



# Análise e Desenvolvimento de Sistemas

Governança de TI

Aula 15

# O que é iterativo?

**Iterativo** significa que o processo de desenvolvimento é dividido em ciclos curtos e repetitivos, chamados de iterações.

Cada iteração inclui uma parte do trabalho de desenvolvimento, desde a análise de requisitos até a implementação e testes.

Após cada iteração, o sistema é testado e avaliado para determinar se atende aos requisitos estabelecidos.

Em seguida, é possível adicionar, modificar ou remover requisitos e funcionalidades com base nos resultados da iteração anterior.

O processo iterativo permite que a equipe de desenvolvimento reaja rapidamente a mudanças e ajuste o projeto ao longo do tempo.

# Diferença entre iteração e interação

A diferença entre **iteração** e **interação** está no contexto em que essas palavras são usadas, embora elas possam parecer semelhantes.

## **Iteração:**

Refere-se ao ato de repetir um processo ou uma ação várias vezes, geralmente em uma sequência ou ciclo.

É um termo comumente usado em programação, matemática, e desenvolvimento de software, onde um conjunto de instruções é executado repetidamente até que uma condição seja atingida.

Exemplo em programação: "O loop for realiza 10 iterações para processar os dados."

## **Interação:**

Envolve a troca ou comunicação entre duas ou mais entidades, como pessoas, sistemas, ou objetos. É o processo de ação mútua.


O termo é amplamente utilizado em áreas como ciências sociais, design de interface, e comunicação, referindo-se à forma como elementos se conectam e respondem uns aos outros.

Exemplo: "A interação entre o usuário e o sistema deve ser intuitiva."

Resumindo:

**Iteração** = Repetição de um processo.

**Interação** = Troca ou ação mútua entre entidades.



# O que é incremental?

---

**Incremental** significa que o processo de desenvolvimento é dividido em partes menores e independentes, chamadas de incrementos.

Cada incremento é uma versão funcional do sistema que adiciona novas funcionalidades ou melhora as funcionalidades existentes.

Cada incremento é testado e validado antes de ser incorporado ao sistema principal.

O processo incremental permite que a equipe de desenvolvimento entregue valor para o cliente em pequenas partes, ao invés de esperar até que todo o sistema esteja pronto.

