

Modelo de Ciclo de Vida



É O CONJUNTO DE ETAPAS OU FASES QUE EXISTEM PARA O DESENVOLVIMENTO DE SISTEMAS APLICATIVOS.

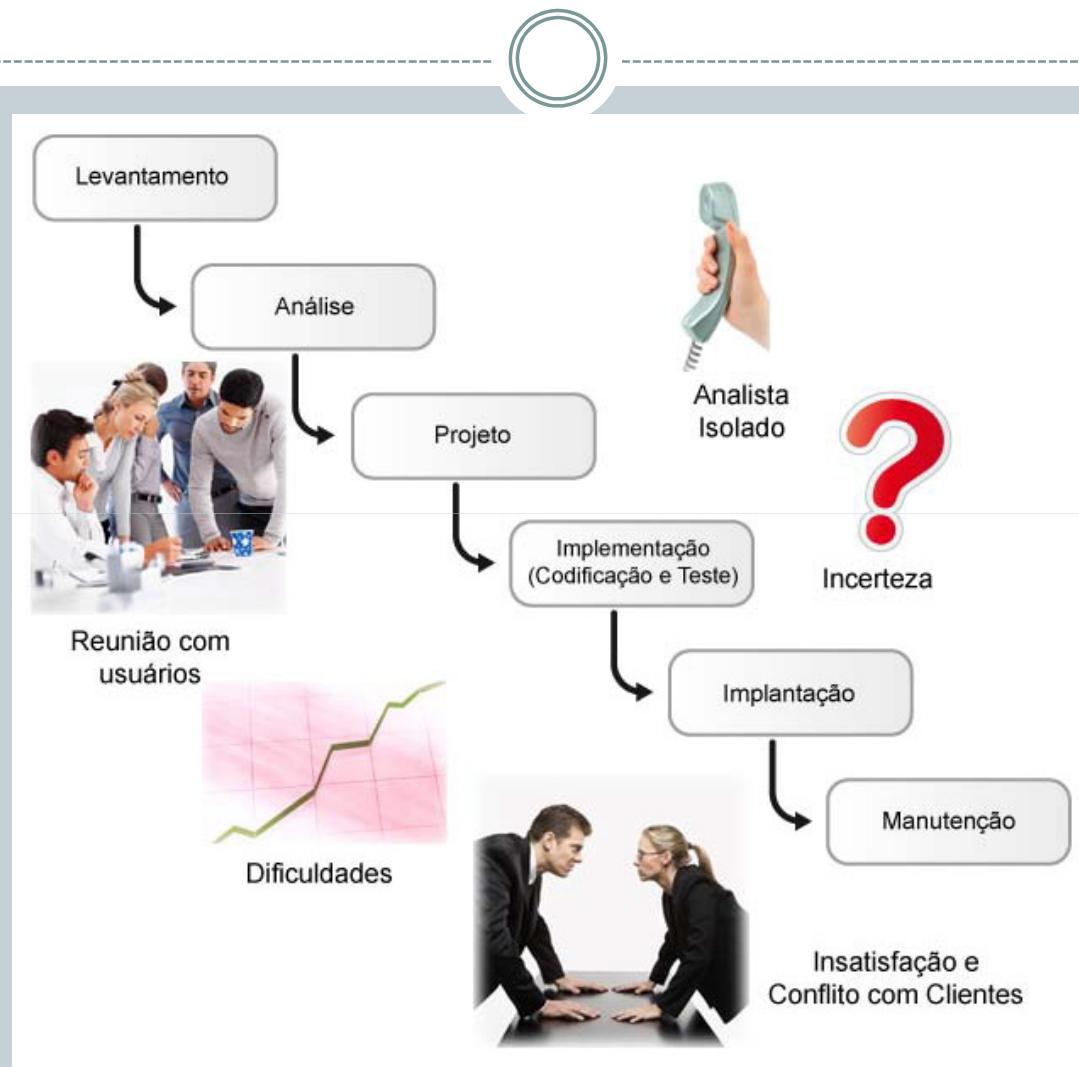
FORAM SENDO INFLUENCIADOS PELO SURGIMENTO DE NOVAS LINGUAGENS DE PROGRAMAÇÃO, POR NOVAS TÉCNICAS DE MODELAGEM DE DADOS, PELAS NECESSIDADES QUE DIA A DIA SÃO MODIFICADAS E EM FUNÇÃO DE NOVAS TECNOLOGIAS.

