

Prof. esp. Thalles Canela

- **Graduado:** Sistemas de Informação - Wyden Facimp
- **Pós-graduado:** Segurança em redes de computadores - Wyden Facimp
- **Professor:** Todo núcleo de T.I. (Graduação e Pós) - Wyden Facimp
- **Diretor:** SCS
- **Gerente de Projetos:** Motoca Systems

Redes sociais:

- **Linkedin:** <https://www.linkedin.com/in/thalles-canela/>
- **YouTube:** <https://www.youtube.com/aXR6CyberSecurity>
- **Facebook:** <https://www.facebook.com/axr6PenTest>
- **Instagram:** https://www.instagram.com/thalles_canela
- **Github:** <https://github.com/ThallesCanela>
- **Github:** <https://github.com/aXR6>
- **Twitter:** <https://twitter.com/Axr6S>

Modelos de Processos Prescritivos Aprofundados

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Modelos de desenvolvimento que oferecem um caminho claramente definido e sequencial para a construção de software.

Características principais:

- Previsibilidade: Cada fase é mapeada de forma estruturada e acontece em uma sequência específica.
- Documentação: Enfatiza a produção de artefatos detalhados em cada etapa.
- Controle: Mudanças são minimizadas e, quando necessárias, são rigorosamente gerenciadas.

Por que usar modelos prescritivos?

- Em ambientes onde a precisão e a previsibilidade são vitais. Exemplo: Desenvolvimento de software para aviação, onde uma mudança inesperada pode ter consequências críticas.

Métodos comuns:

- Modelo em Cascata: Onde cada fase (Requisitos, Design, Implementação, etc.) é concluída antes da próxima começar.
- Modelo Espiral: Que combina o design e a prototipagem em estágios, permitindo uma abordagem mais flexível e focada em riscos.
- Modelo V (V-Model): Que associa cada fase do desenvolvimento a uma fase de teste, enfatizando a verificação e validação.

Visão Geral dos Modelos Prescritivos

- Modelos de desenvolvimento que seguem uma sequência definida e clara de fases.
- Enfatizam a especificação detalhada e a documentação.

Características Fundamentais

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Sequencial: Cada fase deve ser concluída antes de passar para a próxima.
- Específico: Fases claramente definidas com entradas e saídas específicas.
- Controlado: Mudanças são gerenciadas rigorosamente.

Sequencialidade

- Os processos são executados em uma ordem específica.
- Cada fase tem um início e um fim claramente definidos.
- Exemplo: Pense em uma linha de montagem. Primeiro, o chassi do carro é construído, depois o motor é instalado, seguido pela instalação dos interiores e, finalmente, a pintura. Não se pode pintar o carro antes de instalar o motor.

Especificidade

- As fases e atividades são claramente definidas com entradas e saídas específicas.
- São estabelecidos critérios claros para a transição entre as fases.
- Exemplo: Antes de iniciar a fase de design em um projeto, os requisitos devem ser coletados e documentados. A saída da fase de coleta de requisitos (documentação de requisitos) é a entrada para a fase de design.

Controlado

- Mudanças são gerenciadas rigorosamente para evitar desvios e garantir a conformidade com os requisitos iniciais.
- Mecanismos de controle de qualidade são implementados para garantir que os produtos de trabalho atendam aos padrões.
- Método: Revisão Formal - um método onde um grupo de especialistas avalia um produto de software para identificar defeitos, violações de padrões de desenvolvimento e outros problemas.
- Exemplo: Após a codificação, uma revisão de código pode ser feita por pares para garantir que o código atenda aos padrões e não contenha erros.

Documentado

- Cada fase produz documentação detalhada para garantir a clareza e a transferência de conhecimento.
- Facilita a manutenção e a evolução do software a longo prazo.
- Exemplo: Diagramas UML para design, Documentação de Requisitos de Software (SRS), Planos de Teste, e Documentação de Usuário.