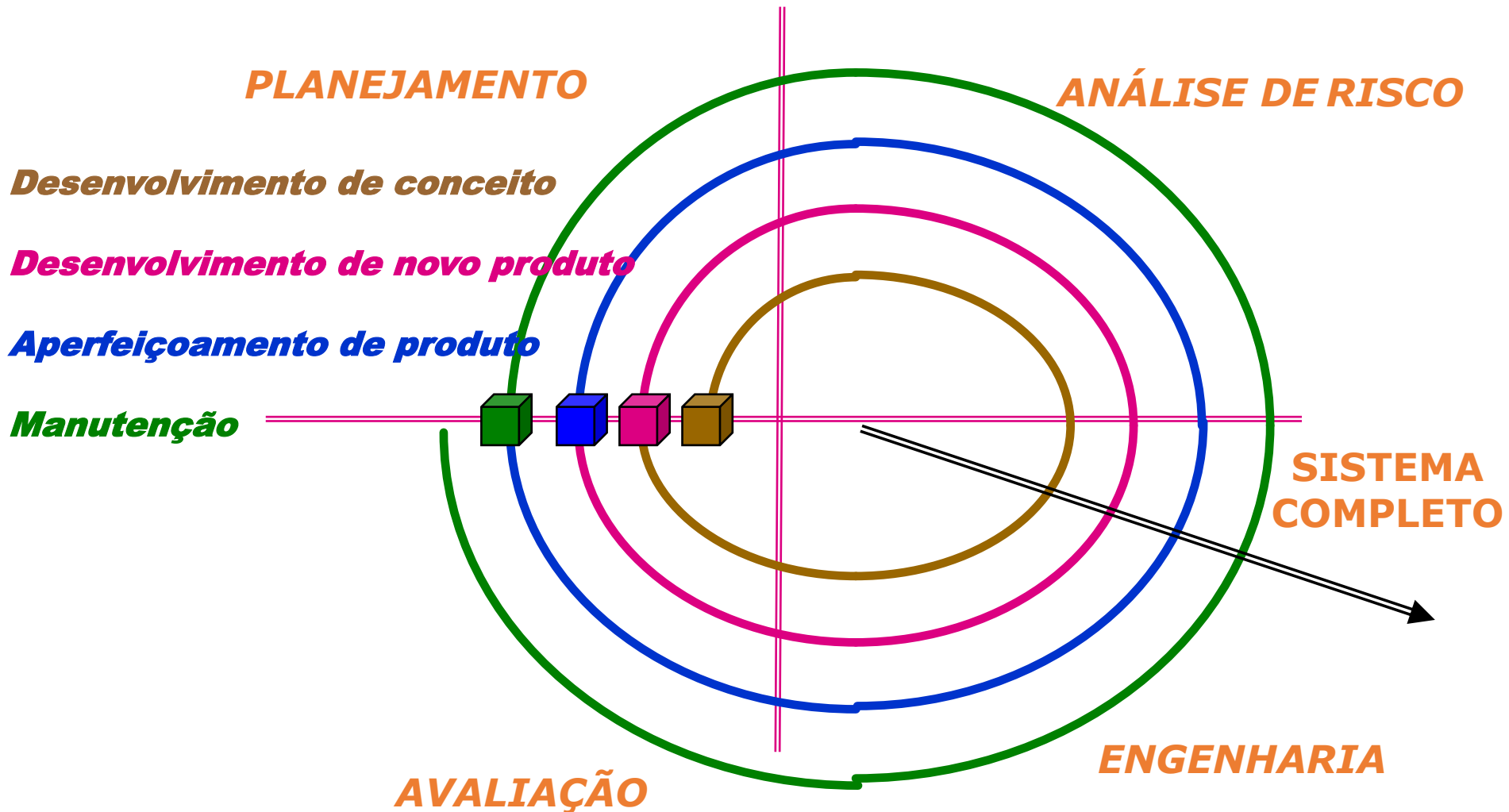


# Em Resumo

---

O processo iterativo e incremental permite que o desenvolvimento de software seja mais flexível e adaptável a mudanças, ao mesmo tempo que entrega valor de forma mais rápida e eficiente para o cliente.

# Modelo Espiral





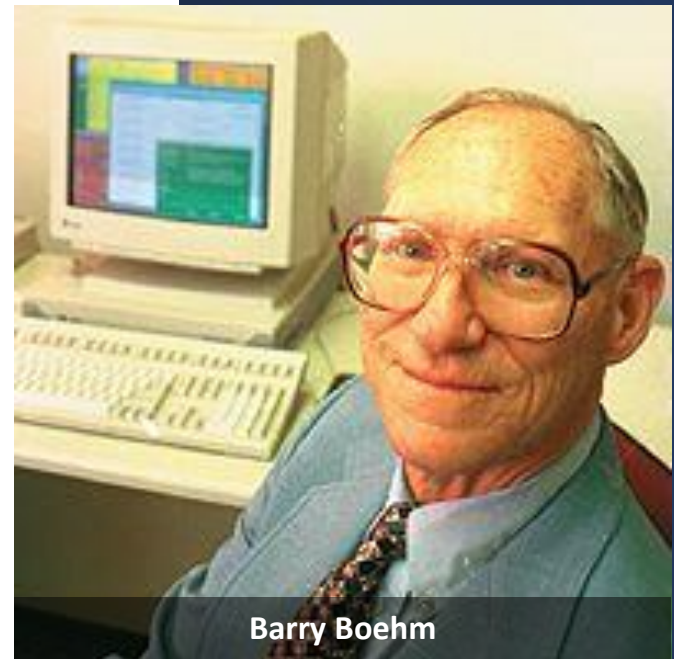
# Modelo Espiral - Conceito



O modelo espiral é um modelo de desenvolvimento de software que foi proposto por Barry Boehm em 1986. Ele foi criado para fornecer uma abordagem mais flexível para o desenvolvimento de software em comparação com modelos mais tradicionais, como o modelo cascata.



O modelo espiral é caracterizado por uma série de iterações em que o projeto é dividido em partes menores e mais gerenciáveis, cada uma com seus próprios objetivos, riscos e requisitos. Cada iteração passa por uma série de fases, incluindo planejamento, análise de riscos, desenvolvimento e avaliação.



Barry Boehm

## Modelo Espiral - Características

Uma das principais características do modelo espiral é que ele coloca uma ênfase significativa na identificação e gerenciamento de riscos.

Isso é feito por meio da avaliação contínua dos riscos associados ao projeto em cada iteração.

A equipe de desenvolvimento é capaz de identificar e gerenciar riscos com mais eficácia do que em modelos mais tradicionais, o que leva a uma maior probabilidade de sucesso do projeto.

Outra característica do modelo espiral é o uso de **marcos (milestones)** para orientar o progresso do projeto.



# Modelo Espiral – Marcos(Milestones)



Marcos são pontos de referência importantes no ciclo de vida do projeto, que geralmente são alcançados após a conclusão de uma fase importante do projeto. Os marcos ajudam a avaliar o progresso do projeto, identificar problemas e planejar a próxima fase.



No modelo espiral, os marcos são definidos para cada iteração do ciclo de desenvolvimento. Cada iteração passa por quatro fases principais: planejamento, análise de risco, implementação e avaliação. Ao final de cada iteração, um marco é atingido e um relatório é elaborado para avaliar o progresso do projeto e definir a próxima iteração.



