

# Unidade I – Conceitos Introdutórios

## **Título:** Garantia de Qualidade de Software – Parte II

Profa. Ana Carolina Gondim Inocêncio

# Roteiro Aula

- Metas e Métricas da SQA (*Software Quality Assurance*)
- Estatísticas da Garantia da Qualidade de Software
- Confiabilidade de Software
- O Plano de SQA

# Metas e Métricas da SQA (*Software Quality Assurance*)

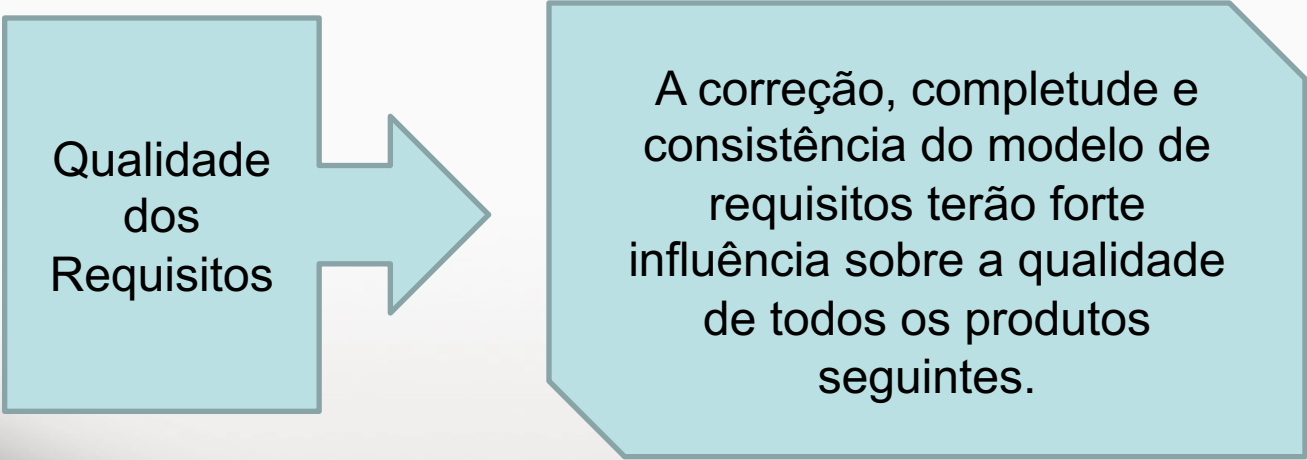
# Elementos de Garantia da Qualidade de Software

- ***Metas, atributos e métricas***
  - As ações descritas, são realizadas para atingir um conjunto de metas pragmáticas:

# Elementos de Garantia da Qualidade de Software

- ***Metas pragmáticas:***

Qualidade  
dos  
Requisitos



```
graph LR; A[Qualidade dos Requisitos] --> B[A correção, completude e consistência do modelo de requisitos terão forte influência sobre a qualidade de todos os produtos seguintes.];
```

A correção, completude e consistência do modelo de requisitos terão forte influência sobre a qualidade de todos os produtos seguintes.



# Elementos de Garantia da Qualidade de Software

- ***Metas pragmáticas:***

Qualidade  
do Projeto



Todo elemento do modelo de projeto deve ser avaliado pela equipe de software para garantir que apresente alta qualidade e que o próprio projeto esteja de acordo com os requisitos.



# Elementos de Garantia da Qualidade de Software

- ***Metas pragmáticas:***

Qualidade  
do Código

O código-fonte e os produtos relacionados (ex. outras informações descritivas) devem estar em conformidade com os padrões locais de codificação e apresentar características que irão facilitar a manutenção.



# Elementos de Garantia da Qualidade de Software

- ***Metas pragmáticas:***

```
graph LR; A[Eficácia do Controle de Qualidade] --> B["A equipe de software deve aplicar os recursos limitados de forma a obter a maior probabilidade possível de atingir um resultado de alta qualidade. A SQA analisa a alocação de recursos para revisões e realiza testes para verificar se eles estão ou não sendo alocados da maneira mais efetiva."];
```

Eficácia do  
Controle de  
Qualidade

A equipe de software deve aplicar os recursos limitados de forma a obter a **maior probabilidade possível de atingir um resultado de alta qualidade**.

A SQA **analisa a alocação de recursos para revisões e realiza testes** para verificar se eles estão ou não sendo alocados da maneira mais efetiva.



