

Aula 16 - Gestão de Qualidade

Disciplina: Gestão de Projetos Tecnológicos

Prof. Me. João Paulo Biazotto

Áreas de Conhecimento Projeto

- Escopo
- Tempo
- Qualidade
- Recursos Humanos
- ...

Ciclo de Vida do Projeto

- Como monitorar e manter registros de todos esses aspectos?



Plano de Gerenciamento do Projeto






Documento que descreve como o projeto será **executado, monitorado, controlado e encerrado.**

Este plano garante que os projetos sejam entregues dentro do prazo, orçamento e com a qualidade esperada.







Plano de Gerenciamento do Projeto

Redução de riscos, melhor alocação de recursos e maior **previsibilidade** dos resultados.

Plano de Gerenciamento do Projeto

- Escopo 
- Tempo 
- Custos 
- Qualidade
- Recursos Humanos
- Comunicação 
- Riscos
- ~~Aquisições~~
- Partes Interessadas 

Plano de Gerenciamento do Projeto

- Escopo 
- Tempo 
- Custos 
- Qualidade 
- Recursos Humanos
- Comunicação 
- Riscos
- ~~Aquisições~~
- Partes Interessadas 

Plano de Gerenciamento de Qualidade

O Plano de Gerenciamento da Qualidade estabelece os **padrões de qualidade relevantes para o projeto** e descreve **como** esses padrões serão **alcançados, monitorados e controlados**.

Ele é parte integrante do Plano de Gerenciamento do Projeto e estabelece as diretrizes e procedimentos para garantir que o nível esperado de qualidade no projeto.

Plano de Gerenciamento de Qualidade

- Definir os **requisitos de qualidade** para o produto, serviço ou resultado do projeto.
- Estabelecer **critérios para aceitação** das entregas.
- Selecionar **práticas**, ferramentas e **padrões de qualidade** a serem aplicados.
- Descrever **responsabilidades** da equipe relacionadas à qualidade.
- **Integrar a qualidade ao processo** do projeto, não apenas ao produto final.

Plano de Gerenciamento de Qualidade

- Definir os **requisitos de qualidade** para o produto, serviço ou resultado do projeto.
 - Cobertura de código
 - Tempo de resposta
 - Acessibilidade

Plano de Gerenciamento de Qualidade

- Estabelecer **critérios para aceitação** das entregas.
 - Pelo menos 95% de cobertura de teste
 - Resposta em até 2 segundos
 - Existência de opção de alto contraste

Plano de Gerenciamento de Qualidade

- Selecionar **práticas**, ferramentas e **padrões de qualidade** a serem aplicados.
 - ISO/IEC 25010
 - SOLID
 - Clean Code

Plano de Gerenciamento de Qualidade

- Descrever **responsabilidades** da equipe relacionadas à qualidade.
 - Desenvolvedor: aplicar boas práticas, escrever testes unitários
 - QA/Testador: elaborar casos de teste, executar testes manuais e automatizados
 - Product Owner: validar critérios de aceitação e coletar feedback de usuários

Plano de Gerenciamento de Qualidade

- **Integrar a qualidade ao processo** do projeto, não apenas ao produto final.
 - Revisões de código contínuas durante o desenvolvimento
 - Execução de testes em cada entrega incremental (ex: em sprints)
 - Coleta de feedback após cada deploy em produção

Plano de Gerenciamento de Qualidade

- Beleza, mas e na prática como é que isso funciona?



Qualidade de Produto vs Qualidade de Processo

- **Qualidade de Produto:** foca no que é entregue.
- **Qualidade de Processo:** foca em como se trabalha para entregar.