Os marcos no modelo espiral são projetados para avaliar não apenas o progresso do projeto em termos de cronograma e orçamento, mas também em termos de qualidade, desempenho e conformidade com as especificações do cliente. Os marcos são definidos com base nos resultados de cada fase, e a equipe do projeto deve alcançar cada marco antes de avançar para a próxima fase.



O modelo espiral é amplamente utilizado em projetos de grande escala e de longo prazo, onde os requisitos e objetivos do projeto podem mudar ao longo do tempo. Os marcos ajudam a manter o projeto no caminho certo e permitem que a equipe do projeto avalie regularmente o progresso e faça ajustes para alcançar os objetivos do projeto.

# Aspectos gerenciais do Processo de Desenvolvimento de Software (PDS)

Os aspectos gerenciais do Processo de Desenvolvimento de Software (PDS) são todas as atividades relacionadas à gestão de projetos de desenvolvimento de software, incluindo o planejamento, a organização, a liderança e o controle dos recursos e atividades do projeto.

A gestão de projetos de desenvolvimento de software é essencial para garantir que o projeto seja concluído com sucesso, dentro do prazo e do orçamento definidos, e que atenda às necessidades e expectativas do cliente.

Algumas das atividades gerenciais essenciais no PDS incluem:

# Aspectos gerenciais do Processo de Desenvolvimento de Software (PDS)

- 1) **Planejamento:** Esta atividade gerencial envolve a definição dos objetivos do projeto, estabelecimento de metas, identificação dos recursos necessários e criação do cronograma. O objetivo é estabelecer um plano claro para o projeto que permita que a equipe trabalhe de forma eficiente e eficaz para atingir os objetivos.
- 2) **Organização:** A atividade de organização envolve a criação da equipe de projeto e a alocação de recursos para as diferentes tarefas do projeto. Isso inclui a identificação de habilidades necessárias e a atribuição de tarefas de acordo com a experiência e competências individuais.
- 3) **Liderança:** A liderança envolve o direcionamento da equipe para alcançar os objetivos do projeto. Isso inclui a definição de metas, fornecimento de orientação e suporte para a equipe e a garantia de que a equipe trabalhe de forma eficaz e eficiente.

# Aspectos gerenciais do Processo de Desenvolvimento de Software (PDS)

- 4) **Controle:** A atividade de controle envolve a monitoração do projeto para garantir que ele esteja avançando conforme planejado. Isso inclui a revisão do progresso do projeto, a identificação e gerenciamento de riscos, o monitoramento do uso de recursos e o ajuste do plano do projeto conforme necessário.
- 5) **Gerenciamento de recursos:** O gerenciamento de recursos envolve a alocação adequada de recursos, incluindo pessoal, orçamento e tempo, para garantir que o projeto seja concluído dentro do prazo e do orçamento previstos. Isso inclui a identificação de recursos necessários, a alocação desses recursos e o monitoramento do uso desses recursos.
- 6) **Gestão de riscos:** A gestão de riscos envolve a identificação e avaliação dos riscos potenciais associados ao projeto e a implementação de medidas para minimizar seu impacto negativo no projeto.

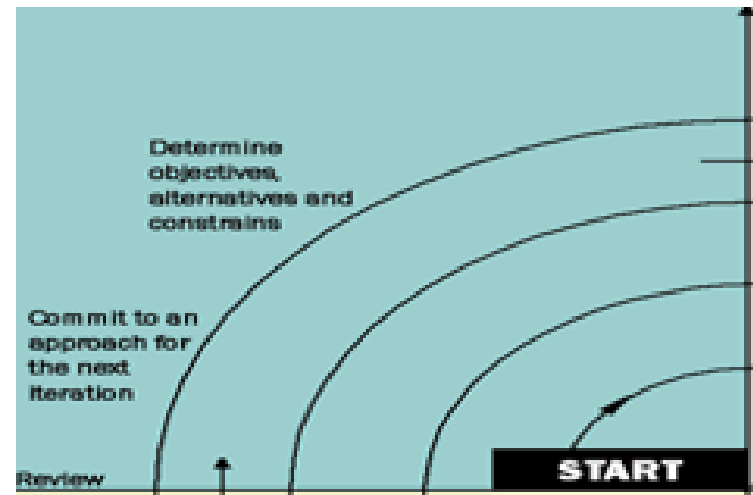
# Modelo Espiral – Tipo de Abordagem

O modelo espiral é considerado uma abordagem adaptativa, pois permite que o projeto se adapte às mudanças e desafios que surgem durante o desenvolvimento.

Ele também é útil para projetos em que os requisitos são mal definidos ou estão em constante mudança.