Modelo em Cascata

- Ilustração gráfica mostrando o fluxo descendente: Requisitos → Design → Implementação → Verificação → Manutenção.
- Ideal para projetos com requisitos bem definidos e estáveis.

Conceito Básico

- Representação gráfica do modelo em cascata, mostrando o fluxo descendente: Requisitos → Design → Implementação → Verificação → Manutenção.
- Definição: Um processo sequencial onde cada fase depende da conclusão da fase anterior.

Características do Modelo em Cascata

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Linearidade: Cada etapa começa apenas quando a anterior termina.
- Clareza: Cada etapa tem um resultado bem definido e um processo claro.
- Rigidez: Mudanças no meio do processo são complicadas e caras.

Fases Detalhadas

- Requisitos: Captura detalhada do que o sistema deve fazer. Exemplo: Um documento de requisitos para um sistema bancário online.
- Design: Projetar a solução do sistema, subdividido em design arquitetônico e design detalhado. Exemplo: UML (Unified Modeling Language) para representar a estrutura do sistema.
- Implementação: Codificação real do software. Exemplo: Usando a linguagem de programação Java para desenvolver um aplicativo de e-commerce.
- Verificação: Testar o sistema para garantir que ele atenda aos requisitos. Exemplo: Testes de unidade para verificar funções individuais, testes de integração para verificar fluxos de trabalho completos.
- Manutenção: Corrigir bugs e fazer melhorias no software entregue. Exemplo: Lançar patches para corrigir vulnerabilidades de segurança identificadas após a entrega.

Quando usar o Modelo em Cascata?

- Projetos com requisitos claramente definidos desde o início.
- Projetos onde o cliente tem uma visão clara e não espera mudanças no meio do caminho.
- Exemplo: Desenvolvimento de um software de controle para um equipamento médico, onde os requisitos e regulamentações são rigorosos e bem definidos.

Métodos e Ferramentas Associadas

- Ferramentas de modelagem, como ERD (Entity Relationship Diagram) para design de banco de dados ou UML para design de software.
- Ferramentas de gerenciamento de requisitos como IBM Rational DOORS.
- Softwares de gerenciamento de projetos como Microsoft Project para rastrear o progresso.
- Ferramentas de teste como JUnit para testes de unidade ou Selenium para testes de interface do usuário.

Vantagens e Desvantagens

- Vantagens: Processo claro e estruturado, documentação robusta, fácil de gerenciar se os requisitos são bem conhecidos.
- Desvantagens: Difícil de adaptar a mudanças, risco de descobrir problemas tarde no processo, pode ser mais demorado que outras abordagens.


Estudos de Caso

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Exemplo de Sucesso: Um sistema de gerenciamento de estoque para uma grande cadeia de lojas, onde os requisitos eram bem conhecidos e inalterados.
- Exemplo de Desafio: Um aplicativo de mídia social inovador, onde o mercado e os requisitos dos usuários estavam em constante mudança.

Modelo Espiral

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Ilustração em forma de espiral, destacando as fases: Planejamento, Análise de Riscos, Engenharia, Avaliação.
 - Enfatiza a avaliação e mitigação de riscos em cada iteração.
- 
- A thin grey horizontal bar is located at the bottom of the slide.

Fases do Modelo Espiral:

- Planejamento: Definição de objetivos, alternativas e restrições do projeto.
- Exemplo: Para um software de gestão de bibliotecas, nesta fase, definimos as principais funcionalidades, como registro de novos livros, rastreamento de empréstimos e gestão de usuários.
- Método: Brainstorming com a equipe e stakeholders para identificar e priorizar funcionalidades.

Fases do Modelo Espiral:

- Análise de Riscos: Avaliação de opções e identificação de riscos, propondo soluções para mitigá-los.
- Exemplo: Ao considerar a integração de um sistema de pagamento online para multas, identificamos riscos como falhas de segurança e fraude.
- Método: SWOT (Forças, Fraquezas, Oportunidades, Ameaças) para avaliar riscos e potenciais soluções.