Qualidade de Produto vs Qualidade de Processo

| Aspecto | Qualidade de Produto | Qualidade de Processo |
|--------------------------------|---|--|
| Foco | O software entregue ao cliente | O caminho seguido para desenvolver o software |
| Objetivo | Atender ou exceder as expectativas do usuário final | Garantir que as atividades do projeto sejam bem executadas |
| Exemplos de Métricas | <ul style="list-style-type: none">- Defeitos por KLOC- Desempenho- Usabilidade | <ul style="list-style-type: none">- Aderência ao processo ágil- Frequência de revisões de código- Taxa de cobertura de testes |
| Ferramentas comuns | Testes de aceitação, testes funcionais, pesquisa de usuários | Auditorias, checklist de boas práticas, métricas de processo |
| Responsáveis principais | QA, testadores, usuários finais | Gerente de projeto, equipe de desenvolvimento, Scrum Master |
| Avaliação | Avaliada ao final ou durante as entregas | Avaliada continuamente ao longo do projeto |

Qualidade de Produto vs Qualidade de Processo

- **Qualidade de Produto:** ISO 25010
- **Qualidade de Processo:** CMMI e MPS.BR

ISO 25010

- [Norma internacional](#) para **avaliação de qualidade** de software
- Define **8 características** de qualidade e subcaracterísticas
- Aplicável em desenvolvimento, teste, avaliação e manutenção de software
- Exemplo: Ao criar um app bancário, a norma ajuda a garantir que ele seja seguro, confiável, usável e manutenível.

ISO 25010

functional suitability

... provide functions that meet stated and implied needs of intended users when it is used under specified conditions

performance efficiency

... perform its functions within specified time and throughput parameters and be efficient in the use of resources under specified conditions

compatibility

... exchange information with other products, and/or to perform its required functions while sharing the same common environment and resources

interaction capability

... to be interacted with by specified users to exchange information between a user and a system via the user interface to complete the intended task

reliability

... perform specified functions under specified conditions for a specified period of time without interruptions and failures

security

... protect information and data so that persons or other products have the degree of data access appropriate to their types and levels of authorization, and to defend against attack patterns by malicious actors

maintainability

... be modified by the intended maintainers with effectiveness and efficiency

flexibility

... be adapted to changes in its requirements, contexts of use, or system environment

safety

... under defined conditions to avoid a state in which human life, health, property, or the environment is endangered

From ISO/IEC 25010:2023(en), (c) ISO.org

ISO 25010

- Funcionalidade (Adequação Funcional)
- Desempenho/Eficiência
- Compatibilidade
- Usabilidade
- Confiabilidade
- Segurança (*Security*)
- Manutenibilidade
- Portabilidade/Flexibilidade
- Segurança (*Safety*)

Funcionalidade (Adequação Funcional)

- O software faz o que se espera?
- A funcionalidade atende aos requisitos do usuário?

Exemplo: Um sistema de vendas que calcula corretamente descontos conforme a política da empresa.

Funcionalidade (Adequação Funcional)

Completude funcional: tudo o que foi solicitado está presente?

Corretude funcional: as funções estão corretas?

Adequação funcional: as funções realmente atendem ao propósito?

Desempenho e Eficiência

O software responde rápido e usa bem os recursos?

Exemplo: Um site de e-commerce carrega em menos de 2 segundos, mesmo com 1000 acessos simultâneos.

Desempenho e Eficiência

Tempo de resposta: quanto tempo demora para responder?

Uso de recursos: quanto CPU, memória, rede?

Capacidade: quanto tráfego ou carga ele suporta?

Exemplo: Aplicativo de videoconferência que funciona bem mesmo com conexão 3G.

Compatibilidade

O software funciona bem com outros sistemas?

Exemplo: Um plugin do navegador funciona em Chrome, Firefox e Edge, sem conflitos.

Compatibilidade

- **Coexistência:** não interfere em outros sistemas
- **Interoperabilidade:** consegue trocar dados com outros sistemas

Exemplo: Sistema de gestão escolar que exporta dados para a Receita Federal.

Usabilidade

- O sistema é fácil de usar, aprender e entender?

Exemplo: Um app de delivery onde o usuário consegue fazer pedidos sem ler tutoriais.

Usabilidade

- **Reconhecimento e operabilidade:** fácil de entender e operar
- **Estética:** aparência agradável
- **Acessibilidade:** acessível para PCD

Exemplo: Site que permite navegação via teclado e leitor de tela.

