

**ANDRÉ OLIVIO MALAFAIA DA CUNHA - 824139394**

**JEREMIAS HENRY CALLATA QUISPE - 824113042**

**KAIQUE NUNES DICHEMAN - 824132510**

**MATHEUS TERCEIRO DANIEL - 824121622**

**PEDRO HENRIQUE GOMES PEREIRA - 823148032**

**SAMUEL GUTIERREZ - 824150619**

**ASAFE ALVES OCTAVIANI - 825113049**

**Modelos de Processos**

Modelos de processos são formas visuais que mostram como as ações, tarefas e trabalhos são arranjados e feitos dentro de um lugar ou sistema. Eles ajudam a ver, analisar, melhorar e controlar os passos de um trabalho. Aqui estão os ti͏po͏s mais importantes de modelos de processos:͏

**Modelos Tradicionais**

**Modelo Cascata**

A principal vantagem do modelo cascata reside em sua organização e documentação detalhada. Cada etapa possui um propósito claro, o que é ideal para projetos com requisitos bem definidos desde o início e que não sofrem alterações significativas ao longo do desenvolvimento.

Entretanto, o modelo cascata apresenta limitações, especialmente quando há necessidade de mudanças após a conclusão de uma etapa. A dificuldade de retroceder e a descoberta tardia de problemas graves no produto final são desafios potenciais.

Em suma, o modelo cascata é comparável a seguir uma receita de bolo: se todos os passos forem seguidos corretamente, o resultado será excelente. No entanto, qualquer mudança ou erro durante o processo pode comprometer o projeto. Por isso, essa abordagem é mais adequada para projetos estáveis, com requisitos claros e imutáveis.

**Modelo Espiral**

O Modelo em Espiral, desenvolvido por Barry Boehm em 1986, é uma abordagem inovadora no desenvolvimento de software que integra elementos de diferentes modelos de processo, com um foco significativo na análise de riscos e na natureza interativa do desenvolvimento. Visualmente representado por uma espiral, cada ciclo do modelo corresponde a uma fase do projeto, permitindo revisões e refinamentos contínuos.

As atividades principais em cada ciclo do modelo incluem a comunicação com o cliente para assegurar um entendimento claro dos objetivos e requisitos, o planejamento dos recursos e cronogramas, a análise e avaliação de possíveis riscos técnicos e de gestão, o desenvolvimento e validação dos componentes do produto, a construção, teste, instalação e fornecimento de suporte ao usuário, e, finalmente, a avaliação do cliente, cujo feedback orienta as próximas iterações.

Uma variação notável do Modelo em Espiral é o Modelo em Espiral WIN-WIN, que destaca a importância da negociação entre desenvolvedores e clientes, garantindo que ambos os lados alcancem seus objetivos, promovendo uma relação de benefício mútuo.

Entre as vantagens do Modelo em Espiral estão a análise contínua de riscos, a combinação dos melhores aspectos de outros modelos de desenvolvimento e a redução dos riscos do projeto, incorporando objetivos de qualidade. No entanto, o modelo pode aumentar o tempo de desenvolvimento e requer mais recursos e experiência na identificação de riscos.

Em resumo, o Modelo em Espiral oferece uma estrutura flexível e iterativa para o desenvolvimento de software, permitindo melhorias contínuas e uma gestão eficaz de riscos ao longo do ciclo de vida do projeto.

**Modelo Incremental**

O modelo incremental é uma abordagem de desenvolvimento de software que se caracteriza pela entrega progressiva e iterativa de funcionalidades. Diferente de modelos tradicionais, como o cascata, que exigem a conclusão integral de todas as etapas antes da entrega final, o modelo incremental divide o projeto em partes menores, denominadas incrementos. Cada incremento representa uma versão funcional do produto, que incorpora novas funcionalidades ao sistema de maneira gradual.

O modelo incremental é particularmente adequado para projetos de grande escala e complexidade, nos quais a divisão em partes menores facilita o gerenciamento e a execução. Também é recomendado em cenários onde a entrega rápida de funcionalidades é essencial ou quando há expectativa de mudanças nos requisitos ao longo do desenvolvimento. Em síntese, o modelo incremental oferece uma abordagem adaptável e interativa, que permite a entrega contínua de valor ao cliente, ao mesmo tempo em que reduz riscos e aumenta a capacidade de resposta às necessidades em evolução.

