**Modelos Clássicos**

**Cascata**

O modelo Cascata segue um fluxo linear e sequencial de desenvolvimento, onde cada fase deve ser concluída antes do início da próxima. As fases incluem: levantamento de requisitos, análise, projeto, implementação, testes, implantação e manutenção.

Vantagens:

* Estrutura simples e bem definida.
* Facilidade de gerenciamento devido à documentação clara.
* Adequado para projetos com requisitos bem estabelecidos e pouco sujeitos a mudanças.

Desvantagens:

* Pouca flexibilidade para mudanças durante o desenvolvimento.
* Problemas só são detectados no final do ciclo

**V-Model**

O **V-Model** é uma extensão do Modelo Cascata que enfatiza a verificação e validação em cada etapa do desenvolvimento. Visualmente, o modelo se assemelha a um "V", onde o lado esquerdo representa as fases de definição e design, e o lado direito corresponde às fases de teste e validação. Para cada fase de desenvolvimento, há uma fase de teste correspondente, garantindo que cada componente seja rigorosamente verificado antes de avançar. Essa abordagem estruturada é particularmente útil em projetos onde a qualidade e a confiabilidade são críticas, como em sistemas embarcados ou aplicações de segurança.

Vantagens:

* Melhor controle sobre testes e qualidade do software.
* Definição clara das atividades de validação e verificação.
* Adequado para sistemas críticos, como software de segurança ou embarcado.

Desvantagens:

* Assim como o Cascata, é inflexível para mudanças.
* Alto custo se erros forem encontrados tardiamente.
* Desenvolvimento pode ser lento devido à necessidade de testes rigorosos.

**Espiral**

Combina elementos dos modelos Cascata e de prototipagem, enfatizando a análise de riscos e a evolução contínua do software. O desenvolvimento ocorre em ciclos iterativos, cada um passando por quatro fases principais: planejamento, análise de riscos, desenvolvimento e avaliação. Essa abordagem permite a incorporação de feedback e a adaptação às mudanças ao longo do processo, tornando-a especialmente adequada para projetos complexos e de grande escala, onde os requisitos podem evoluir ou não ser completamente compreendidos desde o início.

Vantagens:

* Flexibilidade para mudanças.
* Identifica e gerencia riscos desde o início.
* Ideal para projetos de grande porte e com alto grau de incerteza.

Desvantagens:

* Processo complexo e de difícil gestão.
* Pode ser caro e demorado devido à análise de riscos em cada ciclo.
* Não recomendado para projetos pequenos ou de curta duração.

**Metodologias Ágeis**

**Scrum**

Metodologia ágil que organiza o trabalho em ciclos curtos e fixos chamados Sprints, geralmente com duração de duas a quatro semanas. Durante cada Sprint, a equipe trabalha em um conjunto específico de funcionalidades prioritárias, com reuniões diárias (Daily Scrums) para monitorar o progresso e identificar impedimentos. Ao final de cada Sprint, há uma revisão para apresentar o que foi desenvolvido e uma retrospectiva para discutir melhorias no processo. Essa abordagem iterativa e incremental permite uma rápida adaptação às mudanças e entrega contínua de valor ao cliente.

Vantagens:

* Flexibilidade para adaptações rápidas.
* Transparência e colaboração entre os membros da equipe.
* Entregas incrementais aumentam a previsibilidade do progresso.

Desvantagens:

* Requer alto comprometimento da equipe.
* Pode ser difícil de implementar corretamente sem experiência.
* Nem sempre adequado para projetos com requisitos muito estáveis.

**XP (Extreme Programming)**

Metodologia ágil que enfatiza práticas de desenvolvimento de software destinadas a melhorar a qualidade e a capacidade de resposta às mudanças dos requisitos dos clientes. Entre suas práticas destacam-se: feedback constante, abordagem incremental e encorajamento da comunicação entre as pessoas envolvidas. Os cinco valores fundamentais do XP são: comunicação, simplicidade, feedback, coragem e respeito. Esses valores são implementados através de práticas como o Jogo de Planejamento, pequenas releases, metáforas, design simples, testes automatizados, testes de aceitação, semana de 40 horas, propriedade coletiva, programação pareada, padronização do código, desenvolvimento orientado a testes, refatoração e integração contínua.

Vantagens:

* Alta qualidade do código devido à revisão constante.
* Rapidez na entrega e resposta às mudanças.
* Maior envolvimento do cliente.

Desvantagens:

* Exige alto nível de disciplina e colaboração.
* Pode ser difícil de aplicar em equipes grandes.
* Necessita de testes automatizados eficazes.

**Lean**

O Lean é uma filosofia que busca eliminar desperdícios e maximizar o valor entregue ao cliente por meio da melhoria contínua e de práticas eficientes. Embora tenha origem na manufatura, sua aplicação se estende a áreas como serviços e tecnologia.

Vantagens:

* Redução de desperdícios e custos.
* Processo ágil e eficiente.
* Foco na melhoria contínua.

Desvantagens:

* Difícil de implementar em empresas com cultura tradicional.
* Exige envolvimento ativo da equipe e do cliente.
* Pouca estrutura formal, o que pode levar a dificuldades na gestão.

**Kanban**

Utiliza um sistema visual para gerenciar o trabalho à medida que ele passa por um processo. As tarefas são representadas em cartões e colocadas em colunas que refletem as diferentes etapas do fluxo de trabalho, como "A Fazer", "Em Andamento" e "Concluído". Essa visualização permite que a equipe veja o status de cada tarefa em tempo real, identifique gargalos e otimize o fluxo de trabalho. A flexibilidade do Kanban facilita sua implementação em diversos contextos e equipes, permitindo ajustes contínuos conforme necessário.

Vantagens:

* Visualização clara do progresso das tarefas.
* Fácil de implementar em qualquer equipe.
* Permite mudanças contínuas sem impacto significativo.

Desvantagens:

* Pode gerar gargalos se mal gerenciado.
* Não define papéis e responsabilidades de forma estruturada.
* Menos adequado para projetos muito complexos.

**Crystal**

Adapta-se ao tamanho da equipe e criticidade do projeto. Existem diferentes variantes, como Crystal Clear (equipes pequenas) e Crystal Red (projetos críticos).

Vantagens:

* Flexível e adaptável a diferentes contextos.
* Enfatiza comunicação e colaboração.
* Valoriza pessoas e interação sobre processos rigidamente definidos.

Desvantagens:

* Falta de estrutura pode dificultar a aplicação em equipes grandes.
* Requer alto nível de maturidade da equipe.
* Pode ser menos eficaz em projetos com muitas regras e requisitos formais.

**Alternativos**

**Go Horse**

Go Horse é uma abordagem sem metodologia definida, na qual a execução é priorizada em detrimento de planejamento, análise ou estratégia inicial.

A lógica é direta: iniciar um projeto ou solucionar um problema sem muita reflexão, fazendo ajustes conforme os desafios aparecem. Isso contrasta com métodos que demandam planejamento detalhado antes da ação.