

Qualidade de Software

Luiz Leão - luizleao@gmail.com

<http://www.luizleao.com>



Estácio

UNIDADE II : Fatores e Métricas de Qualidade

- Uma visão geral dos fatores que afetam a qualidade
- Métrica dos fatores de qualidade
- Garantia da Qualidade de Software
 - Atividades SQA
 - Revisões de software
 - Revisões técnicas formais
 - Indicadores de qualidade
 - Garantia estatística da qualidade
 - Medidas e modelos de confiabilidade de software

O que pode afetar a qualidade do software

- Segundo **McCall**, a qualidade pode ser afetada pelos seguintes fatores:
 - Correção
 - Confiabilidade
 - Eficiência
 - Integridade
 - Usabilidade
 - Facilidade de Manutenção
 - Flexibilidade
 - Testabilidade
 - Portabilidade
 - Reusabilidade
 - Interoperabilidade

Visão Geral

- Por ser difícil de quantificar a qualidade do software, é necessário que seja analisado vários fatores que podem servir de parâmetro de avaliação

Correção

- O quanto um programa satisfaz a sua especificação e atende aos objetivos da missão do cliente
- Segundo Crosby, quando buscamos o produto com **Erro Zero**, e totalmente aderente aos requisitos, temos uma aplicação com a máxima qualidade

Confiabilidade

- O quanto se pode esperar que um programa realize a função pretendida com a precisão exigida
- Quanto maior a confiabilidade do software, menores são os custos com manutenção, o que eleva a qualidade do produto

Eficiência

- A quantidade de recursos computacionais e código exigidos por um programa para desempenhar a sua função
- O uso otimizado e racional dos recursos de TI, visando a eficiência, contribui para o aumento da qualidade do projeto

Integridade

- O quanto o acesso ao software ou dados por pessoas não autorizadas pode ser controlado
- Deve contar com políticas de segurança, perfis de acesso, criptografia de dados relevantes (Ex. Senha), entre outros fatores.

Usabilidade

- Esforço necessário para aprender, operar, preparar, a entrada de dados e interpretar a saída de um programa
- Um bom projeto de IHC pode contribuir para que o custo com treinamento seja minimizado

Facilidade de Manutenção

- Esforço necessário para localizar e corrigir um erro em um programa
- Não é tão simples de mensurar, pois, envolve diversos tipos de manutenções que o software pode sofrer, onde cada tipo tem suas particularidades

Flexibilidade

- Esforço necessário para modificar um programa em operação
- Sistemas em produção não podem parar para que a modificação seja efetuada, logo, o contingenciamento deve ser bem planejado

Testabilidade

- Esforço necessário para testar um programa de modo a garantir que ele desempenhe a função destinada
- “Um bom testador é tão burro quanto um usuário” autor desconhecido
- Planos de teste e testes unitários são instrumentos que, quando bem elaborados, identificam o escopo de problemas de maneira mais eficaz

Portabilidade

- Esforço necessário para transferir o programa de um ambiente de hardware e/ou software para outro
- Aplicativos desenvolvidos em linguagens multiplataforma terão altos níveis de qualidade nesse quesito

Reusabilidade

- O quanto um programa (ou partes de um programa) pode ser reutilizado em outras aplicações – relacionado com o empacotamento e o escopo das funções que o programa executa.
- Em termos de implementação, o uso de programação orientação a objetos (POO) e modularização da solução facilita o reuso de artefatos de software
- Trabalhar com arquitetura de serviço (SOA), também, pode contribuir para esse quesito, quando são identificadas funcionalidades em comum com outros sistemas da organização

Interoperabilidade

- Esforço necessário para integrar um sistema a outro
- Arquitetura de serviço (SOA) desde a sua concepção, será um diferencial nesse quesito
- É um dos grandes problemas a serem resolvidos no ambientes de TI das grandes empresas, atualmente

Garantia da Qualidade de Software

“Atividade e técnica operacional que é utilizada para satisfazer os requisitos de qualidade”
(McDermid).

