

USJT - UC Gestão de Qualidade de Software - Turma: ADS1AN-BUC1-6272430
RA 823159742 - Igor Cordeiro de Souza Pereira - 823159742@ulife.com.br
RA 823217461 - Lucca Palmieri Dittrich - 823217561@ulife.com.br

USJT-GQS-ADS1AN-BUC1-6272430-Atividade da Prática 02

Atividade 1 - Conceitos de Técnicas de Revisão de Software

Atividade 1:

- Tema:
Conceitos de Técnicas de Revisão de *Software*;
- Objetivo:
Realizar uma pesquisa sobre o tema e fazer uma explanação sobre o resultado encontrado;
- Entrega:
Até o prazo estipulado pela atividade proposta pelo *Google Classroom*, criar um documento contendo os seguintes dados dos integrantes do grupo:
 - Nome para o Grupo;
 - RA dos integrantes do Grupo;
 - Nome completo dos integrantes do Grupo;
 - Turma e curso de cada integrante do grupo; e
 - E-mails institucionais de cada Integrante do Grupo.
- Observação:
Gerar um PDF desse documento e, cada integrante do grupo, publicar como atividade, uma cópia desse documento.

Introdução

O processo de revisão de software é essencial para garantir a qualidade do do serviço, diminuindo a necessidade de correção de bugs do desenvolvimento. Ajuda a prevenir problemas e torna o projeto eficiente. Analisar métricas de revisão e entender a diferença entre revisões permite que o processo seja mais eficaz, maximizando os benefícios e minimizando gastos.

Do contrário, os problemas no projeto podem afetar a confiabilidade, gerar atrasos, insatisfação do cliente e atingir a saúde e integridade da equipe.

Custos

Um dos principais afetados quando há atraso ou falha na entrega do software é o custo financeiro, nos últimos anos empresas no mundo inteiro perderam milhões em valores devido a problemas no desenvolvimento.

Um dado de 2017 estima uma quantia de aproximadamente \$1,7 trilhões.

Como não é possível ter 100% de eficiência, uma boa iniciativa é concentrar 20% do tempo de trabalho para encontrar e corrigir erros, e os 80% restantes em criar recursos e melhorar o produto.

Processo de Inspeção e Revisão

O processo de inspeção descrito por Sommerville envolve várias etapas, começando com:

Planejamento, onde se organizam os detalhes da inspeção.

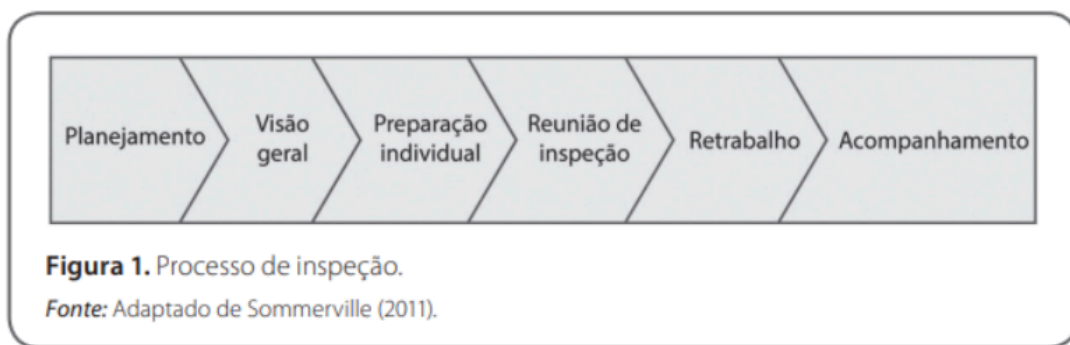
Visão geral, onde todos entendem o que será inspecionado.

Preparação individual, onde cada desenvolvedor analisa o material.

Reunião de inspeção, onde os erros encontrados são discutidos em grupo.

Retrabalho, onde os defeitos identificados são corrigidos.

Acompanhamento, para garantir que tudo foi corrigido corretamente.



Referências:

OLIVEIRA, C. Utilização de checklist para validação de requisitos de software. 30 abr. 2014. Disponível em:

<https://imasters.com.br/devsecops/utilizacao-de-checklist-para-validacao-de-requisitos-de-software>.

Acesso em: 02 set. 2024.

PRESSMAN, R. S.; MAXIM, B. R. Engenharia de software: uma abordagem profissional. 8. ed. Porto Alegre: Bookman, 2016. Acesso em: 02 set. 2024.

