

SSC721 – Teste e Inspeção de Software

Qualidade de Software e Atividades de Verificação e Validação

Profa. Ellen Francine Barbosa
francine@icmc.usp.br

Instituto de Ciências Matemáticas e de Computação — ICMC/USP

SSC721 – Teste e
Inspeção de Software

Casos de Falhas em
Sistemas

Engenharia de Software

Qualidade de Software

Garantia de Qualidade
(SQA)

Atividades de V&V

SQA no Processo de
Desenvolvimento de
Software

Exercício

- Casos de Falhas em Sistemas
- Engenharia de Software
- Qualidade de Software
- Garantia de Qualidade (SQA)
- Atividades de V&V
- SQA no Processo de Desenvolvimento de Software
- Exercício

- **Software**

- Fator chave que diferencia produtos modernos.
 - Sistemas de transportes, médicos, telecomunicações, militares, processos industriais, entretenimento, ...
- Presente no dia-a-dia das pessoas.
 - Importante que esteja correto!!!



- Por que devemos nos preocupar com a **qualidade** do software?



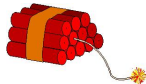
- Em 1995 registraram-se falhas nos sensores das linhas férreas situadas em túneis nos canais ingleses.
- Os trens paravam devido à suspeita de outro trem na linha.
- A causa disso era a névoa de água salgada que “confundia” os sensores.
- Motivo da falha:
 - Provavelmente os requisitos do sistema não levavam em conta a névoa de água salgada como um possível evento.
 - Testes e validação apropriados poderiam detectar o problema.



- Em 1995 um trem bateu atrás de outro, matando o maquinista e ferindo 54 pessoas.
- A distância entre os sinais (projetado em 1918) era menor que a distância de parada necessária para os trens atuais (maiores, mais pesados e mais rápidos). Os trens foram atualizados sem modificação no sistema de controle.
- Motivo da falha:
 - Atualização de uma parte do sistema sem testar e validar o sistema como um todo.



- A explosão a bordo do Challenger é uma das falhas mais notáveis da tecnologia moderna.
- Em 1986, após 73 segundos do seu lançamento, uma explosão envolveu o foguete matando 7 astronautas.
- As investigações concluíram que o problema estava em algumas juntas do motor que não estavam projetadas para a temperatura e pressão ocorrida.
- Motivo da falha:
 - A especificação da pressão não estava de acordo com os requisitos do sistema.
 - Os testes e a validação realizados foram inadequados para detectar a falha.



- Em 1996, o veículo espacial Ariane 5 saiu do curso e explodiu segundos após o seu lançamento.
- Levou uma década de desenvolvimento e custou 7 bilhões de dólares.
- Os testes e a validação insuficientes em componentes reutilizados do veículo Ariane 4 foram a causa do acidente.
- Motivo da falha:
 - Erro de software no cálculo da velocidade horizontal do foguete.
 - A variável que armazenava este valor tinha 64 bits (floating point) e foi erroneamente modificada para 16 bits (signed integer).
 - O valor era maior que 32.767 (maior inteiro), gerando uma falha de conversão!!!



- Por que é difícil construir software com qualidade?
- Existe algum mecanismo para garantir a qualidade do software? Como utilizá-lo eficientemente?



Conjunto de princípios, métodos e técnicas que tratam o software como produto de engenharia que requer planejamento, projeto, implementação e manutenção.

- Objetivo Principal
 - Produzir software de **alta qualidade**, com um baixo custo.

Qualidade é a totalidade de características e critérios de um produto ou serviço que exercem suas habilidades para satisfazer às necessidades declaradas ou envolvidas.

- Conformidade com **requisitos funcionais e de desempenho, padrões de desenvolvimento documentados, e características implícitas** esperadas de todo software profissionalmente desenvolvido.

- Os **requisitos de software** são a base a partir da qual a qualidade é medida.
 - A falta de conformidade aos requisitos significa falta de qualidade.
- **Padrões especificados** definem um conjunto de **critérios** de desenvolvimento que orientam a maneira segundo a qual o software passa pelo trabalho de engenharia.
 - Se os critérios não forem seguidos, o resultado quase que seguramente será a falta de qualidade.

- Existe um conjunto de **requisitos implícitos** que frequentemente não são mencionados na especificação.
 - Por exemplo, o desejo de uma boa manutenibilidade e a facilidade de uso.
- Se o software atende aos requisitos explícitos, mas falha nos requisitos implícitos, a qualidade é suspeita.
- Existe, ainda, uma visão de qualidade de software do ponto de vista **gerencial**.
 - O software é considerado de qualidade desde que possa ser desenvolvido dentro do prazo e do orçamento especificados.