São funções gerenciais e estão relacionadas às atividades de **verificação** e **validação**.

Garantia da Qualidade de Software

- Consome tempo no desenvolvimento de sistemas de software e vai além da entrega do sistema (entra na fase de manutenção);
- Técnicas usadas para cada atividade podem contribuir para o respectivo controle de qualidade;
- Algumas técnicas têm controle embutido, outras não.

Garantia da Qualidade de Software

- Gerentes querem os melhores projetistas para projetar o produto, mas em geral não podem tê-los;
- Existe então a necessidade de concentrar esforços em métodos de SQA (**Software Quality Assurance – Garantia de Qualidade de Software**);
- O papel de SQA é monitorar os métodos e padrões que os engenheiros de software usam;
- Pessoas podem ser experientes em SQA sem, no entanto, serem experientes em projetos de software.

Atividades de SQA

- Em SQA temos uma variedade de tarefas, as quais podemos dividir em dois grandes grupos:
 - **Engenheiros de Software**
 - **Grupo de SQA**

Engenheiro de Software

- Trabalham na elaboração dos artefatos de software que compõem o projeto (documentação, software, etc.);
- Além da aderência aos requisitos funcionais e não-funcionais, esses profissionais devem estar atentos ao processo adotado;

Grupo de SQA

- Responsabilidades sobre o plano de qualidade, inspeção, conservação de registros históricos, análise do produto desenvolvido e *reporting* das atividades de SQA ao gerente do projeto.
- Trabalham no refinamento das entregas

Atividades de SQA

- O SEI (Software Engineering Institute) recomenda as seguintes atividades para o grupo de SQA
 - Preparar uma plano de SQA;
 - Participar da descrição do projeto de software;
 - Revisar as atividades dos engenheiros de software;
 - Documentar e consertar os desvios;
 - Registrar discordâncias e reportar para o gerente;
 - Gerenciar mudanças e métricas de software.

Atividades de SQA: Revisões de Software

- São um filtro no processo de desenvolvimento de software;
- Não são limitadas à especificação, projeto e código.
- Visa a detecção de Defeitos
 - Anomalia do produto, segundo IEEE;
 - Um defeito implica em um problema na qualidade que é descoberto depois que o software foi entregue ao usuário final.

Atividades de SQA: Revisões de Software

- **Atividades de Revisão de Software:**
 - **Inspeções no projeto e no código:** têm a finalidade de detectar erros no projeto ou código e checar quando padrões têm sido seguidos.
 - **Revisões Gerenciais:** esse tipo de revisão é feita para fornecer informações aos gerentes sobre todo o processo no desenvolvimento do projeto de software.
 - **Revisões de Qualidade:** o trabalho de um indivíduo ou de um time é revisado por um grupo composto por membros do projeto e gerentes técnicos.

Atividades de SQA: Revisões Técnicas Formais

- O objetivo primário de revisões técnicas formais (RTF) é encontrar erros durante o processo antes que eles se tornem defeitos após o *release* do software.
- Um benefício óbvio de revisões técnicas formais é a descoberta de erros antes que eles se propaguem para as próximas fases do processo de software.

Atividades de SQA: Revisões Técnicas Formais

- Estudos de algumas indústrias indicam que as atividades de desenvolvimento do software introduzem entre **50% a 60%** do total de erros do projeto;
- *RTF* podem descobrir cerca de 75% desses erros.

Atividades de SQA: Indicadores de Qualidade

- São informações coletadas nas várias fases do projeto
- Servem de parâmetro comparativo com outros projetos de natureza similar
- Esses indicadores podem ser comparados a bases de conhecimento públicas, ou bases internas da equipe desenvolvedora

Atividades de SQA: Indicadores de Qualidade

Exemplo:

Se na base histórica há computado que projetos de e-commerce costumam ter uma produtividade média, por analista de 20 PF/Mês, e o projeto atual temos 25 PF/Mês, no quesito **produtividade de desenvolvimento** a equipe está superando as expectativas

Atividades de SQA: Medidas de Produtividade de Programação

- A qualidade do software depende da produtividade de programação, a qual é afetada por:
 - Qualidade da documentação;
 - Linguagem de programação;
 - Disponibilidade de ferramentas;
 - Experiência do programador;
 - Comunicação no projeto;
 - Grau de dependência entre módulos;
 - Práticas de programação bem definidas.