Ciclo de vida em cascata



- foi a primeira forma de desenvolvimento de sistemas, surgiu no final dos anos 1960 e início dos anos 1970
- não é criado algum tipo de modelo, não são utilizadas técnicas de estruturação e praticamente não existe oportunidade para o usuário realizar alguma alteração em pontos específicos.
- Uma vez fechados os requisitos, eles são quase imutáveis.
- As atividades são realizadas em sequência e não existe retorno entre as atividades.
- Toda a documentação, quando feita, é produzida após o término do projeto.
- não existe o conceito de segmentar o sistema em módulos e, a partir dessa segmentação, tratar cada módulo como um problema menor a ser resolvido.
- caracterizam pela alta incidência de manutenção, pois estão sujeitos a poucas alterações durante o desenvolvimento.

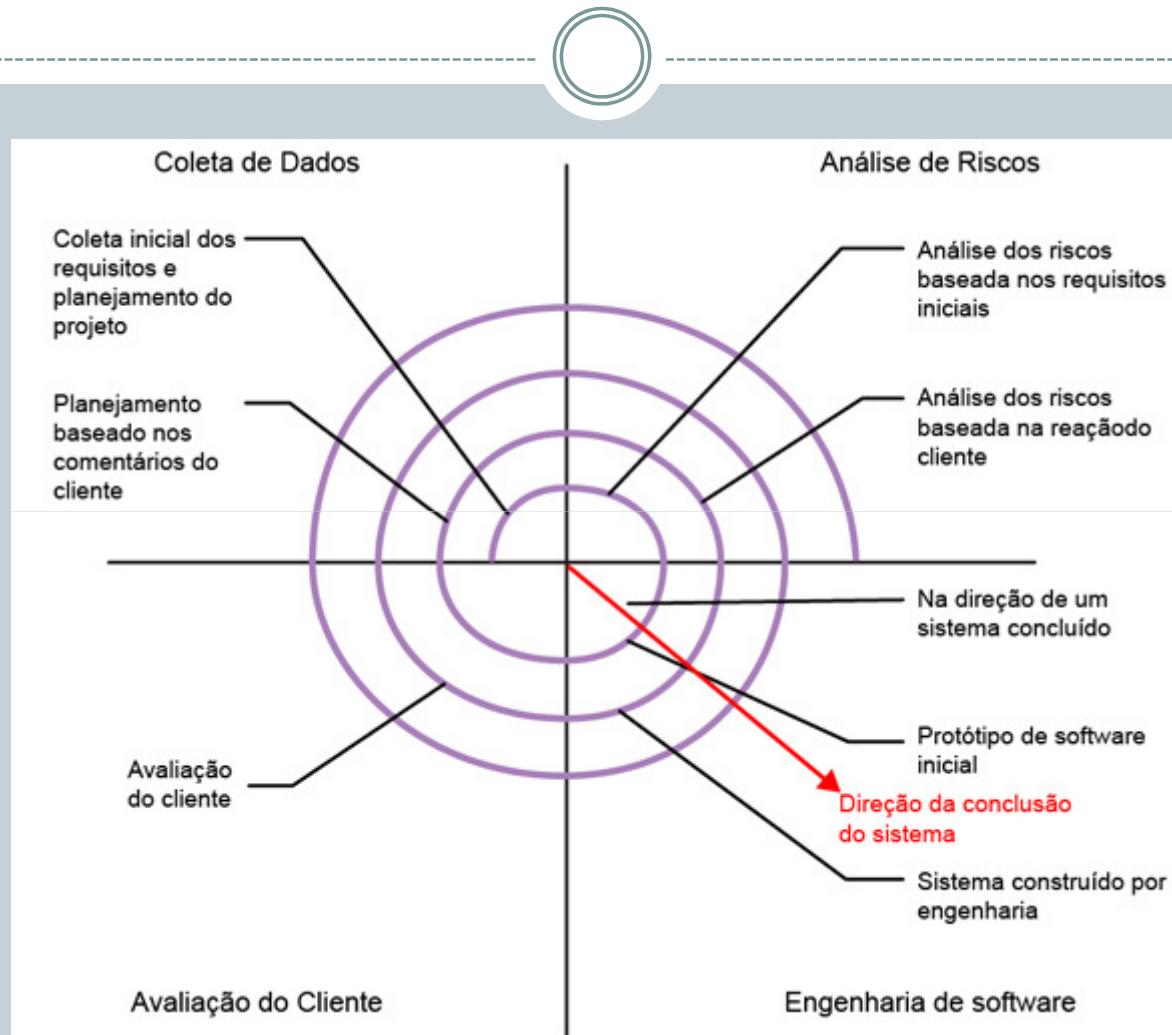
Modelo de Ciclo de Vida em Cascata



Ciclo de vida em espiral

- 
- É um tipo de desenvolvimento de sistemas mais maleável e adaptável, se comparado ao ciclo de vida em cascata.
 - Consiste em agregar funcionalidades ao sistema de forma evolutiva, apresentando resultados parciais em relação à meta desejada (final do desenvolvimento do sistema).
 - Os resultados que são apresentados devem ser operacionais, isto é, devem ser implantados para utilização do cliente.
 - Esse tipo de ciclo de vida permite, ainda, que sejam adaptados novos métodos ao longo do desenvolvimento do sistema.

Ciclo de vida em espiral



Ciclo de vida da prototipação



Muitas vezes, o cliente define um conjunto de objetivos gerais para o software, mas não identifica requisitos de entrada, processamento e saída detalhados. Em outros casos, o desenvolvedor pode não ter certeza da eficiência de um algoritmo, da adaptabilidade de um sistema operacional ou da forma que a interação homem-máquina deve assumir. Nessas, e em muitas outras ocasiões, uma abordagem de prototipação à engenharia de software pode representar a melhor abordagem.

Modelo de prototipação

O modelo pode assumir uma das três formas:

- **Operacionais** – quando aprovados pelo usuário, estão prontos para utilização (geração de produtos acabados. Exemplo: access).
- **Semioperacionais** – quando são necessárias poucas modificações no protótipo
- **não operacionais** – Quando são voltados para análise de pontos específicos.

