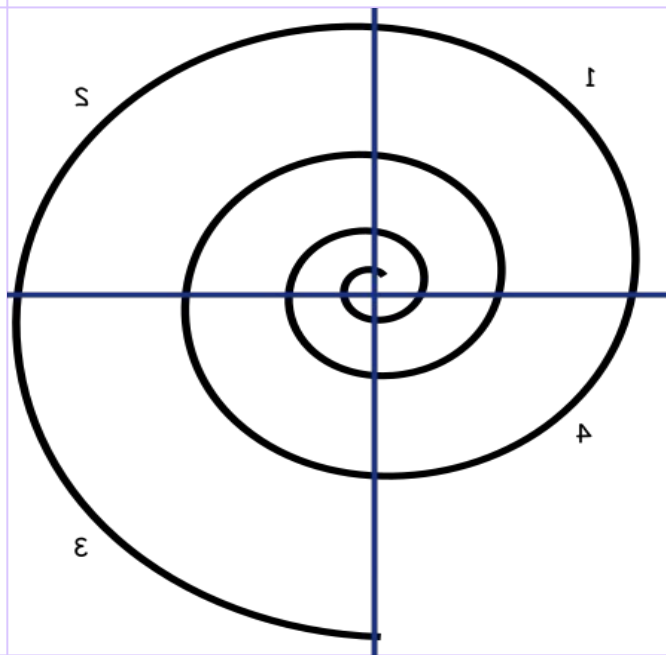




Modelo de desenvolvimento Espiral

Modelagem de Sistemas
Professor: Bruno B. Fernandes





#1

Processo de Desenvolvimento de Sistemas





Processo de Desenvolvimento de Sistemas

Para se criar um novo software é necessário um processo de desenvolvimento, que pode ser um conjunto de atividades organizadas, usadas para definir, desenvolver, testar e manter um software. O desenvolvimento de um software precisa de um projeto de manutenção, para evitar problemas e otimizar possíveis pontos do processo da atividade.

Como funciona o Desenvolvimento de Software?

Dentro do desenvolvimento de software existem diversos processos diferentes. O Levantamento de requisitos como a etapa mais importante do processo, para que o investimento com o projeto tenha o retorno esperado.



Processo de Desenvolvimento de Sistemas

Análise de requisitos:

A etapa de análise de requisitos é onde os desenvolvedores fazem o estudo aprofundado daqueles dados de requisitos coletados na etapa anterior.

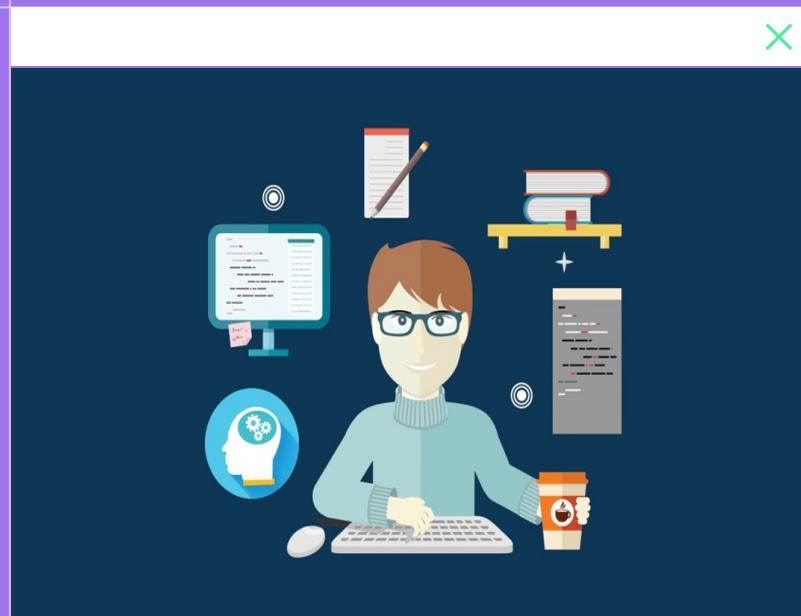
Projeto:

É o momento de se iniciar o projeto de forma mais concreta, isto é, definir alguns aspectos importantes.



#2

Etapas genéricas
de um processo de
desenvolvimento





Etapas do processo de desenvolvimento

1. Conhecer as necessidades do cliente

- a. Através de pesquisas, reuniões ou formulários

2. Definir os requisitos:

- a. Elabora-se uma lista de prioridades após uma profunda análise de requisitos. Esta lista mostrará o que o software precisa ter, o que pode limitar cada função e quanto tempo aquilo poderá demorar, em teoria.