Confiabilidade

- O software é consistente e estável em diferentes condições?

Exemplo: Aplicativo bancário que não falha em transações, mesmo com conexões instáveis.

Confiabilidade

- **Maturidade:** baixa taxa de falhas
- **Disponibilidade:** está disponível quando necessário
- **Tolerância a falhas:** continua funcionando mesmo com erros parciais

Exemplo: Sistema de pagamento com circuit breaker em caso de falha.

Segurança (*Security*)

- O software protege dados e impede acessos indevidos?

Exemplo: Aplicativo de saúde que criptografa dados do paciente e exige autenticação de dois fatores.

Segurança (*Security*)

- **Confidencialidade:** protege dados sensíveis
 - LGPD, GDPR.
- **Autenticidade:** garante a identidade do usuário
- **Auditorabilidade:** permite rastrear acessos

Exemplo: Sistema ERP que registra quem acessou e alterou relatórios financeiros.

Manutenibilidade

- O software é fácil de modificar e corrigir?

Exemplo: Sistema modular onde é fácil alterar regras fiscais sem mexer no sistema todo.

Manutenibilidade

- **Modularidade:** partes isoladas e independentes
- **Reusabilidade:** código pode ser reaproveitado
- **Analísabilidade:** fácil de entender falhas
- **Testabilidade:** fácil de testar

Exemplo: Microserviços bem isolados e com testes automatizados por componente.

Portabilidade

- O software pode ser instalado e usado em diferentes ambientes?

Exemplo: Aplicativo desktop que funciona em Windows, macOS e Linux.

Portabilidade

- **Adaptabilidade:** fácil adaptação a novas plataformas
- **Instalabilidade:** instalação simples
- **Substituibilidade:** pode ser trocado facilmente por outro similar

Exemplo: Web app com responsivo, que roda no navegador e no celular.

Segurança (*Safety*)

- O sistema mitiga possíveis riscos a vida humana, a saúde, a propriedade ou o meio ambiente?

Exemplo: Um sistema aviônico previne falhas na execução sistema?

Segurança (*Safety*)

- **Restrição operacional:** O sistema funciona só dentro de limites seguros.
- **Identificação de riscos:** O sistema percebe quando algo pode causar perigo.
- **Falha segura:** O sistema se protege sozinho quando algo dá errado.

Exemplo: O software de controle de voo limita os comandos do piloto para manter o avião dentro dos parâmetros seguros de velocidade e altitude.

Estudo de Caso - App de Banco

Um sistema bancário digital é uma aplicação de software projetada para gerenciar, automatizar e disponibilizar serviços financeiros oferecidos por uma instituição bancária.

Ele permite que clientes realizem operações como consulta de saldo, transferências, pagamentos, investimentos e contratação de produtos financeiros, de forma segura e eficiente, por meio de interfaces web ou mobile.

Estudo de Caso - App de Banco

| Característica ISO 25010 | Item de Verificação (Checklist) |
|--------------------------|--|
| Funcionalidade | O app permite realizar transferências, pagamentos e visualização de saldo? |
| Usabilidade | A navegação pode ser feita sem necessidade de tutoriais? |
| Segurança | O app exige autenticação por biometria ou 2FA para transações financeiras? |
| Desempenho | O tempo médio de carregamento da tela inicial é inferior a 3 segundos? |
| Confiabilidade | O app mantém a sessão do usuário ativa de forma estável durante o uso? |

Sumarizando...

- A ISO/IEC 25010 ajuda a construir software com qualidade **multidimensional**
- Pode ser usada para:
 - Definir critérios de aceitação
 - Guiar o desenvolvimento
 - Avaliar e comparar produtos

Na próxima aula...

- Modelos de Qualidade de **Processo**
 - MPS.BR
 - CMMI

Atividade

- Utilizar a **ISO 25010** e definir um checklist com 15 itens para checar a qualidade do sistema da escola de T.I.
- Utilize o exemplo em aula como base
- **Não avaliativa :)**

DÚVIDAS?