# Fases do Modelo Espiral - Planejamento

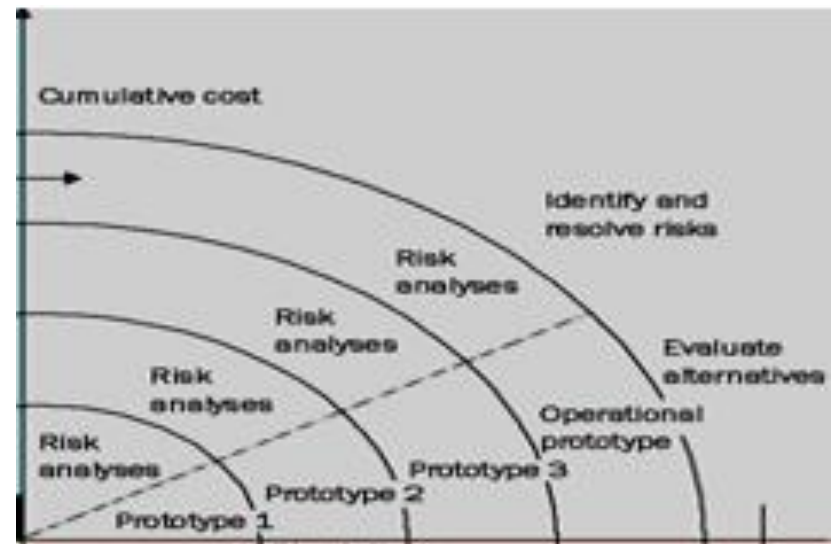
Na fase de planejamento, a equipe de desenvolvimento identifica os objetivos do projeto, as alternativas de abordagem, as restrições do projeto e as possíveis soluções. Além disso, nesta fase, é realizado o planejamento do processo de desenvolvimento, definindo-se as atividades a serem realizadas, os recursos necessários e o cronograma do projeto.





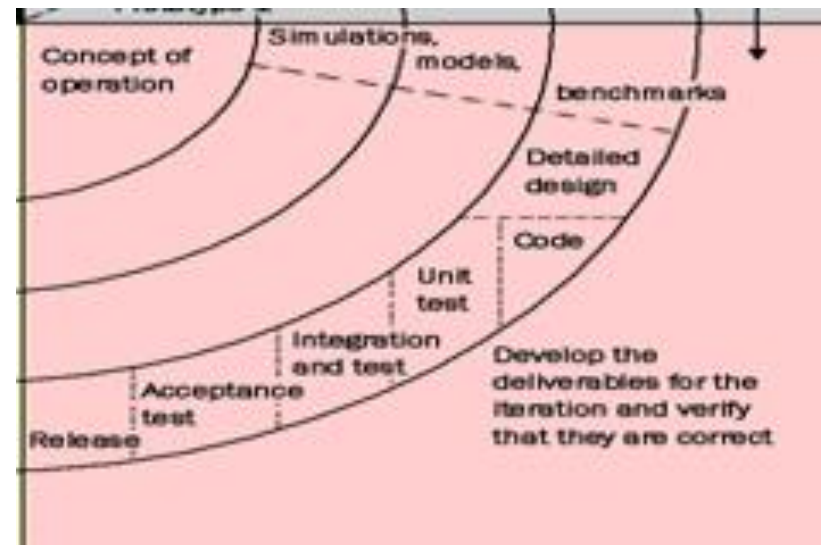
# Fases do Modelo Espiral – Análise de Riscos

A análise de riscos é uma etapa crítica do modelo espiral. Nessa fase, a equipe de desenvolvimento identifica e avalia os riscos associados ao projeto. Os riscos podem ser técnicos, financeiros, de prazo, de recursos ou de requisitos. Com base nessa análise, a equipe pode tomar decisões informadas sobre o projeto, como priorizar determinadas funcionalidades, alocar recursos ou ajustar o cronograma.



# Fases do Modelo Espiral – Engenharia

A fase de engenharia é quando a equipe de desenvolvimento realmente cria o software. Essa fase é dividida em várias subfases, incluindo design, implementação, testes e integração. Cada subfase é planejada e executada de acordo com o cronograma e as especificações definidas na fase de planejamento.

