**Pesquisa sobre as definições de   
modelo de processo**

Maria Eduarda Medeiro Porto - 824144948

Matheus Alves Santana - 824144952

Erick Domingues Soares - 82414486

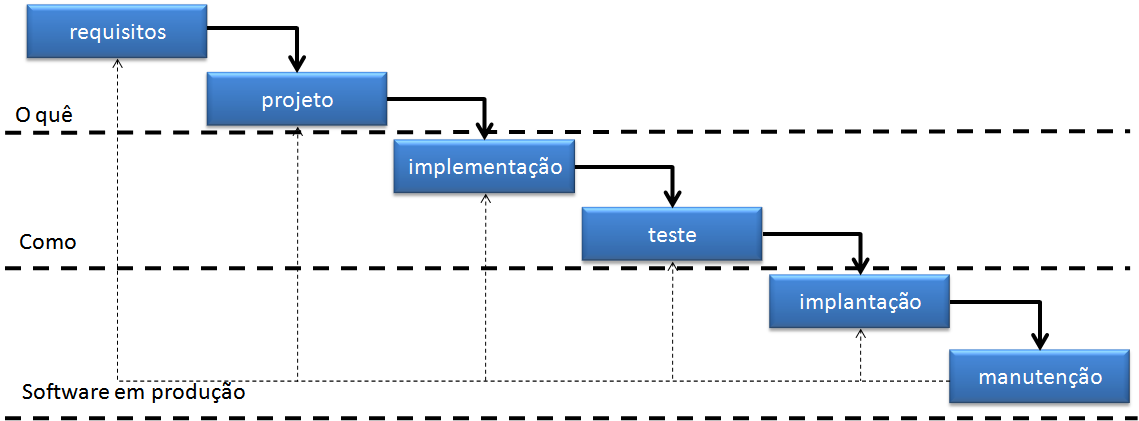
Wellington de Oliveira Sousa - 825240209

Matheus Henrique da costa e silva - 82410661

Gabriel Prieto lima- 824142064

Eduardo Baptistela - 824147595

**Modelo Cascata (Waterfall)**

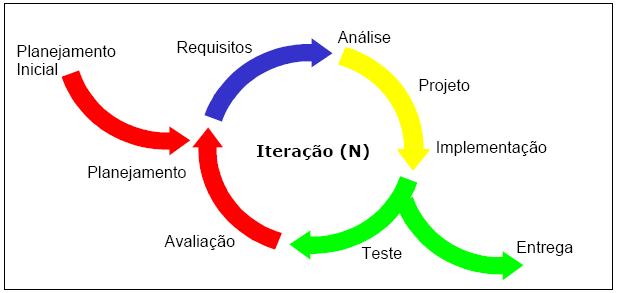


O modelo cascata é um dos mais tradicionais na engenharia de software, caracterizado por um fluxo sequencial e linear de desenvolvimento. Cada fase do projeto é concluída completamente antes que a próxima se inicie. As etapas geralmente incluem: levantamento de requisitos, análise, design, implementação, testes, entrega e manutenção. Esse modelo assume que todos os requisitos podem ser levantados logo no início, o que o torna adequado apenas para projetos onde há pouca ou nenhuma incerteza.

Apesar de sua simplicidade e organização clara, o modelo cascata é pouco flexível. Mudanças nos requisitos após a fase inicial são difíceis e custosas, pois o processo não prevê retorno às etapas anteriores. Outro problema é que o cliente só vê o produto funcionando no final do processo, o que pode resultar em entregas que não atendem às suas expectativas. É mais usado em projetos onde os requisitos são bem definidos e estáveis, como em sistemas regulatórios ou de infraestrutura.

**Exemplo prático:**  
Desenvolvimento de um sistema de controle de ponto eletrônico para uma empresa pública, com requisitos definidos por lei e prazos rígidos.

