**Atividades de garantia de qualidade de software**

Tudo o que move o mundo está envolto em padrões. No mundo tecnológico não é diferente. Os softwares ao serem desenlvidos trazem sem si uma série de questionamentos, tais como funcionalidade, segurança e um fator cada vez mais frequente em tudo o que consumimos: a qualidade, que se liga, intimamente, à satisfação de usuários, à confiabilidade; ao cumprimento do prazo estabelecido e ao objetivo especificado, ou seja, às funcionalidades do produto.

Estudos feitos pelo Departamento Nacional de Comércio do Instituto de Padrões e Tecnologia – NIST (2002) apontam que “bugs de software, ou erros são tão comuns e tão prejudiciais que eles custam a economia dos EUA cerca de US$ 59,5 bilhões anuais, ou cerca de 0,6 por cento do produto interno bruto”.

Para se alcançar metas de segurança, padrões para atingir a Garantia da Qualidade de Software (SQA) foram introduzidos. Os mais conhecidos são IEEE e ISO, desenvolvem softwares em conformidade com a SQA. A ISO 9000, por exemplo, descreve elementos da garantia da qualidade, em termos gerais, que podem ser aplicados a qualquer empresa, independentemente do tipo de produto ou serviço.

Revisões técnicas e testes são realizados constantemente para se revelar e descobrir erros. E os estudos não param por aí. Compreender como os erros são introduzidos nos softwares é objeto de coletas e análises com o claro intuito de eliminá-los e garantir a qualificação do produto final. Toda mudança, seja em códigos ou no projeto, ocasionalmente trazem confusões. A SQA faz essa análise com a finalidade de não ocasionar erros buscando melhorar práticas.

A SQA quer um produto final de alta qualidade e por isso suas tarefas são realizadas por um grupo para alcançar essa meta, seguindo um plano de ação: primeiramente, prepara-se um plano de SQA para um projeto; a equipe participa no desenvolvimento da descrição de qualidade do projeto, revisa as atividades de engenharia de software e inspeciona softwares resultantes para verificar sua conformidade com a gestão da qualidade definida, garante, do mesmo modo, que o projeto seja documentado e registra qualquer problema para que seja resolvido. Essas tarefas servem para atingir qualidade dos requisitos, qualidade do projeto, qualidade do código e eficácia do controle de qualidade.

A garantia da qualidade está intimamente ligada, ao efeito que o software é capaz de proporcionar à sociedade. Sabe-se que um bug ou erro pode trazer uma série de complicações em determinadfos ambientes. Sendo assim, quando se alcança a qualidade tem-se a confiabilidade de que as funcionalidades estarão sendo eficazes, funcionarão corretamente e falhas não serão encontradas ao ser usadas. Em suma, avaliar a qualidade de um produto significa proceder a uma investigação científica minuciosa sobre a mesma. Implica em medir também, estatísticas de SQA.

Se um projeto computacional começa a falhar frequentemente e falhas são associadas a problemas de projeto ou de implementação, quem o desenvolve perde a confiabilidade. Quando o usuário final percebe a frequência de erros fica preocupado e perde a confiança que existia. A proteção do software é uma atividade que identifica e avalia problemas que provocam falhas.

Conclui-se, que a preocupação em construir softwares, vai além, de implementar códigos em computadores. Existe a necessidade de revisão constante para garantir a qualidade.

**Conceitos de Qualidade** (Algumas etapas para garantir a qualidade do software)

Garantia de Qualidade de Software – SQA, é uma atividade guarda-chuva (atividade aplicável durante todo o processo de software, tal como medição e gestão de risco).

**SQA engloba**

(1) Uma abordagem de gestão da qualidade

(2) Tecnologia de engenharia de software eficaz

(3) Revisões técnicas formais

(4) Estratégia de teste multi-camadas

(5) Controle de mudanças documento

(6) Padrão de desenvolvimento de software e seu processo de controle

(7) Mecanismo de medição e elaboração de relatórios

**Qualidade –** refere-se as características mensuráveis de um software. Estes itens podem ser comparados com base em uma determinada norma.

Há dois tipos de controle de qualidade:

**Qualidade do projeto:** as características que os projetistas especificam para um item. Inclui:requisitos, especificações e concepção do sistema.

**Qualidade de conformidade:** o grau em que a especificação de concepção são seguidas. Centra-se em implementação com base no design.