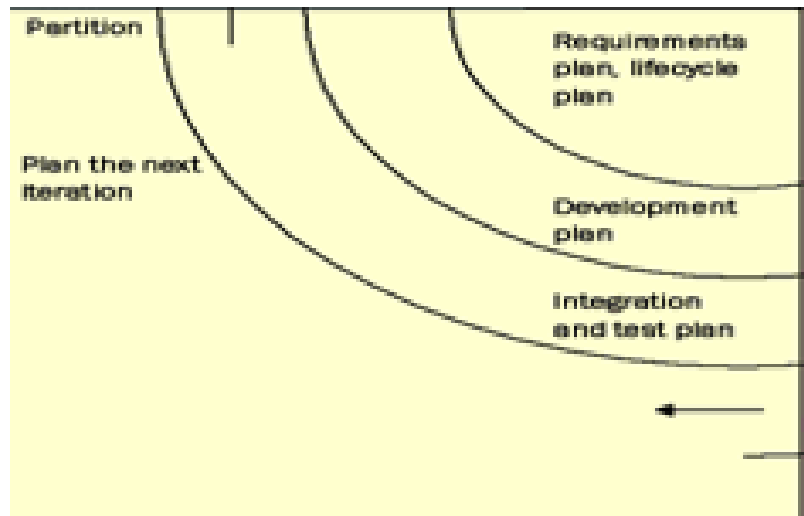


# Fases do Modelo Espiral – Avaliação do Cliente

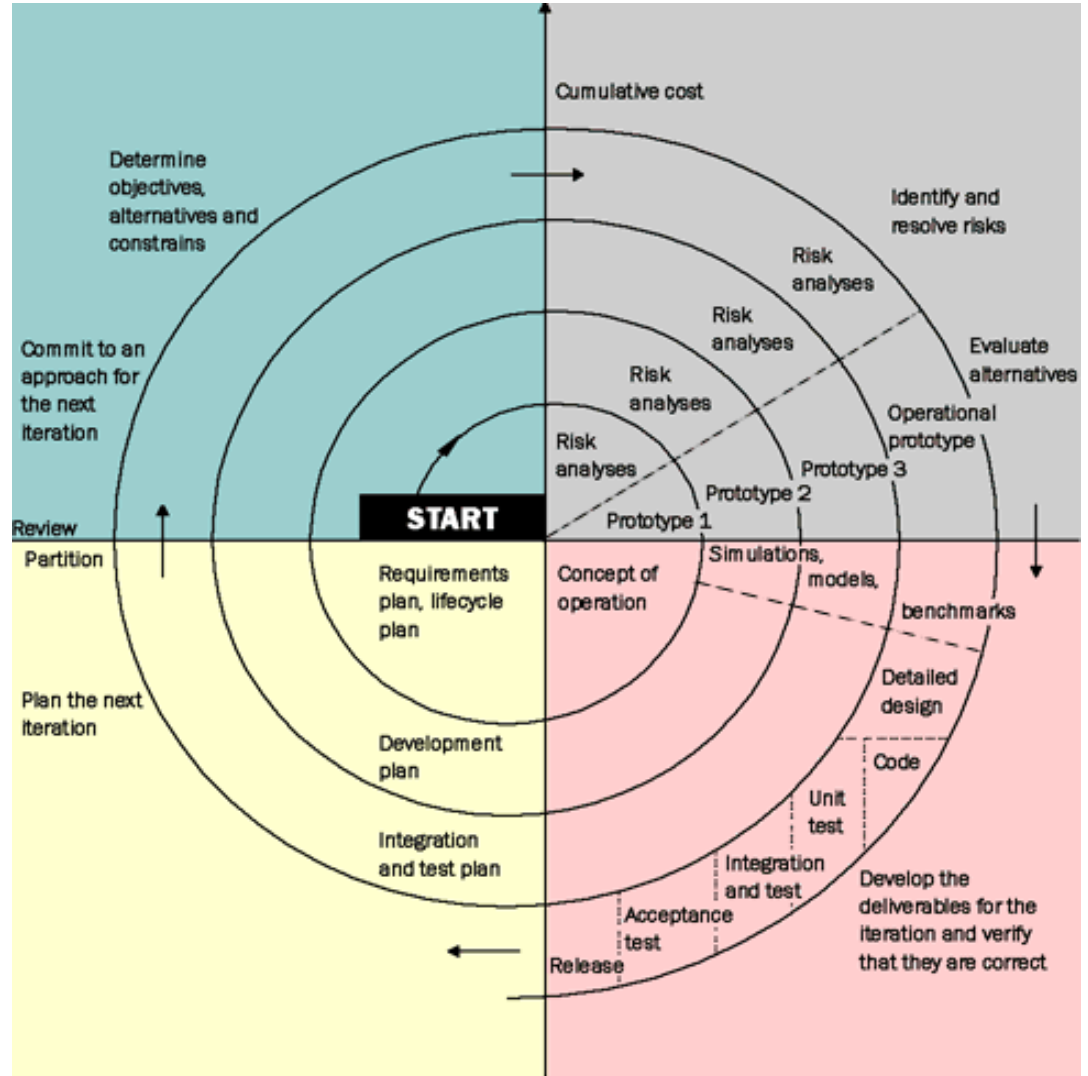
A fase final do ciclo iterativo é a avaliação do cliente.

Nessa fase, o software é entregue ao cliente e avaliado em relação aos requisitos estabelecidos na fase de planejamento.

O feedback do cliente é usado para informar o próximo ciclo do processo de desenvolvimento.



# Modelo Espiral



## Prototipagem e Mecanismos de Risco

**Prototipagem** é um processo que envolve a criação de uma versão inicial de um produto ou sistema para testar e validar suas funcionalidades, design e usabilidade.

O objetivo é identificar e corrigir problemas antes de investir recursos significativos na produção do produto final.

Existem vários tipos de prototipagem, desde protótipos de papel e caneta até protótipos funcionais que simulam o produto final.