**Modelo Iterativo**

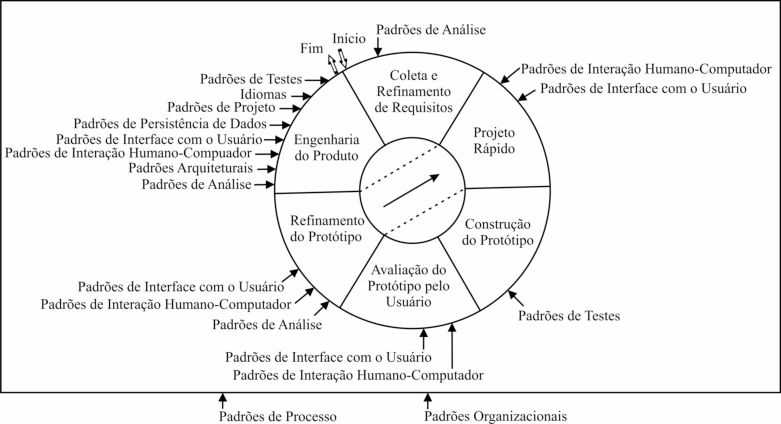


O modelo iterativo propõe o desenvolvimento do sistema em ciclos sucessivos, onde cada ciclo entrega uma parte funcional do software. Diferente do modelo cascata, aqui não é necessário ter todos os requisitos definidos desde o início. As funcionalidades vão sendo construídas, testadas e refinadas com o passar das iterações, permitindo adaptações contínuas conforme o feedback do cliente e da equipe.

Esse modelo traz diversas vantagens, como a redução de riscos, adaptação às mudanças e entrega de valor mais rápido ao cliente. No entanto, exige uma gestão mais dinâmica e maior envolvimento do cliente, o que pode ser desafiador em projetos com comunicação limitada. É bastante utilizado em ambientes ágeis e projetos com forte pressão por tempo de entrega e flexibilidade.

**Exemplo prático:**  
Criação de um sistema de gestão escolar online, onde funcionalidades como matrícula, notas e boletins foram entregues em diferentes versões ao longo do tempo.

**Modelo de Prototipação**

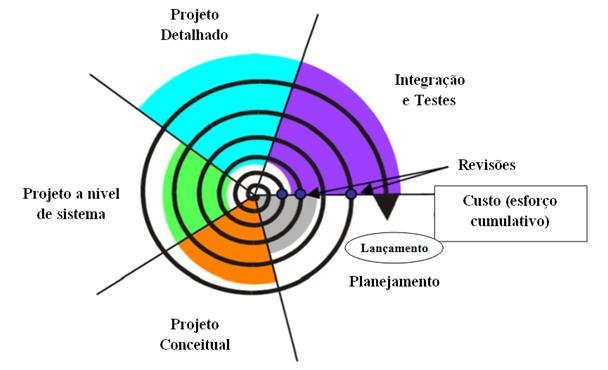


A prototipação é um modelo de desenvolvimento centrado na construção de protótipos, versões iniciais e simplificadas do sistema final, com o objetivo de validar ideias, levantar requisitos e obter feedback do cliente. Existem dois tipos principais: a prototipação descartável, que serve apenas para testes de conceito e depois é abandonada, e a prototipação evolutiva, que é refinada continuamente até se tornar o sistema final.

Esse modelo é muito útil quando os requisitos do projeto são incertos ou difíceis de definir. Permite que o cliente visualize o sistema antes de sua construção completa, contribuindo para um entendimento mais claro das necessidades. Por outro lado, a prototipação pode gerar expectativas irreais se não for bem gerenciada, especialmente se o cliente esperar que o protótipo funcione como o produto final. Também pode haver negligência na qualidade da arquitetura se o foco for apenas em funcionalidades visíveis.

**Exemplo prático:**  
Desenvolvimento de uma plataforma de e-commerce para um cliente sem experiência digital, usando protótipos navegáveis para definir a interface e funcionalidades.