# Estatísticas da Garantia da Qualidade de Software

# Estatísticas da Garantia da Qualidade de Software

- A estatística da garantia de qualidade reflete **uma tendência crescente** em toda a indústria de software para tornar mais **quantitativa a análise da qualidade**.
- Para software, a estatística da garantia da qualidade implica as seguintes etapas:

# Estatísticas da Garantia da Qualidade de Software

- Informações sobre erros de software são coletadas e classificadas.
  - É feita uma tentativa de **associar cada erro a sua causa subjacente**
  - Usando o **princípio de Pareto** (80% dos defeitos podem ser associados a 20% de todas as possíveis causas), são isoladas as 20 % (as causas vitais).
  - Assim que as **poucas causas vitais tiverem sido identificadas**, prossegue-se **para a correção dos problemas que provocaram os defeitos**



# Estatísticas da Garantia da Qualidade de Software

- Animação Princípio de Pareto:
- <https://www.youtube.com/watch?v=3VZDKi5W9fQ>



# Estatísticas da Garantia da Qualidade de Software

- A seguir alguns fatos que ilustram o Princípio de Pareto:
  - 80% do total de vendas estão relacionados com 20% dos produtos;
  - 80% dos lucros estão relacionados com 20% dos clientes;
  - 80% dos usuários de computador usam apenas 20% dos recursos disponíveis;
  - 80% do tempo usamos 20% de nossas roupas;
  - 80% dos resultados são obtidos por 20% dos funcionários.



# Estatísticas da Garantia da Qualidade de Software

- Este conceito representa um **importante passo para a criação de uma gestão de qualidade adaptativa** em que mudanças são feitas para melhorar aqueles elementos do processo que introduzem erros.
- A aplicação de estatística de SQA e o princípio de Pareto podem ser sintetizados em uma única sentença:

***“Invista seu tempo concentrando-se em coisas que realmente importam, mas primeiramente, certifique-se de ter entendido aquilo que realmente importa!”***



# Estatísticas da Garantia da Qualidade de Software

- ***Seis Sigma para Engenharia de Software***
  - É a estratégia para a estatística da garantia da qualidade **mais utilizada na indústria atual**.
  - É uma **metodologia rigorosa e disciplinada** que usa análise estatística e de dados para **medir e melhorar o desempenho operacional** de uma empresa
  - Através da **identificação e da eliminação** de defeitos em processos de fabricação e relacionados a serviços.





# Estatísticas da Garantia da Qualidade de Software

- Sigma (a 18ª. letra do alfabeto grego:  $\sigma$ ) é uma medida de **variação utilizada em estatística**. Esta medida, aplicada a um processo empresarial, diz respeito à **frequência com que determinada operação ou transação utiliza mais do que os recursos mínimos necessários para satisfazer o cliente**.



“ O conceito seis sigma é uma nova forma para medir o quanto um produto é bom. Quando um produto tem seis sigma isto nos diz que sua qualidade é excelente, significando que a probabilidade de produzir defeitos é extremamente baixa” (CAMPOS, 1999, p. 74).



# Estatísticas da Garantia da Qualidade de Software

- ***Seis Sigma para Engenharia de Software***
  - A metodologia Seis Sigma define três etapas essenciais:
    - **Definir** as **necessidades do cliente** e os **artefatos passíveis de entrega**
    - **Medir** o **processo existente** e **seu resultado** para determinar o **desempenho da qualidade atual**
    - **Analisar** as **métricas para defeitos** e determinar as **poucas causas vitais**



# Estatísticas da Garantia da Qualidade de Software

- ***Seis Sigma para Engenharia de Software***
  - Se já existir uma gestão de qualidade, e for necessário um aperfeiçoamento, a estratégia Seis Sigma sugere duas etapas adicionais:
    - **Melhorar o processo** por meio da eliminação das causas fundamentais dos defeitos
    - **Controlar o processo** para garantir que trabalhos futuros não reintroduzam as causas dos defeitos.



# Estatísticas da Garantia da Qualidade de Software

- ***Seis Sigma para Engenharia de Software***
  - Essas etapas essenciais e adicionais são, algumas vezes, conhecidas como método DMAAC ( **D**efinir, **M**edir, **A**nalisar, **A**perfeiçoar e **C**ontrolar)



# Confiabilidade de Software

# Confiabilidade de Software

- A confiabilidade de software, pode ser medida diretamente e estimada usando-se **dados históricos** e **de desenvolvimento**.
- A confiabilidade de software é definida em termos estatísticos como:

***“a probabilidade de operação sem falhas de um programa de computador em um dado ambiente por um determinado tempo.”***

# Confiabilidade de Software

- Medidas de Confiabilidade
- Se considerarmos um sistema computacional, uma medida de confiabilidade simples é o tempo médio entre falhas (MTBF, *mean-time-between-failure*)
  - **MTBF = MTTF + MTTR**
    - Em que os acrônimos MTTF e MTTR são, respectivamente, **tempo médio para falhar** (mean-time-to-failure) e **tempo médio para reparar** (mean-time-to-repair)

# Confiabilidade de Software

- Medidas de Confiabilidade
- De maneira simples, um **usuário final se preocupa com falhas perceptíveis** e não com o número total de defeitos que ocasionaram as falhas.
  - Como **cada defeito** contido em um programa **não tem a mesma taxa de falhas perceptíveis**, o **número total de defeitos** fornece **pouca indicação da confiabilidade** de um sistema.