NUZZI, J. R. Quanto os erros em software podem custar para a sua empresa? Medium, 05 jul. 2017. Disponível em:

<https://medium.com/@jonesroberto/quanto-os-erros-em-software-podem-custar-para-a-sua-empresa-409b77d08bd6>. Acesso em: 02 set. 2024.

Atividade 2 - Conceitos de Garantia de Qualidade de Software

O que é?

SQA(Software Quality Assurance) é um conjunto de tarefas, objetivos, métricas e planos, com o intuito de garantir e padronizar a qualidade de desenvolvimento de um projeto de software.

Por que é importante?

Ao enfatizar a qualidade em todas as etapas do desenvolvimento, é reduzido a quantidade de retrabalho necessário, reduzindo o tempo de entrega e os custos. A equipe SQA(software quality assurance) funciona como um representante do cliente dentro da empresa, e devem implementar as atividades SQA do ponto de vista dele. Verificando se o software corresponde adequadamente aos fatores de qualidade e se seu desenvolvimento foi conduzido conforme os padrões pré-estabelecidos de controle de qualidade..

Plano de SQA

O plano de SQA oferece um roteiro para a implementação das atividades de garantia da qualidade em projetos de software. Este plano inclui a definição do escopo, padrões aplicáveis, ações de SQA, ferramentas de suporte, e responsabilidades organizacionais.

Elementos de Garantia da Qualidade de Software

A SQA abrange uma ampla gama de atividades, incluindo conformidade com padrões, revisões e auditorias, teste de software, coleta e análise de erros, gerenciamento de mudanças, e educação contínua de todos os envolvidos no processo.

Processos da SQA e Características do Produto

A SQA envolve tarefas de planejamento, supervisão, manutenção de registros, análise e relatórios. As características de qualidade do produto incluem requisitos, design, código e a eficácia do controle de qualidade, sendo todos monitorados através de métricas específicas.

Tarefas, Metas e Métricas de SQA

As tarefas da SQA incluem a preparação de planos de SQA, participação no desenvolvimento dos processos de software, auditoria de produtos de trabalho, e garantia de conformidade com os processos estabelecidos. As metas incluem garantir a qualidade dos requisitos, do design e do código, além da eficácia do controle de qualidade.

Goal	Attribute	Metric
Requirement quality	Ambiguity	Number of ambiguous modifiers (e.g., many, large, human-friendly)
	Completeness	Number of TBA, TBD
	Understandability	Number of sections/subsections
	Volatility	Number of changes per requirement Time (by activity) when change is requested
	Traceability	Number of requirements not traceable to design/code
	Model clarity	Number of UML models Number of descriptive pages per model Number of UML errors
Design quality	Architectural integrity	Existence of architectural model
	Component completeness	Number of components that trace to architectural model Complexity of procedural design
	Interface complexity	Average number of pick to get to a typical function or content Layout appropriateness
	Patterns	Number of patterns used
Code quality	Complexity	Cyclomatic complexity
	Maintainability	Design factors (Chapter 8)
	Understandability	Percent internal comments Variable naming conventions
	Reusability	Percent reused components
	Documentation	Readability index
QC effectiveness	Resource allocation	Staff hour percentage per activity
	Completion rate	Actual vs. budgeted completion time
	Review effectiveness	See review metrics (Chapter 14)
	Testing effectiveness	Number of errors found and criticality Effort required to correct an error Origin of error

Estatística da Garantia da Qualidade de Software

A SQA estatística envolve a coleta e categorização de erros e defeitos para identificar as causas principais dos problemas. A técnica de Six Sigma é um exemplo de abordagem estatística amplamente utilizada para melhorar a qualidade através da eliminação de defeitos.

Confiabilidade de Software

A confiabilidade do software é medida em termos de probabilidade de operação livre de falhas em um ambiente específico durante um período de tempo específico. É calculada com métricas como o tempo médio entre falhas (MTBF) e a disponibilidade do software.

Padrões ISO 9000

Os padrões ISO 9000 fornecem um sistema genérico de garantia da qualidade que pode ser aplicado em qualquer tipo de empresa. Para se tornar registrado sob o ISO 9001:2000, uma organização de software deve estabelecer políticas e procedimentos para garantir que os requisitos de qualidade sejam atendidos.

Referências:

[PRESSMAN, R. S. Software Engineering: A Practitioner's Approach. 7. ed. New York: McGraw-Hill, 2010. cap. 16. p. 432.](#)