Modelo de prototipação



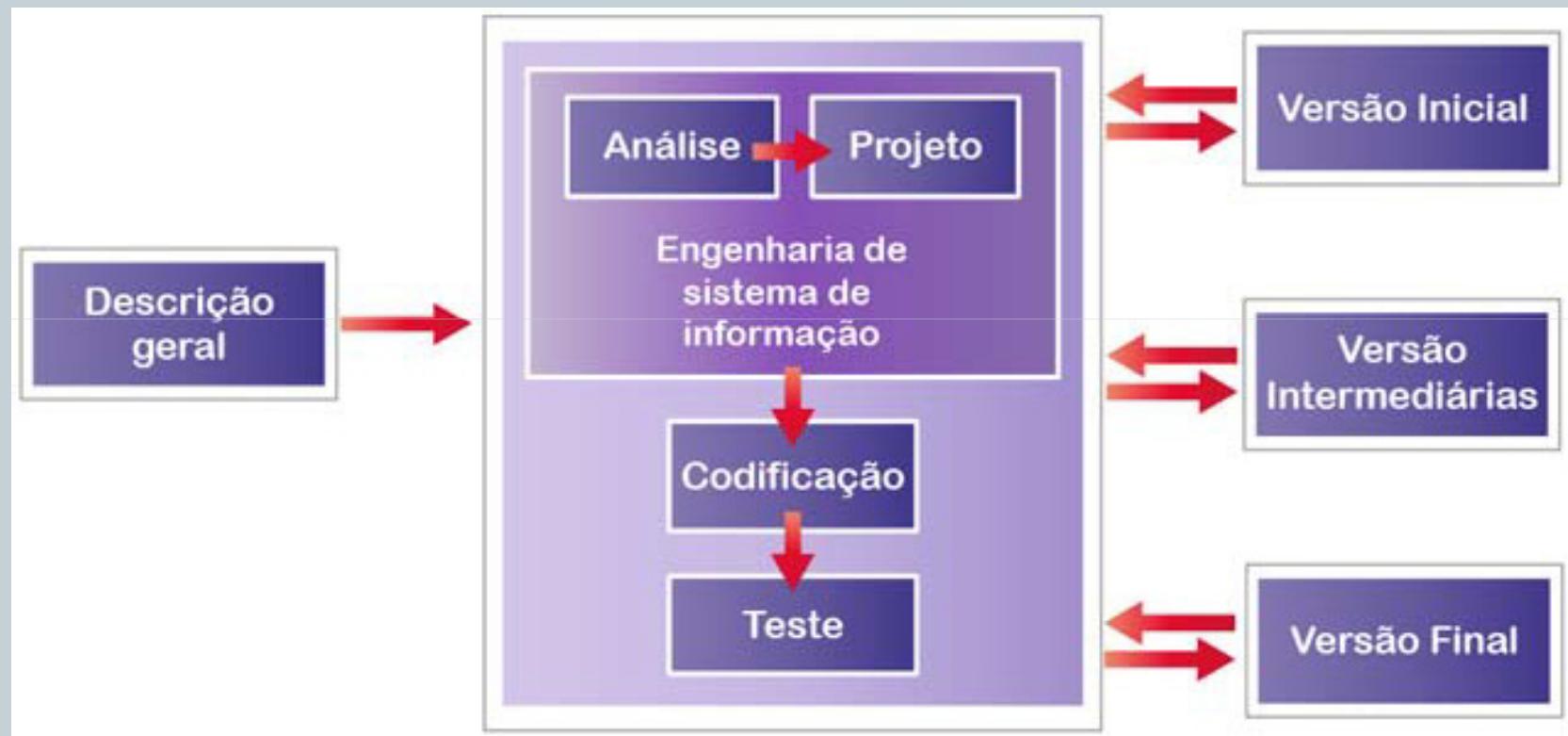
Modelo de Prototipação



- O projeto rápido leva à construção de um protótipo que é avaliado pelo cliente/usuário e é usado para refinar os requisitos para o software a ser desenvolvido. Um processo de interação ocorre quando é feita uma "sintonia fina" do protótipo para satisfazer às necessidades do cliente, capacitando, ao mesmo tempo, o desenvolvedor a compreender melhor aquilo que precisa ser feito.

Modelo incremental

- pode combinar elementos do modelo cascata (aplicado repetidamente) com a proposta iterativa da prototipação.
- O objetivo principal é trabalhar com o usuário para identificar corretamente os requisitos necessários, de maneira incremental, até que o produto final esteja consolidado.
- A versão inicial, geralmente, é onde se encontram as principais regras de negócio, que são a essência do sistema.
- a evolução ocorre quando novas características são adicionadas, de acordo com as necessidades do cliente.
- indicado quando é difícil estabelecer de imediato uma especificação detalhada dos requisitos, sendo indicado para sistemas pequenos, em que os problemas de mudança podem ser contornados.



Referências

- 
- DEMARCO, T. *Análise estruturada e especificação de sistema*. Rio de Janeiro: Campus, 1989.
 - GANE, Chris e SARSON, Trish. *Análise estruturada de sistemas*. Rio de Janeiro: LTC, 1983.
 - GANE, Chris. *Desenvolvimento rápido de sistemas*. Rio de Janeiro: LTC, 1989.
 - AMARAL FILHO, Antônio Rubens Aciães. *Projeto estruturado: fundamentos e técnicas*. Rio de Janeiro: LTC, 1988.
 - SOMMERVILLE, Ian. *Engenharia de Software*. 8. ed. São Paulo: Addison Wesley.
 - PRESSMAN, ROGER. *Engenharia de software*. 6.ed. São Paulo: Mc-Graw Hill.