**Controle de Qualidade**

O que é controle de qualidade? Uma série de inspeções, avaliações e testes utilizados em todo o ciclo de desenvolvimento de um produto de software.

Controle de Qualidade inclui um loop de feedback para o processo.

**Objetivo:** Minimizar os defeitos produzidos e aumentar a qualidade do produto.

**Abordagens de implementação:**

Totalmente automatizado;

Inteiramente manual;

Combinação de ferramentas automatizadas e interações humanas

**Conceito-chave de controle de qualidade:**Comparar os produtos de trabalho com as normas especificadas e mensuráveis

**A garantia de qualidade consiste em:**Função de auditoria e reporte para suportar o gerenciamento

**Objetivo:**

Promover o gerenciamento dos dados necessários para a qualidade do produto.

Obter a percepção e a confiança da qualidade do produto

**Custo da Qualidade**

**Custo da Qualidade:** inclui todos os custos incorridos na busca da qualidade ou trabalhos realizados, relacionados com a qualidade.

**Custo da qualidade inclui:**

**Custo de prevenção:**

* Planejamento da qualidade
* Revisões técnicas formais
* Equipamento de teste
* Formação

**Custo de avaliação:**

* Em processo e inter-processo de inspeção
* Equipamentos de calibração e manutenção
* testes

**O custo de falha**

**Custo de falha interna:**Retrabalho, reparo e Análise de Modos de Falhas e Efeitos

**Custo de falha externa:**

* Resolução de queixas
* Retorno e substituição do produto
* Help desk / Sustentação
* Serviço de Garantia

**Garantia da Qualidade de Software –**Software Quality Assurance – SQA

**Objetivo:** conceber produtos de software de alta qualidade

**Definição de qualidade:**

“Conformidade com requisitos funcionais e de desempenho, explicitamente estabelecidos, padrões de desenvolvimento explicitamente documentados, e características implícitas sobre o esperado para todo o software profissionalmente desenvolvido.”

**Três pontos de importância para a medição da qualidade:**

Utilizar exigências como base;

Utilizar normas especificadas como os critérios;

Considerar os requisitos implícitos

**Sobre a garantia de qualidade:**

A primeira garantia formal de qualidade e função de controle, foi introduzido no mundo da produção no Bell Labs em 1916.

Durante os anos 1950 e 1960, os programadores controlaram a qualidade do produto.

Durante os anos 1970, as normas de garantia de qualidade foram introduzidas primeiramente em contratos militares de desenvolvimento de software.

Em 1987, uma definição estendida é dada em (SCH87).

**Software Reviews**

**O que é software reviews (revisões de software)?** Um “filtro” para o processo de engenharia de software.

**Finalidade**: serve para descobrir erros em análise, projeto, codificação e teste.

**Por que revisões de software?**

Errar é humano

É fácil pegar os erros no trabalho dos engenheiros

**Uma revisão é uma forma de:**

Identificar as melhorias necessárias das partes de um produto;

Confirmar as melhorias nas partes de um produto;

Realizar um trabalho técnico mais uniforme, previsível e gerenciável.

**Diferentes tipos de Revisões:**

**Revisões informais:** reunião informal e recepção informal de verificação

**Revisões formais:** (para uma plateia de clientes, gestores e pessoal). Passo a passo, inspeção e revisões round-robin (ciclos)

**Os termos defeitos e falhas são sinônimos de:** problemas de qualidade encontrados após lançamento do software

**Software com “erro”**: refere-se a um problema de qualidade encontrado pelos engenheiros antes do lançamento do software

**Revisões Técnicas Formais (FTR)**

**Objetivos da FTR:**

* descobrir erros de função, lógica ou implementação
* verificar se o software em análise corresponde às suas exigências
* garantir que o software tenha sido representado de acordo com padrões pré-definidos
* desenvolver software de maneira uniforme
* fazer projetos mais gerenciáveis

**Proposta da FTR:**

Servir como um campo de treinamento para engenheiros juniores (lições aprendidas);

Promover o backup e continuidade

**Reunião Formal de Análise Técnica**

**A preparação de uma reunião de avaliação:**

Agenda e horário da reunião;

Material de revisão e distribuição;

Revisão com antecedência

**Resultados da reunião de Revisões:**

Uma revisão da lista de problemas;

Uma revisão simples do relatório sintético (chamado de atas de reuniões).