Atividades de SQA: Garantia Estatística da Qualidade

- A **confiabilidade de software** usa a análise estatística para determinar a probabilidade de que uma falha de software venha a ocorrer.
- Entretanto, a ocorrência de uma falha não necessariamente resulta num risco ou deformação.

Atividades de SQA: Garantia Estatística da Qualidade

- A **segurança de software** examina as maneiras segundo as quais as falhas resultam em condições que podem levar a uma deformação.
- Ou seja, as falhas não são consideradas no vazio, mas são avaliadas no contexto de um sistema inteiro computadorizado.

Atividades de SQA: Garantia Estatística da Qualidade

Em resumo...

Confiabilidade x Segurança

- **Confiabilidade**

- Usa a análise estatística para determinar a probabilidade de que uma falha **venha a ocorrer**.

- **Segurança**

- Examina as maneiras segundo as quais as falhas resultam em condições que podem levar a uma deformação.

Atividades de SQA: Medidas de Confiabilidade

- “Probabilidade de uma operação de programa de computador ser livre de **falha**”.

Não conformidade com os requisitos de software

- São métricas que visam quantificar o nível de confiabilidade do produto
- Exemplo: um software que opera corretamente em 96 das suas 100 execuções, tem uma confiabilidade de 0.96.

Atividades de SQA: Medidas de Confiabilidade

- Além da medida de confiabilidade, deve-se desenvolver uma **medida de disponibilidade**, a qual é a probabilidade de que um programa esteja operando de acordo com os requisitos num dado ponto no tempo, e é definida como:

$$\text{Disponibilidade} = \text{MTTF} / (\text{MTTF} + \text{MTTR}) \times 100\%$$

MTTF = tempo médio para a falha

MTTR = tempo médio para o reparo

Atividades de SQA: Medidas de Confiabilidade

- A medida de disponibilidade é mais sensível ao MTTR, uma medida indireta da **manutenibilidade de software**.

Plano de SQA

- Especifica os objetivos, as tarefas de SQA a serem realizadas, os padrões, os procedimentos a estrutura organizacional e os mecanismos de auditoria;
- Documentos de ES exigidos:
 - Especificação de Requisitos,
 - Descrição de Projeto, Plano (e Relatório) de Verificação e Validação,
 - Documentação do Usuário.

Plano de SQA

- O padrão IEEE (Padrão ANSI/IEEE 730-1984 e 983-1986) para a preparação do SQAP contém os seguintes tópicos, nos slides a seguir [Humphrey89]:

Plano de SQA (Cont.)

- I. Propósito do plano
- II. Documentos de referência
- III. Administração
 - A. Organização
 - B. Tarefas
 - C. Responsabilidades
- IV. Documentação
 - A. Propósito
 - B. Documentos de engenharia de software exigidos
 - C. Outros documentos

Plano de SQA (Cont.)

- V. Padrões, práticas e convenções
 - A. Propósito
 - B. Convenções
- VI. Revisões auditoriais
 - A. Propósito
 - B. Requisitos de revisão
 1. Revisão dos requisitos de software
 2. Revisões de projeto
 3. Verificação de software e revisões de validação
 4. Auditoria funcional
 5. Auditoria física
 6. Auditorias in-process
 7. Revisões administrativas

Plano de SQA (Cont.)

- VII. Gerenciamento de configuração de software
- VIII. Reportagem de problemas e ações corretivas
- IX. Ferramentas, técnicas e metodologias
- X. Controle de código
- XI. Controle de mídia
- XII. Controle de fornecedores
- XIII. Coleta, manutenção e retenção de registros