## Prototipagem e Mecanismos de Risco

Já o **Mecanismo de Redução de Riscos** é uma estratégia utilizada por empresas para minimizar os riscos associados ao desenvolvimento de novos produtos ou serviços.

O objetivo é identificar e gerenciar os riscos ao longo do ciclo de vida do produto, desde a concepção até a comercialização.

Essa abordagem ajuda as empresas a antecipar possíveis problemas e a tomar medidas para evitá-los ou mitigá-los, o que pode reduzir os custos e aumentar a probabilidade de sucesso do projeto.



## Prototipagem e Mecanismos de Risco

A **prototipagem** e o **mecanismo de redução de riscos** estão intimamente relacionados, uma vez que a prototipagem é uma ferramenta importante para identificar e mitigar riscos em projetos de desenvolvimento de produtos.

Ao criar protótipos, as empresas podem testar diferentes ideias e cenários, e assim obter feedback dos usuários e stakeholders.

Isso pode ajudar a identificar problemas de usabilidade, funcionalidade ou viabilidade técnica antes que o produto seja lançado no mercado, permitindo que as empresas tomem medidas para corrigir ou melhorar o projeto.

## **Principal diferença entre o Modelo Incremental e o Espiral**

O **Modelo Incremental** e o **Modelo Espiral** são dois modelos de processo de software que têm como objetivo melhorar a eficiência e eficácia do processo de desenvolvimento de software.

A principal diferença entre eles é a forma como o processo de desenvolvimento é executado.

## Principal diferença entre o Modelo Incremental e o Espiral

O **Modelo Incremental** segue um processo linear, onde o software é desenvolvido em etapas incrementais e entregues em partes.

Cada incremento consiste em uma funcionalidade específica, e a próxima etapa é construída com base no incremento anterior.

O modelo incremental é iterativo e permite que o desenvolvimento seja realizado em pequenas partes, o que permite que o software seja entregue mais rapidamente e permite que as mudanças sejam feitas com mais facilidade.

# Principal diferença entre o Modelo Incremental e o Espiral

Já o **Modelo Espiral** é um modelo iterativo que utiliza a abordagem de ciclos para desenvolver o software.

Esse modelo envolve quatro fases principais: planejamento, análise de riscos, engenharia e avaliação.

Cada fase é composta por uma série de atividades e, após cada ciclo, o software é avaliado antes de prosseguir para o próximo ciclo.

O Modelo Espiral é mais flexível e adaptável a mudanças, pois permite que as equipes de desenvolvimento de software avaliem e ajustem o software em cada ciclo.

## **Principal diferença entre o Modelo Incremental e o Espiral**

Em resumo, a principal diferença entre o Modelo Incremental e o Modelo Espiral é a abordagem que cada um adota no processo de desenvolvimento de software.

O Modelo Incremental segue uma abordagem linear e constrói o software em etapas incrementais, enquanto o Modelo Espiral segue uma abordagem iterativa e utiliza ciclos para avaliar e ajustar o software em cada fase.



# Vantagens do Modelo Espiral

---

- **Foco no Gerenciamento de Riscos:** Riscos são identificados e tratados logo no início, prevenindo possíveis problemas mais adiante.
- **Iteratividade:** A evolução contínua do software em ciclos permite que o sistema seja ajustado e melhorado conforme o projeto avança.
- **Engajamento dos Stakeholders:** O envolvimento contínuo das partes interessadas em cada fase ajuda a garantir que o software atenda às suas necessidades e expectativas.



# Desvantagens do Modelo Espiral

- **Complexidade:** O método pode ser difícil de implementar em projetos menores devido à sua complexidade.
- **Custos Altos:** A análise contínua de riscos e o uso de protótipos podem aumentar o custo do desenvolvimento.
- **Requisitos Não Claros:** Se os requisitos não forem claramente definidos no início, o processo iterativo pode se tornar prolongado e pouco eficiente.

# Aplicação na Governança de TI

Na **Governança de TI**, o modelo espiral pode ser uma ferramenta valiosa para lidar com grandes projetos tecnológicos, que requerem um alto nível de flexibilidade e gerenciamento de riscos.

Sua ênfase na mitigação de riscos e nas revisões contínuas também se alinha aos princípios da governança, onde a transparência e a responsabilidade no uso de tecnologia são fundamentais.

# Comparação com Outros Modelos

**Cascata:** O modelo espiral é mais flexível que o modelo cascata, que segue uma abordagem linear. No cascata, mudanças nos requisitos são mais difíceis de gerenciar.

**Ágil:** Enquanto o método ágil foca em entregas rápidas e iterações curtas, o espiral tem uma abordagem mais estruturada e controlada, especialmente no que diz respeito ao gerenciamento de riscos.