Etapas do processo de desenvolvimento

- **Avaliar a viabilidade do projeto:**
 - **Pré-existência de soluções** que façam o mesmo e por um preço menor. O que, no final das contas, pode ser mais vantajoso para o cliente;
 - **Pré-existência de frameworks** e códigos que possam ser reaproveitados de alguma forma;



Etapas genéricas do processo de desenvolvimento

- **Número de pessoas e capacidades necessárias para a formação de uma equipe;**
- **Quais tecnologias serão usadas, como o banco de dados, serviço de cloud, linguagens de programação, etc.**
- **Documentar todos os procedimentos:** Nessa etapa, são criadas as documentações que listam o que será desenvolvido e como o processo terá que acontecer.



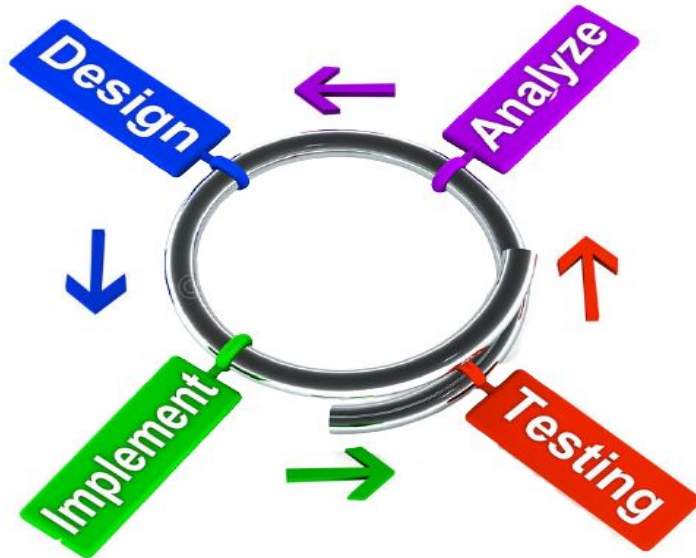
Etapas genéricas do processo de desenvolvimento

- **Escolher a metodologia de desenvolvimento:** Uma decisão muito importante pois ela direciona a forma como o projeto será desenvolvido, as pessoas que farão parte, quando e quais entregas serão feitas, além de outros detalhes importantes.
- **Testar as funcionalidades criadas:** Os testes devem ser muito bem documentados e realizados, buscando emular o uso real da aplicação.



#3

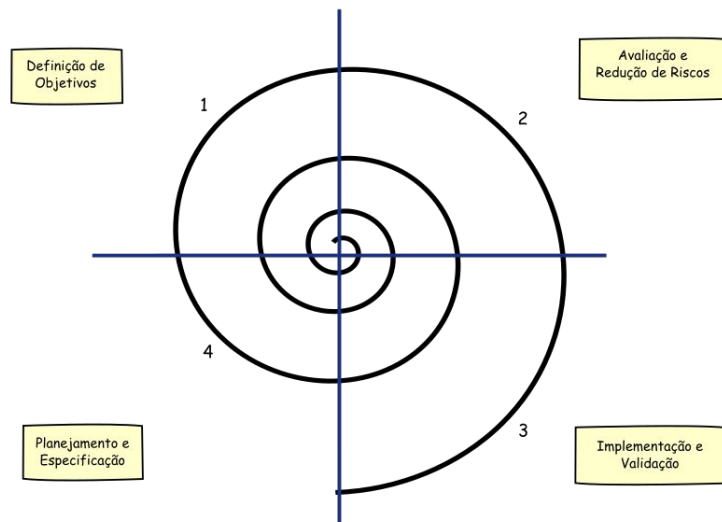
Modelo Espiral





O que é?

Criado em 1988, é uma melhora do Modelo Incremental. Nele, cada volta do espiral passa pelas fases do processo de software. Essas voltas devem ser feitas até que o modelo esteja completo. Idealizado para projetos que precisam passar por várias evoluções ao desenvolver certo projeto.





Etapas:

Constituído por 4 quadrantes principais:

1. Definição de Objetivos:

Os objetivos específicos para esta etapa são identificados e alternativas para realizar os objetivos e restrições são encontradas.