**Modelo Espiral**

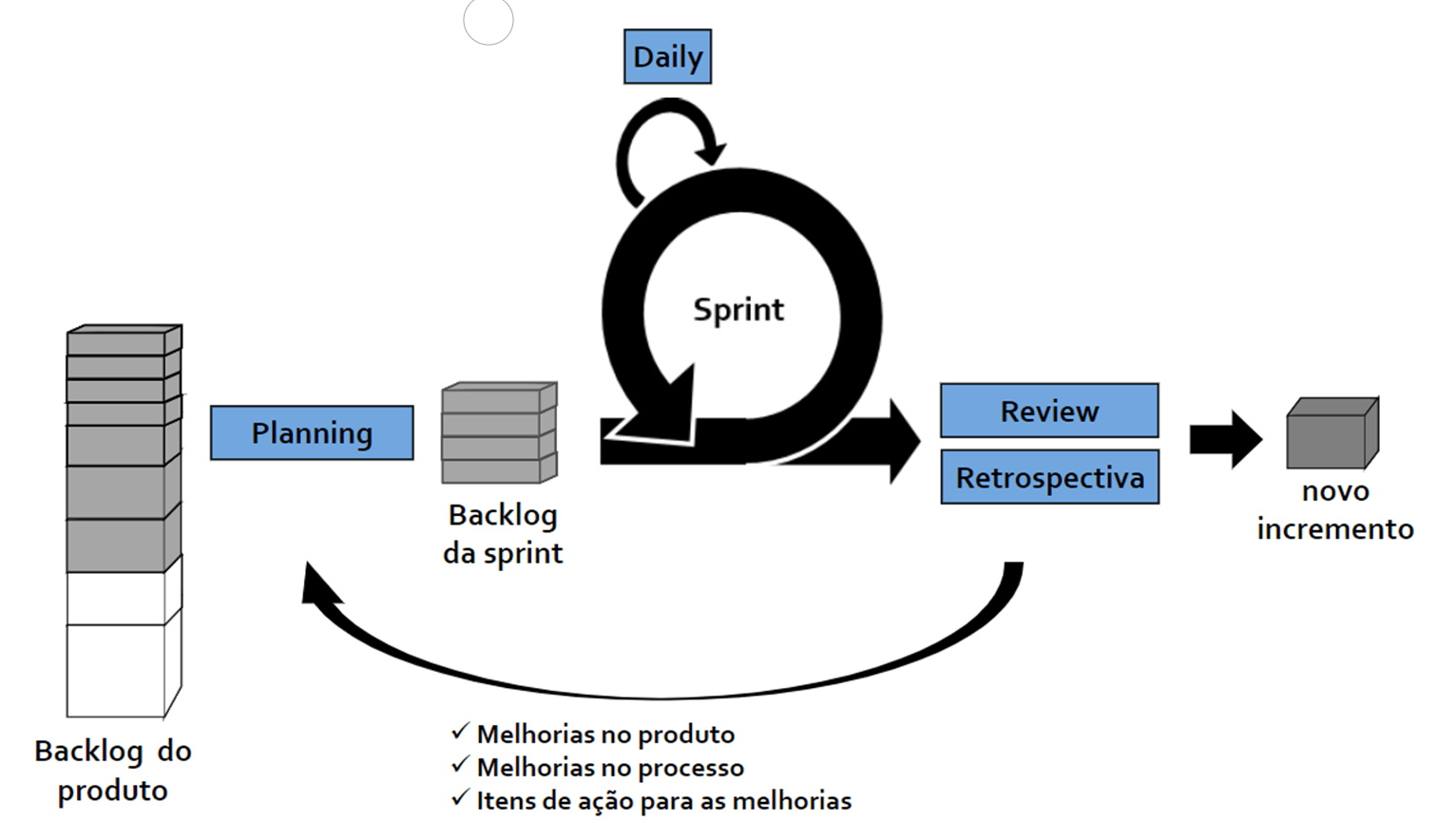


O modelo espiral é uma abordagem evolutiva que combina o desenvolvimento iterativo com uma forte ênfase na análise e gestão de riscos. O processo é representado como uma espiral com várias voltas (ciclos), onde cada volta envolve planejamento, análise de riscos, desenvolvimento e avaliação. Isso permite revisões frequentes e controle sobre as incertezas, tornando o modelo ideal para projetos complexos e de longo prazo.

A principal vantagem do modelo espiral é sua capacidade de identificar e mitigar riscos desde o início, antes que se tornem problemas críticos. Cada ciclo inclui a participação do cliente, o que melhora a qualidade dos requisitos e do produto final. Contudo, é um modelo mais caro e difícil de gerenciar, exigindo equipes experientes e com bom conhecimento de análise de riscos. É mais apropriado para grandes sistemas, como softwares militares, aeroespaciais e financeiros.

**Exemplo prático:**  
Criação de um sistema de controle de voo para uma empresa de aviação, onde falhas podem gerar riscos sérios e os testes precisam ser rigorosos e iterativos.

**Scrum (Metodologia Ágil)**



Scrum é um framework ágil que organiza o trabalho em ciclos curtos e repetitivos chamados sprints, geralmente com duração de 1 a 4 semanas. O objetivo é entregar incrementos funcionais do produto ao final de cada sprint. A equipe é composta por três papéis principais: o Product Owner, responsável pelo que será desenvolvido; o Scrum Master, que garante o bom andamento do processo; e o Time de Desenvolvimento, que executa o trabalho técnico.

A principal força do Scrum está em sua flexibilidade, colaboração constante com o cliente e capacidade de adaptação rápida às mudanças. O modelo valoriza a entrega contínua de valor e a melhoria contínua do processo. Entretanto, sua eficácia depende da maturidade da equipe e da cultura da organização. Sem comprometimento e comunicação eficaz, o Scrum pode resultar em desorganização e perda de foco. É amplamente adotado por startups e empresas de tecnologia, onde a agilidade é uma vantagem competitiva crucial.

**Exemplo prático:**  
Desenvolvimento de um aplicativo de delivery por uma startup, com entregas quinzenais e reuniões semanais com o cliente para revisar prioridades.