# Exemplo: Desenvolvimento de um Sistema de Gerenciamento Escolar (SGE)

## 1. Descrição do Projeto

O projeto tem como objetivo desenvolver um **Sistema de Gerenciamento Escolar (SGE)** que permita a administração eficiente de uma escola, incluindo o cadastro de alunos, controle de notas, emissão de relatórios, gestão de professores, e acompanhamento do desempenho acadêmico. O sistema será acessado por alunos, professores e administradores via web.

## 2. Aplicação do Modelo Espiral

Cada ciclo da espiral será detalhado com base nas quatro fases: **Planejamento, Análise de Riscos, Desenvolvimento e Validação, e Avaliação e Planejamento da Próxima Fase.**

# Primeira Espiral (Ciclo 1): Planejamento Inicial e Protótipo

## Fase 1: Planejamento

### Objetivos:

Desenvolver um **protótipo básico** do sistema com funcionalidades de cadastro de alunos e professores.

Recolher feedback inicial de administradores e professores.

### Alternativas Consideradas:

Utilizar uma abordagem baseada em web com banco de dados relacional.

**Stakeholders:** Diretores da escola, equipe de TI, professores, e alguns alunos para testar o protótipo.

# Primeira Espiral (Ciclo 1): Planejamento Inicial e Protótipo

## Fase 2: Análise de Riscos

### Principais Riscos:

**Incerteza nos Requisitos:** Os requisitos podem não estar claramente definidos nesta fase inicial.

**Problemas de Usabilidade:** O protótipo pode não ser intuitivo para os usuários finais (professores e administradores).

**Riscos Técnicos:** Falta de experiência da equipe no uso de algumas tecnologias web.

### Estratégia de Mitigação:

Criar protótipos simples para validar requisitos e realizar testes de usabilidade com usuários.



# Primeira Espiral (Ciclo 1): Planejamento Inicial e Protótipo

## Fase 3: Desenvolvimento e Validação

Desenvolver um protótipo com funcionalidades básicas de:

- ✓ Cadastro de alunos (nome, idade, série, turma).
- ✓ Cadastro de professores (nome, matérias lecionadas).

Utilizar um banco de dados simples para armazenar essas informações.

Validar com alguns professores e administradores da escola.

# Primeira Espiral (Ciclo 1): Planejamento Inicial e Protótipo

## Fase 4: Avaliação e Planejamento da Próxima Fase

**Feedback:** Os usuários gostaram da interface simples, mas sugeriram melhorias na navegação e maior detalhamento nas informações de alunos e professores.

**Decisão:** Continuar o projeto, expandindo o escopo para incluir o gerenciamento de notas e relatórios no próximo ciclo.

# Segunda Espiral (Ciclo 2): Expansão do Sistema com Módulo de Notas

## Fase 1: Planejamento

### Objetivos:

- ✓ Adicionar a funcionalidade de **gerenciamento de notas**.
- ✓ Permitir que os professores registrem e atualizem notas dos alunos.
- ✓ Gerar relatórios de desempenho.

**Stakeholders:** Administradores, professores, alunos (para visualização de notas).

# Segunda Espiral (Ciclo 2): Expansão do Sistema com Módulo de Notas

## Fase 2: Análise de Riscos

### Principais Riscos:

**Erro de Armazenamento de Dados:** Falhas na gravação das notas podem comprometer a integridade dos dados.

**Segurança:** O sistema agora manipula dados sensíveis, então problemas de segurança devem ser considerados.

### Estratégia de Mitigação:

Implementar backups automáticos e controles de acesso para garantir que apenas professores possam alterar notas. Introduzir criptografia para proteger os dados sensíveis.

# Segunda Espiral (Ciclo 2): Expansão do Sistema com Módulo de Notas

## Fase 3: Desenvolvimento e Validação

Implementar o módulo de notas, permitindo:

- ✓ Cadastro de notas por matéria.
- ✓ Cálculo automático de médias e exibição de relatórios de desempenho.

Validar o funcionamento com um grupo piloto de professores e administradores.

# Segunda Espiral (Ciclo 2): Expansão do Sistema com Módulo de Notas

## Fase 4: Avaliação e Planejamento da Próxima Fase

**Feedback:** Os professores acharam útil o gerenciamento de notas, mas solicitaram a inclusão de **gráficos de desempenho** e maior automação no cálculo de médias.

**Decisão:** Avançar com a inclusão de novas funcionalidades, como relatórios gráficos e gestão de presença.

# Terceira Espiral (Ciclo 3): Relatórios Gráficos e Gestão de Presença

## Fase 1: Planejamento

### Objetivos:

- ✓ Incluir relatórios gráficos de desempenho dos alunos, comparando médias por turma e por disciplina.
- ✓ Adicionar um módulo de **gestão de presença**, permitindo que os professores registrem a presença dos alunos em cada aula.

**Stakeholders:** Alunos, professores, administradores.

# Terceira Espiral (Ciclo 3): Relatórios Gráficos e Gestão de Presença

## Fase 2: Análise de Riscos

### Principais Riscos:

**Sobrecarga de Dados:** O aumento no número de funcionalidades pode sobrecarregar o sistema.

**Performance:** Gerar relatórios gráficos pode afetar o tempo de resposta do sistema.

### Estratégia de Mitigação:

Otimizar consultas no banco de dados para garantir uma boa performance ao gerar relatórios.