Etapas:

2. Avaliação e redução de riscos:

Para cada risco que for identificado na etapa anterior, será realizada uma descrição dos possíveis riscos, este serve para criar estratégias para evitá-los ou amenizá-los.





Etapas:

3 Implementação e validação:

Com as estratégias definidas, é escolhido um modelo de desenvolvimento. Pode-se utilizar modelos diferentes em cada volta de implementação, conforme a necessidade.

4 Planejamento e Especificação:

O projeto é revisto e tomada uma decisão de continuidade, se é decidido continuar, são projetados planos para a próxima fase/protótipo do projeto.



Pontos Fortes

- Inclui a **interação** entre os **desenvolvedores e o cliente**;
- Apresenta uma **abordagem sistemática**;
- Melhora o **tempo de implementação** do sistema;
- **Mais versátil** para lidar com **mudanças**, que podem acontecer, durante **qualquer fase do projeto**;

- Reflete as **práticas reais** da engenharia atual;
- **Fácil** de decidir o **quanto e quando testar**;
- À medida que os **custos aumentam**, os **riscos diminuem**;
- As primeiras iterações são as mais baratas (**tempo e recursos**), desde que o protótipo seja feito em pequenos pedaços.



Pontos Fortes

- **Estimativas** tornam-se mais **realísticas** com o progresso do trabalho, porque **problemas importantes** são **descobertos mais cedo**;
- Engenheiros de software podem começar o **trabalho** no sistema **mais cedo**;
- Suporta **diversos mecanismos** de redução de **risco**;
- Os **desenvolvedores e programadores** descrevem as características com **alta prioridade** em primeiro lugar e, então, desenvolvem um protótipo **baseado sobre estes**.



Pontos Fracos

- Muita ênfase a parte **funcional**;
- É necessária uma **habilidade** mais **elevada**, para a avaliação de **incertezas** ou **riscos associados** ao projeto e sua **redução**;
- Tem um **melhor funcionamento** em softwares que necessitam de requisitos de **maiores complexibilidades**, onde os custos envolvidos são **muito elevados**;
- É **bem aplicado** somente a sistemas de **larga escala**;
- Elaboração de um **script** (Protocolo), que deve ser **seguido estritamente** para que se tenha um **bom funcionamento**, sendo **muitas** das vezes **difícil** segui-lo;
- Esse modelo possui **movimentação** entre **fases complexas** o que torna complexo o **gerenciamento do projeto**;



Pontos Fracos

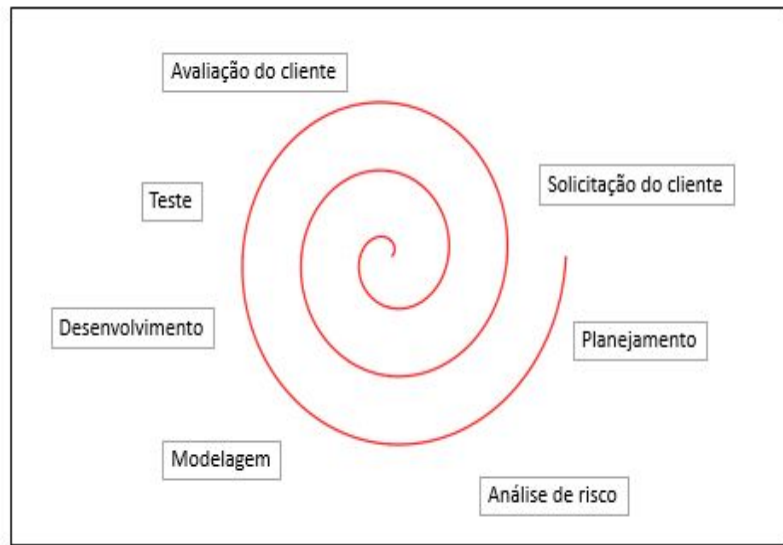
- O modelo **não é usado** na mesma extensão que o **linear** e o de **prototipação**, e, por isso, não foi **"testado"** o **suficiente**;
- Ele **não fornece** indicações suficientes sobre **quantidade** de trabalho **esperada** em cada **ciclo**;

- O **tempo** de **desenvolvimento** (prazo) se torna **imprevisível**;
- Avaliar os **riscos** no projeto pode acabar **aumentando o custo**, e o custo ser **maior** que o **orçamento** inicial.
- Se um **risco importante** não for **descoberto**, fatalmente **ocorrerão problemas**.