Fases do Modelo Espiral:

- Engenharia: Desenvolvimento e verificação do produto.
- Exemplo: Criar e testar a interface do usuário para o sistema de pagamento online mencionado anteriormente.
- Método: Prototipagem para visualizar a solução e testes unitários para verificar sua funcionalidade.

Fases do Modelo Espiral:

- Avaliação: Obter feedback do cliente e dos stakeholders, refinando o produto.
- Exemplo: Apresentar o sistema de pagamento integrado a uma seleção de usuários e coletar suas opiniões.
- Método: Grupos focais ou entrevistas para obter feedback direto.

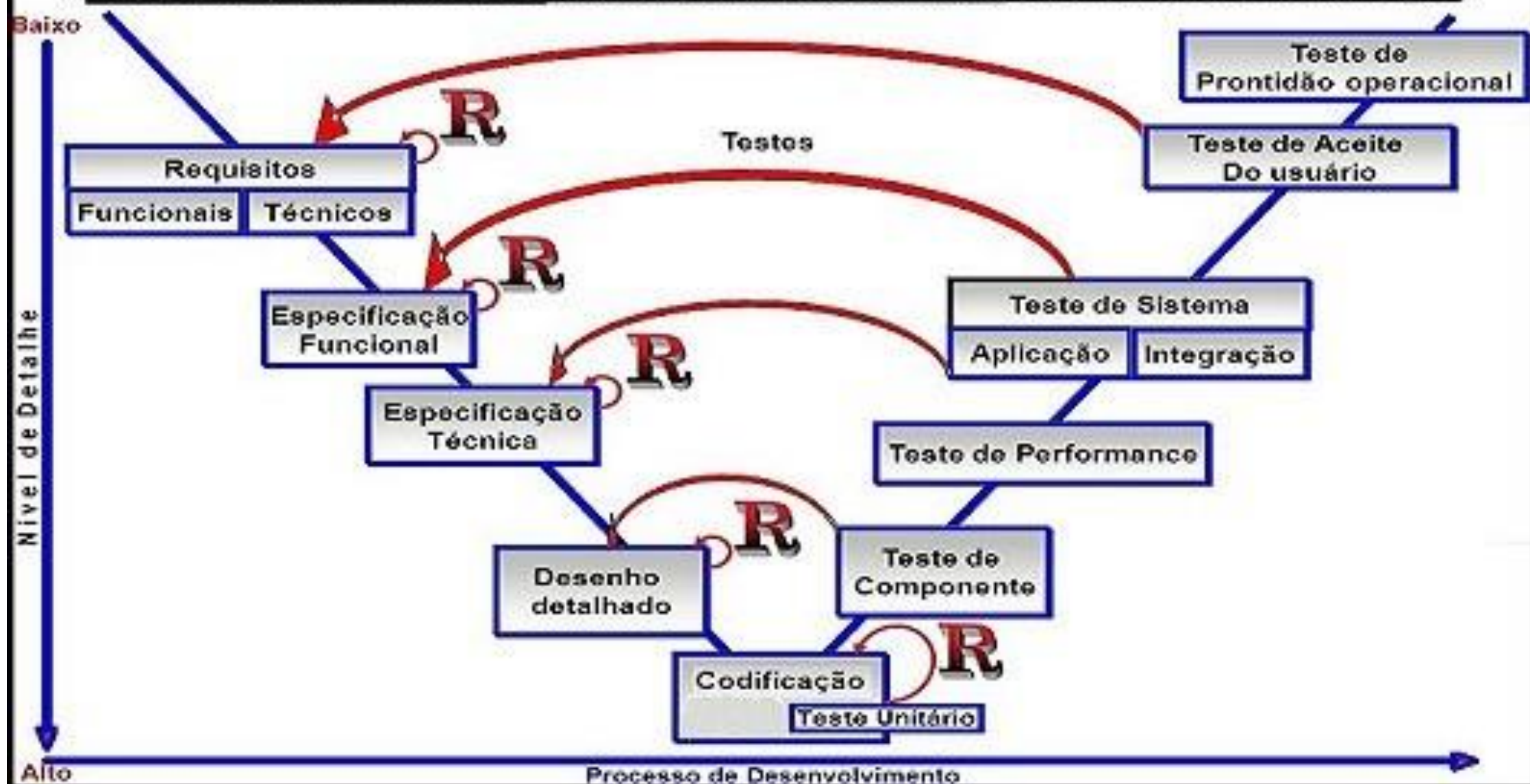
Modelo V (V-Model)