Essa metodologia permite que o desenvolvimento seja realizado em ciclos iterativos, nos quais cada incremento passa por fases de análise, design, implementação e testes. Dessa forma, o cliente recebe versões utilizáveis do produto ao longo do processo, podendo fornecer feedback e utilizar as funcionalidades disponíveis antes da conclusão total do projeto. Essa entrega progressiva reduz os riscos associados ao desenvolvimento, pois problemas podem ser identificados e corrigidos nas fases iniciais.

**Modelos Ágeis**

O que os modelos Ágeis têm em comum é flexibilidade, colaboração e entrega rápida, mas cada um tem seu foco. A escolha da melhor opção depende do tipo de projeto, do tamanho da equipe e das necessidades da organização. Scrum é o mais popular, mas Kanban é mais simples, XP é mais técnico e SAFe é para quem precisa de agilidade em escala

**Modelo Scrum**

No Scrum, as cerimônias são essenciais para garantir a organização e a comunicação eficaz dentro do time. Tudo começa com a Montagem do Backlog, onde a equipe se reúne para esclarecer dúvidas, fornecer detalhes e planejar colaborativamente as tarefas a serem executadas. Diariamente, durante a Revisão do Sprint Backlog, a equipe alinha atividades e melhora a comunicação, garantindo que todos estejam na mesma página.

Ao final de cada Sprint, ocorre a Review Meeting, onde o resultado do trabalho é apresentado. O produto pode ser aceito, aceito com ressalvas ou recusado, dependendo da avaliação do time e dos stakeholders. Após essa avaliação, a equipe realiza uma Reunião de Retrospectiva para discutir os pontos positivos e negativos do Sprint, fazer os ajustes necessários e se preparar para o próximo ciclo.

A execução contínua dessas cerimônias é o que torna o Scrum uma metodologia diferenciada, permitindo alcançar os melhores resultados nos projetos. Ao seguir esse ciclo de planejamento, execução, revisão e melhoria, a equipe mantém um ritmo produtivo e adaptável, sempre focada na entrega de valor e na melhoria contínua.

**Modelo Kanban**

O Método Kanban é uma abordagem de gestão de trabalho que visa aprimorar a eficiência e a produtividade das equipes. O termo "kanban" origina-se do japonês, significando "sinalização" ou "cartão", e foi inicialmente utilizado no Sistema Toyota de Produção para indicar um sistema de sinalização que controla de modo harmônico a fabricação dos produtos necessários na quantidade e tempo adequados.

O processo é caracterizado por sua flexibilidade e adaptabilidade, permitindo que as equipes visualizem o fluxo de trabalho, identifiquem gargalos e implementem melhorias contínuas. Uma de suas principais ferramentas é o quadro Kanban, que auxilia na gestão visual das tarefas, facilitando a compreensão do status de cada atividade e promovendo a transparência entre os membros da equipe.

Entre as vantagens do Kanban destacam-se a capacidade de limitar o trabalho em andamento, o que ajuda a evitar sobrecarga de tarefas; a promoção de entregas contínuas e incrementais; e a facilidade de adaptação a mudanças, uma vez que o método não exige alterações drásticas nos processos já existentes.

Para iniciar a implementação do Kanban, recomenda-se começar pelo que já existe, ou seja, mapear o fluxo de trabalho atual e, a partir daí, identificar oportunidades de melhoria. É fundamental estabelecer políticas explícitas, definir limites para o trabalho em progresso e utilizar métricas para monitorar o desempenho e orientar decisões informadas.

**Modelo Xp**

XP, ou Extreme Programming, é um modelo de processo ágil que se concentra principalmente no desenvolvimento de software. Surgiu como uma resposta às abordagens tradicionais, que tendiam a ser muito rígidas e burocráticas. A XP valoriza a simplicidade, a comunicação constante e o feedback rápido. A ideia é formar equipes com programadores, o que ajuda a melhorar a qualidade do código e compartilhar conhecimento.

Um dos pilares do XP é a entrega frequente de pequenas porções de software, o que permite aos clientes ver o progresso e fornecer feedback rápido. Isto é muito importante porque a XP reconhece que as necessidades do projeto podem mudar e o software precisa ser flexível o suficiente para se adaptar. Outra coisa legal é que o XP dá grande ênfase à qualidade do código, então práticas como testes automatizados, integração contínua e refatoração são essenciais.