#4

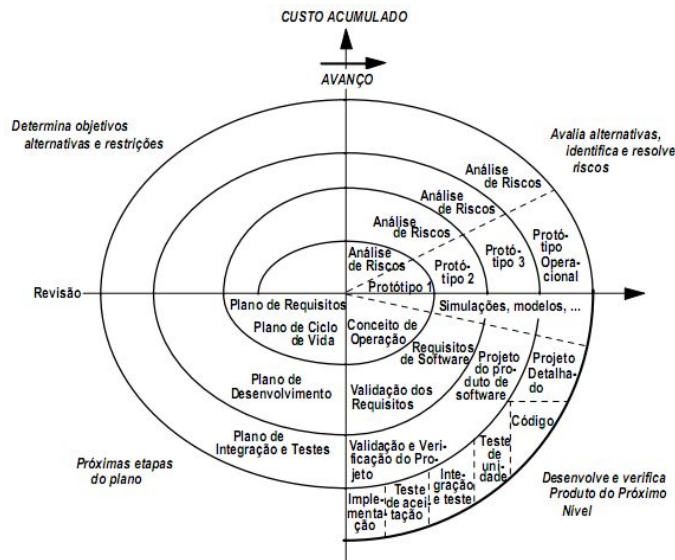
Exemplo em um sistema





Exemplo em um sistema

- O modelo espiral é usado com mais frequência em projetos mais extensos. Para pequenos projetos, é mais viável a utilização dos sistemas de desenvolvimento de software ágil.





Exemplo em um sistema

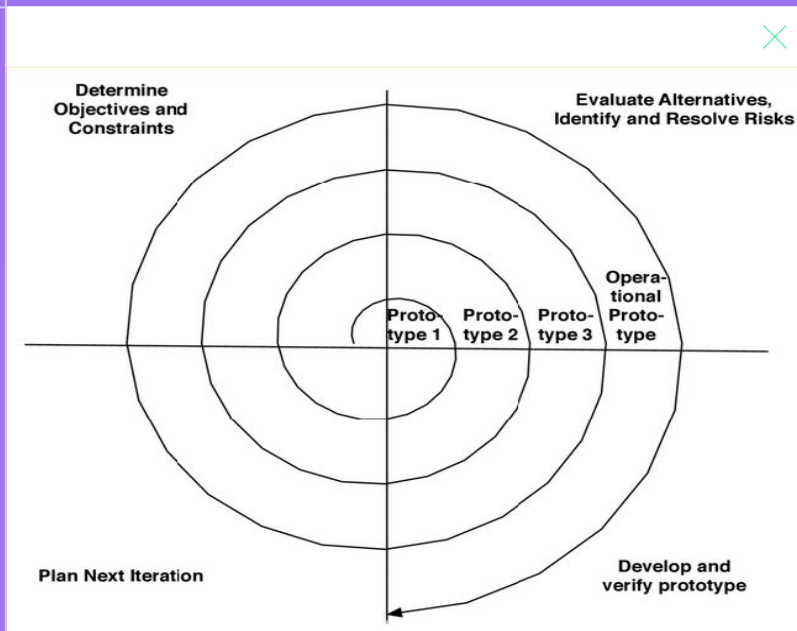
- Um exemplo no mínimo diferente é o Sistemas de combate do futuro.
- Um programa feito pelo Exército dos Estados Unidos que têm adotado o modelo em espiral.
- *“Exército, os Sistemas de Combate do Futuro (FCS, na sigla em inglês) pretendem proporcionar uma capacidade sem precedentes para detectar e derrotar o inimigo, tanto nos conflitos atuais quanto nos vindouros”*





#5

Considerações Finais





Considerações Finais

Mesmo que muito eficaz, o modelo de desenvolvimento em espiral por ser muito maçante e repetitivo, acaba por ser pouco utilizado em projetos de pequeno porte, e também por vezes, é visto como um modelo ineficaz.

Porém quando aplicado da forma correta em projetos de larga escala como equipe grande, o modelo em questão acaba por ser muito eficiente e ao contrário de projetos pequenos, reduz consideravelmente os gastos.



Trabalho feito por:

*Cezar Peterson,
Emanuelli de Oliveira,
Emili da Costa,
Jhonatan Pereira,
Kauan Fernandes,
Kayla Zaja,*

*Larissa Pluijlaar,
Luana Furini,
Pedro Sarmento,
Rafael Rakochinski,
Sofia Benedet Brandl*

<https://forms.gle/HV9rpktNLuu04czK7>