- Visões de Qualidade de Software



Usuário



Desenvolvedor



Organização

**Facilidade de uso, desempenho,
confiabilidade dos resultados,
preços do software, etc.**

**Taxa de defeitos, facilidade de
manutenção e conformidade em relação
aos requisitos dos usuários, etc.**

**Cumprimento de prazo, boa previsão de
custo, boa produtividade.**

- Norma ISO/IEC 9126

- Padronização mundial para a qualidade do produto.
- Características
 - Funcionalidade: Satisfaz as necessidades?
 - Confiabilidade: É imune a falhas?
 - Usabilidade: É fácil de usar?
 - Eficiência: É rápido e “enxuto”?
 - Manutenibilidade: É fácil de modificar?
 - Portabilidade: É fácil de usar em outro ambiente?

Teste de Software

Atividade relevante para avaliação das características
Funcionalidade e Confiabilidade.

- **Processo de Software**

Consiste em uma série de atividades, práticas, eventos, ferramentas e métodos que garantem, técnica e administrativamente, que o software pode ser desenvolvido com qualidade e de maneira organizada, disciplinada e previsível.

- Norma ISO/IEC 12207



● Modelo de Processo de Software

- Procura descrever formalmente e de maneira organizada todas as atividades que devem ser seguidas para a obtenção segura de um produto de software.
- Padrões relacionados a Processo de Software
 - CMM, CMMI
 - ISO/IEC 15504 (SPICE)
 - Bootstrap
 - ...

- Modelo CMM (Capability Maturity Model)
 - Modelo de referência para a qualidade de processo de produção de software.
 - As organizações são classificadas em níveis de maturidade.
 - Processo formal de avaliação.
 - Cada nível de maturidade possui um grupo de atividades correlatas.
 - Áreas-chave do processo (KPA's).

INICIAL (1) → REPETÍVEL (2)



INICIAL (1) → REPETÍVEL (2)

OTIMIZADO

6 - Gerenciamento da Configuração de Software
5 - Garantia da Qualidade de Software

Esta kpa tem o duplo objetivo de assegurar a qualidade tanto do produto sendo construído (o software) quanto do processo de sua construção.

Em termos de produto, isso significa que o CMM exige que a organização defina métodos, técnicas e padrões para verificação da qualidade, estabelecendo, por exemplo, quais os **tipos de teste** que deverão ser realizados, o uso de **revisões, inspeções**, etc.

Já em relação ao processo, esta kpa procura garantir que todas as outras kpas estejam sendo respeitadas, garantindo, por exemplo, que o gerente de projetos esteja fazendo estimativas corretamente usando o modelo definido para a organização.

INICIAL

C
Caóicas

REPETÍVEL (2) → DEFINIDO (3)



REPETÍVEL (2) → DEFINIDO (3)





Qualidade de Processo X

SSC721 – Teste e
Inspeção de Software

Casos de Falhas em
Sistemas

Engenharia de Software

Qualidade de Software

Qualidade de Produto

Qualidade de Processo

Garantia de Qualidade
(SQA)

Atividades de V&V

SQA no Processo de
Desenvolvimento de
Software

Exercício

Teste e Inspeção

Atividades essenciais para ascensão ao Nível 3
do Modelo CMM.



Garantia de Qualidade (SQA – Software Quality Assurance) I

SSC721 – Teste e
Inspeção de Software

Casos de Falhas em
Sistemas

Engenharia de Software

Qualidade de Software

Garantia de Qualidade
(SQA)

Atividades de V&V

SQA no Processo de
Desenvolvimento de
Software

Exercício

- Conjunto de atividades técnicas aplicadas durante todo o processo de desenvolvimento.
 - Garantir que tanto o processo de desenvolvimento quanto o produto de software atinjam os níveis de qualidade especificados.
- Atividades de SQA
 - Aplicação de **métodos técnicos**.
 - Ajudar o analista a conseguir uma especificação de qualidade.
 - Ajudar o projetista a desenvolver um projeto de qualidade.
 - Realização de **revisões**.
 - Avaliar a qualidade da especificação, do projeto, do código, ...



Garantia de Qualidade (SQA – Software Quality Assurance) II

SSC721 – Teste e
Inspeção de Software

Casos de Falhas em
Sistemas

Engenharia de Software

Qualidade de Software

Garantia de Qualidade
(SQA)

Atividades de V&V

SQA no Processo de
Desenvolvimento de
Software

Exercício

- Atividades de **teste de software**.
 - Ajudar a garantir que a detecção de erros seja efetiva.
- Aplicação de **padrões e procedimentos formais**.
 - Garantir que estes sejam seguidos durante o desenvolvimento.
- Processo de **controle de mudanças**.
 - Atividade associada ao gerenciamento de configuração de software.
- Mecanismos de **medição**.
 - Apoio no acompanhamento da qualidade de software.
 - Avaliar o impacto de mudanças metodológicas e procedimentais.
- Anotação e manutenção de **registros**.
 - Procedimentos para coleta e disseminação de informações de garantia de qualidade.