- Gráfico em forma de "V" mostrando as fases de desenvolvimento à esquerda e as fases de teste à direita.
- Enfatiza a verificação e validação.

Estrutura Gráfica:

- Lado esquerdo (Desenvolvimento): Análise de Requisitos → Design de Sistema → Design Arquitetônico → Design de Módulo → Codificação.
- Base do V: Codificação.
- Lado direito (Teste): Teste de Unidade → Teste de Integração → Teste de Sistema → Teste de Aceitação.

O V-Model ilustra o "VER&VAL" (Verificação e Validação) para cada fase do ciclo de desenv.



Vantagens do Modelo V:

- Clareza de Fases: Cada fase de desenvolvimento tem uma fase de teste correspondente.
- Detecção Precoce de Erros: Problemas podem ser identificados e corrigidos mais cedo no ciclo de vida.
- Enfoque Rigoroso: O modelo garante que o produto final esteja bem testado em todos os níveis.

Exemplo Prático:

- Projeto de uma Loja Online:
- Análise de Requisitos: Definição da capacidade de suportar 10.000 usuários simultâneos, pagamento seguro, etc.
- Design de Sistema: Escolha da arquitetura do servidor, integração com gateways de pagamento.
- Design Arquitetônico: Esquema do banco de dados, design do backend e frontend.
- Design de Módulo: Desenvolvimento de módulos individuais como carrinho de compras, página de produto, etc.

Exemplo Prático:

- Codificação: Escrita de código para cada módulo.
- Teste de Unidade: Teste do módulo do carrinho de compras isoladamente.
- Teste de Integração: Teste do carrinho de compras em conjunto com a página de produto.
- Teste de Sistema: Testando a loja online como um todo.
- Teste de Aceitação: Teste final onde um grupo selecionado de usuários tenta realizar compras para garantir que tudo funciona como esperado.

Métodos e Ferramentas Associadas:

- Diagramas de Fluxo de Dados (DFD): Utilizados durante as fases de design para visualizar o fluxo de informações.
- JUnit: Uma ferramenta de teste de unidade popular para Java.
- Selenium: Ferramenta de teste de integração para testar interfaces de usuário.
- JIRA: Para rastrear bugs e problemas encontrados durante os testes.

Quando usar Modelos Prescritivos?

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Projetos com requisitos bem definidos.
- Quando há uma compreensão clara do domínio do problema.
- Em ambientes onde as mudanças são mínimas e caras.

Contextos Adequados:

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Projetos com Requisitos Estáveis e Bem Definidos:
- Exemplo: Um sistema de contabilidade para um setor que segue normas rígidas e estáveis.

Contextos Adequados:

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Ambientes Regulatórios ou com Alta Conformidade:
- Exemplo: Desenvolvimento de software para dispositivos médicos que precisam de aprovação de órgãos reguladores.

Contextos Adequados:

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Projetos de Grande Escala que Exigem Coordenação Rigorosa entre Equipes:
- Exemplo: Sistemas de controle de tráfego aéreo ou sistemas bancários integrados.

Contextos Adequados:

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Quando a Estrutura e a Documentação são Críticas:
- Exemplo: Software para usinas nucleares onde a falha não é uma opção e cada decisão de design precisa ser documentada e revisada.

