

# **Organizar o Processo de Trabalho no Desenvolvimento de Aplicações**

**Prof. Me. Gabriel Caixeta Silva**

# Introdução

- **Importância da organização no desenvolvimento de aplicações.**
  - A organização é a espinha dorsal de qualquer projeto de desenvolvimento de aplicativos. Ela assegura que todos os elementos do projeto se encaixem de maneira eficiente, minimizando problemas e riscos.
- **Objetivo:**
  - Nossa meta é explorar como organizar efetivamente o processo de trabalho no desenvolvimento de aplicações, fornecendo insights práticos para melhorias significativas nos resultados.

## O que é Desenvolvimento de Aplicações?

- O desenvolvimento de aplicações envolve a criação de programas de software projetados para atender às necessidades específicas dos usuários. Essas aplicações podem variar de aplicativos móveis a sistemas de gerenciamento empresarial.
- **Exemplos** incluem aplicativos de redes sociais, aplicativos de e-commerce, sistemas de gerenciamento de projetos, entre outros.
- O desenvolvimento de aplicativos é complexo devido à necessidade de compreender requisitos variados, tecnologias em constante evolução e garantia de qualidade. Desafios incluem prazos apertados, orçamento limitado e requisitos em constante mudança.

# Por que a Organização é Importante?

- **Evitar erros e retrabalho.**
  - A organização ajuda a reduzir erros, o que economiza tempo e recursos, já que correções e retrabalhos frequentemente são dispendiosos.
- **Gerenciar recursos efetivamente.**
  - Ao organizar o processo de desenvolvimento, você pode alocar recursos (como equipe, tempo e orçamento) de forma eficiente, evitando desperdícios.
- **Cumprir prazos e metas.**
  - A organização auxilia na definição de metas realistas e no cumprimento de prazos, garantindo que o projeto siga um curso suave e não ultrapasse o cronograma.

# Métodos Tradicionais vs. Métodos Ágeis

## Métodos Tradicionais

- **Waterfall.**
  - O modelo em cascata é uma abordagem sequencial que divide o desenvolvimento em etapas distintas, como requisitos, design, implementação, testes e manutenção.
- **Vantagens e Desvantagens.**
  - As vantagens incluem uma estrutura clara e documentação sólida, mas a rigidez pode dificultar a adaptação a mudanças nos requisitos.

# Métodos Tradicionais vs. Métodos Ágeis

## Métodos Ágeis

- **Scrum:** é um framework ágil que promove a colaboração, a adaptabilidade e a entrega iterativa de funcionalidades.
- **Kanban:** é um sistema de gerenciamento visual que ajuda a otimizar o fluxo de trabalho, permitindo que a equipe se concentre em tarefas prioritárias.
- **Vantagens e Desvantagens.**
  - As vantagens dos métodos ágeis incluem maior flexibilidade e capacidade de resposta a mudanças, mas podem requerer adaptação contínua.

# Fluxo de Trabalho Organizado

## Passos Básicos

### 1. Levantamento de Requisitos.

- Esta fase envolve a identificação das necessidades dos stakeholders, incluindo usuários finais e clientes.

### 2. Planejamento.

- O planejamento é onde você define os objetivos, recursos necessários e um cronograma para o projeto.

### 3. Design.

- Nesta etapa, você cria a arquitetura de software e o design da interface do usuário.

# Fluxo de Trabalho Organizado

## Passos Básicos

### 4. Desenvolvimento.

- A equipe de desenvolvimento traduz o design em código funcional.

### 5. Testes.

- Testes rigorosos são realizados para identificar bugs e garantir a qualidade.



# Fluxo de Trabalho Organizado

## Passos Básicos

### 6. Implantação.

- O aplicativo é lançado em um ambiente de produção.

### 7. Manutenção.

- A fase de manutenção inclui correções de bugs, atualizações e melhorias contínuas.

## Uso de Ferramentas de Gestão de Projetos

- **Trello.**
  - O Trello é uma ferramenta de gerenciamento de projetos que usa cartões e quadros para ajudar a organizar tarefas e colaborar em equipe.
- **Jira.**
  - O Jira é uma ferramenta corporativa que fornece recursos avançados de gerenciamento de projetos, incluindo rastreamento de problemas e relatórios personalizáveis.
- **Asana.**
  - O Asana é uma plataforma de colaboração que permite rastrear tarefas, projetos e metas, facilitando a coordenação da equipe.

# Comunicação Efetiva

- **Importância da comunicação na organização.**
  - A comunicação eficaz é essencial para manter todos os membros da equipe alinhados, informados e motivados.
- **Reuniões regulares.**
  - Reuniões regulares, como reuniões diárias de stand-up, podem manter a equipe atualizada e identificar problemas rapidamente.
- **Uso de ferramentas de comunicação.**
  - Ferramentas como Slack e Microsoft Teams podem facilitar a comunicação e a colaboração entre membros da equipe, mesmo em ambientes remotos.

# Documentação

- **Criação e manutenção de documentação.**
  - Documentar requisitos, design, arquitetura e código é fundamental para garantir que todos na equipe estejam na mesma página.
- **Especificações de projeto.**
  - Especificações detalhadas de projeto ajudam a definir claramente os requisitos e metas do projeto.
- **Documentação de código.**
  - A documentação de código ajuda os desenvolvedores a entender e manter o código, facilitando futuras alterações.

## Automação de Processos

- **Automação de tarefas repetitivas.**
  - Automatizar tarefas rotineiras, como compilação, testes e implantação, pode economizar tempo e minimizar erros humanos.
- **Exemplo: Integração Contínua (CI) e Entrega Contínua (CD).**
  - A CI e a CD são práticas que automatizam o processo de integração, teste e implantação de código, permitindo entregas frequentes e consistentes.

## Gerenciamento de Mudanças

- **Adaptação a mudanças nos requisitos.**
  - A organização inclui a capacidade de se adaptar a mudanças nos requisitos do projeto, o que é comum em muitos desenvolvimentos de aplicativos.
- **Uso de controle de versão (ex: Git).**
  - O controle de versão, como o Git, permite rastrear todas as alterações no código-fonte, facilitando a colaboração da equipe e o gerenciamento de mudanças de forma eficaz.