- Dentre as atividades de SQA estão as atividades de **verificação** e **validação** de software.
- O objetivo é minimizar a ocorrência de erros e riscos associados.
 - Detectar a presença de erros nos produtos de software.

- Verificação

- Assegurar consistência, completitude e corretitude do produto **em cada fase** e **entre fases** consecutivas do ciclo de vida.

Estamos construindo corretamente o produto?

- Assegurar que o produto, ou uma determinada função do mesmo, esteja sendo implementado corretamente.
 - Verifica-se inclusive se os métodos e processos de desenvolvimento foram adequadamente aplicados.

- Validação

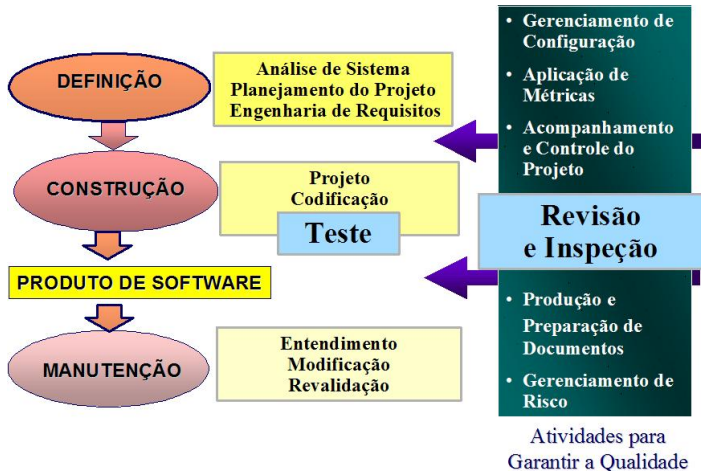
- Assegurar que o produto sendo desenvolvido corresponde ao produto correto, conforme os **requisitos do usuário**.

Estamos construindo o produto certo?

- V&V abrangem um amplo conjunto de atividades de SQA:
 - **Revisões técnicas formais**
 - Auditoria de qualidade e configuração
 - Monitoramento de desempenho
 - Simulação
 - Estudo de viabilidade
 - Revisão da documentação
 - Revisão da base de dados
 - **Testes**
- V&V envolvem atividades de **análise estática** e de **análise dinâmica**.

- Não requerem a execução propriamente dita do produto.
- Podem ser aplicadas em qualquer produto intermediário do processo de desenvolvimento.
 - Documento de requisitos, diagramas de projeto, código-fonte, planos de teste, ...
- As **revisões** são o exemplo mais clássico de análise estática.
 - **Inspeção**
 - Walkthrough
 - Peer Review

- Requerem a execução do produto.
 - Código ou quaisquer outras representações executáveis do sistema.
- Exemplos de atividades que constituem uma análise dinâmica do produto:
 - **Teste de Software**
 - Simulação





SQA no Processo de Desenvolvimento de Software II

SSC721 – Teste e
Inspeção de Software

Casos de Falhas em
Sistemas

Engenharia de Software

Qualidade de Software

Garantia de Qualidade
(SQA)

Atividades de V&V

SQA no Processo de
Desenvolvimento de
Software

Exercício

● Aspectos Positivos

- O software terá menos defeitos latentes.
- Maior confiabilidade resultará em maior satisfação do cliente.
- O custo do ciclo de vida global do software pode ser reduzido.
- Os custos de manutenção podem ser reduzidos.

● Aspectos Negativos

- Difícil de ser instituída em pequenas empresas.
- Representa uma mudança cultural.
 - Mudança nunca é fácil!!!

- Especifique um conjunto de casos de teste adequado para testar o programa a seguir.
- **O programa lê três valores inteiros que representam os lados de um triângulo. A partir dos valores, o programa informa se os lados formam um triângulo isósceles, escaleno ou equilátero.**
 - **Condição:** a soma de 2 lados tem que ser maior que o terceiro lado.

- ① Existe c.t. para triângulo escaleno válido?
- ② Existe c.t. para triângulo isósceles válido?
- ③ Existe c.t. para triângulo equilátero válido?
- ④ Existem pelo menos 3 c.t. para isósceles válido contendo a permutação dos mesmos valores?
- ⑤ Existe c.t. com um valor zero?
- ⑥ Existe c.t. com um valor negativo?
- ⑦ Existe c.t. em que a soma de 2 lados é igual ao terceiro lado?
- ⑧ Para o item 7, existe um c.t. para cada permutação de valores?
- ⑨ Existe c.t. em que a soma de 2 lados é menor que o terceiro lado?

- 10 Para o item 9, existe um c.t. para cada permutação de valores?
 - 11 Existe c.t. para os 3 valores iguais a zero?
 - 12 Existe c.t. com valores não inteiros?
 - 13 Existe c.t. com número de valores errados, por exemplo, 2 valores ao invés de 3?
 - 14 Para cada c.t. você especificou a saída esperada para a entrada projetada?
-
- **Questões baseadas em erros encontrados em implementações deste programa!!!**