Métodos e Ferramentas Associadas:

- Especificação Formal: Método que utiliza uma linguagem matemática para especificar, desenvolver e verificar sistemas.
- Uso: Garante que o sistema desenvolvido atenda a todos os requisitos especificados.

Métodos e Ferramentas Associadas:

- Modelagem CASE (Computer-Aided Software Engineering): Conjunto de ferramentas que fornecem suporte automatizado para muitas atividades do processo de desenvolvimento de software.
- Uso: Ajuda na criação de documentação robusta e na manutenção da consistência entre modelos e especificações.

Métodos e Ferramentas Associadas:

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar on the right side.

- Revisões Técnicas Formais: Avaliações sistemáticas do software ou do processo de desenvolvimento para descobrir erros.
- Uso: Garante que o produto em desenvolvimento atenda aos padrões e requisitos definidos.

Vantagens dos Modelos Prescritivos

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Estrutura clara e bem definida.
- Facilidade na coordenação e gestão.
- Documentação robusta.

Clareza e Previsibilidade

- Descrição: Uma vez que os modelos prescritivos seguem um conjunto específico de fases sequenciais, o progresso pode ser facilmente medido e as expectativas podem ser gerenciadas de acordo.
- Exemplo: Em um projeto de desenvolvimento de um software bancário, os requisitos de segurança são críticos. Usando o modelo em cascata, cada fase do projeto é meticulosamente planejada e documentada. Portanto, antes de se mover para a fase de design, todos os requisitos de segurança são estabelecidos, garantindo que a equipe saiba exatamente o que precisa ser feito e evitando surpresas.

Documentação Robusta

- Descrição: Modelos prescritivos geralmente envolvem uma documentação abrangente em cada fase, garantindo que todas as especificações, designs e decisões sejam registradas para referência futura.
- Exemplo: Ao desenvolver um sistema de gestão hospitalar, é essencial ter uma trilha de documentação clara para atender aos regulamentos e auditorias do setor de saúde. Ao adotar o modelo V, a equipe garante que cada fase de desenvolvimento é mapeada à sua respectiva fase de teste, com documentos detalhados descrevendo o fluxo.

Facilidade de Coordenação

- Descrição: Devido à sua natureza linear e à especificidade de cada fase, a coordenação entre equipes e stakeholders torna-se mais gerenciável.
- Método: Uso de Milestone Reviews. Ao final de cada fase, uma revisão é realizada para garantir que tudo foi cumprido conforme o planejado antes de avançar para a próxima fase. Isso garante que todos os stakeholders estejam alinhados e cientes do progresso.

Redução de Ambiguidades

- Descrição: Como cada fase é detalhada e documentada antes de passar para a próxima, há menos chances de mal-entendidos ou ambiguidades.
- Exemplo: Ao desenvolver um sistema ERP para uma grande corporação, a fase de requisitos em um modelo em cascata define claramente o escopo do projeto. Isso evita que os desenvolvedores interpretem os requisitos de maneira diferente ou adicionem funcionalidades não solicitadas.

Controle Rígido de Mudanças

- Descrição: Os modelos prescritivos muitas vezes têm mecanismos rígidos de controle de mudanças, o que pode ser benéfico em ambientes onde as mudanças são caras ou arriscadas.
- Método: Uso de Change Control Boards (CCB). Se uma mudança for proposta após a fase de requisitos ser concluída, ela passará por uma revisão rigorosa por uma CCB, garantindo que as mudanças sejam avaliadas em termos de custo, risco e impacto antes de serem aprovadas.

Desafios e Limitações

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar extending downwards from its right end.

- Pouca flexibilidade para acomodar mudanças.
- Ciclo de feedback longo.
- Pode levar a um excesso de documentação e burocracia.

Flexibilidade Limitada

- Definição: Os modelos prescritivos são rígidos e têm uma sequência pré-definida de fases.
- Exemplo: Imagine um projeto de construção de uma casa. Se você seguir um modelo prescritivo, decidirá sobre o design da casa, o material a ser usado, a cor das paredes, etc., no início. Se, no meio do caminho, você quiser mudar a cor das paredes, isso seria problemático e caro.

