnicas de controle sem execução do artefato
  - Revisões
    - leitura do artefato, com ou sem narrações para terceiros
  - Inspeções
    - semelhante a revisões, mas realizadas segundo um procedimento definido, documentado e controlado
  - Desenvolvimento em pares
    - duas pessoas trabalhando juntas em uma mesma estação de trabalho
    - uma digita e a outra controla o que está sendo digitado, dá sugestões, verifica a aderência a padrões, ...

- Técnicas de controle sem execução do artefato
  - Teste estático, análise estática
    - exame de propriedades de um artefato sem pô-lo em operação:
      - verificar se os padrões de programação estão sendo observados
      - verificar se, para cada throw, existe um catch capaz de interceptar a exceção sinalizada
  - Medição estática
    - obtenção de medidas estruturais relativas ao artefato
      - as medidas indicam a probabilidade da presença de problemas
        - ex. complexidade (número ciclomático – McCabe) é (supostamente) proporcional ao número de defeitos injetados

- Técnicas de controle com execução indireta
  - Simulações
    - modelos que permitem prever ou avaliar propriedades do artefato (especificação)
  - Protótipos
    - versões experimentais e descartáveis de aspectos do artefato
      - não são liberações (*releases*) em um desenvolvimento incremental!



- Técnicas de controle com execução direta
  - Testes
    - condução de experimentos controlados envolvendo a execução do artefato
  - Medição dinâmica
    - obtenção de medidas relativas ao comportamento do artefato durante a execução
  - Instrumentação
    - código de controle da integridade ou de medição contido nos artefatos
    - código de controle da cobertura dos testes contido nos artefatos
  - Aprovação a cada iteração
    - teste realizado pelo usuário a fim de verificar se o construto corresponde às suas expectativas explícitas e implícitas
    - viabiliza o controle da qualidade de especificações antes de se dispor do sistema completo

# Considerações Finais

- SQC tem foco no produto
  - Importante para aprovação de artefatos ao longo do processo de software
- SQC engloba uma série de atividades de verificação, validação e teste