**As decisões da reunião:**

Aceitar o produto do trabalho e ou modificação adicional;

Rejeitar o produto do trabalho devido a erros ;

Aceitar o trabalho sob condições (por exemplo, mudança e revisão);

Folha de sign-off.

**Relatório sintético da Revisão (um registro histórico do projeto) responde às seguintes perguntas:**

O que foi revisado?

Quem analisou isso?

Quais foram os resultados e conclusões

**Revisão da lista de problemas serve a dois propósitos:**

Para identificar áreas problemáticas no projeto;

Para servir como um check-list de itens para ação (é necessário um processo de follow- up) .

**Diretrizes de Revisão (para FTR)**

**Um conjunto mínimo de diretrizes para FTR:**

* Rever o produto, e não o produtor;
* Definir uma agenda e mantê-la;
* Limitar debates e refutações;
* Declarar áreas problemáticas, mas não tentar resolver todos os problemas observados;
* Tomar notas escritas;
* Limitar o número de participantes e insistir na preparação prévia;
* Elaborar uma lista de verificação para cada parte do produto que é susceptível de ser revisado;
* Alocar recursos e calendário para FTRs;
* Realizar treinamento significativo para todos os revisores;
* Rever suas primeiras revisões.

**Statistical Quality Assurance**

**Garantia de qualidade estatística:**reflete uma tendência crescente em todo o setor para a qualidade se tornar mais quantitativa.

**Garantia de qualidade estatística implica nas seguintes etapas:**

Informações sobre defeitos de software são coletadas e categorizadas;

É feita uma tentativa de rastrear cada defeito na sua causa raiz;

Usando o princípio de Pareto (80 por cento dos defeitos podem ser rastreados para 20 por cento, e isolar os 20 por cento);

Uma vez que as poucas causas vitais foram identificados, corrigir os defeitos.

**Causas de erros:**

* Especificação incompleta ou errada (IES)
* Interpretação de comunicação com o cliente (MCC)
* Desvio intencional da especificação (IDS)
* Violação dos padrões de programação (VPS)
* Erro na representação de dados (EDR)
* Módulo de interface inconsistente (IMI)
* Erro na lógica do design (EDL)
* Testes incompletos ou errados (IET)
* Documentação inexata ou incompleta (IID)
* Erro na tradução da linguagem de programação para o design (PLT)
* Interface homem- computador ambígua ou inconsistente (HCI)
* Diversos (MIS)

**Statistical Quality Assurance**

Em conjunto com a coleta de informações de defeitos, os desenvolvedores de software podem calcular um índice de erro (EI) para cada etapa nova no processo de engenharia de software.

**Após análise, projeto, codificação, testes e lançamento, são coletados os seguintes dados:**

Ei = o numero total de erros encontrados durante uma etapa no processo.

Si = o número de erros graves

Mi = o número de erros moderadas

Ti = o número de pequenos erros

PS = o tamanho do produto, durante uma etapa.

Em cada etapa do processo de engenharia de software, um índice de fase ( Pl i ) é computado:

PI i = ws (Si / Ei) + wm (Mi / Ei) + peso (Ti / Ei)

**Índice de erro (El) pode ser calculado como se segue :**

EI = (PI 1 + 2 PI 2 + 3 PI 3 + IPI I) / PS

**O Plano de SQA**

**O plano de SQA fornece um roteiro para instituir a Garantia de Qualidade de Software.** (segue um esboço de planos SQA feito pelo IEEE [ IEEE94 ] .)

**Itens básicos:**

* Objetivo do plano e seu escopo
* Gerenciamento
* Estrutura organizacional, as tarefas de SQA, a sua aplicação no processo
* Papéis e responsabilidades relacionadas à qualidade do produto
* Documentação
* Documentos do projeto, modelos, documentos técnicos, documentos de usuários.
* Normas, práticas e convenções
* Revisões e auditorias
* Teste – plano de teste e procedimento
* Relatório de problemas e ações de correção
* Ferramentas
* Controle de código
* Controle da mídia
* Controle de fornecedor
* Coleção de registros, manutenção e retenção
* Treinamento
* Gerenciamento de risco

**Referência**

# SQA no Processo de Software. Qualidade de Software. Acessado em 25 Mai de 2019. Disponivel em https://rpnsuporte.wordpress.com/category/qualidade-de-software/.