Ciclo de Feedback Longo

- Definição: O feedback do cliente ou usuário é obtido depois que o software é completamente desenvolvido.
- Exemplo: Voltando ao projeto da casa, seria como mostrar a casa completa ao cliente apenas depois de ter sido totalmente construída, ao invés de envolvê-lo em cada etapa.

Propensão a Documentação Excessiva

- Definição: Devido à sua natureza sequencial, pode haver uma tendência a produzir documentação detalhada em cada fase.
- Método para Aliviar: Uso de técnicas de documentação "just-in-time" ou "lean", onde a documentação é produzida apenas quando é absolutamente necessária e com o nível de detalhe necessário.

Dificuldade em Acomodar Mudanças

- Definição: As mudanças nos requisitos ou no design podem ser difíceis de acomodar depois que a fase relevante foi concluída.
- Exemplo: Se, após a fase de design em um projeto de software, surgir a necessidade de uma nova funcionalidade, incorporá-la pode exigir um retrabalho significativo.
- Método para Aliviar: Realizar revisões frequentes e checkpoints para garantir que os requisitos e o design estejam alinhados com as necessidades do negócio.

Risco de Não Cumprir Prazos e Orçamentos

- Definição: A natureza sequencial pode levar a subestimações e, conseqüentemente, a ultrapassagens de tempo e custo.
- Método para Aliviar: Uso de técnicas de estimativa como o "Wideband Delphi" ou a "Function Point Analysis" para obter estimativas mais precisas.

Comparação com Modelos Ágeis

- Contraste com abordagens ágeis para dar uma perspectiva equilibrada.
- Discussão sobre quando um modelo prescritivo pode ser mais adequado do que um ágil e vice-versa.

Principais Diferenças

- Abordagem e Flexibilidade:
 - Prescritivos: Sequenciais, planejamento detalhado no início.
 - Ágeis: Iterativos, adaptação contínua às mudanças.
- Documentação:
 - Prescritivos: Enfatizam documentação detalhada.
 - Ágeis: Priorizam o software funcional mais do que documentação extensa.
- Feedback:
 - Prescritivos: Ciclos mais longos entre etapas.
 - Ágeis: Feedback rápido através de iterações curtas.

Exemplos de Métodos

- Prescritivos:
- Cascata: Progressão linear através das fases.
- Espiral: Foco na análise e mitigação de riscos a cada iteração.

Exemplos de Métodos

- Ágeis:
- Scrum: Organiza o desenvolvimento em sprints com revisões regulares.
- Kanban: Enfatiza fluxo contínuo e limita o trabalho em progresso.
- Extreme Programming (XP): Foco na qualidade do código e práticas de programação.


Aplicabilidade

- Projetos de Infraestrutura:
- Prescritivos podem ser preferíveis quando há requisitos rígidos e mudanças são caras (ex.: sistemas críticos de segurança).
- Projetos de Desenvolvimento de Produto:
- Ágeis são benéficos quando o mercado é volátil e os requisitos mudam frequentemente (ex.: aplicativos para smartphones).

Exemplo Prático: Desenvolvimento de um App

- Cenário 1: Um app para aeronaves, onde os requisitos e normas são rígidos. Modelo em cascata pode ser mais adequado devido à necessidade de certificação e testes rigorosos.
- Cenário 2: Um app de redes sociais para uma startup. Scrum ou Kanban pode ser mais adequado para responder rapidamente às mudanças do mercado e feedback dos usuários.

Atividade Interativa - Análise de Cenário

A thick yellow horizontal bar spans the width of the slide, with a vertical yellow bar on the right side.

- Apresentação de um cenário de projeto hipotético.
- Os alunos discutem e decidem se um modelo prescritivo seria adequado e justificam suas escolhas.