Testar escalabilidade do sistema para garantir que o número de alunos e professores possa crescer sem comprometer a performance.



# Terceira Espiral (Ciclo 3): Relatórios Gráficos e Gestão de Presença

## Fase 3: Desenvolvimento e Validação

Implementar gráficos de desempenho baseados em notas.

Incluir a funcionalidade de gestão de presença, com relatórios de frequência.

Validar com mais usuários finais (professores e administradores).

# Terceira Espiral (Ciclo 3): Relatórios Gráficos e Gestão de Presença

## Fase 4: Avaliação e Planejamento da Próxima Fase

**Feedback:** O sistema está atendendo bem, mas foi sugerido que os relatórios de presença sejam integrados ao módulo de notas para facilitar o acompanhamento completo do aluno.

**Decisão:** Avançar para a fase de integração e otimização.

# Quarta Espiral (Ciclo 4): Integração e Otimização do Sistema

## Fase 1: Planejamento

### Objetivos:

- ✓ **Integrar os módulos de notas e presença**, de modo que os professores possam visualizar e relacionar facilmente a frequência dos alunos com o desempenho acadêmico.
- ✓ **Otimizar a performance** do sistema, garantindo que ele suporte um número crescente de usuários e dados, sem comprometer a velocidade de resposta.
- ✓ Adicionar permissões mais detalhadas, permitindo níveis de acesso diferenciados para administradores, professores e alunos.

### Stakeholders:

Professores, administradores, equipe de TI e alunos (principalmente para a visualização de dados integrados e relatórios).

# Quarta Espiral (Ciclo 4): Integração e Otimização do Sistema

## Fase 2: Análise de Riscos

### Principais Riscos:

- ✓ **Complexidade na Integração:** A integração de diferentes módulos (notas e presença) pode causar inconsistências nos dados ou exigir mudanças profundas na estrutura do banco de dados.
- ✓ **Problemas de Escalabilidade:** Com o aumento do número de usuários e dados, o sistema pode sofrer lentidão ou falhas.
- ✓ **Segurança e Privacidade:** Como o sistema agora lida com dados mais integrados e sensíveis, há um risco maior de vazamento de informações.

### Estratégias de Mitigação:

Realizar **testes extensivos de integração** para garantir que os dados de notas e presença sejam combinados corretamente.

Otimizar as consultas e operações no banco de dados para garantir a **escalabilidade**.

Implementar **criptografia avançada** e monitoramento de acessos para proteger os dados sensíveis.

# Quarta Espiral (Ciclo 4): Integração e Otimização do Sistema

## Fase 3: Desenvolvimento e Validação

### Integração dos Módulos:

- ✓ Desenvolver um painel integrado onde professores podem visualizar gráficos e relatórios que mostram a relação entre a frequência dos alunos e suas notas.
- ✓ Implementar filtros para que os professores possam visualizar dados por turma, matéria ou aluno específico.

### Otimização do Sistema:

Revisar o código e o banco de dados para melhorar a performance. Implementar caching para consultas frequentes e otimizar o carregamento de relatórios.

### Níveis de Permissão:

Adicionar diferentes níveis de acesso no sistema:

- ✓ **Administradores:** Têm acesso total a todos os dados e funcionalidades, incluindo relatórios gerenciais.
- ✓ **Professores:** Podem visualizar e modificar dados de suas turmas e gerar relatórios.
- ✓ **Alunos:** Têm acesso restrito, podendo visualizar apenas suas próprias notas e frequência.

### Validação:

- ✓ Realizar uma bateria de testes com diferentes tipos de usuários (administradores, professores e alunos) para garantir que todos os níveis de permissão estão funcionando corretamente.
- ✓ Testar a performance do sistema com uma carga simulada de muitos usuários simultâneos, para verificar a escalabilidade.

# Quarta Espiral (Ciclo 4): Integração e Otimização do Sistema

## Fase 4: Avaliação e Planejamento da Próxima Fase

- **Feedback:**

- O painel integrado de notas e frequência foi bem recebido, facilitando o acompanhamento dos alunos pelos professores.
- Os administradores elogiaram a otimização, relatando que o sistema está funcionando de maneira mais ágil e sem travamentos.
- Alguns alunos pediram que o sistema enviasse notificações automáticas quando novas notas fossem lançadas ou quando estivessem próximos de perder o limite de faltas permitido.

- **Decisão:**

- Considerar a adição de um **sistema de notificações automáticas** para alunos e professores, como uma próxima etapa de desenvolvimento.
- Continuar monitorando a performance e escalabilidade do sistema conforme o número de usuários cresce.
- Planejar futuras melhorias com base no feedback contínuo dos usuários.