# Confiabilidade de Software

- Medidas de Confiabilidade
- Ex.:
  - Programa em operação 3.000 horas – sem falhas;
  - Vários defeitos podem não ser detectados por dezenas de horas antes de serem descobertos
  - O MTBF com esses erros obscuros poderia ser de 30.000 ou até mesmo 60.000 horas de processador.
  - Outros defeitos, embora ainda não descobertos, poderiam ter uma taxa de falhas de 4.000 ou 5.000 horas.
  - Mesmo se cada um dos erros da primeira categoria fosse eliminado, o impacto sobre a confiabilidade de software seria desprezível.



# Confiabilidade de Software

- Medidas de Confiabilidade
- O MTBF pode ser problemático por duas razões:
  - 1) Projeta um período de tempo entre falhas, mas **não fornece uma projeção da taxa de falhas**
  - 2) o MTBF **pode ser mal interpretado como sendo o tempo de vida médio**, muito embora não seja esse o significado.

# Confiabilidade de Software

- Medidas de Confiabilidade
  - Uma medida alternativa de confiabilidade é **falha ao longo do tempo** ( FIT, failures-in-time) – uma medida estatística de **quantas falhas um componente terá ao longo de um bilhão de horas de operação**.
  - Consequentemente 1 FIT equivale a uma falha a cada bilhão de horas de operação.

# Confiabilidade de Software

- Medidas de Confiabilidade e Disponibilidade
  - Além de uma medida de confiabilidade, deve-se também desenvolver uma **medida de disponibilidade**.
  - Disponibilidade de software é a **probabilidade de que um programa esteja operando** de acordo com os **requisitos** em um **dado instante** e é definida da seguinte forma:

$$\text{Disponibilidade} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} \times 100\%$$

**MTTF** - tempo médio para falhar  
**MTTR** - tempo médio para reparar

# Confiabilidade de Software

- Proteção de Software
  - É uma garantia da qualidade de software que se concentra na **identificação** e na **avaliação de potenciais problemas** que podem **afetar negativamente um software** e provocar falha em todo o sistema.
  - Um processo de **modelagem e análise** é efetuado como **parte de proteção do software**.
  - Inicialmente, os problemas são identificados e classificados por criticalidade e risco.

# Confiabilidade de Software

- Proteção de Software

- Ex.:

- Problemas associados a um controle computadorizado de um automóvel podem:
      - 1) Provocar uma aceleração descontrolada que não pode ser interrompida
      - 2) Não responder ao acionamento do pedal do breque
      - 3 )Não operar quando a chave é ativada
      - 4) Perder ou ganhar velocidade lentamente.
    - Uma vez identificados esses perigos no nível de sistema, técnicas de análise são utilizadas para atribuir gravidade e probabilidade de ocorrência.

# Confiabilidade de Software

- Proteção de Software
  - Análises técnicas devem ser feitas para prever a cadeia de eventos que podem causar problemas e a probabilidade que cada um dos eventos irá ocorrer para criar a cadeia.
  - Ao identificar e analisar os problemas, deve ser elaborada a lista de eventos indesejáveis e as respostas desejadas pelo sistema para esses eventos.
  - O papel do software em administrar eventos indesejáveis é então indicado.

# Confiabilidade de Software

- Confiabilidade X Proteção

Confiabilidade: análise estatística para determinar a probabilidade de ocorrência de uma falha de software

X

Proteção: examina as maneiras em que as falhas resultam em condições que podem levar a problemas

# O Plano de SQA



# O Plano de SQA

- O plano de SQA fornece um **roteiro para instituir a garantia da qualidade** de software.
- Desenvolvido pelo grupo de SQA, o plano serve como um **gabarito para atividades de SQA** que são instituídas para cada projeto de software.

# O Plano de SQA

- Foi publicada pela IEEE um padrão para planos de SQA.
- O padrão recomenda uma estrutura que identifique:
  - 1) propósito e escopo do plano;
  - 2) descrição de todos os artefatos resultantes de ES (modelos, documentos, código-fonte) que caem dentro do âmbito da SQA
  - 3) todos os padrões e práticas que são aplicados durante a gestão de qualidade
  - 4 ) as ações e tarefas da SQA (incluindo revisões e auditorias e sua aplicação na gestão de qualidade
  - 5 ) as ferramentas e os métodos que dão suporte às ações e tarefas da SQA
  - 6 ) procedimentos para administração de configurações de software
  - 7) métodos para montagem, salvaguarda e manutenção de todos os registros relativos à SQA
  - 8) papéis e responsabilidades dentro da organização relacionados com a qualidade do produtos.

# Kahoot assíncrono

- [https://kahoot.it/challenge/04026898?challenge-id=dd701e52-7000-49c6-a9d2-33d726c8c00f\\_1638212114800](https://kahoot.it/challenge/04026898?challenge-id=dd701e52-7000-49c6-a9d2-33d726c8c00f_1638212114800)
- Termina em: 12/12/2021 às 23h

# Questões

# Questões

1. Por que normalmente existe tensão entre um grupo de engenharia de software e um grupo de garantia da qualidade de software independente?
2. Dada a responsabilidade de melhorar a qualidade de software na organização. Qual é a primeira coisa a ser feita? E a seguinte?